Musical Instrument Recognition using Convolutional Neural Networks and Spectrograms

Rujia Chen

A research component submitted to Auckland University of Technology in fulfillment of the requirements for the degree of Doctor of Philosophy in Computer and Information Sciences.

2024

School of Engineering, Computer & Mathematical Sciences

Abstract

This thesis presents a study on musical instrument recognition, leveraging Convolutional Neural Networks (CNNs) and a One-vs-All (OvA) classification framework. The primary goal is to accurately identify individual instruments within complex polyphonic music under varying noise conditions. The research introduces a novel approach by incorporating multiple spectrogram features and attention mechanisms, aiming to enhance classification accuracy and robustness in real-world scenarios.

The study includes nine interrelated and progressively developed experiments, starting with the development of binary classifiers for specific instruments and progressing to large-scale evaluations using the NSynth and Open-MIC datasets. Early experiments validated the feasibility of the OvA approach on small datasets, while subsequent experiments expanded the model's scope to encompass ten instrument families, demonstrating its scalability and adaptability.

Contributions of this work include a detailed analysis of various spectrogram techniques, which highlighted the superiority of a combined spectrogram approach in capturing diverse acoustic features. This combination was effective in handling challenging polyphonic contexts, where single spectrogram techniques often fell short. The integration of attention mechanisms further refined the model's focus on the critical spectral and temporal patterns, improving instrument recognition in complex environments.

Key findings revealed that while the CNN-OvA models excelled in recognizing individual instruments and moderately complex mixes, they encountered challenges with larger ensembles.

The thesis also offers some insights through feature map and heatmap analyses, which contribute to a better understanding of the interpretability of the CNN model's responses to spectrogram inputs. These analyses elucidate the underlying decision-making processes and highlight potential areas for further refinement and optimization in model design.

This research contributes to the field of Music Information Retrieval (MIR) by exploring a novel classification approach, conducting empirical analyses, and providing insights that could

inform future developments. It underscores the potential of CNNs, enhanced by multi-spectrogram features and attention mechanisms, to tackle the complexities of polyphonic music recognition, paving the way for further advancements in the field.

Table of Contents

List of Figures:

List of Tables:

Attestation of Authorship

I hereby declare that this submission is my own work and that, to the best of my knowledge and belief, it contains no material previously published or written by another person (except where explicitly defined in the acknowledgements), nor used artificial intelligence tools or generative artificial intelligence tools (unless it is clearly stated, and referenced, along with the purpose of use), nor material which to a substantial extent has been submitted for the award of any other degree or diploma of a university or other institution of higher learning.

2024-08-01

Acknowledgements

I would like to express my heartfelt gratitude to my supervisor, Professor Ajit, for his exceptional guidance and support throughout my research on musical instrument classification. His insightful suggestions were invaluable, especially when I was struggling with finding a novel approach for the study.

I am also deeply thankful to Dr. Akbar for providing me with few projects to practice and refine my experimental skills, which also greatly contributed to the success of this research.

Additionally, I extend my appreciation to the CMS PhD Team Postgraduate Coordinator, Jan, for arranging my submission extension during a challenging phase of my research.

Chapter 1. Introduction

Imagine being in a cocktail lounge with people talking around us. A piano starts playing in the background. How is that we can instantly focus on the sound and identify it as a piano despite the background noise? Now imagine that, instead of a piano, a string quartet starts playing. How is that we can tell, after a few moments, that there are two violins, a viola and a cello despite the chatter? What computational methods and techniques exist so that intelligent computers can also do this quickly and effectively?

In the field of music technology, accurately identifying musical instruments in audio recordings is a significant challenge that combines the beauty of music with advanced computational techniques.

Two methods come to mind as to how we humans might be able to identify musical instruments. First, our brain takes in all the music information (signals) and filters the music into distinct instruments for classification (filtering into several, or FiS). Second, our brain has specific 'yes/no' classifiers that trigger if a particular instrument is present (one versus all), and then integrates the results to produce a list of instruments.

This thesis explores the area of Music Information Retrieval (MIR) (Downie, 2003), with a particular focus on using the OvA approach (Rifkin & Klautau, 2004) or in the other word, binary classifiers, to recognize different musical instruments. This research is at the crossroads of advanced machine learning algorithms and the detailed study of musical sounds. It aims to improve how digital music systems categorize and recommend music and to provide useful tools for music experts, teachers, and artists in their work.

Central to this thesis is the idea of combining technology with our understanding of how humans perceive music. It investigates how sophisticated computer models can replicate and enhance our ability to identify various musical instruments. By looking at the importance of instrument recognition in the wider field of MIR, the thesis lays the groundwork for understanding its significance and possible applications. From the early days of audio analysis to the recent

developments in deep learning, the following chapters present a story that aligns with the technological and methodological progress in this area.

The objective of this thesis is to offer a musical instrument software based on the OvA, presenting a method for building these models and evaluating how effective they are in real situations. It goes beyond theoretical discussions, touching on actual applications and challenges, and ends with thoughts on the future direction of MIR and its influence on our digital interaction with music. Essentially, this thesis combines research, innovation, and a deep appreciation for music, with the hope of contributing to the evolving field of music technology.

1.1. Overview of Musical Instrument Identification

Musical instrument identification involves recognizing and categorizing the sounds produced by different instruments. This field combines music, technology, and acoustics, blending artistic expression with scientific analysis. It explores how each instrument produces unique sounds, characterized by distinctive pitches, timbres, and intensities (Downie, 2003).

Traditionally, the ability to identify musical instruments was a skill developed by musicians, composers, and music enthusiasts (chapter 2.3). It required a deep understanding of how instruments sound in different settings, like solo performances, ensembles, and orchestras. This knowledge has been important for music composition, performance, and appreciation throughout history.

Despite considerable progress (chapter 2.4 and 2.5), the field continues to face challenges, particularly in accurately identifying instruments in complex and polyphonic pieces. However, these challenges also present opportunities for further research and innovation, driving the continuous evolution of this fascinating domain.

1.2 Objectives

Real-world polyphonic music often has multiple instruments playing simultaneously, making identification tricky. The broad objective of the thesis is to develop a deep learning model to classify musical instrument in polyphonic music. This objective is explained in more detail below

(chapter 1.2.1) and achieving this broad objective will be through six specific research questions (chapter 1.2.2).

1.2.1 Aim of the Study

The principal aim of this study is to explore and refine the OvA approach for the identification of musical instruments in audio files. This research seeks to understand how effectively individual instruments can be recognized and classified within diverse musical contexts using OvA models. The approach adopted will be brain-inspired (a convolutional neural network (CNN) approach). The purpose of using the OvA approach is to see how well the system learns new instruments and to find out what it means for future music recognition models inspired by the brain. One benefit of the OvA approach is that it is modular: learning a new instrument just means adding a new binary module. This is different from the FiS (filtering into several) approach, where adding a new instrument might require retraining the whole system.

A key aim is to tackle the inherent challenges in musical instrument recognition, especially in polyphonic compositions where multiple instruments are present. Achieving this aim will require investigation of how OvA models can differentiate between overlapping and harmonically complex sounds.

Another important goal is to help improve Music Information Retrieval (MIR) technology. By using and testing the OvA approach, the study aims to expand what is possible in automated musical instrument recognition.

In particular, while previous research has focused to a large extent on identifying single instruments clearly presented in an audio sample, very little research in comparison has been performed on how a recognition and identification system performs under conditions of noise (the cocktail lounge effect). So another aim of this thesis is to identify the limits of an OvA approach under increasingly noisy conditions to learn more about the limits of the approach as well as what is happening within the system.

Finally, the thesis assumes that nothing more than raw signals is required for musical instrument recognition. This is in contrast to most previous approaches, where prior feature

extraction and feature selection are performed before recognition. CNNs in image and speech analysis are proving effective in extracting the relevant features for classification using their own kernels and other architectural features. Therefore, for the purpose of this thesis, raw signals are interpreted as spectrograms, which provide a visual representation or descriptions of the musical signals, with no feature analysis performed on the spectrograms before input to the CNNs used in this study.

1.2.2 Research Questions

The broad objective of developing a deep learning model for musical instrument recognition is broken down into the following six research questions to determine how effective are at identifying musical instruments in different conditions:

1. What is the capability of Binary Classifiers (OvA model) in accurately identifying specific instruments from their spectrogram representations in a clear environment?
2. To what extent can Binary Classifiers (OvA model) discern and correctly identify musical instruments within a spectrogram that contains noisy backgrounds?
3. Under what noise conditions does the performance of OvA models degrade?
4. Are Binary Classifiers (OvA model) capable in recognizing each instrument present within samples of both synthetic and real-world audio recording polyphonic music, where multiple instruments are played simultaneously?
5. What type of spectrogram algorithm is suitable for identifying different types of musical instruments? Additionally, how do various algorithms perform when applied to different instruments?
6. What features are extracted from the convolutional layer for each instrument, and can these features be visualized and quantified?

Metrics for assessing whether the objectives associated with these questions will be introduced later in Chapter 1.3, the research objective chapter.

1.2.3 Expected Contributions to the Field

During the journey to answer the research questions, we aim to make several significant contributions. These contributions are expected to advance the field and have practical implications in various areas:

1. Improving MIR Techniques: This research may make constructive contributions to the field of Music Information Retrieval (MIR). By improving the accuracy and speed of instrument identification, it can offer new ideas and methods that are useful for both academic researchers and professionals working in this area. This can help in developing better tools and applications for analysing and understanding music.
2. Practical Applications: The findings are expected to have practical applications in various areas such as digital music libraries, music education, and audio production. Improved instrument recognition capabilities may lead to more intuitive music search and discovery features, enriched educational tools, and advanced audio editing software.
3. Interdisciplinary Impact: The study also aims to bridge gaps between disciplines. By combining techniques from machine learning, audio signal processing, and cognitive science, it can offer a holistic view of how technology can mimic and enhance human musical perception. This interdisciplinary approach can lead to innovations that benefit multiple fields and provide a deeper understanding of the interplay between technology and music.

1.2.4 Overview of Thesis Achievements

In this section, we present an overview of the key achievements of this thesis, encapsulated by our final model, illustrated in Figure 1. This model, a multi-spectrogram input attention network, exemplifies the culmination of our research efforts.

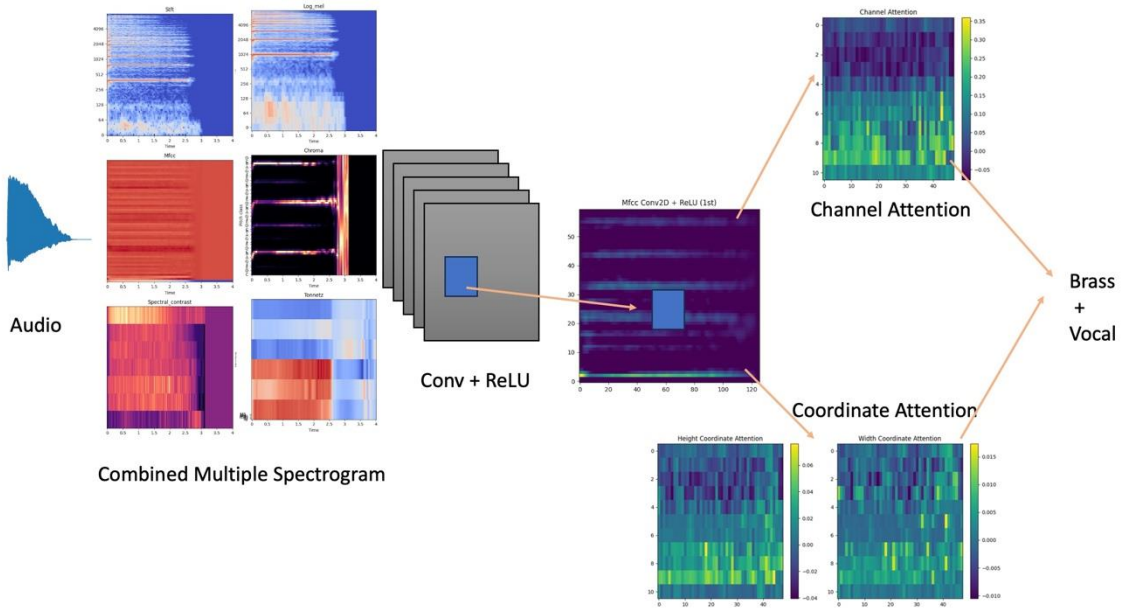


Figure 1. Multi Spectrogram Attention Convolutional Neural Network.

Figure 1 showcases the architecture of our Multi-Spectrogram Attention Convolutional Neural Network. The model begins with audio input, which is transformed into multiple types of spectrograms, each capturing different aspects of the audio features. These spectrograms are then combined to create a representation of the input, providing a rich set of features for the network to process.

However, the increased number of features can lead to challenges in determining which features are most important. To address this, our model incorporates attention mechanisms to dynamically focus on the most relevant features, thereby improving the model's performance and interpretability. The attention layers, including both channel attention and coordinate attention, act as a mechanism to selectively emphasize important features while filtering out less relevant ones.

This process can be likened to a "distract and refocus" approach, where the initial broad focus on multiple features is refined through the attention mechanisms to concentrate on the most critical aspects of the input data. By applying these attention layers, the model effectively enhances its ability to learn and generalize from the complex spectrogram inputs, ultimately leading to better performance in recognizing and classifying musical instruments. Each part of this model is detailed in the experimental sections of this thesis:

1. The creation and combination of multiple spectrograms are discussed in Chapter 5.6.
2. The implementation and impact of convolutional layers with ReLU activation are explored in Chapter 5.1 to 5.4
3. The application of channel and coordinate attention mechanisms and their effect on model performance are detailed in Chapter 5.8.
4. Chapter 5.5 investigates the feature map and heatmap analysis of synthetic data, while Chapter 5.9 presents the analysis of the real recording dataset.

This integrated approach not only improves classification accuracy across various instruments but also provides a robust framework for handling complex audio datasets, achieving the research objectives.

1.2.5 Structure of the Thesis

The thesis is organized to provide a detailed exploration of the OvA approach in musical instrument recognition conducting an Iterative Experiment Methodology. It starts with an introduction to the topic, followed by a comprehensive background review, methodology, experiments, results, and analysis.

Each chapter builds on the previous one, guiding the reader through the research process. The structure is designed to be logical, clear, and informative, taking the reader from theoretical foundations to practical implementations and future possibilities.

Specifically, how the research questions described in chapter 1.2.2 have been answered are mainly presented in Chapter 5, the Experiment chapter,

1. **Research Question 1** has been answered in chapter 5.1, and chapters 5.2 and 5.4 demonstrate that the OvA model can effectively recognize musical instruments in polyphonic settings.
2. **Research Question 2 and 3** have been addressed in chapter 5.3, which shows that while the OvA model can function in simple noisy backgrounds, its performance degrades when faced with complex noise levels such as traffic, crowds, and natural interferences.

3. **Research Question 4** has been answered in chapters 5.6 and 5.8, which evaluate the effectiveness of the model on the open-mic dataset with different network layer structures and attention algorithms.
4. **Research Question 5** has been answered in chapter 5.7, discussing the efficiency of six different spectrogram types.
5. **Research Question 6** has been answered in chapters 5.5 and 5.9, both statistically and visually.

Lastly, the final sections of the thesis will summarize the findings, discuss their implications, and suggest directions for future research. This includes a critical evaluation of the study's impact and contributions to the broader fields of Music Information Retrieval (MIR) and Artificial Intelligence (AI).

1.3 Objectives and Proposed Approach

This chapter presents the core research objectives and proposed approaches that form the foundation of this thesis. Our work aims to advance the field of musical instrument recognition through a series of interconnected experiments, each addressing specific challenges in audio signal processing and machine learning. We begin with the development of a prototype binary classifier and progressively build towards more sophisticated models capable of handling complex, real-world audio scenarios. Our research spans from evaluating performance on curated datasets to assessing robustness under various noise conditions and polyphonic settings.

1.3.1 Research Objectives

The interconnected research objectives are as follows:

- RO-1: Develop and evaluate an OvA model for instrument recognition in clear acoustic conditions, assessing its accuracy in identifying specific instruments from spectrograms.
- RO-2: Evaluate the scalability of the OvA model by increasing the number of instrument classes and assessing its performance and efficiency as the dataset grows.

- RO-3: Systematically evaluate the OvA model's performance under various noise conditions, identifying and characterizing specific conditions that lead to performance degradation.
- RO-4: Assess the OvA model's ability to identify multiple instruments in polyphonic music samples, using both synthetic and real-world audio recordings.

- RO5: Compare and evaluate the performance of various spectrogram algorithms, including the combination of multiple spectrogram features, in identifying different types of musical instruments to determine the most effective approach for each instrument category.
- RO-6: Extract, visualize, and quantify the features from the convolutional layers for each instrument, developing methods to interpret these features and gain insights into the model's decision-making process.

In summary, these objectives directly address the research gaps identified and align with the hypotheses proposed in chapter 3.1. They provide a clear roadmap for the research, focusing on key aspects of instrument recognition in various conditions and exploring the underlying mechanisms of the OvA model. This structured approach enables a thorough exploration of binary classification, robustness to noise, polyphonic recognition, model interpretability, and advanced spectrogram analysis techniques. By setting specific, quantifiable targets for each objective, the research ensures evaluation and clear benchmarks. The objectives span multiple datasets, including synthetic and real audio recording, to demonstrate the generalizability and real-world applicability of the developed models. Together, these objectives create a cohesive research narrative that systematically improves upon existing methods, culminating in an attention-based CNN model with enhanced accuracy and interpretability. The definitions and methodologies for measuring accuracy, EMR, mAP, and other metrics are detailed in the chapter 3 and chapter 4, providing a foundation for understanding and replicating the research outcomes.

1.3.2 Proposed Approach

Based on foundational works that set up the deep-learning-based MIR research conventions (Benetos et al., 2013; Choi, Fazekas, Cho, et al., 2017; E. J. Humphrey et al., 2012; Kong et al., 2019; McFee & Ellis, 2014), to evaluate the research objectives, we develop an approach as follows:

1. Dataset Preparation:
    - Training Data: Collect and prepare datasets of spectrograms for various instruments, ensuring a mix of solo, polyphonic, and noisy recordings.
    - Testing Data: Prepare separate datasets for testing the models, including samples with varying levels of background noise and polyphonic complexity.
2. Model Development:
    - CNN Architecture: Design and implement CNN architectures tailored to the unique characteristics of each instrument's spectrogram.
    - Binary Classifiers (OvA Model): Develop individual binary classifiers for each instrument, trained to distinguish between the presence and absence of the target instrument in the audio sample.
3. Feature Extraction and Analysis:
    - Spectrogram Types: Evaluate the effectiveness of different spectrogram types (STFT, log-mel, MFCC, chroma, spectral contrast, tonnetz) for instrument recognition.
    - Feature Visualization: Visualize and quantify the features extracted by the convolutional layers to understand their contributions to the recognition task.
4. Model Evaluation:
    - Noise Robustness: Test the models' performance in various noisy environments to determine the conditions under which their performance degrades.

- Polyphonic Settings: Evaluate the ability of the models to recognize instruments in polyphonic music, where multiple instruments are played simultaneously.

5. Advanced Techniques:
    - Attention Mechanisms: Implement and assess attention algorithms to enhance the models' focus on relevant parts of the spectrogram.
    - Network Layer Structures: Experiment with different network layer structures to optimize model performance.

By addressing these aspects, the proposed approach aims to provide an evaluation of the hypothesis.

1.4 Outputs Related to the Research

This research has yielded five major outcomes: three have been peer-reviewed, while two are currently under review.

Initially, my aim was to gain a better understanding of convolutional neural networks (CNNs). This led to the creation of the work titled "Evolving Convolutional Filter Using Genetic Algorithm for Image Classification" (Chen & Narayanan, 2021), which extends my master's study. This project provided an in-depth overview of the internal workings of CNNs, particularly focusing on the backpropagation mechanism. Through this work, we investigate alternative optimization techniques rather than backpropagation. By employing genetic algorithms (GAs) to evolve filter weights, we demonstrated that evolutionary techniques could serve as a viable substitute for gradient descent, particularly for simple classification tasks like geometric shape recognition. This study not only deepened our understanding of CNNs but also laid the groundwork for more complex applications.

Recognizing the capabilities of CNNs in handling complex tasks, we next applied CNNs as a backbone for drone images classification task in seal identification, resulting in another work titled "Seal Identification using CNNs" (Chen et al., 2023). This study aimed to identify individual seals from a set of images, leveraging the powerful feature extraction capabilities of CNNs. By

training the model on a dataset of seal images, we achieved high accuracy in distinguishing between different seals, showcasing the potential of CNNs in wildlife monitoring and conservation efforts. This work demonstrated how CNNs could be effectively used in practical applications beyond theoretical research, emphasizing their versatility and robustness.

Building on my exploration of backpropagation mechanisms and using CNNs as a backbone for image classification tasks, I gained sufficient experience to extend my knowledge to music applications. This effort culminated in the creation of another work, "OvA for Musical Instrument Classification" (Chen et al., 2024). In this study, we addressed the challenge of identifying musical instruments within polyphonic music using binary classification strategies. By employing CNNs to analyse various spectrogram representations of musical pieces, we achieved significant improvements in instrument recognition accuracy. This research highlighted the importance of feature selection and the ability of CNNs to adapt to different types of input data, further extending the applicability of CNNs to the field of music information retrieval.

Additionally, two more papers are currently under review for a conference and a journal. These works investigate the extension of single spectrogram analysis to multiple spectrograms, conducting t-tests (submitted, under review), and a preliminary study aimed at opening the "black box" of CNNs by understanding the features extracted from convolutional filters (submitted, under review). The t-test analysis explores the statistical significance of features extracted from different spectrogram types, providing insights into their contributions to the overall classification accuracy. Meanwhile, the preliminary study delves into the interpretability of CNNs, using heatmaps and feature maps to visualize the regions of spectrograms that influence the model's decisions the most. By examining these visualizations, we aim to identify the most critical features for accurate instrument classification and understand how CNNs process complex auditory signals.

Through these studies, we have systematically explored the capabilities of CNNs in various domains, from image classification and wildlife identification to musical instrument recognition. Each work has built upon the findings of the previous studies, employing an iterative experimental approach to refine our understanding and improve the performance of our models.

This research journey not only demonstrates the versatility and effectiveness of CNNs across different applications but also provides few insights into the inner workings of these neural network models.

1.5. Scope and Relevance

In this section, we introduce why musical instrument identification is a crucial area of study in Music Information Retrieval (MIR). Our goal is to show how recognizing different musical instruments in audio recordings can have a big impact on various aspects of music and technology. The importance of musical instrument recognition in; MIR is discussed. We will see that identifying instruments is not just about naming them; It is about opening ways for people to find and enjoy music. This can mean being able to search for songs by instrument or understanding music better through detailed analysis. "Scope and Relevance" aims to give a clear picture of how influential musical instrument identification is in the world of music and technology.

1.5.1 Importance of Music Recognition

Today, automated musical instrument identification has diverse applications, ranging from enhancing user experience on digital music platforms to aiding in music education and research. It plays a vital role in music information retrieval (MIR), helping to organize, categorize, and recommend music based on instrumental composition (Hu & Downie, 2007).

Automated musical instrument identification significantly enhances various aspects of music information retrieval (MIR), providing practical benefits in multiple domains. Two major areas where this technology shows its importance are in improving search and retrieval capabilities on digital music platforms and enriching music education and academic research.

1) Enhancing Search and Retrieval Capabilities

One of the most significant impacts of instrument identification in MIR is its ability to refine how we search and access music. Platforms like Spotify(Spotify AB, 2023) and SoundHound (SoundHound Inc., 2023) have implemented advanced search functionalities that allow users to find music based on various criteria, including instrumental content.

With advanced recognition techniques, users can search for music not just by artist or genre but also by specific instruments from Metadata. This capability is particularly beneficial in large digital music collections and libraries, where finding a piece of music featuring a particular instrument can be like searching for a needle in a haystack. By enabling more precise search criteria, this technology greatly enhances the user experience, making music discovery both easier and more personalized.

2) Enriching Music Education and Research

Instrument recognition technology holds immense value in the realms of education and academic research. For students and teachers, it offers a new dimension of learning – understanding the role and sound of various instruments in compositions. This can be a powerful tool in music education, enriching students' understanding of different music styles and compositions. Software like Ear-Master (EarMaster ApS, 2023) as an educational tool can be used for ear training and music theory practice. It includes exercises that help students recognize different instruments by their sound, an application directly benefiting from advances in instrument recognition technology. This tool is widely used in music education, demonstrating the practical application of MIR in teaching and learning environments.

The role of instrument recognition extends to making music more accessible, especially in educational settings. Tools that can identify and isolate instruments in a composition are invaluable for teaching music theory and practice. For instance, applications like Yousician (Yousician Ltd, 2023) use real-time feedback for music learning, which involves recognizing the instrument being played. This technology can be particularly beneficial for visually damaged students, who rely more on auditory cues. By enabling a deeper understanding of the structure and composition of music, these tools make learning more interactive and inclusive.

Educational applications, such as those developed by Avid Technology with their Pro Tools software, leverage this technology to provide students with interactive and immersive learning experiences, where they can isolate and study individual instruments within a mix, fostering a deeper understanding and appreciation of music.

For researchers and musicologists, the ability to automatically identify instruments in recordings can aid in more detailed music analysis, opening doors to new insights in music theory and history(Herrera-Boyer et al., 2006; Schedl et al., 2014).

3) Archiving and Preservation

The cataloguing and archiving of music also benefit significantly from the ability to identify musical instruments. Tools like Melodyne and Sibelius offer analysis features that include instrument recognition, aiding in music transcription and categorization.

In digital music libraries, correctly tagging recordings with the instruments they feature not only aids in organization but also in preservation. For instance, the British Library Sound Archive utilizes categorization of recordings, including the types of instruments, to preserve musical heritage. For example, organizations like the Smithsonian Folkways Recordings use such technologies to archive and study traditional music from around the world, ensuring that the rich musical traditions of various cultures are captured and preserved for future generations.

This aspect is particularly crucial for historical and cultural music recordings, where identifying and documenting the instruments used can provide valuable context and information for future generations (Casey et al., 2008). Such enhanced cataloguing improves both the accessibility and educational value of music archives.

4) Supporting the Music Industry and Artists

In the professional music industry, the application of instrument recognition can streamline several production and creative processes. For instance, Digital Audio Workstations (DAWs) like Ableton Live and Logic Pro X offer features that analyse and categorize instrument sounds for easier editing and mixing.

For producers and audio engineers, it simplifies tasks such as mixing and mastering by automatically identifying and isolating instrument tracks. For composers and musicians, this technology can inspire new creative pathways, allowing them to experiment with different instrumental sounds and combinations (Tindale et al., 2005). In an industry where time is often of the essence, these efficiencies not only save time but also open up new creative possibilities.

5) Facilitating Music Recommendation Systems

The integration of instrument identification in music recommendation systems has revolutionized how listeners discover new music (Aucouturier & Pachet, 2003). Streaming services like Pandora and Spotify have employed sophisticated algorithms that take into account the types of instruments played in a song. This advancement allows for a more nuanced recommendation system, catering to the specific tastes of listeners. For instance, a jazz enthusiast interested in tracks featuring a saxophone or a classical music lover looking for piano-centric compositions can easily find their preferences met. Such targeted recommendations significantly enhance the user experience, making music discovery both engaging and personalized.

6) Real-time Applications in Live Performances

The application of instrument identification is not limited to recorded music; it also has significant implications for live performances and interactive learning applications. In live concerts, real-time recognition of instruments can enhance audience engagement through interactive displays or augmented reality experiences(O'Hanlon & Plumbley, 2014).

7) Revolutionizing Music Recommendation Algorithms

Instrument identification plays a critical role in refining music recommendation algorithms. Platforms like Spotify and Apple Music now incorporate instrument-based data into their recommendation engines, allowing for recommendations that are tailored to the listener's instrumental preferences. This means users can discover new music not just based on genres or artists they like but also based on their favourite instruments. The impact here is two-fold: listeners enjoy a more personalized experience, and artists, especially those known for specific instrumental skills, gain increased visibility.

The advent of instrument identification technology has markedly changed how digital music platforms classify and organize their content(Celma Herrada & others, 2009; Radio, 2015). By accurately tagging tracks with specific instrument data, these platforms can offer more sophisticated categorization, moving beyond basic genre or artist classifications. For instance, a track can be tagged not just as "jazz" but more specifically as "jazz with prominent saxophone,"

enhancing the granularity of music categorization. This advancement aids in the creation of more nuanced and varied playlists, catering to the diverse preferences of listeners.

8) Enhancing Audio Content Analysis

In the realm of content analysis, instrument identification technologies enable more in-depth analysis of audio files. For example, music producers and sound engineers can use these technologies to automatically detect and isolate specific instruments within a track for remixing or mastering purposes. This capability is particularly beneficial in complex compositions where manual identification and recognition of instruments would be time-consuming and challenging.

9) Facilitating Music Metadata Management

The accurate identification of instruments also plays a significant role in the management of music metadata. In digital archives and libraries, instrument data can be used to enhance the metadata of each track, making the search and retrieval process more efficient and user-friendly. This is especially valuable for researchers, educators, and enthusiasts who are looking for specific types of music or conducting comparative musical studies.

10) Improve Audio File Compression

On a more technical level, instrument identification can influence audio quality and file compression techniques. By understanding the instrumental makeup of a song, audio compression algorithms can be optimized to preserve the quality of specific instruments while compressing the file. This leads to a better balance between file size and audio quality, which is crucial in streaming services where bandwidth and storage are considerations

1.5.2 One-vs-All Approach Overview

The OvA approach (Rifkin & Klautau, 2004), also known as one-vs-rest, is a strategy used in machine learning for multi-class classification problems.

The rationale behind employing binary classifiers lies in the inherently multi-class nature of the classification problem at hand, as opposed to scenarios that involve single-class classification. For instance, in the case of the MNIST dataset, an image of a handwritten digit, such as the handwritten image on Figure 2(a) labelled as '2', distinctly represents the number '2' and cannot simultaneously signify '2' and '4', barring instances where the handwriting is ambiguous or overlapping (refer to figure 2(b) for illustration).

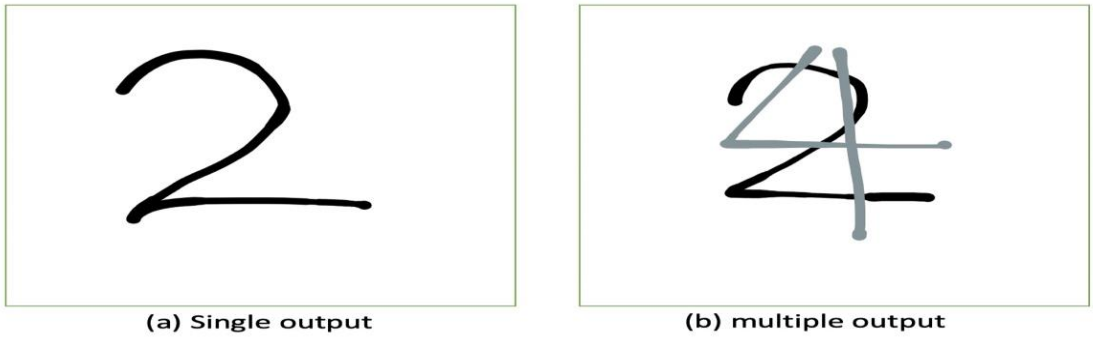


Figure 2. Comparison of single-label and multi-label output classification. (a) Single-label output where the classifier predicts a single digit. (b) Multi-label output where the classifier identifies overlapping digits, recognizing both '2' and '4'.

However, in the context of polyphonic music, as illustrated in Figure 3, the scenario diverges significantly.

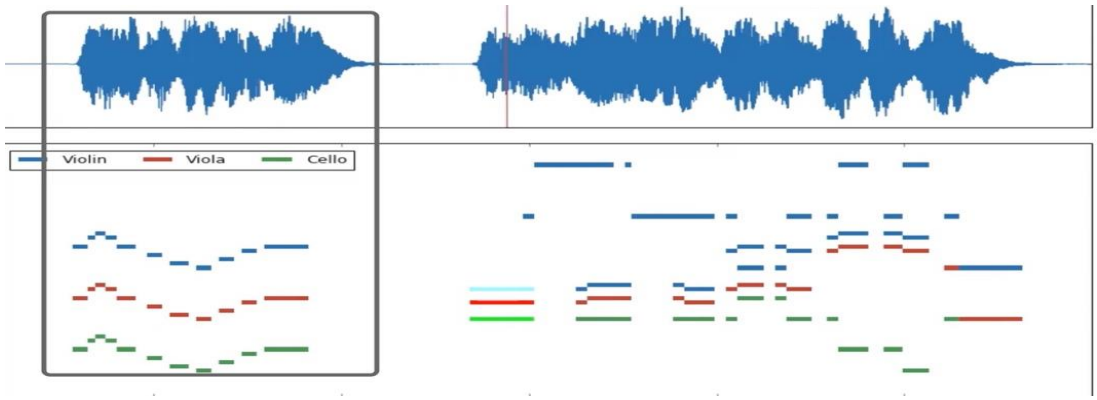


Figure 3. Polyphonic music often contains overlapping signals in the waveform, necessitating a multi-label classification strategy from source (Bittner et al., 2018).

Polyphonic compositions inherently feature overlapping sounds where a single segment of a waveform can encapsulate the presence of multiple instruments, such as a violin, viola, and cello simultaneously (Bittner et al., 2018). This complexity underscores the necessity for OvA binary classifiers in distinguishing and accurately classifying each instrument within such densely layered audio environments.

Multi-label learning algorithms, which tackle the challenge of associating an instance with multiple labels simultaneously, can be approached through various methodologies. One prevalent strategy involves transforming the multi-label problem into multiple binary classification tasks, a technique that allows for individual assessment of each label's presence or absence in an instance (Zhang & Zhou, 2013). Another approach is converting the multi-label problem into a label ranking task, where the goal is to predict a ranking of labels for each instance based on their relevance, further enriching the model's interpretability and applicability (Fürnkranz et al., 2008). These methods exemplify the diversity and adaptability of machine learning algorithms in handling complex, multi-faceted data scenarios.

In this thesis, we only design the model based on transferring a multi-label problem into a multiple binary classification problem.

1.5.3 Definition and Concept

The binary classifier approach involves creating a separate binary classifier for each musical instrument. Each classifier is trained to recognize whether its specific instrument is present in a given audio segment, effectively distinguishing that instrument from all others. This means that for each instrument, the classifier outputs a 'yes' or 'no' regarding the presence of that instrument in the audio sample. By using multiple binary classifiers, each dedicated to a single instrument, the system can accurately identify multiple instruments within a complex audio signal.

In practical terms, if our models are classifying three different instruments, the OvA approach would create three distinct classifiers (see Figure 4). Each binary classifier is trained on data where its target instrument is labelled as positive, and all other instruments are labelled as

negative. This method simplifies the complex problem of multi-class classification into manageable binary classification tasks.

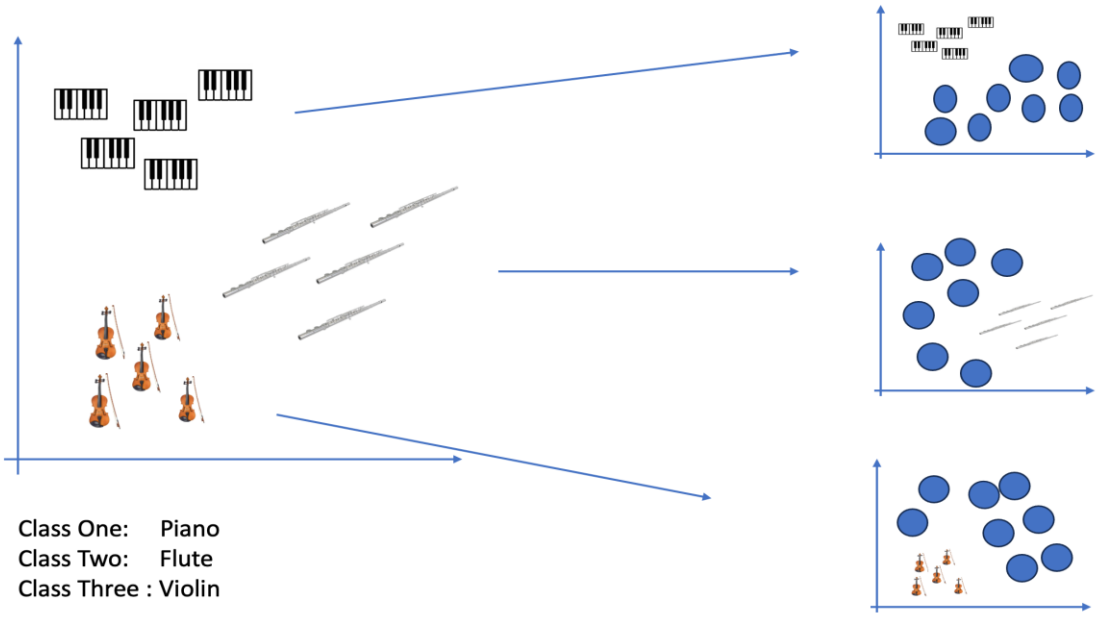


Figure 4. Illustration of the binary classifier approach. Three binary classifiers are developed to represent and distinguish between piano, flute, and violin. Each classifier is responsible for identifying whether its respective instrument is present in the audio segment, effectively separating it from all other instruments. This approach allows for accurate identification of multiple instruments within a complex audio signal.

The rationale for employing the OvA approach in our study stems from the inherent complexity of polyphonic music analysis. In a typical spectrogram of a polyphonic piece, the audio segment may not consist of a single instrument. Instead, it could simultaneously feature multiple instruments, such as a combination of a violin, flute, and vocal, all playing together. This scenario presents a challenge for traditional multi-class classification methods, as the input data (in this case, the spectrogram) would require multiple positive class labels for each segment, corresponding to each instrument present.

To effectively address this challenge, and as noted earlier, we propose transforming the multi-label classification problem into a series of binary classification problems through the OvA approach. This method allows us to create individual models for each instrument, where each model is trained to distinguish whether its respective instrument is present in the audio segment or not, against the backdrop of other sounds. By doing so, we can more accurately analyze complex musical pieces where multiple instruments are playing concurrently, ensuring each

instrument's presence is identified effectively in the spectrogram. This binary simplification is crucial for handling the overlapping and diverse nature of sounds in polyphonic music.

So, what we do with the OvA approach is we break down this big problem into smaller, easier parts. Instead of trying to figure out everything at once, we make a separate 'detector' for each instrument. Each detector's job is to find out whether its instrument – and only its instrument – is playing in a part of the music.

For example, one detector focuses only on identifying the violin. It checks each part of the music and answers 'yes' if it detects a violin and 'no' if it does not. Simultaneously, another detector does the same for the flute, and yet another for the voice. This way, instead of tackling one big, complicated problem, we address several smaller, simpler problems all at once. All detectors work at the same time, ensuring comprehensive analysis of the audio in one go.

1.5.4 Advantages over Other Methods

OvA approach is not only a valid choice but also an advantageous one for our specific research context in MIR. It has many advantages over other methods.

1) Focused Learning and Specialization:

One of the primary advantages of the OvA approach is its ability to focus on one instrument at a time, allowing for more specialized and detailed learning. This focus can lead to more accurate recognition, especially in cases where specific characteristics of an instrument need to be closely analysed.

2) Alignment with Human Intuition and Cognitive Processes

Other advantages of the OvA approach is to mirror human cognitive processes in recognizing musical instruments. Just as a person might focus on identifying one instrument at a time in a piece of music, the OvA model specializes in recognizing individual instruments against a diverse auditory backdrop. This alignment with human perception can be a compelling reason, especially in applications aimed at mimicking or complementing human listening experiences.

3) Flexibility in Adding New Instruments

Also, other advantages of the OvA approach is the modularity and scalability of the OvA approach when it comes to expanding the model to include new instruments. In a multi-label or complex neural network setting, adding a new instrument often requires retraining the entire model, which can be resource intensive. In contrast, with OvA, we can add or update individual models dedicated to each instrument without disturbing the existing system, making it more adaptable and efficient for evolving musical databases, especially in fields like music, where new instruments or sounds might emerge

4) Isolated Training and Updating for Specific Instruments

In addition, other advantages of the OvA approach is it can isolate model training in the OvA setup. If one instrument model underperforms or needs updating due to new data or improved techniques, we can retrain just that specific model without having to retrain models for all the other instruments. This not only saves computational resources but also allows for targeted improvements, ensuring that updates or refinements are more manageable and less disruptive to the overall system.

5) Computational Efficiency in Model Training and Deployment

Also, OvA is a computational efficiency approach, particularly relevant if dealing with a limited computational budget or aiming for deployment in less powerful environments (like mobile apps or web services). OvA models, being simpler and more focused, can require less computational power to train and run compared to more complex multi-label classifiers, making them more suitable for scenarios where computational resources are a concern.

In traditional multi-class classification problems, especially with a large number of classes, the model complexity and computational requirements can be quite high. OvA simplifies this by breaking down the problem to multiple single problems, which can be particularly beneficial in cases with limited computational resources.

6) Potential in Polyphonic Music Analysis:

Finally, while recognizing instruments in polyphonic settings (where multiple instruments play simultaneously) is challenging, the OvA model offers a strategic way to tackle this problem. By training classifiers to recognize individual instruments in complex audio, the approach holds potential for advanced polyphonic music analysis.

1.6 Summary of Chapter One

Chapter One presents the research question and hypothesis of the study on musical instrument identification, outlining the aim, objectives, and expected contributions. It introduces the proposed OvA approach, highlighting its advantages in modularity and scalability for recognizing multiple instruments. The chapter also discusses the importance and relevance of music recognition, setting the stage for the detailed methodologies and experiments to be covered in subsequent chapters. The next chapter, the Literature Review, will discuss the necessary literature in music recognition, providing a foundation for understanding the current state of the field and the innovations introduced by this research.

Chapter 2. Literature Review

In the previous chapter, we presented the research question and hypothesis of this study, outlining the aim, objectives, and hypothesis. We introduced the proposed OvA approach for musical instrument identification, highlighting its advantages in modularity and scalability. The importance and relevance of music recognition were discussed, setting the stage for the detailed methodologies and experiments to follow.

And this section provides a broad overview of the early stages in the development of musical instrument recognition, from manual efforts to the first electronic and digital processing attempts. It sets the stage for the subsequent sub-sections, which will delve into the evolution from manual to automated recognition, key milestones in the field, and the transformative shift towards machine learning. This historical perspective not only provides context but also highlights the remarkable progress made in this area of music technology.

2.1 Auditory Perception Theory

Table 1 summarizes key auditory perception theories. Each theory is briefly described, followed by a brief description of the human auditory processing mechanism being modelled or simulated by the theory

Although the field of human auditory perception remains an open and ongoing area of research (Plack & Moore, 2010; Van Opstal, 2016; Warren, 2013) , theories in biological and psychological sciences including Auditory Scene Analysis (ASA) (Bregman, 1994), The Gestalt Principle of Perception (Koffka, 1922; Wertheimer, 1938), Temporal Coherence Theory (Elhilali et al., 2009; Shamma et al., 2011), Stream Segregation (B. C. Moore & Gockel, 2002) , Auditory Attention (Shinn-Cunningham, 2008) , Multisensory Integration (Calvert et al., 2004) Neural Plasticity and Learning (Peretz & Zatorre, 2003) developed by auditory scientists.

Table 1. Overview of Auditory Perception Description

Auditory Perception Theory	Brief Description
Auditory Scene Analysis (ASA)	Auditory Scene Analysis (ASA) (Bregman, 1994) refers to the cognitive process by which the auditory system separates and organizes sounds into perceptually distinct elements or streams. This involves identifying different sound sources in a complex auditory environment, such as distinguishing a conversation partner's voice from background noise at a busy cafe. ASA relies on various cues, such as pitch, timbre, spatial location, and temporal patterns, to group sounds that likely originate from the same source and segregate those from different sources.
Gestalt Principle of Perception	The Gestalt Principle of Perception (Koffka, 1922; Wertheimer, 1938) is a theory that suggests humans naturally perceive visual and auditory stimuli as whole, organized structures rather than as disjointed individual parts. In the auditory domain, this principle manifests in the tendency to group sounds based on proximity, similarity, continuity, and common fate (sounds that change together are perceived together). For instance, when listening to music, we tend to hear melodies and harmonies as coherent wholes rather than as isolated notes.
Temporal Coherence Theory	Temporal Coherence Theory (Elhilali et al., 2009; Shamma et al., 2011) posits that sounds are grouped together perceptually when they exhibit synchronized changes over time. This means that the auditory system uses the temporal relationships between sounds to determine which sounds belong together. For example, a sequence of sounds with consistent timing and rhythm, like the footsteps of a person walking, will be perceived as coming from a single source.
Stream Segregation	Stream Segregation (B. C. Moore & Gockel, 2002) is the process by which the auditory system separates different sound sources into distinct perceptual streams. This allows listeners to focus on a specific sound, such as a single musical instrument in an orchestra or a speaker in a noisy room. Factors influencing stream segregation include pitch, timbre, spatial location, and temporal coherence. Effective stream segregation enables better understanding and processing of complex auditory scenes.
Auditory Attention Models	Auditory Attention (Shinn-Cunningham, 2008) Models describe how the brain selectively focuses on specific sounds while filtering out others. This selective attention is crucial in environments with multiple sound sources, such as crowded places or while multitasking. Auditory attention can be directed voluntarily (top-down) or involuntarily (bottom-up), based on the salience and relevance of sounds. These models help explain phenomena like the "cocktail party effect," where one can focus on a single conversation amidst background noise.
Multisensory Integration	Multisensory Integration (Calvert et al., 2004) refers to the process by which the brain combines information from different sensory modalities to create a comprehensive understanding of the environment. For auditory processing, this means integrating auditory information with visual, tactile, and other sensory inputs. For example, seeing a person’s lips move while they speak can enhance auditory perception and improve speech comprehension, particularly in noisy environments.
Neural Plasticity and Learning	Neural Plasticity and Learning (Peretz & Zatorre, 2003) in the context of auditory perception refer to the brain's ability to adapt and reorganize its neural pathways based on experience and training. This plasticity underlies the improvement of auditory skills over time, such as language acquisition, musical training, and recovery from hearing loss. Through repeated exposure and practice, the auditory system can enhance its ability to recognize and discriminate between different sounds, leading to improved auditory recognition and comprehension skills.

The seven concepts outlined in Table 1 form the foundation of psychoacoustical sciences, providing crucial insights into how humans perceive and process auditory information. These theories and models elucidate the cognitive and neural mechanisms underlying auditory perception, offering a comprehensive framework for understanding complex auditory scenes, attention mechanisms, and the integration of multisensory inputs. auditory science (Bregman, 1994; Calvert et al., 2004; Elhilali et al., 2009; Koffka, 1922; B. C. Moore & Gockel, 2002; Peretz & Zatorre, 2003; Shinn-Cunningham, 2008).

By exploring these, the knowledge-based models (Patterson et al., 1987; Slaney & others, 1993) built directly upon these auditory principles, some are called 'filterbank' models, which require human knowledge to predefine the filters. For example, the Gammatone filterbank, developed by Patterson et al. (1988), mimics the filtering characteristics of the human cochlea and has been extensively used in various applications, such as speech and music analysis, and hearing aid design (B. C. Moore, 2012).

The concepts outlined in Table 1 establish a direct analogy between human auditory perception theories and the computational techniques used in this study. Auditory Scene Analysis (ASA) relies on cues such as pitch, timbre, spatial location, and temporal patterns to group sounds from the same source while distinguishing them from others. This parallels how spectrogram-based techniques function, as they are designed to replicate the way human auditory perception organizes complex soundscapes. A more detailed discussion of this relationship can be found in Chapter 2.3.3.1 (Spectrograms). Similarly, the role of Auditory Attention Models aligns with attention mechanisms in neural networks, both of which selectively enhance relevant information while filtering out background noise. This analogy is further explored in Chapter 2.4.4 (Attention Mechanisms in MIR). Additionally, Neural Plasticity and Learning serve as an inspiration for neural networks, which mimic the brain's ability to adapt and refine auditory recognition based on experience and training. The connection between these concepts is discussed in Chapter 2.4.1 (Neural Networks).

2.2 Related Psychoacoustics Research

2.2.1 Cocktail Party Effect

Many machine learning algorithms draw inspiration from bionics, the study of natural systems as models for designing and engineering technology. Similarly, our model is also inspired by the cocktail party effect (Arons, 1992; Cherry, 1953; Haykin & Chen, 2005), a phenomenon illustrating how humans can focus on a single sound source while filtering out a myriad of background noises, which is a subdomain of selective attention mechanism (Johnston & Dark, 1986). This capability has led us to develop a OvA model, which acts as a binary classifier to identify a focused instrument in polyphonic music or amidst noisy backgrounds, effectively ignoring other sounds. We believe our approach mimics human auditory selective attention (Johnston & Dark, 1986), enabling the enhancement of specific sound recognition in complex acoustic environments. By leveraging this principle, we aim to improve the accuracy and efficiency of sound separation and identification tasks, contributing to advancements in audio processing technologies.

Further extending this concept, the OvA model not only simplifies the multi-class classification problem into multiple binary classification problems (Rifkin & Klautau, 2004) but also provides a framework for understanding how to prioritize and isolate individual sound sources in a manner analogous to human auditory processing. This strategy facilitates more refined control over sound extraction and enhances the model's ability to deal with real-world audio challenges, such as distinguishing a particular instrument within an orchestra.

2.2.2 Challenges in Musical Instrument Identification

Identifying musical instruments within compositions presents a significant challenge, not only for computational models but also for humans. Research conducted by Stöter et al. (Stöter et al., 2013) delves into the human capacity to discern the number of instruments in polyphonic music, revealing that while 69% (Figure 5) of participants could confidently identify a single instrument in a piece, their confidence dwindled when faced with multiple instruments. This illustrates the

intrinsic complexity of recognizing instruments in music, especially as real-world compositions are predominantly polyphonic, complicating the extraction and identification process.

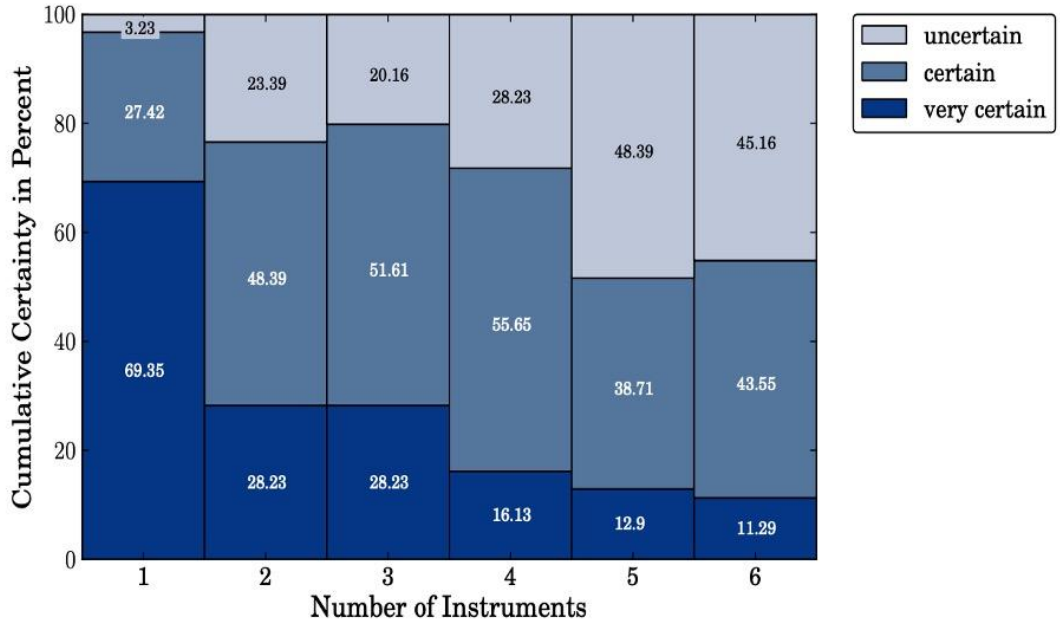


Figure 5. Human ability of counting the number of instruments in polyphonic music (Stöter et al, 2013).

The task of musical instrument identification is further complicated by the diverse nature of instrument sounds, which can vary widely in timbre, quality, and playing style (Eronen, 2001; Essid et al., 2006). These variations introduce additional layers of difficulty for accurate instrument recognition. Furthermore, the multifaceted characteristics of musical sounds impose a significant computational burden due to their high dimensionality (Kitahara et al., 2003). Such studies underscore the inherent challenges in isolating and identifying musical instruments from polyphonic compositions, highlighting both the human and computational struggles with this intricate task(Fuhrmann & others, 2012).

2.3. Early Techniques in Instrument Recognition

2.3.1 Manual Categorization

1) Understanding the Traditional Approach

Initially, the task of recognizing and classifying musical instruments in recordings was a domain reserved for trained musicians and musicologists (Copland, 1952, 2011). Copland (1952) was known for his efforts to make music accessible and understandable to a wider audience. His book, "What to Listen for in Music," serves as a classic guide that helps readers appreciate the complexities of music, including understanding orchestration and the distinctive qualities of different instruments in a composition.

This era, before the advent of advanced computational tools, relied heavily on human perception and expertise. Skilled listeners would identify instruments based on their timbral characteristics (McAdams, 1999, 2013), a process heavily influenced by their training, experience, and subjective interpretation.

The book – "Auditory Scene Analysis: The Perceptual Organization of Sound" (Bregman, 1994), provides valuable insights into how humans perceive and organize sound, including the identification of musical instruments. It offers a foundational understanding of the auditory processing that underpinned early methods in instrument recognition.

Clarke's work (Clarke, 2005) delves into the ecological approach to auditory perception, explaining how humans perceive and categorize sounds, including musical instruments, within their environment. This research is fundamental to understanding the cognitive processes involved in early manual methods of instrument recognition.

The method build upon expert knowledge (Agostini et al., 2003; Martin, 1999), while requiring a deep understanding of musical sounds, was inherently subjective. Different experts might have varying opinions about the instruments used in a piece, especially in complex compositions or when the quality of the recording was poor. The process was not only time-consuming but also lacked consistency, highlighting the need for a more systematic approach.

The subjective nature and inherent limitations of manual methods highlighted the need for more objective, reliable, and scalable solutions, setting the stage for the evolution towards computational approaches in instrument recognition (Casey et al., 2008).

2.3.2 Early Electronic Method

1) Oscilloscopes and Spectrum Analysers:

The introduction of electronic sound analysis marked the beginning of automated recognition (Havelock et al., 2008; Pierce, 2019). It allowed for a more systematic approach, although it was still rudimentary compared to modern standards (Casey et al., 2008). The journey towards automated recognition of musical instruments began with the advent of electronic sound analysis. This period marked a significant departure from the manual, ear-based methods that had dominated the field(Chowning, 1973). Electronic analysis introduced a more structured, systematic way of examining audio signals, laying the groundwork for what would eventually become a more sophisticated automated process(Moorer, 1975).

- **The Onset of the Digital Revolution:** The advent of the digital era marked a significant turning point in the field of musical instrument recognition. This period saw a shift from analogue electronic methods to digital signal processing (DSP), which opened up new possibilities for analysing and interpreting complex audio data (Rabiner & Gold, 1975).
- **Development of DSP Tools:** Digital signal processing tools (Oppenheim & Schafer, 1975) brought with it sophisticated tools and algorithms capable of dissecting sound with greater precision. Computer-based systems began to employ techniques such as Fast Fourier Transforms (FFT) for spectral analysis, digital filtering, and waveform manipulation, offering a more in-depth understanding of the acoustic properties of musical instruments.
- **Oscilloscopes** were among the first electronic tools used to visualize audio signals. They displayed waveforms of sound, allowing researchers to observe the shapes and frequencies of audio signals from musical instruments. Spectrum Analysers (Blackman & Tukey, 1958) expanded on the capabilities of oscilloscopes by providing a more

detailed analysis of the frequency spectrum of sounds. They were crucial in breaking down complex audio into its constituent frequencies. These tools allowed for a more objective look at the properties of sound, which was a significant advancement over relying solely on the human ear.

During the era of early computer recognition system, some of the first computer-based instrument recognition systems were developed (Martin, 1999; Wold et al., 1996). These systems utilized basic DSP tools to analyse audio recordings, identifying instruments based on their characteristic frequency patterns, attack and decay profiles, and other acoustic features.

2.3.3 Feature Engineering in Classical Machine learning

In the realm of classical machine learning, feature engineering is the process of extracting and selecting specific characteristics (or "features") from raw data to make it understandable for machine learning models (Eronen & Klapuri, 2000; Tzanetakis & Cook, 2002). In the context of musical instrument recognition, this involved transforming complex audio signals into a set of identifiable and quantifiable features that could be used for instrument classification.

2.3.3.1 Spectrograms

Humans perceive musical pitch in terms of the logarithm of frequency, rather than the frequency itself (Stevens et al., 1937; Stevens & Volkmann, 1940). That is the basis of the following logarithmic scaling features.

The book "An Introduction to the Psychology of Hearing" by Brian C.J. Moore (2012), provides a comprehensive overview of psychoacoustics, including how humans perceive pitch and timbre. It explains the logarithmic scaling of pitch perception, which is part of the basis for the design of the following spectral features.

1) Discrete Fourier Transform (DFT):

The Discrete Fourier Transform is a mathematical technique used to convert signals from the time domain to the frequency domain (Brigham, 1988). In the context of audio analysis, this means transforming a segment of sound (which naturally varies over time) into a representation that shows the frequencies that make up this sound. DFT decomposes an audio signal into its

constituent frequencies, providing insight into the harmonic content of the signal. This decomposition is fundamental in understanding the spectral characteristics of different musical instruments (Oppenheim & Schafer, 1975).

Its ability to transform audio signals from the time domain to the frequency domain makes it invaluable for analysing the spectral content of musical instruments (Pamuk & others, 2022), In practice, the DFT is applied to small segments of an audio signal, often using a windowing function to mitigate edge effects. This approach allows for the analysis of how the spectral content of a signal change over time, which is particularly important in music where the sound of instruments can vary from moment to moment. The calculation of the DFT, especially for large data sets or long audio files, can be computationally intensive (Brigham, 1988). Thus, Brigham claims the Fast Fourier Transform (FFT) algorithms are commonly used to efficiently compute the DFT, reducing the computational load and making the process more feasible for real-time applications.

Instruments can often be differentiated by their unique spectral signatures – the specific way they produce and combine different frequencies (Eronen & Klapuri, 2000). Many musical instruments produce sound with a rich harmonic structure, including fundamental tones and various overtones. The DFT helps in identifying these harmonic elements, which are crucial in distinguishing one instrument from another(Fletcher & Rossing, 1998). While DFT provides a detailed snapshot of frequency content, it does not inherently capture the dynamic changes in sound over time. This limitation is often addressed by using methods like the Short Time Fourier Transform (STFT), which applies the DFT in a moving window across the signal (Allen & Rabiner, 1977b).

One inherent limitation of the DFT is the trade-off between time and frequency resolution. Higher frequency resolution reduces time resolution and vice versa, which can be a consideration when analysing sounds that have rapid temporal changes (Cohen, 1995).

2) Short Time Discrete Fourier Transform (STFT):

The STFT signal (Allen & Rabiner, 1977b). is an extension of the basic concept of the Discrete Fourier Transform (DFT). It addresses one of the primary limitations of DFT - the lack of temporal information. Allen and Rabiner propose that STFT divides the longer time signal into shorter segments of equal length and then applies the Fourier Transform to each of these segments. This process allows for the analysis of both frequency and time, providing a time-varying frequency representation of the signal.

A key aspect of STFT is the use of a windowing function to select segments of the signal. This window moves across the signal, and at each position, the Fourier Transform is applied, resulting in a two-dimensional representation of time and frequency (Allen & Rabiner, 1977b). Thus, the choice of window size and type (like Hamming, Hanning, etc.) can significantly affect the STFT's ability to capture certain features of the signal (Harris, 1978).

STFT is particularly useful in musical instrument recognition because it captures how the spectral content of an audio signal changes over time. This is essential for identifying instruments, as many have distinctive ways their sound evolves like the attack, sustain, and decay of a note (Tzanetakis & Cook, 2002).

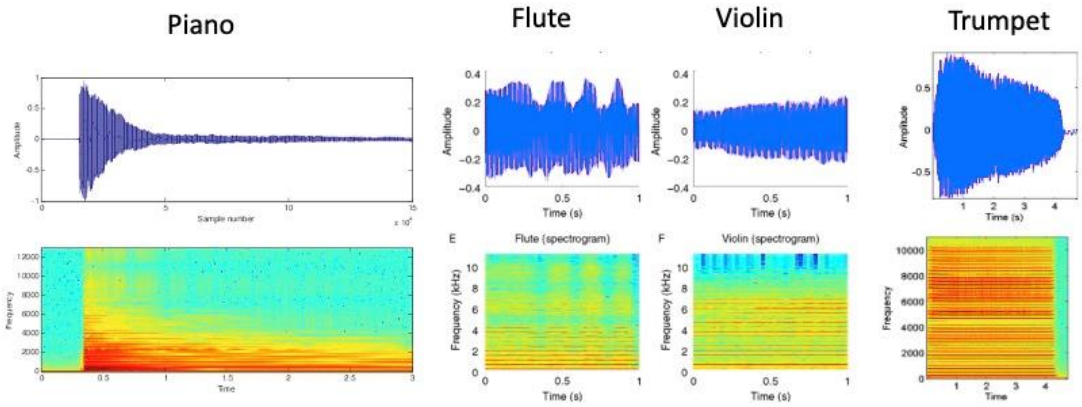


Figure 6. The amplitude waveform and STFT spectrograms of four musical instruments: Piano, Flute, Violin, and Trumpet. Each instrument's amplitude waveform is shown in the top row, highlighting the differences in their attack, sustain, and decay characteristics. The bottom row displays the corresponding STFT spectrograms, illustrating how the frequency content of each instrument evolves over time.

The output of STFT is often visualized in a spectrogram (Figure 6), where the x-axis represents time, the y-axis represents frequency, and the intensity of colours or shades represents the amplitude or energy of a particular frequency at a particular time. Spectrograms are a powerful tool for visualizing and understanding the frequency content of audio signals.

- Piano: The waveform shows a rapid decay after the initial strike, with the spectrogram indicating a rich harmonic structure that diminishes over time.
- Flute: The waveform displays periodic peaks, corresponding to the breath pulses, while the spectrogram shows sustained harmonic overtones.
- Violin: The waveform demonstrates the bowing action, with a relatively steady amplitude. The spectrogram reveals sustained harmonics with slight frequency modulations due to the vibrato.
- Trumpet: The waveform has a strong attack and a gradual decay, with the spectrogram showing prominent harmonics that remain relatively stable over the duration of the note.

Also, from Figure 6 we can see, the instruments in a piece of music often produce sounds that vary greatly over short periods. The STFT's ability to provide a time-frequency analysis makes it ideal for analysing these complex sounds, where both the frequency content and its evolution over time are important for recognition (Griffin & Lim, 1984). The STFT involves a compromise between time and frequency resolution. A wider window gives better frequency resolution but poorer time resolution, and vice versa (Cohen, 1995). This trade-off must be carefully managed based on the specific requirements of the audio analysis task.

The challenges of STFT instrument classification models are,

- **Overlapping Windows:** To improve the continuity and quality of the time-frequency representation, overlapping windows are often used in STFT. This technique helps in capturing more detailed information about the signal, especially in rapidly changing audio environments (Allen & Rabiner, 1977b).
- **Computational Efficiency:** While STFT is more computationally intensive than DFT, the use of efficient algorithms and modern computing resources has made it feasible for real-time analysis and large-scale data processing (Rabiner & Schafer, 1978).

In summary, the Short Time Discrete Fourier Transform is a critical tool in spectral feature engineering, especially for tasks that require a detailed understanding of how an audio signal's frequency content varies over time. Its application in musical instrument recognition is invaluable, allowing for the analysis of the dynamic and evolving nature of musical sounds.

3) Constant Q Transform (CQT):

The Constant Q Transform (Brown, 1991) is a powerful tool for analysing musical audio signals. It stands out from other Fourier-related transforms due to its logarithmic frequency scaling, which is more aligned with the way humans perceive pitch. This scaling means that the ratio of the frequency to the bandwidth (the 'Q' factor) is constant across the spectrum.

The CQT is especially adept at capturing the harmonic structure of sounds, a critical aspect in music (Brown & Puckette, 1992). CQT effectively represents musical notes and scales, making it a valuable tool for recognizing musical instruments (Schörkhuber & Klapuri, 2010).

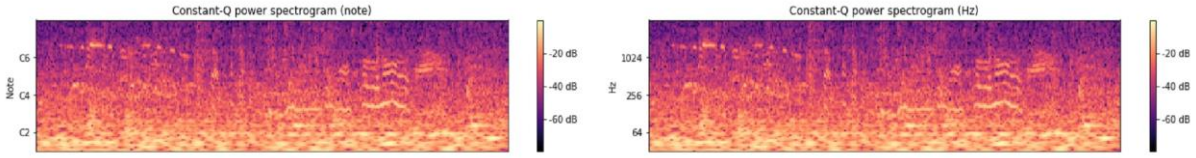


Figure 7. CQT spectrogram.

The output of the CQT can be visualized (Figure 7) as a form of spectrogram, where each filter's output is represented over time. This visualization provides insights into both the pitch content and the temporal evolution of the music (Schörkhuber & Klapuri, 2010), aiding in the detection and recognition of musical instruments.

For polyphonic music, where multiple instruments play simultaneously, the CQT's ability to distinguish overlapping harmonic structures becomes particularly useful (Schörkhuber & Klapuri, 2010). It helps in identifying the contribution of individual instruments within a complex mix. CQT involves dividing the audio signal into frames (like STFT) and then applying a series of filters, each tuned to a specific frequency. These filters cover the entire audible spectrum and are spaced logarithmically to match musical scales (Brown & Puckette, 1992).

The challenges and considerations of CQT of instrument classification may be:

- Computational complexity: One of the challenges with the CQT is its computational complexity, especially when high resolution in both time and frequency is required. Advances in computing power and optimized algorithms have made it more accessible for real-time applications and large-scale analyses (Schörkhuber & Klapuri, 2010).
- Resolution crade-offs: Similar to STFT, CQT also involves a trade-off between time and frequency resolution. However, its logarithmic nature tends to provide a more balanced view, especially in capturing the nuances of musical notes and harmonies (Brown, 1991).

In summary, the Constant Q Transform is an essential technique in spectral feature engineering for music-related applications. Its unique approach to frequency analysis, particularly its alignment with musical scales and harmonic structures, makes it invaluable in tasks such as musical instrument recognition and polyphonic music analysis.

4) Log-mel Spectrograms

A mel-spectrogram is an enhanced version of a spectrogram where the frequency scale is converted to the mel scale (Slaney, 1998). The mel scale more closely approximates human hearing perception than the linear frequency scale used in standard spectrograms. It emphasizes lower frequencies more than higher ones, mirroring how the human ear perceives sound (Stevens et al., 1937).

The "log" in log-mel spectrogram refers to applying a logarithmic transformation to the amplitude (or energy) of the mel spectrogram. This transformation is applied because human perception of sound intensity is also logarithmic. In simpler terms, we perceive differences in loudness in a logarithmic manner rather than linear (Zwicker & Fastl, 1990).

Log-mel spectrograms are widely used as input features for machine learning models (Choi, Fazekas, Sandler, et al., 2017; Graves et al., 2013; Pons et al., 2017), especially deep learning models like Convolutional Neural Networks (CNNs). They provide a compact and efficient representation of the audio signal that captures relevant information for tasks like speech recognition, music genre classification, and instrument identification.

2.3.3.2 Scalogram

1) Wavelet Transform:

Fourier-based methods use a fixed window size. The trade-off between time and frequency resolution is because the size of the tapering window is fixed, and the trade-off between time and frequency resolution may be optimal for a certain frequency, but not for others (Cohen, 1995). Thus, we should set different window sizes for different frequency ranges (Mallat, 1999).

The Wavelet Transform is a mathematical technique used for time-frequency analysis of signals (Mallat, 1999). Unlike the Fourier Transform, which only analyses frequency content, the Wavelet Transform provides a way to analyse both time and frequency components simultaneously (Daubechies, 1992). It does this by decomposing a signal into wavelets - small waves that are localized in time.

One of the key strengths of the Wavelet Transform is its ability to provide good time resolution for high-frequency events and good frequency resolution for low-frequency events (Strang & Nguyen, 1996). This makes it particularly effective for analysing signals with non-stationary or transient characteristics - common in many musical instruments. For example.

- **Capturing Transient Sounds:** Musical instruments, especially percussive and plucked instruments, often produce transient sounds with significant characteristics in a short time frame. The Wavelet Transform is adept at capturing these transient features, which are crucial for accurate instrument identification.
- **Dynamic Range of Instruments:** Different instruments have a wide range of dynamics and tonal qualities. The Wavelet Transform's adaptability in analysing various frequency components over time makes it a versatile tool for analysing this dynamic range.

The output of the Wavelet Transform is often represented in a scalogram (Figure 8), which is similar to a spectrogram but uses wavelet coefficients instead of Fourier coefficients (Addison, 2017). This representation is useful for visualizing both the frequency content and the timing of audio events.

The wavelet scalogram in Figure 8 provides a detailed time-frequency representation of an audio signal. It shows how the energy of the signal is distributed across different scales (frequencies) and time. Time (or Space) Axis (x-axis) represents the time or spatial dimension over which the signal is analysed. Scales Axis (y-axis) represents the different scales of the wavelet transform, which correspond to different frequency components of the signal. Lower scales represent higher frequencies, while higher scales represent lower frequencies. The intensity of the colours indicates the magnitude of the wavelet coefficients. Areas with higher energy are shown in red, while areas with lower energy are shown in blue.

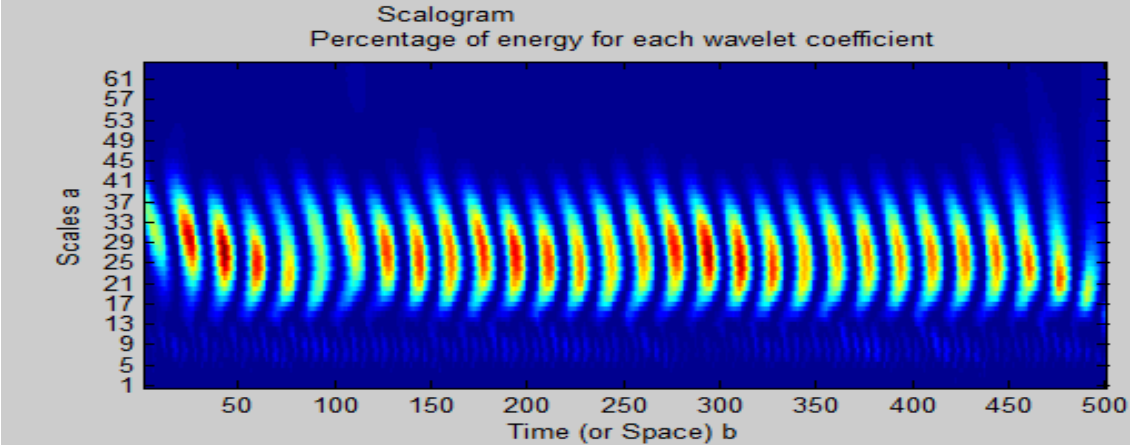


Figure 8. Wavelet Scalogram.

However, the effectiveness of the Wavelet Transform can depend heavily on the choice of wavelets and the parameters used (Addison, 2017). T. This requires a degree of experimentation and expertise to fine-tune the transform for specific audio analysis tasks.

In machine learning, particularly in models analysing audio data, the Wavelet Transform can be used for feature extraction (X.-R. Liu et al., 2010). It provides a set of features that represent both the temporal and frequency characteristics of a sound, which can be highly beneficial for tasks like instrument recognition in complex audio environments (Dhanalakshmi et al., 2011).

2.3.3.3 Conclusion of Feature Selection Method

To summarize all the literature reviews above, spectrograms are excellent for analysing the harmonic content of musical signals, as they provide a clear representation of the frequency spectrum over time. In polyphonic music, where multiple instruments play simultaneously,

spectrograms can help distinguish different instruments based on their spectral signatures. Spectrograms are generally well-suited for instruments with sustained and harmonic sounds, like strings or wind instruments, where the frequency content is relatively stable over time.

Scalograms are particularly adept at capturing transient sounds, which are brief and non-sustained, making them ideal for percussive instruments like drums or plucked strings like guitars. For instruments that produce sounds with rapidly changing frequency content, the adaptability of the Wavelet Transform in scalograms provides a more nuanced analysis. Scalograms are beneficial for analysing instruments that produce complex sounds with quick temporal variations, as well as for music with a lot of dynamic changes in tone and pitch.

2.4. Rise of Deep Learning

2.4.1 Introduction of Neural Networks

The rise of multilayer neural networks (Rosenblatt, 1958; Rumelhart et al., 1986) marked a significant shift in machine learning. Originating from the desire to mimic the human brain's functioning, neural networks are computational models designed to recognize patterns in data through a structure and process similar to human neurons (McCulloch & Pitts, 1943). Neural networks consist of layers of interconnected nodes (neurons), where each connection represents a weight that is adjusted during the learning process (Rumelhart et al., 1986). These networks learn to perform tasks by considering examples, generally without being programmed with any task-specific rules.

In the realm of audio signal processing, the introduction of neural networks opened new possibilities for complex tasks like speech recognition (Graves et al., 2013; Hinton et al., 2012), sound classification (Hershey et al., 2017; Piczak, 2015), and, notably, musical instrument recognition (Choi, Fazekas, Sandler, et al., 2017; Han et al., 2016). They offered a more sophisticated and flexible approach compared to traditional machine learning techniques.

2.4.2 Evolution of Convolutional and Recurrent Networks

Convolutional Neural Networks (CNNs) (LeCun et al., 1995, 2015), primarily used in image processing, found significant applications in audio analysis (Hershey et al., 2017; Piczak, 2015). They are adept at handling data with a grid-like topology, such as spectrograms of audio signals (Choi, Fazekas, Sandler, et al., 2017; Han et al., 2016). CNNs are known for their ability to detect patterns and features in data (LeCun et al., 2015), making them suitable for identifying characteristics in musical instruments.

Recurrent Neural Networks (RNNs) (Elman, 1990) are designed to recognize patterns in sequences of data, making them ideal for time-series analysis like audio signals. They have an internal memory that helps them understand the temporal dynamics of data, crucial for tasks like music generation and temporal aspects of sound classification. The evolution of these networks

led to more advanced architectures like Long Short-Term Memory (LSTM) (Hochreiter & Schmidhuber, 1997) networks (a type of RNN), which are particularly effective in handling long-term dependencies in data, beneficial for analysing extended audio recordings.

Transformer (Vaswani et al., 2017) revolutionized natural language processing (NLP) by models to capture long-range dependencies in data more efficiently than previous recurrent architectures. The key innovation in transformers is the self-attention mechanism. This mechanism allows the model to weigh the importance of different parts of the input data (such as words in a sentence) and focus more on relevant parts while processing. The self-attention mechanism computes a weighted sum of input features, enabling the model to dynamically adjust the focus on different input elements, thereby capturing complex dependencies and relationships. Additionally, transformers employ positional encodings to maintain the order of input sequences, as they process the entire sequence simultaneously rather than step-by-step like RNNs.

Building on the success of transformers, Dosovitskiy et al. introduced Vision Transformers (2020), which adapt the transformer architecture for image analysis. ViTs work by dividing an image into a grid of smaller patches, which are then treated as tokens (mimic the word splits in text). Each patch is flattened and linearly embedded into a vector, and these vectors are fed into the transformer model. Like in NLP, the self-attention mechanism allows the ViT to weigh and integrate information from different patches, effectively capturing spatial relationships and dependencies across the image. This approach enables ViTs to learn rich and detailed representations of visual data, often achieving state-of-the-art performance in image classification tasks.

2.4.3 Breakthroughs in Deep Learning for Audio

2.4.3.1 Automated Feature Learning

One of the major breakthroughs with deep learning in audio analysis is the ability of networks to learn feature representations automatically (Hinton et al., 2012; LeCun et al., 2015). This contrasts with traditional methods where features had to be meticulously handcrafted (Piczak, 2015). With deep learning, particularly Convolutional Neural Networks (CNNs) and Recurrent Neural

Networks (RNNs), models can learn complex patterns and representations directly from the raw audio data (Graves et al., 2013; Hershey et al., 2017). This automated feature learning significantly reduces the need for domain-specific knowledge and labour-intensive feature engineering, allowing the models to discover intricate patterns that may be overlooked by human designers (Aytar et al., 2016).

2.4.3.2 End-to-End Learning Models

Deep learning enabled the development of end-to-end learning models in audio processing(Dieleman & Schrauwen, 2014), where raw audio can be input directly into the neural network, and the system learns to extract relevant features and perform classification or regression tasks. This approach streamlines the pipeline, eliminating the need for intermediate steps such as feature extraction and selection. End-to-end models have proven to be highly effective in various audio-related tasks, including speech recognition (Hinton et al., 2012), (Hershey et al., 2017; Piczak, 2015), and, notably, musical instrument recognition (Choi, Fazekas, Sandler, et al., 2017; Han et al., 2016).

2.4.3.3 Advancements in Model Architectures

The evolution of model architectures such as CNNs, RNNs has played a critical role in the advancements of deep learning for audio. CNNs excel in capturing local patterns in spectrograms and other time-frequency representations (Hershey et al., 2017; Piczak, 2015), making them suitable for tasks like audio classification and sound event detection. RNNs and Long Short-Term Memory (LSTM) networks are adept at handling sequential data (Graves et al., 2013; Hochreiter & Schmidhuber, 1997), making them ideal for tasks involving temporal dependencies, such as speech recognition and music generation. Recently, Transformer networks, known for their attention mechanisms, have been successfully applied to audio tasks, providing superior performance in tasks like automatic speech recognition and music composition (Oord et al., 2016).

2.4.3.4 Real-World Applications

Deep learning has led to significant advancements in various real-world applications, for example,

- Voice Assistants Applications developed by Technologies such as Google Assistant, Amazon Alexa, and Apple Siri (Tulshan & Dhage, 2019) leverage deep learning models to understand and respond to natural language queries. These systems utilize deep learning for tasks like speech recognition, natural language understanding, and speech synthesis, providing users with intuitive and responsive voice interactions.
- Automated Music Transcription is another type of DL application. Deep learning models have improved the accuracy and efficiency of automated music transcription, which involves converting audio recordings into symbolic representations like MIDI(Hawthorne et al., 2018; Thickstun et al., 2016) . This is particularly challenging for polyphonic music, where multiple instruments play simultaneously. Deep learning models can learn to separate and transcribe individual instruments, providing detailed and accurate transcriptions.

3.4.3.4 Transfer Learning and Pre-trained Models:

The concept of transfer learning, where models pre-trained on large datasets are fine-tuned for specific tasks, has been instrumental in advancing audio processing. Pre-trained models like VGGish (Hershey et al., 2017) and OpenL3 (Cramer et al., 2019) provide powerful feature extractors that can be adapted for various audio tasks with relatively small amounts of task-specific data. This approach accelerates the development of effective audio models and makes advanced audio processing accessible even with limited computational resources

2.4.4 Attention Mechanisms in MIR

In recent years, attention mechanisms have been increasingly applied to various Music Information Retrieval (MIR) tasks, leading to significant advancements. A study (Gururani et al., 2019) explored an attention mechanism within a multiple-instance learning framework to address weakly labeled data in multi-label instrument recognition. The proposed method demonstrated improved performance over traditional approaches in handling overlapping instrument sounds.

Also, a self-attention-based deep sequence model (Won et al., 2019) for music tagging, combining convolutional layers with Transformer encoders. This architecture achieved

competitive results on the MagnaTagATune and Million Song Dataset, offering enhanced interpretability through heat map visualizations.

A study (Balke et al., 2019) introduced a soft-attention mechanism to enhance tempo-invariant audio-sheet music retrieval systems. The attention model focused on relevant parts of the input representation, enhancing robustness against tempo variations.

Another study (Dorfer et al., 2018) applies soft-attention mechanism to address tempo variations in audio queries , allowing the model to encode the most informative parts of an audio excerpt. This approach improved retrieval performance and demonstrated intuitive behavior.

A study proposed MATT (X. Liu & Zhang, 2022), a novel multiple-instance attention mechanism designed to enhance the classification of long-tail music genres. The mechanism improved performance in identifying underrepresented genres by focusing on informative segments within music tracks.

The researchers developed a system that utilizes attention mechanisms to identify emotionally salient segments in pop music (C.-H. Huang & Yang, 2020). The model effectively highlighted key emotional points, aiding in tasks such as music summarization and recommendation.

Another study (Tseng & Yeh, 2021) introduced multi-attention neural networks to capture diverse aspects of music for automatic tagging. The model demonstrated improved tagging accuracy by attending to various musical characteristics.

2.5. State-of-the-Art in Instrument Recognition

Recent advancements in musical instrument identification have leveraged deep learning techniques to achieve significant improvements. Wise et al. (2024) investigated attention-augmented convolutional neural networks (CNNs) for musical instrument identification, demonstrating that introducing attention mechanisms to CNNs improves the network's ability to capture the causal structure within spectrograms. Their study, which utilized the Short-Time Fourier Transform (STFT) and Constant-Q Transform (CQT) spectrograms, found that a 25% attention augmentation led to accuracy improvements, achieving up to 95.09% with STFT inputs (Wise et al., 2024).

Choi et al. (2017) developed a Convolutional Recurrent Neural Network (CRNN) that combines CNNs for local feature extraction with Recurrent Neural Networks (RNNs) for temporal summarization, showing that this hybrid architecture is highly effective for music tagging tasks. The CRNN outperformed several existing CNN models, highlighting the importance of temporal feature summarization in music classification(Choi Convolutional recu…).

Blaszke et al. (2022) proposed an innovative approach for musical instrument identification by integrating multiple spectrogram features. Their work demonstrated that combining features from different spectrogram representations, such as CQT and STFT, enhances the model's capability to differentiate between similar-sounding instruments, thereby improving classification accuracy.

Gururani et al. (2018) focused on a CNN-based model designed to classify musical instruments in polyphonic music by utilizing various spectrogram representations. This approach was particularly effective in complex audio environments where multiple instruments overlap, showcasing the importance of diverse feature inputs for accurate classification.

Cakir et al. (2017) introduced a method using convolutional neural networks for musical instrument recognition in polyphonic music, emphasizing the significance of spectral and temporal features in achieving high classification accuracy. Their approach was notable for its

ability to handle the complexity of polyphonic music, where multiple instruments are present simultaneously.

Tan, Wong, and Baskaran (2023) explored the use of deep learning models for instrument identification, specifically focusing on the efficiency of different neural network architectures in processing large-scale audio datasets. Their findings indicated that model architecture plays a critical role in balancing computational efficiency with classification accuracy.

Moore (2007) contributed to the field with a comprehensive analysis of feature extraction techniques for musical instrument identification, comparing traditional methods with emerging machine learning approaches. This work laid the groundwork for subsequent research by highlighting the potential of combining various feature extraction techniques to improve model performance.

In summary, this thesis does not aim to provide a comprehensive survey of all literature in the field of musical instrument identification. Instead, it focuses on key studies that have directly inspired the approach taken in this work. The research discussed here highlights the evolving role of deep learning, particularly CNNs, CRNNs, and the integration of multiple spectrogram features, in advancing the accuracy and efficiency of musical instrument identification systems.

2.5.1 Current Leading Methodologies

Blaszke and Kostek (2022) categorize the machine learning approach of recognizing musical instrument to 2 categories, which are Feature Extraction approach, and 2D Audio Representation approach.

The Feature Extraction approach involves extracting a Feature Vector (FV) full of audio descriptors to inform machine learning algorithms. The 2D Audio Representation approach relies on a two-dimensional audio layout for processing through deep learning models (especially CNNs), or even more advanced methods, like variational or deep SoftMax autoencoders, to enhance audio representation extraction.

2.5.1.2 Feature Extraction Approach:

This approach involves extracting specific features from audio signals, such as spectral, cepstral, and other relevant features that are used to train machine learning models. Feature extraction methods can include techniques like Mel-Frequency Cepstral Coefficients (MFCCs), Chroma features, and others that help capture the distinctive characteristics of different musical instruments.

Liu et al. (2022) proposed a musical instrument classification algorithm based on multi-channel feature fusion and XGBoost. They extracted features from audio signals and combined them using feature fusion techniques to improve classification accuracy . Another feature - extraction research (Eronen & Klapuri, 2000) explores the use of cepstral coefficients and temporal features for musical instrument recognition, showcasing their effectiveness in identifying different instruments. There is also a model (Choi et al., 2016) builds up a method for automatic tagging of musical instruments using deep convolutional neural networks and various spectral features.

2.5.1.3 2-D Audio Representation Approach:

This method involves transforming audio signals into 2D representations, such as spectrograms, which are then used as input to convolutional neural networks (CNNs) or other deep learning models. These 2D representations capture both time and frequency information, providing a more comprehensive view of the audio signal.

Recent advancements have seen the rise of end-to-end learning models that can process raw audio waveforms directly, bypassing the need for manual feature extraction. Models like WaveNet (Oord et al., 2016) and SampleRNN (Mehri et al., 2016) demonstrate this capability, offering improved performance by learning from the raw audio data.

Blaszke and Kostek (2022) utilized convolutional neural networks (CNNs) to identify musical instruments from 2D spectrograms. Their model achieved high accuracy in identifying various instruments by leveraging the detailed information captured in these 2D representations.

Another study (Taenzer et al., 2021) uses a deep CNN with an attention mechanism for automatic instrument recognition (AIR), where the attention mechanism relies on features from a pre-trained CNN. The findings reveal that using log-mel spectrograms as input reduces AIR performance, emphasizing the importance of feature representation quality.

The reason nowadays the deep learning model is replacing the traditional music feature approaches can be found in the paper "Moving beyond feature design: Deep architectures and automatic feature learning in music informatics," (E. J. Humphrey et al., 2012). This paper critiques the reliance on hand-crafted features in music informatics, which often results in diminishing returns. They advocate for the adoption of deep learning architectures that automatically learn feature representations from data, arguing that these methods can overcome the limitations of heuristic-based designs. The paper reviews the benefits of deep learning, emphasizing its potential to revolutionize music information retrieval (MIR) by addressing both new and existing challenges more effectively.

2.5.2 Musical Instrument Identification Models

Patil et al. (2012) delve into the intricate world of timbre perception, investigating the neural underpinnings that enable the auditory system to distinguish the unique characteristics of various musical instruments. Their research emphasizes the complex nature of auditory processing and its role in identifying and differentiating musical sounds. By employing a neuro-computational model based on spectro-temporal receptive fields (STRFs) of neurons in the mammalian primary auditory cortex, the authors shed light on the joint spectro-temporal features that contribute to the rich representation necessary for accurate timbre perception and robust instrument classification. Their findings underscore the significance of cortical neurons in encoding perceptually relevant information, establishing a direct link between neural representations and behaviourally relevant perceptual attributes.

Chi et al. (2005) make a significant contribution to the understanding of sound attributes and structure analysis through their proposed multiresolution spectrotemporal analysis. Their research broadens the comprehension of the auditory system's ability to process and interpret complex acoustic information. By presenting a computational model inspired by psychoacoustical

and neurophysiological findings in the early and central stages of the auditory system, the authors provide valuable insights into the spectral and temporal modulation content of sounds. Their work highlights the importance of considering both spectral and temporal dimensions in the analysis of sound, offering a more comprehensive approach to understanding the intricacies of auditory processing.

Agus et al. (2012) explore the rapid categorization of musical sounds based on their timbre, focusing on the cognitive speed in identifying and distinguishing between different instrument sounds. Their research sheds light on the human auditory system's remarkable ability to quickly recognize and classify musical instruments, even in the presence of complex acoustic scenes. Through a series of psychophysical experiments, the authors demonstrate the fast and accurate nature of timbre-based recognition, particularly for voices. Their findings suggest the existence of efficient neural mechanisms that rely on selectivity to complex spectro-temporal signatures of sound sources, contributing to the understanding of how the brain processes and categorizes musical sounds.

Yang et al. (1992) focus their research on the auditory system's ability to encode and interpret a wide array of acoustic information, studying the auditory representations of acoustic signals. Their work delves into the transformations that occur in the early auditory system, providing insights into how sound is processed and represented in the brain. By proposing a mathematical formulation of these computations, the authors contribute to the understanding of the spectro-temporal modulations in the auditory spectrogram and their role in sound perception. Their research highlights the importance of considering both spectral and temporal aspects of sound in the development of computational models of auditory processing.

Kostek (2004) stands out in the field of musical instrument classification and duet analysis, applying music information retrieval techniques to automate sound recognition and categorization. The author presents a comprehensive approach to musical instrument classification, encompassing pitch extraction, parametrization, and pattern recognition. By employing artificial neural networks as a decision system, Kostek demonstrates the effectiveness of machine learning techniques in accurately classifying musical instruments. Furthermore, the

research extends to the separation of duet sounds, introducing the frequency envelope distribution (FED) algorithm for decomposing and analyzing musical duets. Kostek's work showcases the potential of music information retrieval techniques in advancing the field of automatic music analysis and classification.

Agostini, Longari, and Pollastri (2003) focus their research on the classification of musical instrument timbres using spectral features. The authors evaluate a set of features to recognize musical instruments from monophonic musical signals, aiming for a compact representation by limiting the number of descriptors to spectral characteristics. Various classification methods, including support vector machines, quadratic discriminant analysis, canonical discriminant analysis, and k-nearest neighbours, are implemented and tested on a dataset of 1007 tones from 27 instruments. The system achieves impressive results, with support vector machines and quadratic discriminant analysis showing comparable success rates close to 70% for individual instrument classification. The authors highlight that string instruments are the most challenging to classify, while brass and woodwind instruments yield more satisfactory results. The study also identifies the most relevant features for timbre classification, including inharmonicity, spectral centroid, and the energy contained in the first partial. Agostini et al.'s work demonstrates the effectiveness of machine learning methods in musical instrument classification using a limited set of spectral features. The study provides valuable insights into the relative importance of different spectral descriptors and the performance of various classification algorithms in the context of musical timbre recognition.

Krishna and Sreenivas (2004) address the problem of musical instrument recognition, extending their approach from isolated notes to solo instrumental phrases. The authors employ speech and audio processing techniques alongside statistical pattern recognition principles to develop a scalable system that does not rely on explicit temporal segmentation of musical phrases. The study proposes the use of Line Spectral Frequencies (LSFs) as robust features for musical instrument recognition, comparing their performance with Mel-Frequency Cepstral Coefficients (MFCCs) and Linear Predictive Cepstral Coefficients (LPCCs). Gaussian Mixture Models and K-Nearest Neighbour classifiers are used for classification. The experimental dataset includes the

University of Iowa's Musical Instrument Samples and the RWC database. The proposed system achieves impressive results, with best accuracies of around 95% at the instrument family level and 90% at the individual instrument level when classifying 14 instruments. The authors demonstrate the scalability of their approach by successfully classifying short segments of solo music using models built from isolated notes, achieving 74% accuracy in a forced 3-way classification task.

Krishna and Sreenivas's work highlights the potential of using frame-level features, such as LSFs, for robust musical instrument recognition across both isolated notes and solo phrases. The study showcases the effectiveness of statistical modelling techniques in capturing the timbral characteristics of musical instruments and the feasibility of extending isolated note models to solo music classification.

2.5.3 Multilabel Classification of Musical Instrument

Kenarsari-Anhari (2020) tackles the challenging task of multi-instrument recognition using the OpenMIC dataset, which contains weakly-labelled and partially-labelled polyphonic audio clips. The study addresses the scarcity of labelled data, a significant hurdle in applying deep learning to this domain. The OpenMIC dataset provides only the presence or absence of instruments for each clip (weak labels), while only a subset of instruments is labelled (partial labels).

To handle the weakly-labelled problem, Kenarsari-Anhari proposes an attention-based recurrent neural network that learns to attend to specific segments within an audio clip. This approach allows the model to focus on the most relevant parts of the signal for each instrument. Data augmentation techniques, such as mixup and concatenation of random examples, are employed to mitigate the partially-labelled issue by exposing the model to a more diverse set of examples.

The proposed model architecture combines a VGGish CNN for local feature extraction, a bidirectional LSTM layer for temporal modelling, and an attention layer for aggregating predictions across time. A modified focal loss function is introduced to address class imbalance by placing more emphasis on harder-to-classify instruments. The study demonstrates state-of-the-

art results on the OpenMIC dataset, highlighting the effectiveness of the attention mechanism and data augmentation strategies in handling weak and partial labels.

Kenarsari-Anhari's work underscores the importance of developing specialized techniques to overcome the limitations of available training data in music information retrieval tasks. The proposed attention-based recurrent neural network and data augmentation methods provide a promising approach for multi-instrument recognition in real-world scenarios where complete and strong labels are often unavailable.

Spyromitros-Xioufis, Tsoumakas, and Vlahavas (2011) present two multi-label learning approaches for musical instrument recognition, which were the winning solutions in the ISMIS 2011 contest on Music Information. The main challenge in this task was the heterogeneity and scarcity of the available training data, which consisted of a small dataset of instrument pair mixtures and a larger dataset of single instrument recordings.

Spyromitros-Xioufis et al.'s work showcases the potential of multi-label learning techniques in music information retrieval tasks, particularly in scenarios with limited and heterogeneous training data. The proposed solutions provide valuable insights into data preprocessing, model design, and post-processing strategies for tackling multi-instrument recognition challenges.

The work of Kenarsari-Anhari and Spyromitros-Xioufis, Tsoumakas, and Vlahavas contribute significantly to the field of multi-label musical instrument classification by addressing the challenges of weakly-labelled and partially labelled data, as well as the heterogeneity and scarcity of training examples. Kenarsari-Anhari (2020) proposes an attention-based recurrent neural network and data augmentation techniques to handle weak and partial labels, while Spyromitros-Xioufis et al. (2011) demonstrate the effectiveness of the Binary Relevance approach combined with strong base classifiers and data engineering strategies. Both studies highlight the importance of developing specialized techniques to overcome the limitations of available training data in music information retrieval tasks, providing valuable insights and methodologies for future research in this domain.

2.6 Challenges in Modern Instrument Recognition

The challenges encompass handling the complexity of polyphonic music, dealing with the inherent variability in instrument sounds, and addressing issues related to data scarcity and quality. Understanding and tackling these challenges are crucial for developing effective and robust instrument recognition systems.

2.6.1 Handling Polyphony

1) Complexity of Multiple Instruments:

One of the primary challenges in modern instrument recognition is the accurate identification of individual instruments in polyphonic music, where multiple instruments are playing simultaneously (Bosch et al., 2012). This complexity poses a significant challenge for pattern recognition and feature extraction. In polyphonic settings, sounds from different instruments can overlap, leading to masking effects where the presence of one instrument obscures the sound of another. This can make it difficult to isolate and identify individual instruments accurately (Vincent et al., 2006).

Addressing polyphony often involves sophisticated signal processing and machine learning techniques such as source separation, blind signal separation, and the use of deep neural networks capable of disentangling complex audio scenes (Chandna et al., 2017; Z. Huang et al., 2014).

2.6.2 Challenges of Variability and Diversity for Machine Learning

Musical instruments can produce a wide range of sounds. This variability in timbre, caused by differences in playing style, technique, or instrument construction, adds to the complexity of accurate instrument recognition (Herrera et al., 2000). The acoustic characteristics of a recording environment and the presence of background noise also contribute to the variability in instrument sounds, posing additional challenges in recognition tasks (Tzinis et al., 2019). Effective instrument recognition systems must be robust to these variations, requiring comprehensive and

diverse training datasets, along with algorithms capable of generalizing across different playing styles and recording conditions (Lostanlen et al., 2018).

2.6.3 Challenges of Data Scarcity

One of the significant hurdles in training machine learning models for instrument recognition is the scarcity of high-quality, labelled datasets. Annotated data, especially for less common instruments or specific genres, can be limited. The quality of audio recordings and the diversity represented in the datasets are crucial for training effective models. Poor quality or unrepresentative data can lead to models that do not perform well in real-world scenarios (Bogdanov et al., 2013).Strategies to address data scarcity and quality issues include data augmentation techniques (Schlüter & Grill, 2015), synthetic data generation, and semi-supervised or unsupervised learning approaches that can leverage unlabelled data.

In addition to these data and computational challenges, there is also the challenge of how and whether musical recognition models should be inspired by human perception of sound.

2.7 Human Perception and Machine Learning: Bridging the Gap

Human ears can separate different frequencies of sound along the basilar membrane (Klinke, 1987) in a manner that is somewhat analogous to how a spectrogram visualizes sound frequencies over time because the development of spectrograms was inspired by this biological process (Roberts & Mullis, 1987; Schafer & Rabiner, 1973), as they provide a visual representation of the frequency content of sounds over time, aiding in the analysis and understanding of auditory signals. This process is a fundamental aspect of how we hear and understand complex sounds and is central to the field of psychoacoustics (B. C. Moore, 2012).

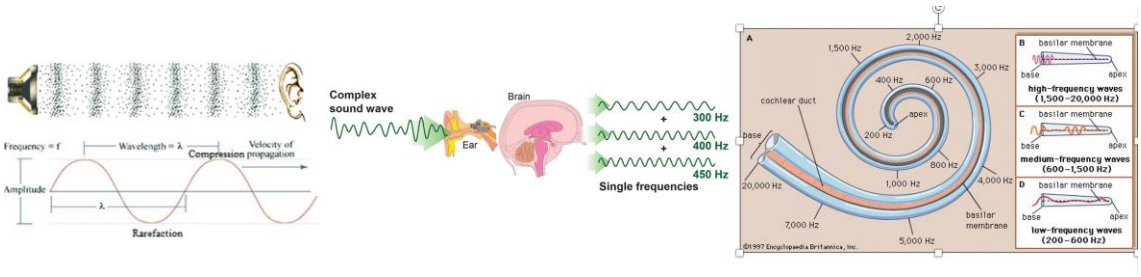


Figure 9. Human hearing mechanism.

The basilar membrane (Figure 9) within the cochlea of the inner ear acts as a sort of natural spectrogram. It is able to mechanically separate incoming sound waves into their component frequencies through a process known as tonotopic organization (Pickles, 2012). From the research "Processing of acoustic stimuli in the inner ear--a review of recent research results"(Klinke, 1987) we know how it works:

- High-frequency waves (1500-20000 Hz) are detected near the base of the cochlea. This area of the basilar membrane is stiffer and narrower, which makes it responsive to these higher frequencies.
- Medium-frequency waves (600-1500 Hz) are detected in the middle regions of the cochlea. The properties of the basilar membrane change progressively from base to apex, allowing it to resonate with medium frequencies at this location.
- Low-frequency waves (200-600 Hz) are detected near the apex of the cochlea. The apex is wider and more flexible, which enables it to respond to lower frequencies.

This tonotopic mapping ensures that different frequencies stimulate different hair cells along the basilar membrane, allowing the auditory system to distinguish sounds based on their frequency composition. The brain interprets these different stimulations as sounds of varying pitches.

In the context of machine learning and music information retrieval, using a spectrogram to represent audio signals is meaningful and mimics an aspect of human auditory processing (McFee et al., 2015). A spectrogram is a visual representation of the spectrum of frequencies in a sound or other signal as they vary with time. It effectively captures the time-varying spectral properties of audio signals, making it a powerful tool for analysing and processing sounds in a way that is somewhat analogous to how our ears and brain perceive and analyse sound (Boashash, 2015).

Therefore, according to the psychoacoustics research, we believe that by using spectrograms to analyse and recognise musical instruments or other audio features, our ML model is inspired by a method that parallels human auditory perception, although in a simplified and computational form. This approach is foundational in many applications involving sound analysis, including speech recognition (Hinton et al., 2012), music recommendation (Schedl et al., 2014), and environmental sound analysis (Piczak, 2015), among others.

Also, as noted earlier, humans perceive musical pitch in terms of the logarithm of frequency- rather than the frequency itself (Stevens et al., 1937; Stevens & Volkmann, 1940). That is the basis of adopting logarithmic scaling features. And it makes our spectrogram analysis meaningful because spectrogram is logarithm of frequency.

The book "An Introduction to the Psychology of Hearing" by Brian C.J. Moore (2012) also provides a comprehensive overview of psychoacoustics, including how humans perceive pitch and timbre. It explains the logarithmic scaling of pitch perception, which is part of the basis for the design of the following spectral features.

Clarke's work (Clarke, 2005) delves into the ecological approach to auditory perception, explaining how humans perceive and categorize sounds, including musical instruments, within their environment. This research is fundamental to understanding the cognitive processes involved in early manual methods of instrument recognition.

2.7.1 Human Instrument Recognition Mechanisms

When employing Convolutional Neural Networks (CNNs) to classify spectrograms for instrument recognition based on the literature presented from chapter 2.1 to chapter 2.6, the technique primarily leverages the capacity to recognize distinct sound characteristics and track changes over time within the audio. Spectrograms visually capture these elements, reflecting both the intensity and frequency spectrum of sounds as they evolve. Through this approach, CNNs learn to identify specific signatures associated with various musical instruments, mirroring the human ability to recognize different sounds based on subtle variations and temporal dynamics, though without directly mimicking the human ear's complex mechanisms for spatial or precise frequency-based separation. This process enhances the model's ability to differentiate between instruments by focusing on visual patterns within the spectrogram, highlighting the role of machine learning in capturing and interpreting complex auditory information.

Thus, understanding how humans perceive and categorize musical instruments provides insights into and inspiration for designing machine learning models for instrument recognition. The human auditory system has evolved to not only detect sound but also to interpret and categorize it in complex ways.

1) Cognitive processes in sound identification

Humans naturally perform auditory scene analysis where they can isolate and focus on specific sounds, such as a single instrument in an orchestra (Bregman, 1994). This cognitive ability to separate and identify sounds is a process that binary classifiers in machine learning aim to emulate. The human brain is adept at recognizing patterns in sound, such as the distinct timbre of instruments. This is similar to how a machine learning model learns to identify patterns from audio features to classify different instruments.

2) Role of timbre, pitch, and rhythm

Timbre is key in human identification of different instruments. It encompasses qualities of sound that enable distinction beyond pitch and loudness (McAdams, 2013; McAdams & Giordano, 2008). In machine learning, extracting features related to timbre allows models to differentiate

between instruments, akin to human auditory perception. Pitch and rhythm are also crucial in how humans perceive music. They contribute to the recognition of melodies and rhythms played by different instruments (Patel, 2008). Similarly, machine learning models can be trained to recognize these elements, further enhancing their classification capabilities.

3) Influence of learning and exposure

Just as humans learn to recognize instruments better with experience and exposure, machine learning models require training (chapter 2.4) on diverse datasets to improve their recognition accuracy. Humans adapt their auditory perception (chapter 2.1) based on memory and context, something that machine learning models mimic through training and adaptation to new data.

2.7.2 Comparison with Machine Learning Approaches

In this chapter, we compare the literature of acoustic which presented in chapter 2.1 and chapter 2.7.1 with machine learning algorithms.

1) Pattern recognition in AI

In artificial intelligence (AI) as noted in chapter 2.4, particularly in machine learning, pattern recognition involves the identification and classification of patterns (Bishop, 2006) within data. AI models for musical instrument recognition are trained to detect patterns in audio signals that correspond to different instruments.

Unlike the human brain, which relies on innate capabilities and experiential learning, AI models learn from large datasets (Hastie et al., 2009). These datasets contain various examples of sound patterns, which the models use to 'understand' and predict the characteristics of different instruments. AI models use algorithms to dissect and analyse sound, employing techniques like neural networks to mimic some aspects of human pattern recognition, albeit in a more structured and mathematical way.

2) Similarities in feature extraction

Both humans and AI systems rely on extracting key features from sound to identify instruments. Humans unconsciously process features like pitch, timbre, and rhythm to differentiate sounds,

while AI systems computationally extract similar features (Ellis, 2009). Spectral features (like frequency and harmonics) and temporal features (such as the duration and attack of notes) are crucial in both human and AI-based recognition processes (Tzanetakis & Cook, 2002). In AI, techniques like MFCCs and spectrograms serve a similar purpose to the human auditory system's feature extraction process (Logan, 2000).

3) Differences in contextual processing

Humans have a remarkable ability to use contextual information (like the type of music or the expected set of instruments) to aid in sound recognition (Bregman, 1994). AI models, however, typically lack this level of contextual understanding. The human auditory system is inherently adaptable and can learn from a minimal set of examples or even a single instance (Ericsson et al., 1993). In contrast, machine learning models require extensive data to learn and might not adapt as quickly to new or unseen types of sound environments (Goodfellow et al., 2016).

Thus, humans interpret music not just through the sound itself but also through emotional and cultural lenses. AI systems do not possess this level of interpretative understanding, focusing instead on objective data analysis

2.7.3 Justification for OvA Modelling

In this section, the justification for using the OvA approach is rooted in its alignment with human auditory processing capabilities, particularly in terms of specialization and selective attention. Additionally, the advantages of targeted recognition, scalability, and adaptability in the context of machine learning for musical instrument recognition are highlighted, underscoring the practical benefits and effectiveness of the OvA (Binary Classifier) strategy.

2.7.3.1 Mimicking human specialization

1) Selective Attention in Human Perception:

Humans have the ability to selectively focus on specific sounds in an environment, a process akin to the auditory scene analysis (Bregman, 1994). This ability to isolate a single sound source from

a complex auditory scene is similar to the OvA approach in machine learning, where each model is trained to recognize one specific instrument.

The concept of selective attention in human auditory perception is a fundamental principle in cognitive psychology. It refers to the ability to focus on a specific sound source while filtering out a range of other stimuli. This concept was extensively studied by cognitive psychologists like Donald Broadbent (2013) and Anne Treisman (1980).

Donald Broadbent's pioneering work, described in his book "Perception and Communication," laid the groundwork for understanding selective attention. He proposed the 'Filter Model' of attention, suggesting that humans selectively process information based on its physical characteristics.

Albert Bregman in his book "Auditory Scene Analysis: The Perceptual Organization of Sound." (Bregman, 1994). Bregman's work delves into how the auditory system separates and organizes sounds into perceptually meaningful elements, akin to the OvA approach's goal of isolating individual instruments from a complex audio mix.

2) Emulating Cognitive Processes:

The OvA strategy emulates this aspect of human cognition by creating individual classifiers for each instrument. Each classifier is trained to be highly specialized, akin to how humans can develop a keen ear for specific sounds or instruments with experience and practice.

The idea that the human brain can develop specialized skills, such as recognizing specific sounds or instruments, is supported by numerous studies in cognitive psychology and neuroscience (Kraus & Chandrasekaran, 2010; Patel, 2008; Peretz & Zatorre, 2003; Zatorre, 2003). For instance, research on expert musicians has shown enhanced auditory skills and brain adaptations for processing their instrument's sounds (e.g., "The musician's brain as a model of neuroplasticity," by Thomas et. al. (2002))

The OvA approach in machine learning mimics this aspect of human cognition by creating specialized models for each task (or instrument in this case). Each focused training on a

single instrument parallel how humans, through experience and practice, become attuned to specific auditory cues, enhancing their ability to distinguish and recognize particular sounds.

2.8 Summary

In this chapter we have reviewed previous work in musical instrument recognition including the use of spectrograms and deep learning models. We have also briefly described current understanding of how humans perceive music and may be able to distinguish between different musical instruments from a psychoacoustic perspective. We have identified the relatively under-explored problem of recognizing musical instruments in polyphonic music samples as worthy of further investigation. In the next chapter, we propose a method for tackling these problems given what we have learned from the literature review presented in this chapter.

Chapter 3. Methodology

Based on the literature review presented in the previous chapter, which examined current instrument recognition techniques, auditory perception theories, and recent advancements in deep learning for audio analysis, we have identified two significant research gaps. First, there is a persistent challenge in accurately recognizing individual instruments within polyphonic music, particularly due to overlapping and harmonically complex sounds. Second, existing models often face scalability issues when new instruments are introduced, typically requiring complete retraining. Addressing these gaps forms the foundation of our research methodology. Therefore, this chapter outlines our approach to investigating these challenges through the development and evaluation of a OvA model for musical instrument recognition. Our methodology aims to assess the OvA model's effectiveness in improving recognition precision for individual instruments in complex audio environments and its scalability when incorporating new instruments without retraining.

3.1 Research Gaps and Research Objectives

Despite numerous studies presented in the literature review, there remains a significant challenge in accurately recognizing individual instruments in polyphonic music. This complexity arises due to overlapping and harmonically complex sounds, which current models struggle to differentiate effectively. The hypothesis and objectives of this research gap are presented in this chapter.

3.1.1 Research Gaps, Hypothesis and Objectives.

3.1.1.1 Instrument Recognition in Clear Environments

Current research lacks a comprehensive evaluation of Binary Classifiers (OvA model) in identifying specific instruments from spectrogram representations in clear acoustic conditions.

- **Hypothesis one:** Binary Classifiers in an OvA model can accurately identify specific instruments from their spectrogram representations in a clear environment.

- **Objective one (RO-1):** To develop and evaluate an OvA model for instrument recognition, assessing its accuracy in identifying specific instruments from spectrograms in clear acoustic conditions.

3.1.1.2 Scalability of Binary Classifiers in Instrument Recognition

Current research lacks a comprehensive understanding of how Binary Classifiers (OvA model) perform when the number of instrument classes increases significantly.

- **Hypothesis 2:** Binary Classifiers in an OvA model can maintain high accuracy in discerning and correctly identifying musical instruments within spectrograms even as the number of instrument classes increases.
- **Objective 2:** To evaluate the scalability of the OvA model by:
    a. Progressively increasing the number of instrument classes in the dataset.
    b. Assessing the model's accuracy and computational efficiency as the number of classes grows.
    c. Identifying any performance bottlenecks or limitations that emerge with larger numbers of instrument classes on state-of-art benchmarks.
    d. Comparing the scalability of the OvA approach to other multi-class classification methods.

3.1.1.3 Performance Degradation Under Noise Conditions

The specific noise conditions that lead to performance degradation in OvA models for instrument recognition are not well documented.

- **Hypothesis three:** The performance of OvA models for instrument recognition degrades under specific noise conditions, which can be identified and characterized.
- **Objective three:** To systematically evaluate the OvA model's performance under various noise conditions, identifying and characterizing the specific conditions that lead to performance degradation.

3.1.1.4 Instrument Recognition in Polyphonic Music

The effectiveness of Binary Classifiers (OvA model) in recognizing multiple instruments in polyphonic music, including both synthetic and real-world recordings, is not fully explored.

- **Hypothesis four:** Binary Classifiers in an OvA model can recognize each instrument present within samples of both synthetic and real-world audio recordings of polyphonic music, where multiple instruments are played simultaneously.
- **Objective four:** To evaluate the OvA model's ability to identify multiple instruments in polyphonic music samples, using a dataset that includes both synthetic and real-world audio recordings.

3.1.1.5 Optimal Spectrogram Algorithms for Different Instruments

There is a lack of comprehensive comparison of different spectrogram algorithms' effectiveness for various musical instrument types.

- **Hypothesis five:** Different spectrogram algorithms are more suitable for identifying certain types of musical instruments, with varying performance across instrument categories.
- **Objective five:** To compare and evaluate the performance of various spectrogram algorithms (e.g., STFT, Mel, CQT) in identifying different types of musical instruments, determining the most effective algorithm(s) for each instrument category.

In detail, the Mathematical hypothesis of the instrument classification model lies in the fusion of two key theorems: the Fourier Transform, which forms the basis for spectrograms, and the Universal Approximation Theorem (Park & Sandberg, 1991), which underlies the power of artificial neural networks (including CNNs) built upon Multi-Layer Perceptron's (MLPs)(Rumelhart et al., 1986). By combining these two theorems, we can develop a understanding of the model's architecture and capabilities.

To delve deeper into the mathematical details, we will expand equation by incorporating the specific operations performed within each layer of the neural $\mathrm{MLP}(x) = \mathrm{ReLU}(xW_1 +$

$b_1)W_2 + b_2$ network. The expanded network architecture consists of multiple layers, each utilizing Rectified Linear Unit (ReLU) activations.

First layer: $$H^{(1)} = \text{ReLU}(XW_1 + b_1)$$

$$Y^{(1)} = H^{(1)}W_2 + b_2$$

Second layer: $$H^{(2)} = \text{ReLU}\left(Y^{(1)}W_3 + b_3\right)$$

$$Y^{(2)} = H^{(2)}W_4 + b_4$$

$$\vdots$$

$n - th$ layer: $$H^{(n)} = \text{ReLU}\left(Y^{(n-1)}W_{2n-1} + b_{2n-1}\right)$$

$$Y^{(n)} = H^{(n)}W_{2n} + b_{2n}$$

- $H(i)$: The hidden layer outputs at the $i - th$ layer.
- $Y(i)$: The output of the network at the $i - th$ layer.
- X: The input data matrix.
- W: The weight matrix for the $i - th$ layer.
- b: The bias vector for the $i - th$ layer.
- ReLU: The Rectified Linear Unit activation function, which is defined as $\text{ReLU}(x) = \max(0, x)$

Thus, a neural network N with sufficient capacity can approximate any continuous function. The network takes the combined feature vector $F(t)$ as input: $y = \mathcal{N}(\boldsymbol{F}(t); \boldsymbol{W})$

where $\boldsymbol{W}$. represents the weights and biases of the neural network.

Given an example of STFT spectrogram, the math of it is $y = \mathcal{N}(\text{STFT}(t); \boldsymbol{W})$ to achieve our task of musical instrument classification. This approach essentially represents an approximation based on a universal approximation theorem using the STFT equation.

Also, if we use y to classify the music audio based on six features, the equation is:

$y = \mathcal{N}(\text{STFT}(t), \text{Log-Mel}(t), \text{MFCC}(t), \text{Chroma}(t), \text{Spectral Contrast}(t), \text{Tonnetz}(t); \boldsymbol{W})$.

This fusion of six spectrogram equations is expected to enhance the model's performance by leveraging the complementary information each feature provides. Combining these different spectrogram features allows us to capture a more comprehensive representation of the audio signal. The STFT captures time-varying frequency content, which is crucial for transient sounds

and rapidly changing harmonics. The Log-Mel spectrogram provides a perceptual scaling of frequency, capturing the essence of human auditory perception. MFCCs offer a compact representation of the spectral envelope, essential for distinguishing the timbre of various instruments. Chroma features highlight harmonic and pitch class relationships, useful for instruments with prominent harmonic content. Spectral contrast measures the difference in amplitude between peaks and valleys, capturing the relative levels of harmonics and formants, which is vital for complex timbres. Tonnetz features represent harmonic relationships, making them effective for instruments that play chords and have strong harmonic progressions.

3.1.1.6 Feature Extraction and Visualization

The specific features extracted by convolutional layers for each instrument and their potential for visualization and quantification are not well understood.

- **Hypothesis 6:** The features extracted from the convolutional layer for each instrument can be visualized and quantified, providing insights into the model's recognition process.
- **Objective 6:** To extract, visualize, and quantify the features from the convolutional layers for each instrument, developing methods to interpret these features and gain insights into the model's decision-making process.

3.1.1.7 Summary

The rest of this chapter section presents the methodologies employed in the research for musical instrument recognition using the OvA approach. It details the systematic procedures and techniques utilized in every stage of the model development, from initial data (synthetic data and real audio recording) acquisition to the final implementation of the classifiers. This includes an in-depth exploration of the data collection process, emphasizing the strategies for compiling a diverse and representative dataset, balancing sample classes, and selecting appropriate sources and types of audio data.

chapter 3.3.1 delves into the feature extraction techniques pivotal in transforming raw audio input into informative data for the machine learning models. Here, we describe the processes of converting audio files into digital waveforms, generating spectrograms for time-

frequency analysis, and applying advanced feature extraction methods tailored to capture the unique characteristics of different musical instruments.

In the core of chapter 3.1 Prototype Experiment, the architecture design of the OvA models is proposed, highlighting the rationale behind selecting specific neural network structures, layer configurations, and their functions, as well as the customization process for individual instruments. This is complemented by a detailed description of the training and validation strategies and methods, and techniques employed to prevent overfitting and ensure model robustness.

Additionally, the experimentation and model training subsections provide a step-by-step account of the experimental setup, the training process, including hyperparameter tuning and optimization, and the challenges encountered during model training. This part of the methodology aims to offer transparency and replicability in the research process, ensuring that the findings are grounded in rigorous and well-documented procedures. Lastly, chapter 3.3.3 section outlines the model evaluation criteria, discussing the selection of relevant metrics to assess the performance of the models accurately. This includes a consideration of both traditional evaluation metrics and those specific to multi-label classification scenarios, addressing the unique challenges posed by polyphonic music recognition.

3.2 Iterative Experimental Methodology

To answer the research questions (chapter 1.2.2) and achieve the objectives (chapter 3.3.1), the methodology we are using is "Iterative Experimental Methodology" (George et al., 2005; Thomke & Matters, 2003; Zimmerman et al., 2007). It includes

1. Initial feasibility assessments,
2. Detailed experiments on varied datasets,
3. Comparative analyses with existing models,
4. Robustness tests under different conditions (e.g., noise interference),
5. Evaluations on synthesized and real-world data,
6. Continuous refinement of models based on experimental outcomes.

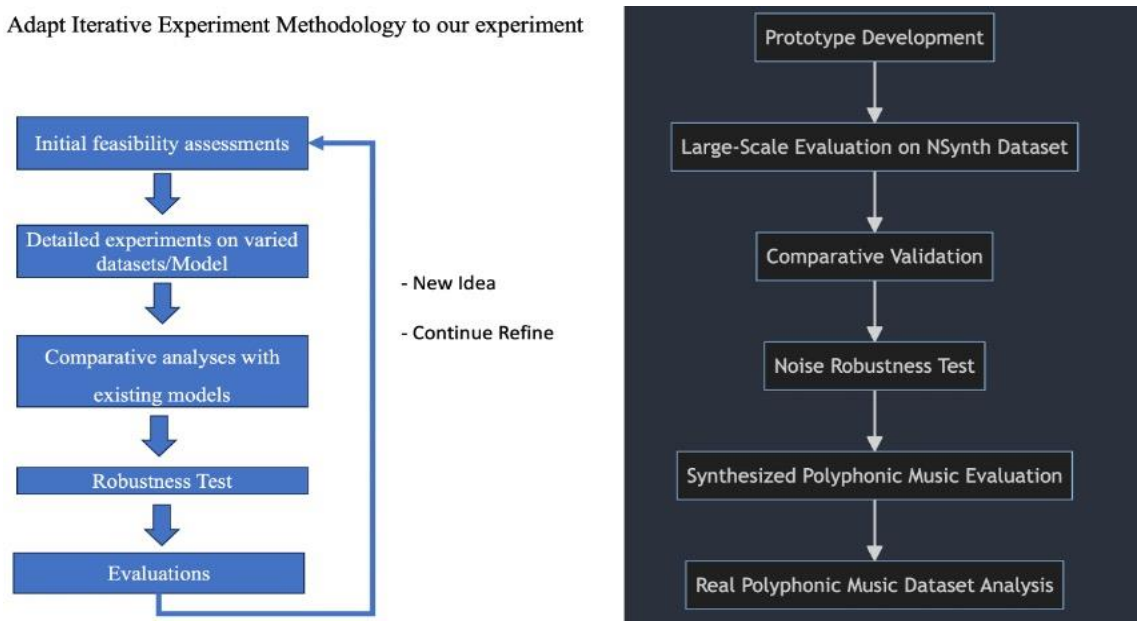


Figure 10. Adapt Iterative Experiment Methodology to Our Model Design.

We will conduct a series of experiments according to Figure 10, though iterative experimental methodology (George et al., 2005; Thomke & Matters, 2003; Zimmerman et al., 2007). This methodology allows us to systematically investigate the feasibility and effectiveness of using CNN-based binary classifiers for instrument identification in polyphonic music which is presented in chapter 3.3.1 By breaking down the research into well-defined steps, we aim to provide a detailed understanding of the process for readers who may be unfamiliar with this area of study.

George et al.'s book, "Statistics for Experimenters: Design, Innovation, and Discovery" (2005), emphasizes the importance of iterative experimentation in the process of scientific discovery and innovation. The authors argue that iterative experiments allow researchers to systematically explore and optimize processes, leading to more effective and efficient solutions. They highlight the role of statistical methods in designing and analysing iterative experiments, enabling researchers to make data-driven decisions and draw meaningful conclusions. In their book, George et al. discuss the concept of sequential experimentation, where each experiment builds upon the findings of the previous one. This iterative approach allows researchers to refine their hypotheses, adjust experimental parameters, and continuously improve their understanding of the system under study. The authors provide practical guidelines and case studies

demonstrating how iterative experimentation can be applied in various fields, such as manufacturing, engineering, and research and development.

Thomke & Matters, in their book "Experimentation Matters: Unlocking the Potential of New Technologies for Innovation" (2003), also emphasize the crucial role of experimentation in driving innovation and technological advancement. They argue that iterative experimentation is essential for understanding and exploiting the potential of new technologies. Thomke & Matters highlight the importance of rapid experimentation cycles, where insights from each experiment are quickly incorporated into the next iteration. This agile approach enables researchers and innovators to learn from failures, adapt to changing circumstances, and converge towards optimal solutions. The authors provide numerous examples from industries such as pharmaceuticals, aerospace, and software development, illustrating how iterative experimentation has led to groundbreaking innovations and improved product development processes.

Zimmerman et al., in their paper "Research through Design as a Method for Interaction Design Research in HCI" (2007), discuss the application of iterative experimental methodology in the context of human-computer interaction (HCI) research. They propose research through design approach that involves iterative cycles of design, implementation, and analysis. In Zimmerman's opinion, iterative experimentation is particularly valuable in HCI research, where the goal is to create interactive systems that meet user needs and provide effective user experiences. By iteratively designing, prototyping, and evaluating interactive systems, researchers can gain insights into user behaviour, preferences, and usability issues. This iterative process allows for the refinement of design concepts, the identification of potential problems, and the incorporation of user feedback into subsequent iterations.

3.2.1 First iteration - Step 1: Instrument Identification in Polyphony

The first step in our iterative experiment methodology is to assess the feasibility of using CNN-based binary classifiers to identify musical instruments within polyphonic compositions (objective one in chapter 3.3.1.1). In polyphonic music, multiple instruments play simultaneously, creating a complex auditory landscape. Our goal is to determine whether binary classifiers can

effectively distinguish individual instruments within this intricate musical context (objective one in chapter 3.3.1.1).

To achieve objective one (chapter 3.3.1.1), we will conduct a prototype experiment where we train individual binary classifiers to recognize specific instruments within polyphonic musical pieces. Each classifier will be designed to identify the presence or absence of a particular instrument, such as a violin, piano, or flute. By focusing on a limited set of instruments initially, we can evaluate the classifiers' performance in a controlled setting.

We will curate a dataset of targeted instruments (objective one in chapter 3.3.1.1). The dataset will be carefully annotated to indicate the presence or absence of each instrument. In addition. We will extract relevant features from the audio data using techniques such as STFT spectrograms, or other suitable representations. These features will capture the unique characteristics of each instrument. It includes,

- **Classifier training:** We will train individual CNN-based binary classifiers using the extracted features. Each classifier will be optimized to recognize a specific instrument within the polyphonic context. We will employ techniques such as data augmentation and regularization to improve the classifiers' generalization capabilities.
- **Evaluation:** We will evaluate the trained classifiers using a separate test dataset. The classifiers' performance will be measured using metrics such as precision and recall rates. Precision measures the proportion of correctly identified instrument instances among all positive predictions, while recall measures the proportion of correctly identified instrument instances among all actual instances of the instrument.

By conducting this prototype experiment, we expect to establish a baseline understanding of the feasibility of using CNN-based binary classifiers for instrument identification in polyphonic music (Objective one in chapter 3.3.1.1). The evaluation metrics, such as precision and recall rates, will provide insights into the classifiers' ability to accurately identify specific instruments within complex musical compositions. This step will lay the foundation for further experiments and refinements in the subsequent steps of our iterative methodology.

3.2.2 First iteration - Step 2: Large-Scale Evaluation on NSynth Data

Upon successfully assessing the feasibility of using CNN-based binary classifiers for instrument identification in polyphonic music, the next step is to extend the study to larger and more complex datasets. The objective two (chapter 3.3.1.2). of this step is to evaluate the scalability and robustness of the binary classifier approach when applied to a diverse range of musical instruments and compositions.

In this step, we will expand our experiment to utilize the NSynth dataset, which is a large-scale dataset of musical notes synthesized from various instruments to address the objective two (chapter 3.3.1.2). The NSynth dataset provides a rich and varied collection of instrumental sounds, allowing us to test the binary classifiers' performance on a broader spectrum of instruments and musical contexts.

The objective two, large-scale evaluation on the NSynth dataset will involve the following steps:

1. **Data preparation:** We will preprocess the NSynth dataset to extract relevant features and organize the data into suitable formats for training and testing the binary classifiers. The dataset will be divided into different instrument families, such as strings, brass, woodwinds, and percussion, to enable focused evaluation of the classifiers' performance on each family.
2. **Classifier training:** We will train a set of 10 binary classifiers, each dedicated to recognizing a specific instrument family. The classifiers will be trained using the pre-processed NSynth data, leveraging the knowledge and insights gained from the prototype experiment in Step 1. We will fine-tune the classifiers' architectures and hyperparameters to optimize their performance on the larger dataset.
3. **Evaluation:** We will evaluate the trained classifiers using a held-out portion of the NSynth dataset. The classifiers' performance will be assessed based on metrics such as precision, recall, and F1 score. These metrics will provide a understanding of the classifiers' ability to accurately identify instruments within each family.

4. **Refinement:** Based on the evaluation results, we will analyse the classifiers' performance and identify areas for improvement. We will refine the training strategies, such as adjusting the network architectures, applying data augmentation techniques, or incorporating domain-specific knowledge, to enhance the classifiers' accuracy and robustness.

By conducting a large-scale evaluation on the NSynth dataset, we expect to gain insights into the scalability and generalization capabilities of the CNN-based binary classifiers for instrument identification to answer the research question two (chapter 1.2.2). The evaluation metrics will provide a quantitative measure of the classifiers' performance on a diverse range of instruments and musical compositions. This step will assess the effectiveness of the binary classifier approach and identify potential challenges or limitations that need to be addressed in subsequent experiments. The refined training strategies and insights gained from this step will inform the design and implementation of the comparative Testing and further experiments in the iterative methodology.

3.2.3 First iteration - Step 3: Comparative Testing

After evaluating the performance of the CNN-based binary classifiers on a large-scale dataset, the next step is to test their effectiveness by comparing them against a traditional single CNN model to address the research objective two (chapter 3.1.1.2) in comparison. The objective of this comparative testing is to investigate whether using an ensemble of binary classifiers offers any advantages over a single CNN model in terms of instrument identification accuracy and computational efficiency.

In this step, we will conduct a comparative analysis between the binary classifier ensemble and a single CNN model. Both approaches will be trained and evaluated on the same dataset, allowing for a fair comparison of their performance. The single CNN model will be designed to identify multiple instruments simultaneously, while the binary classifier ensemble will consist of individual classifiers, each dedicated to identifying a specific instrument. To address the research objective two (chapter 3.1.1.2) The comparative testing will involve the following steps:

1. **Single CNN model development:** We will design and implement a single CNN model capable of identifying multiple instruments within polyphonic music. The model's architecture will be carefully crafted to capture the relevant features and patterns necessary for accurate instrument recognition. We will train the single CNN model using the same dataset used for evaluating the binary classifiers.
2. **Binary classifier ensemble preparation:** We will prepare the ensemble of binary classifiers developed in the previous steps. Each classifier will be specialized in identifying a specific instrument, and their outputs will be combined to provide the final instrument identification results.
3. **Performance evaluation:** We will evaluate both the single CNN model and the binary classifier ensemble on a common test dataset. The evaluation metrics, such as precision, recall, and F1 score, will be calculated for each approach. Additionally, we will measure the computational efficiency of both methods, considering factors like training time, inference time, and resource requirements.
4. **Comparative analysis:** We will analyse the evaluation results to compare the performance of the binary classifier ensemble against the single CNN model. We will examine the trade-offs between accuracy and computational efficiency, as well as the strengths and limitations of each approach in handling different instrument types and musical complexities.

Through the comparative testing, we expect to gain insights into the relative performance of the binary classifier ensemble and the single CNN model for instrument identification in polyphonic music. By comparing their accuracy metrics and computational efficiency, we can determine whether using an ensemble of specialized binary classifiers offers any advantages over a traditional single CNN model. This step will highlight the potential benefits or limitations of the binary classifier approach and provide guidance for further refinements and optimizations in subsequent experiments. The comparative analysis will contribute to a deeper understanding of the most effective strategies for instrument identification in complex musical contexts and address the research objective two (chapter 3.1.1.2).

3.2.4 First iteration - Step 4: Noise Robustness Test

In real-world scenarios, musical recordings often contain various types of noise, such as background sounds, recording artifacts, or environmental disturbances. The objective three (chapter 3.1.1.3) of this step is to investigate the robustness of the CNN-based binary classifiers in extracting instrumental features and accurately identifying instruments in the presence of noise.

In this step, we will evaluate the performance of the binary classifiers under noisy conditions. We will introduce different levels and types of noise to the musical datasets used in previous experiments and assess how well the classifiers maintain their instrument identification accuracy. By testing the classifiers' noise robustness, we can determine their reliability and practicality for real-world applications.

The noise robustness test that can address the objective three (chapter 3.1.1.3) will involve the following steps:

1. **Noise generation:** We will generate different types of noise, such as white noise, pink noise, or environmental sounds, at various signal-to-noise ratios (SNRs). These noise samples will be carefully selected to represent realistic scenarios encountered in musical recordings.
2. **Dataset augmentation:** We will augment the existing musical datasets by adding the generated noise samples to the audio recordings. The noise will be added at different SNRs to create a range of noisy conditions. This augmentation process will expand the dataset and provide a more challenging test environment for the binary classifiers.
3. **Classifier evaluation:** We will evaluate the performance of the binary classifiers on the noise-augmented dataset. The classifiers will be tasked with identifying instruments in the presence of varying levels of noise. We will measure the classifiers' accuracy, precision, recall, and F1 score under different noise conditions.
4. **Threshold adjustment:** Based on the evaluation results, we will analyze the impact of noise on the classifiers' performance. We will identify the noise thresholds at which the classifiers' accuracy starts to degrade significantly. By understanding these thresholds,

we can adjust the models' noise tolerance levels and develop strategies to mitigate the effects of noise on instrument identification.

Through the noise robustness test, we expect to address research objective three (chapter 3.1.1.3) that gain insights into the binary classifiers' ability to extract instrumental features and accurately identify instruments in the presence of noise. The results of this step will provide necessary information for refining the classifiers' noise handling capabilities and developing techniques to enhance their robustness. The evaluation metrics, such as Precision, Recall, F1 Score, and Exact Match Ratio, should be calculated for each test scenario to quantify the impact of noise and distortions on the classifiers' performance that can address research objective 3 (chapter 3.1.1.3) statistically.

We now go into the specific details how our methodology will satisfy the research aims and objectives as listed in chapter 3.1 above.

3.2.5 First iteration - Step 5: Synthesized Polyphonic Music Evaluation

To address the objective four (chapter 3.1.1.4), the model performance should be quantified on synthesized polyphonic music as a start. Synthesized music allows for greater control over the complexity and composition of the musical pieces, enabling the evaluation of the classifiers' ability to handle intricate polyphonic structures. The objective of this step is to examine the classifiers' recognition accuracy on synthesized polyphonic music that mimics real-world complexity.

In this step, we will generate a dataset of synthesized polyphonic music specifically designed to test the binary classifiers' performance. The synthesized music will be created using advanced music synthesis techniques, such as MIDI-based composition or algorithmic music generation. The synthesized pieces will incorporate various instruments, harmonies, and rhythmic patterns to simulate the complexity found in real-world polyphonic music.

The synthesized polyphonic music evaluation will involve the following steps:

1. **Music synthesis:** We will utilize music synthesis software or algorithms to generate a diverse range of polyphonic compositions. The synthesized music will include different

combinations of instruments, varying levels of polyphony, and different musical styles. The synthesis process will be carefully controlled to ensure the generated pieces are representative of real-world polyphonic music.

2. **Dataset creation:** The synthesized polyphonic compositions will be organized into a structured dataset suitable for evaluation. Each composition will be annotated with ground truth labels indicating the presence and timestamps of specific instruments. The dataset will be divided into training, validation, and testing subsets to facilitate an evaluation of the classifiers.
3. **Classifier training and evaluation:** The binary classifiers will be trained on the synthesized polyphonic music dataset using the same training strategies employed in previous steps. The classifiers' performance will be evaluated on the testing subset, measuring metrics such as accuracy, precision, recall, and F1 score. The evaluation will provide insights into the classifiers' ability to recognize individual instruments within the complex synthesized polyphonic compositions.
4. **Analysis and refinement:** Based on the evaluation results, we will analyze the classifiers' performance on the synthesized music dataset. We will identify any challenges or limitations in handling the complexity of the synthesized polyphonic compositions. If necessary, we will refine the classifiers' architectures, training strategies, or feature extraction techniques to improve their recognition accuracy on synthesized music.

The evaluation metrics will provide quantitative measures of the classifiers' performance on synthesized music, allowing for a comparison with their performance on real musical recordings. This step will help identify any gaps or limitations in the classifiers' capabilities and guide further improvements in their design and training. The insights gained from the synthesized polyphonic music evaluation will contribute to the development of more robust and versatile instrument identification systems that can effectively handle both real and synthesized musical compositions. The refinements made based on this evaluation will be carried forward to the subsequent steps in the iterative methodology.

3.2.6 First iteration - Step 6: Real Polyphonic Music Dataset Analysis

To address the objective four (chapter 3.1.1.4), the model performance should also be quantified on real polyphonic music recordings. Recordings represent the ultimate test for instrument identification systems, as they involve numbers of instruments playing simultaneously in a complex and dynamic musical environment. The objective of this step is to evaluate the real-world applicability of the binary classifiers by evaluating their ability to identify individual instruments within real audio.

The recordings include a wide range of instruments such as strings, woodwinds, brass, and percussion. The classifiers will be tasked with identifying and localizing individual instruments within these complex musical compositions.

The real Polyphonic Dataset analysis will involve the following steps:

1. **Dataset acquisition:** We will acquire a dataset of high-quality recordings from Open-Source Music Databases such as Open-MIC. The recordings cover a diverse range of musical pieces, composers, and styles to ensure an evaluation. The dataset will be carefully curated to include accurate annotations of the instruments present in each recording.
2. **Data preprocessing:** The dataset will be pre-processed to extract relevant features and prepare them for input to the binary classifiers. This may involve techniques such as audio segmentation, noise reduction, and feature extraction. The preprocessing steps will be optimized to handle the specific characteristics of audio recordings, such as the presence of reverberation and the wide dynamic range of instruments.
3. **Classifier evaluation:** The binary classifiers will be applied to the pre-processed audio recordings to identify and localize individual instruments. The classifiers' performance will be evaluated using metrics such as precision, recall, and F1 score for each instrument category. Additionally, we will assess the classifiers' ability to accurately estimate the temporal boundaries of instrument activity within the recordings.
4. **Performance analysis:** Based on the evaluation results, we will analyse the binary classifiers' performance on real recordings. We will examine their ability to handle the

complexity and diversity of music, considering factors such as the number of instruments, the presence of overlapping instrument sounds, and the dynamic range of the recordings. We will identify any challenges or limitations in applying the classifiers to real-world recordings and propose strategies for improvement.

By evaluating the binary classifiers on real-world recordings, we expect to evaluate their real-world applicability and assess their performance in complex musical environments. The evaluation metrics will provide a quantitative measure of the classifiers' ability to identify and localize individual instruments within live audio sample. This step will shed light on the classifiers' robustness and reliability in handling the intricacies of real-world musical performances. The insights gained from this analysis will inform the refinement and optimization of the instrument identification system for practical applications in music information retrieval, audio analysis, and musicological research. The real audio recording analysis will serve as a crucial evaluation step, demonstrating the potential of the CNN-based binary classifiers to support various real-world use cases in the field of music technology.

3.2.7 Next Round of Iteration - Step 7: Refine the Model on Real Audio Recording

In the next iteration of our research, if the step 6 cannot achieve a reasonable result, we will conduct another round of iteration, to discover more efficient models, we will focus on testing and refining our CNN-based binary classifiers using the Open-MIC dataset (E. Humphrey et al., 2018). This dataset comprises real audio recordings, presenting new challenges and opportunities compared to NSynth datasets (Engel, Resnick, Roberts, Dieleman, Norouzi, et al., 2017), used previously. We will begin by acquiring and preprocessing the OpenMIC dataset. The preprocessing phase will involve annotating the dataset to indicate the presence or absence of specific instruments at various time intervals, ensuring a well-defined and accurate dataset for our experiments.

In new iteration, we will extract relevant features from the audio data from other spectrograms, mel-frequency cepstral coefficients (MFCCs), and other suitable representations.

These features will capture the unique characteristics of each instrument, providing a solid foundation for the training phase. We will then train the CNN-based binary classifiers using the extracted features from the Open-MIC dataset. To enhance the classifiers' generalization capabilities, we will apply data augmentation and regularization techniques during training.

We may also use new model or new architecture to refine our model in this new iteration. After training the classifiers, we will evaluate their performance using a separate test set from the Open-MIC dataset. The performance will be measured using precision, recall, and F1 score metrics, providing an assessment of the classifiers accuracy. Based on the evaluation results, we will refine the classifiers by adjusting network architectures, training strategies, or incorporating additional domain-specific knowledge. This iterative refinement process aims to improve the classifiers' accuracy and robustness, ensuring their effectiveness in real-world applications.

3.2.8 Next Round of Iteration - Step 8 Summary of All Iterations and Heatmap/Feature Map Analysis

In this final step, we will address the objective six (chapter 3.1.1.6) summarize the findings from all iterations and perform a detailed heatmap and feature map analysis to open the "black box" of our CNN-based binary classifiers. This analysis will help us understand how the classifiers make decisions and identify the most influential features for instrument recognition.

We will begin by generating heatmaps and feature maps for each classifier to visualize the areas of the audio spectrograms that influence their decisions. By analysing these visualizations, we will identify patterns and features critical for instrument identification, examining how the classifiers differentiate between instruments and handle overlapping sounds in polyphonic music. This analysis will provide necessary insights into the classifiers' decision-making processes and highlight the strengths and limitations of our approach.

The summary of iterations and heatmap/feature map analysis will offer an understanding of the performance and decision-making processes of our CNN-based binary classifiers. This step will reveal the most effective strategies for instrument identification and identify areas for further improvement. The insights gained from this analysis will contribute to the development of more

accurate and reliable music information retrieval systems, advancing the field of music technology research and informing future research and practical applications.

Eventually, al the six research objectives (chapter 3.1.1) are all quantified sequentially by this interactive experiment research methodology

3.3 Epistemological Framework and Methodological Paradigms in Deep Learning Research

To conduct deep learning research effectively, we must adhere to the "Epistemological Framework and Methodological Paradigms in Deep Learning Research" which includes,

1) Data Acquisition and Preprocessing (García et al., 2015).
2) Dataset Partitioning (Reitermanova & others, 2010).
3) Experimental Model Design and (Goodfellow et al., 2016; Kohavi, 1995).
4) Hyperparameter Optimization (Bergstra & Bengio, 2012; LeCun et al., 2015) .
5) Performance Metrics and Analytical Methods (Saito & Rehmsmeier, 2015).
6) Analytical Techniques for Result Interpretation (Qi et al., 2019; Zeiler & Fergus, 2014a).

3.3.1 Dataset Acquisition and Preprocessing

The first step in evaluating the performance of binary classifiers on synthetic audio and real recordings is to collect a diverse and representative dataset. This dataset should encompass a wide range of recordings, featuring different composers, musical styles, and recording conditions. To ensure the comprehensiveness of the evaluation, it is essential to include recordings that capture the nuances and complexities of musical instruments, such as the benchmark datasets like NSynth (Engel, Resnick, Roberts, Dieleman, Norouzi, et al., 2017), IRMAS (Instrument Recognition in Musical Audio Signals) (Bosch et al., 2018), and MIR-1K(Hsu & Jang, 2009) are crucial for evaluating and comparing different methodologies. Open-MIC is an open dataset of multilabel classification (E. Humphrey et al., 2018) These datasets provide a standard ground for testing and help in pushing the boundaries of current technologies.

3.3.2 Training and Evaluation

To assess the performance of binary classifiers on real recordings, it is necessary to establish an evaluation framework that takes into account the nuances and complexities of music. This framework should consider various aspects of instrument detection, including the ability to accurately identify the presence or absence of each instrument, the precision of onset and offset detection, and the handling of false positives and false negatives.

When setting up the evaluation framework, it is important to select appropriate metrics that provide an assessment of the classifiers' performance. Commonly used metrics in instrument recognition tasks include Precision, Recall, F1 Score, and Exact Match Ratio. Precision measures the proportion of correctly identified instrument instances among all positive predictions, while Recall measures the proportion of correctly identified instrument instances among all actual instances of the instrument. The F1 Score provides a balanced measure by combining Precision and Recall. Exact Match Ratio evaluates the classifiers' ability to correctly identify the presence and temporal boundaries of each instrument within the recordings.

In addition to these metrics, the evaluation framework should also consider the specific challenges posed by recordings, such as the presence of multiple instruments playing simultaneously and the potential for overlapping or ambiguous instrument sounds. The framework should be designed to assess the classifiers' robustness in handling these scenarios and their ability to accurately distinguish between similar-sounding instruments within the audio sample. By establishing an evaluation framework, we can obtain a detailed understanding of the strengths and limitations of binary classifiers in recognizing instruments within real recordings.

3.3.3 Testing the Deep Learning Model

With the evaluation framework in place, the next step is to test each binary classifier against the acquired dataset of real recordings. This performance testing phase involves applying the classifiers to each recording and documenting their responses to different instruments.

During the testing process, it is important to carefully observe and record how the classifiers respond to various scenarios within the recordings by evaluating their ability to

accurately identify the presence or absence of each instrument. The testing should cover a diverse range of musical passages, including solo sections, ensemble playing, and complex polyphonic textures.

As the classifiers process each recording, their responses should be documented, including the predicted instrument labels, onset and offset times, and any additional relevant information. This documentation will serve as the basis for subsequent analysis and evaluation of the classifiers' performance.

It is crucial to test the classifiers on a sufficiently large and representative subset of the dataset to obtain reliable and generalizable results. The testing should cover a wide range of music styles and recording conditions to assess the classifiers' robustness and adaptability. Throughout the performance testing phase, any notable observations or patterns in the classifiers' responses should be recorded. This may include identifying particularly challenging scenarios where the classifiers struggle to accurately identify instruments or highlighting instances where the classifiers excel in distinguishing between similar-sounding instruments within the audio.

By testing each binary classifier against the dataset and documenting their responses, we can gather insights into their performance and identify areas for further analysis and improvement. This testing approach is essential for evaluating the effectiveness of binary classifiers in recognizing instruments within real recordings and understanding their strengths and limitations in this complex musical context.

3.3.4 Analysis of Results

After conducting the performance testing of the binary classifiers on the real recordings, the next step is to analyse the results in detail. This analysis phase involves examining the classifiers' responses and identifying patterns, strengths, and weaknesses in their performance.

One aspect of the analysis is to evaluate the robustness of each classifier in distinguishing between similar-sounding instruments within the audio sample. Music sample often features instruments with overlapping frequency ranges and timbral characteristics, making it challenging to accurately identify and separate them. By assessing how well the classifiers handle these

scenarios, we can gain insights into their ability to capture subtle differences and make precise instrument classifications.

The analysis should also focus on identifying any scenarios or musical passages where the classifiers consistently perform well or struggle. This may involve examining the classifiers' responses to specific instrument combinations, dynamic ranges, or playing techniques. By pinpointing these patterns, we can better understand the limitations and strengths of the binary classifier approach in the context of music.

Furthermore, the analysis should consider the overall accuracy and reliability of the classifiers across the entire dataset. This involves calculating and interpreting the evaluation metrics such as Precision, Recall, F1 Score, and Exact Match Ratio. These metrics provide quantitative measures of the classifiers' performance and help in comparing their effectiveness against other approaches or benchmark results.

In addition to the quantitative analysis, qualitative observations and insights should also be documented. This may include noting any specific challenges encountered during the testing process, such as ambiguous instrument sounds or complex polyphonic textures that posed difficulties for the classifiers. These observations can provide necessary information for future improvements and refinements of the instrument recognition system.

The analysis of results should be thorough and systematic, taking into account various aspects of the classifiers' performance. It should aim to uncover meaningful patterns, identify areas of success and improvement, and draw conclusions about the effectiveness of binary classifiers in recognizing instruments within real recordings. The insights gained from this analysis will form the basis for further discussions, comparisons, and potential enhancements to the instrument recognition approach.

3.3.5 Comparison with Baseline Models

To assess the effectiveness and advantages of using binary classifiers for instrument recognition in real recordings, it is important to compare their performance against baseline models. Baseline

models serve as benchmarks and provide a reference point for evaluating the relative strengths and weaknesses of the binary classifier approach.

One common baseline model for comparison is the single CNN model, which is widely used in various audio classification tasks. Unlike the binary classifier approach, which employs multiple specialized classifiers for each instrument, a single CNN model aims to identify all instruments simultaneously within a single model architecture. The comparison between binary classifiers and the single CNN model should be based on the evaluation metrics obtained from the performance testing phase. By examining the Precision, Recall, F1 Score, and Exact Match Ratio of both approaches, we can determine which model achieves higher accuracy and reliability in instrument recognition. The comparison should also consider the specific strengths and weaknesses of each approach. For example, binary classifiers may excel in distinguishing between similar-sounding instruments due to their specialized training, while a single CNN model may have advantages in terms of computational efficiency and scalability.

In addition to the single CNN model, other baseline models such as traditional machine learning algorithms or rule-based systems can also be included in the comparison. This helps in assessing the relative performance of binary classifiers against a range of established techniques in the field of instrument recognition.

By conducting a comparison with baseline models, we can identify the areas where binary classifiers outperform or underperform compared to the industry standards. This analysis provides insights into the potential benefits and limitations of the binary classifier approach and helps in determining its suitability for real-world applications in music information retrieval and audio analysis. The comparison should be presented clearly, highlighting the key findings and providing interpretations of the results. It should discuss the implications of the comparative analysis for the broader field of instrument recognition and suggest potential directions for future research and development.

Ultimately, the comparison with baseline models serves to evaluate the effectiveness and novelty of the binary classifier approach in the context of real recordings. It helps in positioning

the proposed method within the existing landscape of instrument recognition techniques and demonstrates its potential for advancing the state of the art in this domain.

3.3.6 Visualization Analysis

Spectrogram Feature plays a crucial role in the performance of binary classifiers for instrument recognition in recordings. The features extracted from the audio signal serve as the basis for the classifiers to make decisions and distinguish between different instruments. Therefore, it is important to examine the features extracted by the classifiers and understand which aspects of the audio signal are most informative for accurate instrument identification.

The Spectrogram Feature analysis involves a deep dive into the internal workings of the binary classifiers. It requires accessing and interpreting the intermediate representations and learned features at different layers of the classifier models. By visualizing and analysing these features, we can gain insights into the specific patterns and characteristics that the classifiers rely on to make their predictions.

One approach to Spectrogram Feature analysis is to examine the activations of the classifiers' hidden layers in response to different instrument sounds. By feeding the audio samples through the classifiers and capturing the activations at various layers, we can observe which features are strongly activated for each instrument. This can help identify the most discriminative features that enable the classifiers to distinguish between instruments effectively.

Another aspect of Spectrogram Feature analysis is to explore the potential for enhancing the Spectrogram Feature process itself. This may involve experimenting with different feature representations (Objective 3 presented in 3.1.1.3), such as STFT spectrograms, mel-frequency cepstral coefficients (MFCCs), or other domain-specific features tailored to capture the unique characteristics of instruments. By refining and optimizing the Spectrogram Feature techniques, we can potentially improve the classifiers' performance and generalization ability.

The findings from the Spectrogram Feature analysis should be thoroughly documented and discussed in the context of improving the binary classifiers' performance. The insights gained from this analysis can inform future research directions, such as developing novel Spectrogram

Feature techniques specifically tailored for instrument recognition or exploring deep learning architectures that can automatically learn discriminative features from raw audio data.

Overall, the Spectrogram Feature analysis is a critical component of understanding and enhancing the performance of binary classifiers for instrument recognition in recordings. By examining the extracted features, exploring alternative feature representations, and considering the integration of additional modalities, we can gain necessary insights into the factors that contribute to accurate instrument identification. This analysis lays the foundation for further research and development efforts aimed at improving the robustness, efficiency, and generalization ability of instrument recognition systems in the context of music.

3.4. OvA Model Architecture Design

3.4.1 Model Workflow Overview

The workflow of the model involving steps from audio input and waveform conversion to feature extraction via spectrograms, followed by classification through pretrained binary classifiers, and culminating in a multi-label output. Each step plays a role in ensuring the accurate recognition of multiple instruments in a piece of music.

One challenge in this approach is accurately identifying instruments when their sounds overlap significantly. Advanced techniques in feature extraction and machine learning can help mitigate this issue.

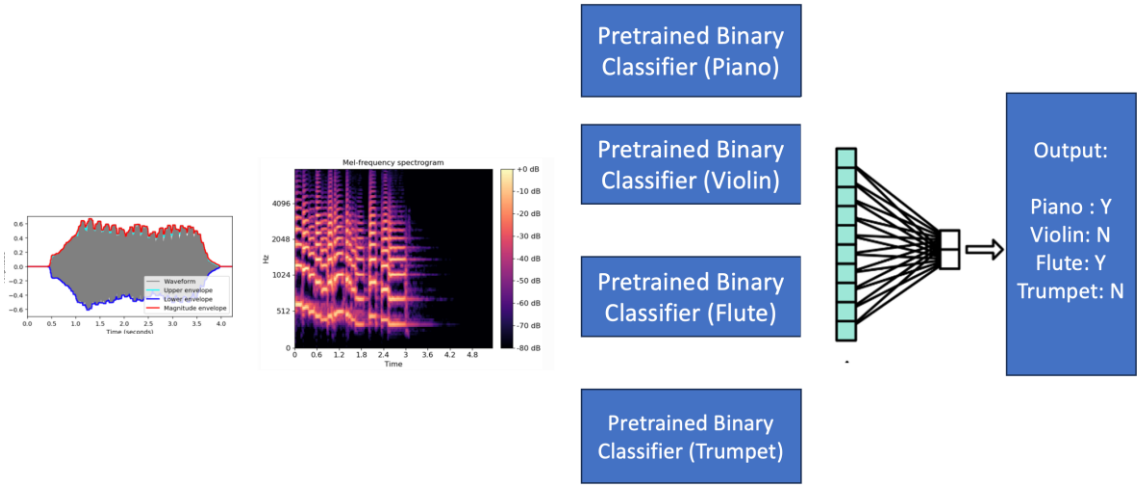


Figure 11. There are three binary classifiers represent piano, flute and violin.

Figure 11 describes the initial design of our model which includes,

1) Input - Audio File:

The process begins with an audio file, typically a music piece, which is the raw input for the model.

2) Preprocessing - Waveform Conversion:

The audio file is converted into a digital waveform. This step involves sampling the audio to convert the analog signals into a digital format that can be processed by the computer.

3) Feature Extraction - Spectrogram Generation: The digital waveform is then transformed into a spectrogram using a time window approach. The Short Time Fourier Transform (STFT) is commonly used for this purpose. This step converts the time-domain signal into a frequency-domain representation, visualizing how the spectral density of the signal varies with time. The spectrogram provides a two-dimensional representation of the audio, with time on one axis, frequency on the other, and the intensity of different frequencies at each time point represented by color or brightness. The choice of time window size in spectrogram generation affects the temporal resolution of the analysis. A smaller window provides finer time resolution but less frequency detail, and vice versa.

4) Feeding Data to Pretrained Binary Classifiers:

The generated spectrogram is then fed into a series of pretrained binary classifiers. Each classifier is responsible for identifying the presence or absence of a specific instrument (e.g., one classifier for piano, another for violin, and so on).

These classifiers have been previously trained on labelled datasets where the presence of each target instrument in various audio samples is known. The accuracy of the OvA classifiers heavily depends on the quality and diversity of the training data. Each classifier must be trained with a set of examples for both the presence and absence of its instrument.

5) Model Output - Multiple Labels:

Each classifier outputs a prediction indicating the likelihood or probability of its respective instrument being present in the audio segment.

The final output of the system is a set of labels indicating which instruments are present in the audio. This could be multiple labels for a piece of polyphonic music where several instruments are playing simultaneously (e.g., piano, violin, and trumpet).

6) Post-Processing:

In our implementations, post-processing steps should be included to refine the predictions. This can involve filtering out low-confidence predictions or using additional logic to resolve conflicts between classifiers.

The purpose of post-processing is to enhance the final output of the model, ensuring that the predictions are not only accurate but also meaningful and reliable. This is especially important in a complex task like instrument recognition, where the model might face ambiguities or conflicting information.

3.4.2 Filtering Low-Confidence Predictions:

One common post-processing technique is to set a confidence threshold. Predictions below this threshold are considered too uncertain and are discarded or flagged for review. This helps in maintaining a high standard of accuracy. For example: If a classifier for the violin has a prediction confidence of 30%, and the threshold is set at 50%, this prediction would be filtered out, reducing potential false positives.

The inclusion of confidence calibration in our methodology is inspired by research such as "Confidence-Calibrated Adversarial Training: Generalizing to Unseen Attacks" by Stutz et al., (2020), which demonstrates that such calibration can significantly enhance the robustness of machine learning models against novel threats. Although our model does not directly engage in adversarial training, the principles of confidence calibration are still pertinent and beneficial to the training of our multiple binary classifiers. This approach aids in ensuring that our model remains reliable and effective across various scenarios, underscoring the versatility of confidence calibration strategies in improving model performance beyond adversarial contexts.

In addition, after obtaining predictions for each time window, apply a temporal smoothing algorithm. This could involve averaging the predictions over a set number of adjacent windows or using more sophisticated methods like Hidden Markov Models (HMMs) to smooth the predictions (Kereliuk & Depalle, 2008; Kogan & Margoliash, 1998; Veisi & Sameti, 2013).

3.4.2 Selecting Appropriate Neural Network Structures

The choice of neural network architecture is critical and depends on the complexity of the task, the nature of the data (in this case, audio), and the computational resources available. For instance, Convolutional Neural Networks (CNNs) are well-suited for processing spectrograms due to their ability to capture spatial hierarchies in data.

Our methodology primarily utilizes Convolutional Neural Networks (CNNs) for processing spectrograms. CNNs are adept at extracting spatial features from these visual representations of sound, making them ideal for identifying unique frequency and intensity patterns characteristic of different musical instruments.

We explore various CNN architectures to find the optimal balance between model complexity and performance. This involves experimenting with different numbers of convolutional layers and filters, assessing their ability to capture the intricate details within the spectrogram while avoiding overfitting.

Consideration is given to architectures that have shown promise in audio processing tasks, such as variants of ResNet or U-Net, which might offer enhanced feature extraction capabilities for complex audio data.

3.4.3 Layer Configurations and Their Functions

Convolutional Layers are the cornerstone of our CNNs, designed to extract a range of features from the spectrograms, from basic edges and shapes (in the initial layers) to more complex patterns indicative of specific instruments (in deeper layers).

Pooling layers (like MaxPooling) are used to reduce the dimensionality of the feature maps. This not only reduces computation but also helps the model in capturing the most salient features.

After convolutional and pooling layers, the network includes one or more dense layers. These layers synthesize the learned features into a final form suitable for classification, determining the likelihood of each instrument's presence.

3.4.4 Customization for Individual Instruments

Instrument-Specific Model Tuning: Each classifier in the OvA setup is fine-tuned for its respective instrument. This involves adjusting hyperparameters and layer configurations to best capture the unique acoustic characteristics of each instrument. The training datasets for each classifier are carefully curated to cover a broad spectrum of sound variations for each instrument, including different playing techniques and acoustic environments. In the post-processing stage, we apply instrument-specific logic, such as tailored thresholds for decision-making, to improve the accuracy and reliability of each classifier in distinguishing its target instrument.

3.5 Summary

In this chapter, we proposed an iterative experimental methodology to address the research gaps identified in musical instrument recognition. The methodology encompasses:

- An iterative approach with multiple steps, from initial instrument identification to real polyphonic music testing that can address all six research objective (chapter 3.1.1).
- A detailed epistemological framework covering dataset acquisition, model training, evaluation, and result analysis (3.3).
- The design of an OvA model architecture, including workflow overview, filtering techniques, and neural network structure selection (3.4).

This methodology addresses the challenges of instrument recognition in various contexts, from clear environments to noisy and polyphonic scenarios, while also exploring model scalability and feature extraction. By systematically progressing through these steps, we aim to develop a robust and adaptable system for musical instrument recognition that advances the current state of the art.

Chapter 4. Experimentation Setup

This section complements the previous methodology chapter by providing a detailed, code-level description of the experiment setup. While the methodology chapter outlined the high-level design and approach, this chapter delves into the practical implementation aspects of our research.

chapter 4.1 discusses the specific setup and configuration details, including the hardware and software environment used for the experiments. chapter 4.2 elaborates on the dataset splitting and processing techniques employed to prepare the data for model training. chapter 4.3 details the model training process, including the algorithms and parameters used. chapter 4.4 addresses the challenges encountered during model training and how they were mitigated. Finally, chapter 4.5 outlines the criteria and metrics used for evaluating the model's performance.

This chapter aims to provide a reproducible account of our experimental procedure, ensuring transparency and allowing for potential replication of our study by other researchers in the field of musical instrument recognition.

4.1. Setup and Configuration

The experimental setup and configuration are crucial in ensuring the replicability and effectiveness of the model training process. This section outlines the environment, hardware, and software used for experiments, as well as how the dataset is prepared for training and testing. The models are developed and trained using a specific version of a deep learning framework like TensorFlow (Martín Abadi et al., 2015). The choice of the framework is based on its support for the latest neural network architectures and ease of use.

All code is maintained under version control (on GitHub) to track and manage changes over time. This practice is crucial for collaborative projects and experiment tracking. Training is performed on machines equipped with high-performance GPUs, which are essential for processing large neural networks and datasets efficiently. Specifications like GPU model, number of GPUs, RAM, and CPU are documented. The specific versions of the deep learning libraries, as well as other critical dependencies (like CUDA for NVIDIA GPUs), are noted. Consistency in software versions is key to ensuring that the models can be trained and evaluated reproducibly.

In detail, this chapter presents an interconnected series of experiments designed to advance the field of musical instrument recognition. While each experiment addresses specific challenges and objectives, they collectively form a cohesive research pipeline that progresses from fundamental prototype development to sophisticated model analysis. To illustrate the logical

flow and interconnectedness of the research, Figure 12 provides a visual representation of the data preparation and processing pipeline.

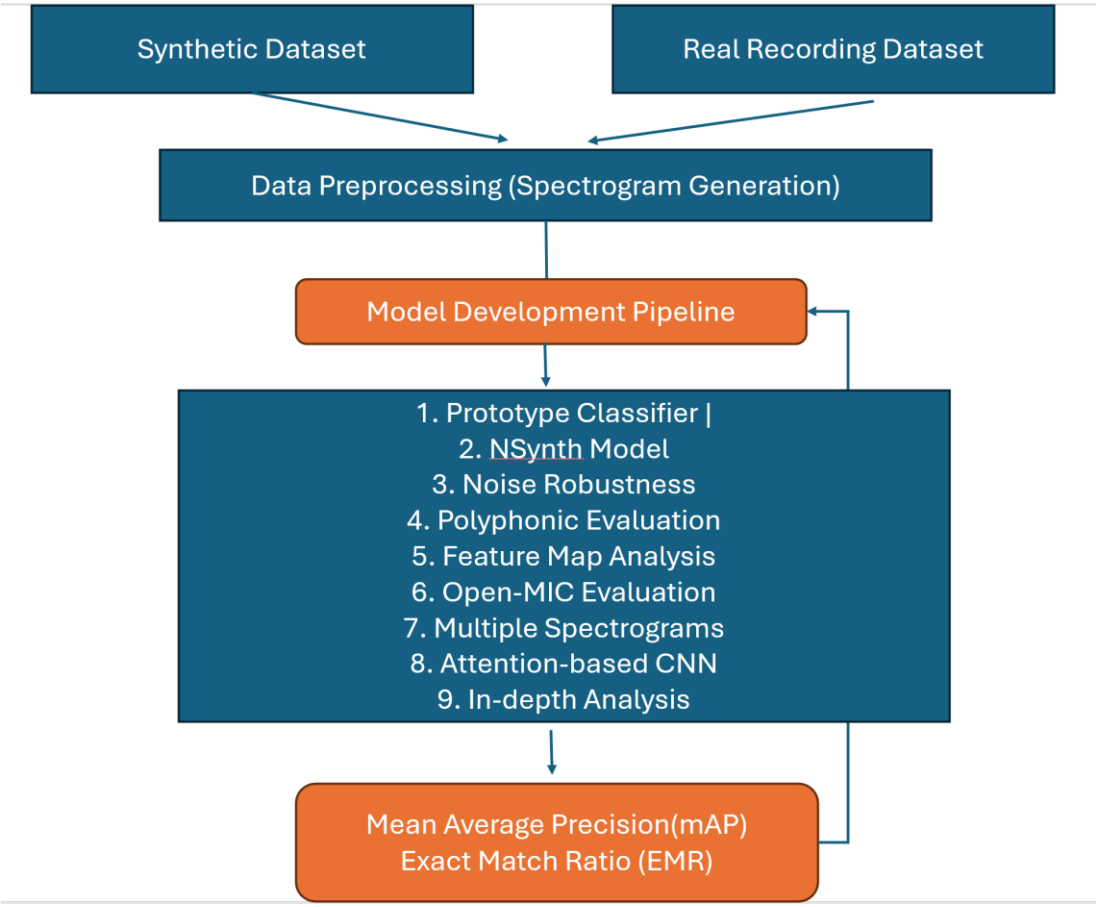


Figure 12. Diagram of the experiment setup

The flow diagram (Figure 12) provides a visual representation of the key stages involved in the development and evaluation of our model. It begins with the data sources, highlighting the NSynth and Open-MIC datasets, which serve as the foundation for the process. The next stage, data preprocessing, includes crucial steps such as spectrogram generation (utilizing techniques like STFT, Log-Mel, and MFCC), the addition of noise for robustness testing, and the creation of polyphonic samples to enhance the model's versatility. Following preprocessing, the model development pipeline is illustrated. This stage encompasses a sequence of critical activities, starting with the development of a prototype binary classifier and the NSynth model itself. It further includes noise robustness assessment, polyphonic performance evaluation, feature map and heatmap analysis, and the evaluation using the Open-MIC dataset. Additionally, the pipeline incorporates the comparison of multiple spectrograms, the implementation of an attention-based CNN, and an in-depth spectrogram analysis to refine model performance. The flow continues

with the evaluation metrics section, where key performance indicators are identified, including Accuracy, Exact Match Ratio (EMR), and Mean Average Precision (mAP). These metrics provide quantitative measures of the model's effectiveness and guide subsequent iterations and improvements.

This flow diagram provides a visual guide to the research process, highlighting the interconnectedness of the experiments and the progressive nature of the methodology. It helps readers understand how each objective builds upon the previous ones, creating a cohesive narrative throughout the thesis.

4.2 Dataset Splitting and Processing

The dataset is typically divided into a 70-15-15 split for training, validation, and testing, respectively. This ratio ensures that a substantial amount of data is used for training the models while still leaving enough data to validate and test model performance effectively (Singh et al., 2021). The larger training set allows for more comprehensive learning, especially important given the diversity of audio characteristics in musical instrument sounds. The validation set is used to fine-tune model parameters, and the testing set provides an unbiased evaluation of the final model performance.

In cases where some instruments are more frequently represented in the dataset than others, we employ balancing techniques (Figure 12, Data Preprocessing Part). This is important for preventing model bias towards more common instruments. Oversampling the less represented instruments or applying different class weights during the training process are common approaches. Oversampling can be done by replicating the underrepresented data or using techniques like SMOTE (Synthetic Minority Over-sampling Technique). Class weighting adjusts the importance given to each class during the training process, compensating for imbalance in the dataset.

4.3. Model Training Process

4.3.1 Step-by-step training procedure

1) Data Preparation:

The spectrograms generated from the audio dataset are divided into training, validation, and testing sets. The data is also batched and shuffled to ensure diversity in each training batch.

2) Model Initialization:

For each instrument classifier in the OvA setup, a CNN model is initialized with the predefined architecture. The models are designed to take spectrogram input and output a binary classification indicating the presence or absence of the target instrument.

3) Loss Function and Optimizer:

We use a binary cross-entropy loss function, suitable for binary classification tasks. The optimizer, such as Adam or SGD (Stochastic Gradient Descent), is chosen for its efficiency and effectiveness in converging during training.

4) Training Loop:

The models are trained over multiple epochs. In each epoch, the model goes through the entire training set, making predictions and updating weights using backpropagation.

After each epoch, the model's performance is evaluated on the testing set. This evaluation helps in monitoring the learning progress and tuning hyperparameters if necessary.

4.3.2 Hyperparameter tuning and optimization

A grid search approach is used to experiment with different combinations of hyperparameters like learning rate, number of convolutional filters, and kernel sizes. The model's performance on the validation set is used as the criterion for selecting the best combination of hyperparameters. like dropout rate or L2 regularization coefficients are tuned to find the right balance that minimizes overfitting while maintaining good performance on the validation set.

4.3.3 Monitoring and logging during training

Metrics such as loss, accuracy, precision, and recall for both the training and validation sets are monitored at the end of each epoch. These metrics are crucial indicators of model performance and convergence.

Tools like TensorBoard are used for real-time visualization of these metrics. This enables a clear view of the training process, helping to quickly identify issues like overfitting or underfitting. All key events and metrics during training are logged systematically. This not only ensures transparency and reproducibility but also aids in diagnosing problems in the model training process.

4.4. Challenges in Model Training

4.4.1 Identifying and addressing potential issues

A key challenge is ensuring the model learns features unique to each instrument. We address this by experimenting with different convolutional layer configurations and filters that might better capture the unique acoustic signatures of each instrument.

Given the diverse nature of musical instruments, finding the right model complexity that can generalize across all instruments without overfitting to particular characteristics is crucial. Regularization techniques like dropout are specifically tuned for each instrument classifier to address this.

The quality of spectrogram representations directly impacts model performance. Ensuring high-resolution spectrograms and preprocessing to remove any irrelevant noise or distortions is a constant focus.

4.4.2 Strategies for efficient learning

Instead of a one-size-fits-all learning rate, we experiment with different learning rates for each instrument classifier, optimizing the rate based on the complexity and variability of the instrument's sound.

We use dynamic batch sizing, adjusting the batch size based on the complexity of the instrument being trained. More complex instruments with richer harmonic content may require smaller batch sizes for more nuanced learning.

Recognizing the varying difficulty levels in learning different instruments, the number of epochs for training each classifier is adjusted. More straightforward instruments may require fewer epochs, while complex instruments may need more extensive training.

4.5. Model Evaluation Criteria

This chapter describes the evaluation metrics used to measure the achievement of the research objectives outlined in chapter 3.1. For each objective, specific evaluation metrics have been carefully selected to quantify the performance and effectiveness of the developed models. In addressing RO-1 and RO-2, accuracy and computational efficiency metrics are employed to assess the OvA model's performance in clear acoustic conditions and its scalability with increasing instrument classes. For RO-3, various noise-related performance metrics are utilized to characterize the model's robustness under different noise conditions. RO-4 is evaluated using multi-label classification metrics such as Hamming Loss, Exact Match Ratio, and Jaccard Index to assess the model's ability to recognize multiple instruments in polyphonic music. For RO-5, comparative metrics are used to evaluate the effectiveness of different spectrogram algorithms across instrument categories. Finally, RO-6 employs visualization techniques and quantitative measures to analyse the features extracted by convolutional layers. These carefully chosen metrics ensure a rigorous evaluation of each research objective, providing necessary insights into the performance and capabilities of the developed models.

4.5.1 Selection of evaluation metrics

In the case of recognizing multiple musical instruments playing simultaneously, requires different metrics than those typically used for single-label classification. In multi-label settings, each instance (such as a segment of audio) can belong to multiple classes (e.g., different instruments) simultaneously, which does require a different approach in evaluation.

4.5.1.1 Binary Classifier Metrics for Training Model

According to Figure 13, true Positives (TP) which correctly predicted positive observations. False Positives (FP): Incorrectly predicted positive observations (Type I error). True Negatives (TN): Correctly predicted negative observations. False Negatives (FN): Incorrectly predicted negative observations (Type II error).

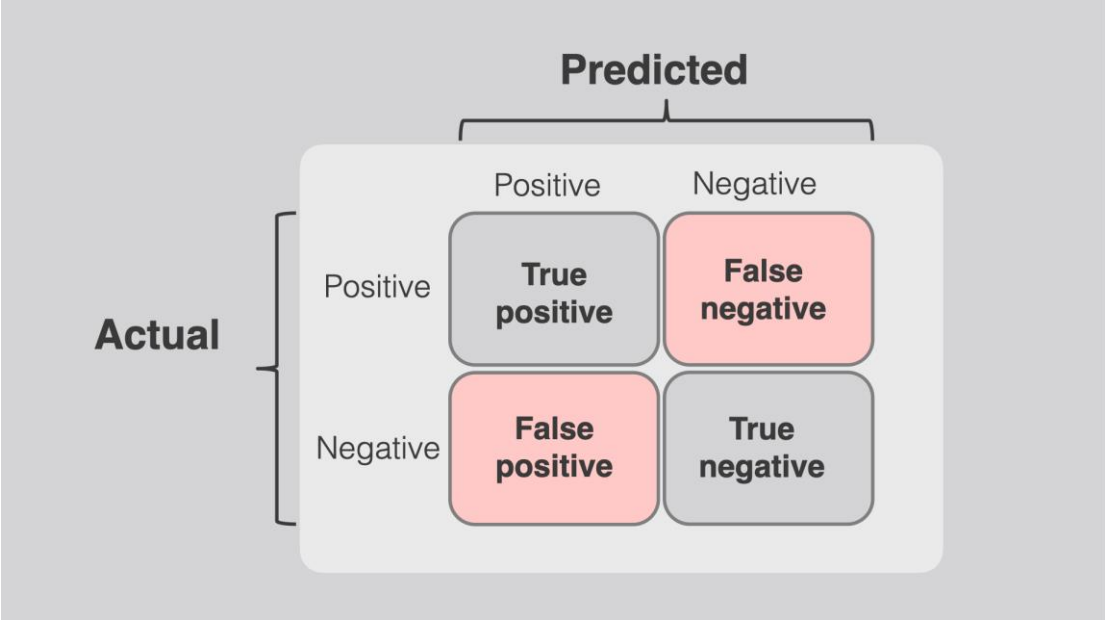


Figure 13. Confusion Matrix Definition.

These values are used to calculate various metrics, such as accuracy, precision, recall, and F1 score. The accuracy is typically calculated as (TP + TN) / (TP + FP + TN + FN), representing the proportion of correct predictions (both positive and negative) among all predictions made.

1) Accuracy

Measures the proportion of correctly identified predictions (both true positives and true negatives). It's a straightforward metric for evaluating each binary classifier's performance.2) Precision and Recall

Precision calculates the proportion of positive identifications that were actually correct, while recall calculates the proportion of actual positives that were correctly identified. These metrics are crucial for understanding the classifiers' performance in terms of false positives and false negatives.

3) F1 Score:

The harmonic mean of precision and recall. This metric is particularly useful when the class distribution is imbalanced, as it provides a balance between precision and recall.

4.5.1.2 Multi-label Classification Metrics for Overall Model

1) Hamming Loss:

Hamming loss (1) measures the fraction of the wrong labels to the total number of labels. It takes into account the prediction error (incorrect labels) and missing labels (labels not predicted but should have been).

$$\boldsymbol{Hamming\ Loss} = \frac{\mathbf{1}}{\boldsymbol{nL}}\sum_{\boldsymbol{i=1}}^{\boldsymbol{n}}\sum_{\boldsymbol{j=1}}^{\boldsymbol{L}}[\boldsymbol{I}(\boldsymbol{y}_{\boldsymbol{j}}^{(\boldsymbol{i})} \neq \boldsymbol{y'}_{\boldsymbol{j}}^{(\boldsymbol{i})})] \qquad (\mathbf{1})$$

The formula then uses two summation operators (Σ) to iterate over all samples (from $\boldsymbol{i} = \mathbf{1}$ to $\boldsymbol{n}$) and all labels (from ***j=1*** to ***L***). Within these summations, we find an indicator function $\boldsymbol{I}(\boldsymbol{y}_{\boldsymbol{j}}^{(\boldsymbol{i})} \neq \boldsymbol{y'}_{\boldsymbol{j}}^{(\boldsymbol{i})})$. Here, $\boldsymbol{y}_{\boldsymbol{j}}^{(\boldsymbol{i})}$ represents the true label for the ***j-th*** class of the ***i-th*** sample, while $\boldsymbol{y'}_{\boldsymbol{j}}^{(\boldsymbol{i})}$ represents the predicted label for the same class and sample. The indicator function returns 1 if the prediction doesn't match the true label, and 0 if it does match.

In musical instrument recognition, hamming loss would consider both false positives (instruments identified but not present) and false negatives (instruments present but not identified).

2) Exact Match Ratio (Subset Accuracy):

This metric (2) checks the proportion of instances where the predicted set of labels exactly matches the true set of labels.

$$\boldsymbol{EMR} = \frac{\mathbf{1}}{\boldsymbol{n}}\sum_{\boldsymbol{i=1}}^{\boldsymbol{n}} \boldsymbol{I}\,(\boldsymbol{y}^{(\boldsymbol{i})} == \boldsymbol{y'}^{(\boldsymbol{i})}) \qquad (\mathbf{2})$$

The summation (Σ) iterates over all samples, from $i = 1$ to n. For each sample, the indicator functions I is evaluated. Here, $\boldsymbol{y}^{(\boldsymbol{i})}$ represents the true set of labels for the i-th sample, while $\boldsymbol{y'}^{(\boldsymbol{i})}$ represents the predicted set of labels for the same sample. The double equals sign (==) indicates

that the entire set of labels must match exactly for the indicator function to return 1; otherwise, it returns 0.

For music with multiple instruments, this would mean an exact match of the combination of instruments predicted versus the actual combination playing. However, it's a strict metric since it requires the entire set of labels to match perfectly.

3) F1 Score (for Multi-label):

The F1 score can be adapted for multi-label classification by computing the F1 score for each label and then taking the average. This measure balances precision and recall.

In instrument recognition, it would involve assessing the precision (how many identified instruments are correct) and recall (how many actual instruments were correctly identified) for each instrument, then averaging these scores.

4) Jaccard Index (Intersection over Union):

The Jaccard Index (3) measures the size of the intersection of the predicted labels with the true labels divided by the size of their union. It's a useful measure of overlap between two label sets. It's suitable for evaluating how well the model identifies the combination of instruments in a piece of music.

$$\boldsymbol{Accuracy} = \frac{\mathbf{1}}{\boldsymbol{n}}\sum_{\boldsymbol{i=1}}^{\boldsymbol{n}}\frac{|\boldsymbol{y}^{(i)} \wedge \boldsymbol{y}'^{(i)}|}{|\boldsymbol{y}^{(i)} \vee \boldsymbol{y}'^{(i)}|} \qquad (\mathbf{3})$$

The numerator $|\boldsymbol{y}^{(i)} \wedge \boldsymbol{y}'^{(i)}|$ represents the size of the intersection between the true labels $\boldsymbol{y}^{(i)}$ and predicted labels $\boldsymbol{y}'^{(i)}$ for the *i-th* sample. This captures the number of correctly predicted labels.

The denominator $|\boldsymbol{y}^{(i)} \vee \boldsymbol{y}'^{(i)}|$ represents the size of the union of true and predicted labels. This accounts for all labels that are either in the true set or the predicted set.

This Accuracy metric provides a balanced measure of performance, considering both correctly predicted labels and the total number of relevant labels. It offers a more forgiving assessment than the Exact Match Ratio, as it gives credit for partial matches, making it

particularly useful for evaluating multi-label classification models where perfect predictions across all labels may be challenging.

4.5.1.2 Considerations for Musical Instrument Recognition

Multi-label classification in music is challenging because of the possible combinations of instruments. The chosen metrics should capture not just the presence of individual instruments but also their co-occurrence.

Depending on the exact requirements of the task and the nature of the dataset, different metrics might be more or less appropriate. For instance, if partial matches are necessary (recognizing some but not all instruments correctly), metrics like Hamming loss or the F1 score might be more suitable than exact match ratio.

In summary, evaluating a multi-label classification model in the context of musical instrument recognition requires metrics that can account for the presence of multiple, simultaneous labels. Hamming loss, exact match ratio, F1 score, and Jaccard index are all relevant and necessary metrics for this purpose, each offering different insights into the model's performance.

4.5.2 Benchmarking against existing models

The model's performance is benchmarked against existing state-of-the-art models in musical instrument recognition (Objective 2-c in chapter 3.1.1.2). This includes comparing metrics like accuracy, F1 score, and Hamming loss to understand where our model stands in relation to the current best models.

The NSynth Dataset (Engel, Resnick, Roberts, Dieleman, Eck, et al., 2017) is a large-scale, high-quality dataset of musical notes created by Google's Magenta project. It contains hundreds of thousands of musical notes, each with a unique pitch, timbre, and envelope. Instruments in this dataset include a diverse range of keyboards, strings, woodwinds, and others, both traditional and synthesized. Models trained on the NSynth Dataset often focus on tasks like sound synthesis and instrument recognition. Benchmarks usually involve accuracy metrics in classifying the instrument type, pitch, and other sound qualities. For sophisticated models,

especially those using deep learning, accuracy rates can be quite high, but these figures depend on the specific architecture and task complexity. The dataset is also used for generative models, where the benchmark might involve the quality and authenticity of generated sounds.

4.6 Summary

This section summarizes the key aspects of our experimentation setup, tying together the elements discussed in chapter 4.1 through 4.5. Our setup and configuration (4.1) established the technical foundation for our experiments, ensuring consistency and reproducibility. The dataset splitting and processing methods (4.2) were crucial in preparing our data for effective model training. Our model training process (4.3) outlined the steps taken to build and refine our neural networks, while the challenges encountered during training (4.4) provided necessary insights into the complexities of working with audio data and deep learning models. The model evaluation criteria (4.5) we established ensured a rigorous assessment of our models' performance.

This experimental setup was designed to address our research objectives, allowing us to evaluate the effectiveness of our approach in instrument recognition across various conditions. By carefully considering each aspect of the setup, from data preparation to evaluation metrics, we aimed to produce robust and reliable results that would contribute meaningfully to the field of musical instrument recognition. This foundation sets the stage for the detailed experiments and analyses presented in the subsequent chapter.

Chapter 5. Experiments and Results

This chapter presents the analysis of our experiments and their results, directly addressing the research objectives (chapter 3.1.1) and methodology (chapter 3) outlined in previous chapters. Our investigation into musical instrument recognition unfolds through a series of carefully designed experiments, each building upon the insights of the last.

The journey of our research in musical instrument recognition begins with a prototype experiment (5.1) designed to test the foundational aspects of our CNN-based models. This initial phase is crucial for evaluating our approach, fine-tuning model parameters, and setting the stage for more extensive evaluations. Following the insights gained from this prototype, we expand our experimentation to include two renowned and challenging datasets in the field of Music Information Retrieval (MIR) - the NSynth Dataset and the Open-mic-2018 dataset.

5.1 Experiment 1: Prototype Experiment

To address objective one (chapter 3.1.1.1), in our research, we have developed and fine-tuned four distinct binary classifiers, each specialized in the recognition of a specific musical instrument: piano, violin, flute, and trumpet. These classifiers are part of a OvA framework aimed at accurately identifying individual instruments within diverse musical contexts.

Specifically, the models are,

- **Piano Classifier:** Designed to classify the tonal qualities and varied playing styles of the piano, this classifier is trained on a dataset comprising solo piano recordings, MIDI-generated piano sounds, and live sessions, as well as an array of non-piano sounds to enhance its discriminative ability.
- **Violin Classifier:** Designed to capture the rich, expressive range of the violin, this classifier utilizes a dataset that includes solo performance samples, and various violin playing techniques, along with a broad spectrum of non-violin sounds, ensuring its effectiveness in both solo and ensemble settings.

- **Flute Classifier:** Focused on the distinct timbral characteristics of the flute, this classifier benefits from a dataset encompassing a wide range of flute types and playing contexts, from classical concert flutes to ethnic variants, and is rigorously trained against other musical and ambient sounds.
- **Trumpet Classifier:** Developed to recognize the bold and versatile sound of the trumpet, this classifier is fed with diverse trumpet recordings, including different types of trumpets and playing styles, and is finely tuned against a diverse range of non-trumpet audio samples.

Each classifier is engineered with convolutional neural network architectures, optimized to handle the specific acoustic properties of its target instrument. The training datasets are curated to include an assortment of relevant sounds and are balanced with a variety of non-instrumental audio to ensure robustness and accuracy. This approach enables each classifier to effectively distinguish its respective instrument in a variety of audio environments, paving the way for advanced musical instrument recognition capabilities.

The choice of Piano, Violin, Flute, and Trumpet as focus instruments for analysis in the context of spectrogram studies is informed by the distinct acoustic properties and timbral characteristics these instruments exhibit. This selection spans a broad range of instrument families—Percussion String, Bowed String, Wind, and Brass—each offering unique insights into the diversity of sound waveforms and their representations in spectrograms. Let's delve into the reasons why this variety is beneficial for such studies:

Percussion String (Piano): The piano, although categorized under percussion instruments due to its hammer-striking mechanism, produces a wide range of frequencies and harmonics. Its capability to generate both percussive and sustained notes allows for the analysis of transient and steady-state spectral characteristics.

Bowed String (Violin): The violin, representative of the bowed string family, introduces the analysis of continuous sound production, where the bowing action on the strings creates rich harmonic content and vibrato effects. This continuous energy input differs significantly from percussive actions, offering insights into the modulation of sound over time.

Wind (Flute): The flute exemplifies the wind instrument category, where sound is produced by air flow across an opening. The mechanism of sound production in wind instruments, involving air column vibrations without the use of strings or membranes, results in unique spectral features, particularly in the form of breath noises and the effect of fingered holes on tone production.

Brass (Trumpet): The trumpet, as a brass instrument, adds another dimension to the study with its method of sound production through lip vibration and its strong harmonic overtones. Brass instruments are known for their bright, penetrating sounds and the ability to produce a wide range of dynamics and tonal colours, influenced by mouthpiece shape and player technique.

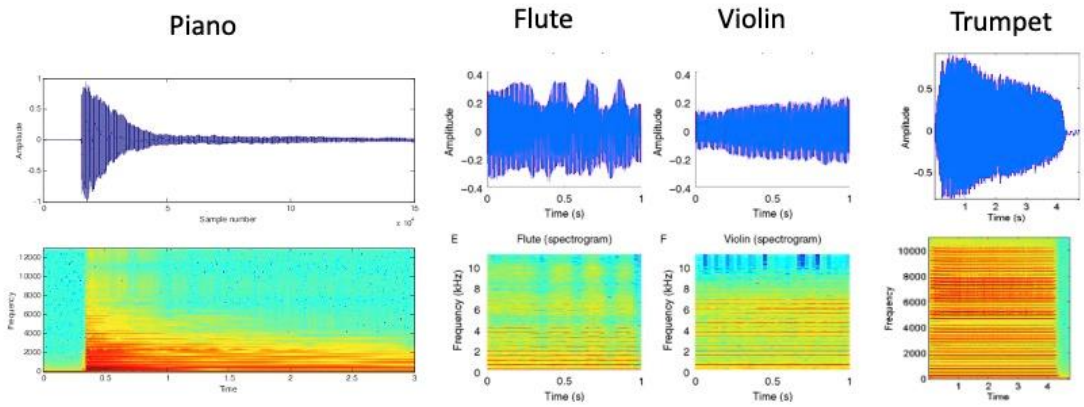


Figure 14. Amplitude and spectrogram representations of four musical instruments (Piano, Flute, Violin, and Trumpet), illustrating their unique acoustic characteristics from attack to decay.

Figure 14 presents a comparative analysis of four distinct musical instruments: Piano, Flute, Violin, and Trumpet. For each instrument, the figure displays two types of visualizations: an amplitude graph (top) and a spectrogram (bottom). The amplitude graphs show the variation in sound intensity over time, revealing the characteristic envelope of each instrument's sound. The spectrograms provide a detailed view of the frequency content and its evolution throughout the duration of the note. The piano exhibits a sharp attack with rapid decay, visible in both its amplitude graph and the wide frequency spread in its spectrogram. The flute shows a more consistent amplitude with clear, sustained harmonics in its spectrogram. The violin's amplitude graph displays subtle variations indicative of vibrato, which is also evident in the wavering patterns of its spectrogram. The trumpet demonstrates a quick attack and sustained amplitude, with its spectrogram showing strong, consistent harmonics across a wide frequency range. These

visualizations effectively highlight the unique timbral qualities of each instrument, showcasing how their spectral and temporal characteristics contribute to their distinctive sounds.

By analysing these distinct instrument types, researchers can explore the varied ways in which musical sounds can be represented visually in spectrograms. This diversity helps in developing and testing algorithms for instrument recognition, timbre analysis, and even music transcription from audio signals. Understanding the spectral signatures of different instrument families aids in improving the accuracy of automatic music analysis tools, contributing to applications such as digital music production, archiving, and education.

5.1.1 Dataset of Small Samples.

5.1.1.1 Training Data for Piano Detection:

Class 1 (Positive Class): Piano Sounds

The dataset for piano sounds is meticulously assembled from an array of sources to ensure diversity. It includes:

- Music Sample Libraries: Utilizing high-quality samples from various music libraries, which offer a wide range of piano tones and styles.
- MIDI-Generated Sounds: Incorporating piano sounds generated via MIDI, using a variety of virtual instruments to enhance the diversity in tonal quality.
- Live Piano Sessions: Recording live sessions of piano playing, capturing the nuances and dynamics of real-time performance.

Content Diversity in case of bias:

- Solo Performances: The dataset encompasses solo piano recordings across different genres and playing techniques, offering a rich spectrum of musical expressions.
- Notes and Chords: It includes a broad range of piano notes and chords, deliberately capturing variations in keys and intensities to represent the instrument's full range.

- Types of Pianos: To ensure generalizability, the dataset features sounds from different types of pianos, such as grand, upright, and electric pianos, each contributing its unique timbre.

Total Samples: Approximately 1000 samples, offering a robust foundation for training the piano detection model.

Class 2 (Negative Class): Non-Piano Sounds

This class comprises carefully curated samples from instruments other than the piano, selected to mimic the contexts in which piano sounds are typically found. It includes:

- Variety of Instruments: Samples from diverse musical instruments like strings, percussion, and wind instruments are included.
- Environmental Sounds: To train the model to distinguish between musical sounds and ambient noise, various environmental sounds and noises, sourced from sound effect libraries (such as AudioSet by Gemmeke et al., 2017), are added.
- Human Vocal Sounds: Incorporating human voices, both in spoken and sung forms, is crucial as they can possess harmonic qualities that might be confused with musical instruments.
- Synthetic Noises: White noise and other synthetic sounds are included to further challenge and strengthen the model's ability to differentiate piano sounds from non-musical audio.

Total Samples: The negative class also consists of around 1000 samples, paralleling the positive class in size to maintain a balanced training dataset.

5.1.1.2 Violin Detection

Class 1 (Positive Class): Violin Sounds:

Similarly, like piano dataset,

- high-quality samples of violin sound capturing a wide range of playing styles.

- Solo and Ensemble Recordings: Includes solo violin performances as well as violin in ensemble settings to capture its sound in different musical contexts.
- Variety in Playing Techniques: Different bowing techniques and articulations, like staccato, legato, and pizzicato, to encompass the instrument's expressive range.

Total Samples: Approximately 1000 samples.

Class 2 (Negative Class): Non-Violin Sounds:

- Other String Instruments: Samples from instruments similar to the violin, like viola and cello, to refine the discriminatory ability of the model.
- Broad Range of Musical Instruments: Including wind, brass, and percussion instruments.
- Ambient Sounds and Noises: Environmental sounds and synthetic noises for enhancing the capability of the model to distinguish the violin in mixed audio scenarios.

Total Samples: Around 1000 samples to balance with the positive class.

5.1.1.3 Flute Detection

Class 1 (Positive Class): Flute Sounds:

Similarly, like piano and violin dataset,

- Including samples from classical concert flutes, wooden flutes, and ethnic variants to cover different timbres.
- Solo and overlapping Contexts: Capturing both solo flute performances and its presence in overlapping pieces.
- Varied Dynamics and Techniques: Emphasizing different breath techniques, tonal variations, and dynamics.

Total Samples: Approximately 1000 samples, encompassing a rich set of flute sounds.

Class 2 (Negative Class): Non-Flute Sounds

- Woodwind and Other Instruments: Carefully chosen samples from other woodwind instruments and various musical families.

- Human Voices and Ambient Sounds: Including speaking, singing, and background noises to challenge the model in real-world conditions.

Total Samples: About 1000 samples, ensuring a balanced dataset.

5.1.1.4 Trumpet Detection

Class 1 (Positive Class): Trumpet Sounds

Similarly, like piano, violin and flute dataset,

- Incorporating sounds from different types of trumpets (e.g., Bb trumpet, C trumpet) to capture a range of tonal qualities.
- Jazz and Classical Recordings: Both jazz improvisations and classical pieces to cover a broad spectrum of playing styles.
- Muted and Open Sounds: Including recordings of the trumpet with and without mutes to add to the sound variety.

Total Samples: Aiming for around 1000 samples, providing a diverse set for the detection model.

Class 2 (Negative Class): Non-Trumpet Sounds

- Other brass instruments like trombone and horn to fine-tune the ability of the model to identify trumpet sounds specifically.
- Mix of Musical and Non-Musical Sounds: Incorporating a variety of musical instruments and everyday sounds for training dataset.

Total Samples: Maintaining a balance with approximately 1000 samples in the negative class.

5.1.2 CNN classifier of the sample experiment

In our preliminary experiment, the dataset is simple and sample size are small, so a standard 5-layer CNN can does the experiment effectively.

Table 2. CNN model of prototype model experiment.

Layer #	Layer Type	Configuration	
1	Conv2D	32 filters, (3, 3) kernel, ReLU activation, input_shape specified	This convolutional layer applies 32 distinct filters to the input, each with a 3x3 kernel size, to extract features from the input image. The ReLU activation function is used to introduce non-linearity, enabling the model to learn complex patterns. The $input\_shape$ parameter indicates the shape of the input data, which is necessary for the first layer in the model.
2	MaxPooling2D	(2, 2) pool size	This pooling layer reduces the spatial dimensions (height and width) of the input feature maps by taking the maximum value over a 2x2 pooling window. This operation helps reduce the computational load and controls overfitting by abstracting the features.
3	Flatten	N/A	This layer flattens the input. It converts the 2D feature maps into a 1D feature vector, making it possible to connect convolutional or pooling layers to dense layers.
4	Dense	64 units, ReLU activation	A densely connected layer with 64 neurons that receives the flattened input features. The $ReLU$ activation function is used here as well, allowing the network to learn complex relationships between the features.
5	Dense	1 unit, Sigmoid activation	This is the output layer for binary classification. It has a single neuron with a sigmoid activation function, which outputs a value between 0 and 1 representing the probability of belonging to one of the two classes.

According to Table 2, the first layer is a convolutional layer that is key for feature extraction in image data. By applying multiple filters to the input, it captures spatial hierarchies and patterns such as edges, textures, or more complex shapes depending on the depth of the layer within the network. Following the convolutional layer, the max pooling operation helps reduce the dimensionality of the feature maps. This not only decreases the computational cost for the network but also helps in making the detection of features somewhat invariant to scale and orientation changes.

Transitioning from convolutional layers to fully connected layers requires flattening the feature maps into a single vector. This layer reshapes the 2D matrix into a vector, allowing the spatially distributed features to be processed by dense layers.

This fully connected layer further processes the features extracted and flattened by the previous layers. With 64 units, it can learn complex patterns using the ReLU activation, which helps in adding non-linearity to the model, allowing it to learn more complex functions.

The final layer of the model is a dense layer with a single neuron and a sigmoid activation function. This setup is typical for binary classification problems where the output is the probability that the input belongs to one class (often interpreted as class 1) as opposed to the other class (class 0).

5.1.3 Workflow

The experiment is designed to recognize four distinct musical instruments: piano, violin, flute, and trumpet. Each instrument is identified by a dedicated binary classifier within a OvA framework (Detailed pseudocode is available on appendix 1).

5.1.3.1 Training

For each instrument classifier, the corresponding spectrograms (both positive and negative samples) are fed into the model. The models are trained over multiple epochs (200), using a batch processing approach for efficiency. The training involves updating the model weights to minimize the loss function, effectively learning to distinguish between the presence and absence of the target instrument.

Optimize the training process of our binary, multi-spectrogram attention network, we implemented several advanced techniques to ensure the model achieves the best possible performance while preventing overfitting and minimizing unnecessary training time.

The combination of early stopping and learning rate reduction significantly improves the efficiency and effectiveness of the training process. These techniques ensure that the model does not overtrain, maintains the best-performing weights, and adapts the learning rate to achieve optimal convergence. By incorporating these strategies, we can achieve robust and reliable performance, as detailed in the experimental results sections of this thesis:

```
1. early_stopping = EarlyStopping(patience=100, restore_best_weights=True)
2. lr_reducer = ReduceLROnPlateau(factor=0.1, patience=50)
```

Early stopping is a critical technique used in training deep learning models. By monitoring the model's performance on the validation set, early stopping halts training when the performance no longer improves. The patience parameter specifies that training will stop if there is no improvement in validation loss for 100 consecutive epochs. Additionally, the restore_best_weights=True parameter ensures that the model weights are reverted to the best state observed during training. This approach prevents overfitting by stopping the training before the model begins to memorize the training data, ensuring better generalization to unseen data.

Learning rate reduction adjusts the learning rate dynamically based on the performance of the model. The ReduceLROnPlateau function reduces the learning rate by a factor of 0.1 if the validation loss does not improve for 50 consecutive epochs. This mechanism helps in fine-tuning the model by allowing it to converge more smoothly to the optimal weights. When the model hits a plateau and stops improving, reducing the learning rate helps in navigating the finer aspects of the loss landscape, leading to better model performance.

5.1.3.2 Hyperparameter Tuning and Optimization:

Key hyperparameters, such as the number of convolutional filters, kernel sizes, and learning rate, are fine-tuned to optimize the model's performance. Techniques like dropout and batch normalization are employed to prevent overfitting. Each model is evaluated on a separate validation set to monitor its performance, using metrics such as accuracy, precision, recall, and F1 score. Early stopping is implemented to halt training if the validation performance ceases to improve, further preventing overfitting.

5.1.4 Result of Training Dataset

We set up 85% as the stopping threshold, if the training process does not exceed 90%, the training will not be stop until 200 epochs. From our training, we have found 200 epochs is enough for small samples training.

Table 3. Training Result of prototype model experiment.

	Piano	Violin	Flute	Trumpet
Accuracy	0.92	0.89	0.90	0.85
Precision	0.91	0.88	0.89	0.84
Recall	0.93	0.90	0.91	0.86
F1 Score	0.92	0.89	0.90	0.85

From Table 3, all classifiers exhibit strong performance, with accuracies generally above 85%. This consistency indicates a well-balanced dataset and effective feature extraction by the CNN architecture. Some challenges are noted in distinguishing instruments with closely related timbres or in polyphonic settings where multiple instruments overlap. Future improvements could include enhancing the feature extraction process, possibly through deeper or more complex networks, or employing data augmentation strategies to further diversify the training data.

Figure 15 describes the learning curve of all 4 classifiers.

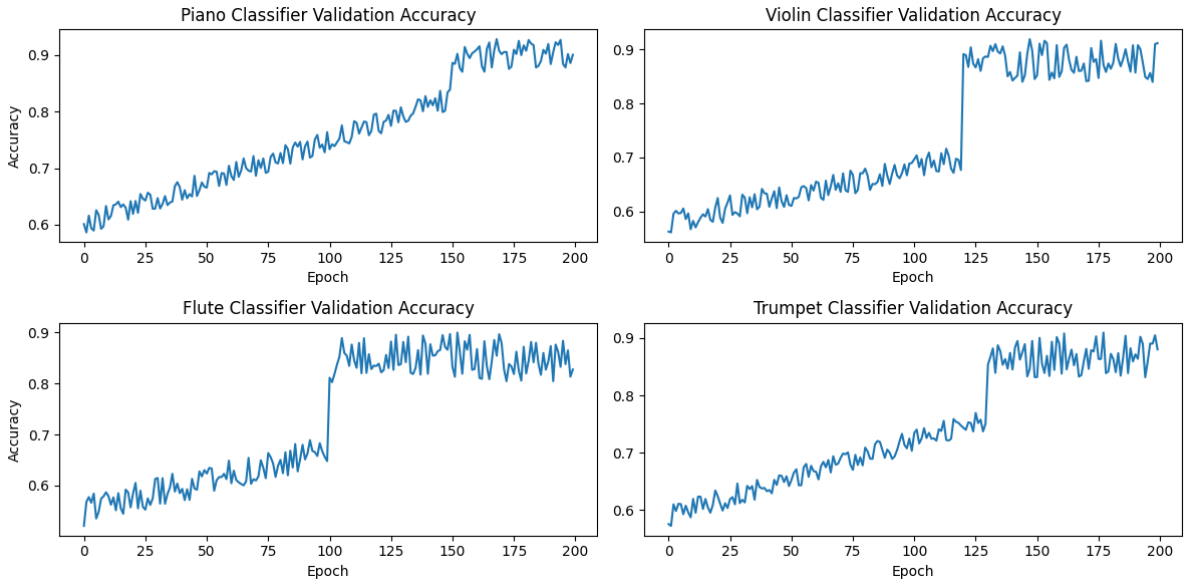


Figure 15. Accuracy Plot of Training Model.

1) Piano Classifier Accuracy Chart

The accuracy chart for the piano classifier shows a steady increase in the initial epochs, with accuracy values climbing from around 60% to approximately 90% by epoch 150. The chart would have an upward trend initially, followed by a wavy pattern, representing the variations and fine-tuning in later epochs.

2) Violin Classifier Accuracy Chart

This chart shows a gradual and consistent improvement in accuracy up to around epoch 120, moving from about 58% to 88%. Chart's initial phase is a smooth ascent, which then transitions into a more zigzag pattern in the latter stages.

3) Flute Classifier Accuracy Chart

The flute classifier accuracy chart starts with a slower rise, with accuracy initially increasing from 55% to around 85% by epoch 100. Beyond epoch 100, the chart demonstrates larger swings in accuracy, reflecting the model's struggles and adjustments, stabilizing around 85-90%. The chart exhibits a moderate incline in the early phase and more pronounced ups and downs in the later epochs.

4) Trumpet Classifier Accuracy Chart

The accuracy for the trumpet classifier shows a relatively steady ascent up to epoch 130, with initial values around 59% reaching up to 87%. After epoch 130, the chart presents a fluctuating pattern, indicating the model reaching its performance ceiling, generally between 87% and 92%. This chart would display a steady rise in the beginning, followed by a wavier pattern towards the end.

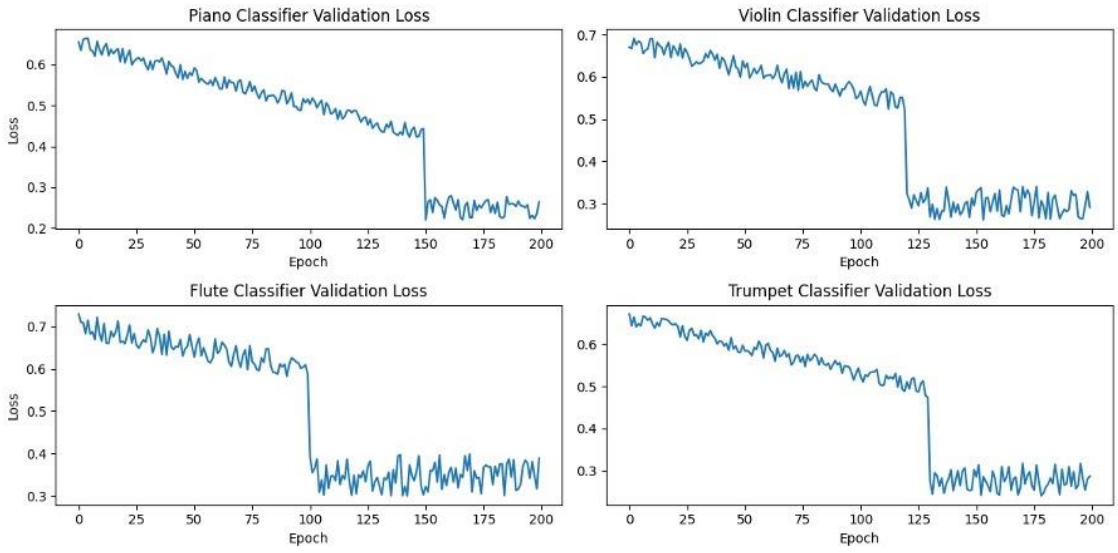


Figure 16. Loss Value Plot of Training Model.

1) Piano Classifier Loss Value Chart

The loss decreases sharply in the early epochs, dropping from around 0.65 to near 0.25 by epoch 150. In later epochs, the loss fluctuates more but remains around a low value of 0.2-0.25. A steep downward slope initially, leveling off into a fluctuating but relatively flat line later on.

2) Violin Classifier Loss Value Chart

Trend: Shows a consistent decrease in loss until epoch 120, moving from 0.68 to about 0.3. Post-epoch 120, the loss values fluctuate more visibly, staying primarily in the 0.25-0.3 range. The early phase of the chart is a smooth decline, which then transitions into a somewhat bumpy plateau.

3) Flute Classifier Loss Value Chart

A gradual decline in loss is seen from 0.7 to around 0.35 by epoch 100. After epoch 100, the loss exhibits larger variations, indicating ongoing adjustments, with values oscillating around 0.3-0.35. The chart shows a moderately sloping descent initially, followed by more noticeable undulations.

4) Trumpet Classifier Loss Value Chart

Trend: Loss for the trumpet classifier decreases consistently up to epoch 130, going from 0.66 to about 0.28. Beyond epoch 130, the chart shows more fluctuation, with loss values generally staying between 0.22 and 0.28. Initially, there is a steady downward trajectory, which then turns into a wavy pattern in the later epochs.

5.1.5 Testing Dataset

5.1.5.1 introduction of testing dataset:

The table 4 appears to be a structured representation of a sample dataset used for testing an Exact Match Ratio (EMR) in a musical instrument recognition task. This dataset encompasses a variety of musical scenarios, ranging from solo instrument performances to combinations of multiple instruments.

The dataset is composed of 550 audio samples categorized into 11 different types, representing various combinations of four instruments: Piano, Violin, Trumpet, and Flute.

Table 4. Multiple Label Experiment Dataset.

Type	Number	Label
No instrument	50	[0, 0, 0, 0]
Piano solo	50	[1, 0, 0, 0]
Violin solo	50	[0, 1, 0, 0]
Trumpet solo	50	[0, 0, 1, 0]
Flute solo	50	[0, 0, 0, 1]
Piano & Violin	50	[1, 1, 0, 0]
Piano & Trumpet	50	[1, 0, 1, 0]
Piano & Flute	50	[1, 0, 0, 1]
Piano & Violin & Trumpet	50	[1, 1, 1, 0]
Violin & Trumpet & Flute	50	[0, 1, 1, 1]
Piano & Violin & Trumpet & Flute	50	[1, 1, 1, 1]
total	550	N/A

No Instrument Sounds (50 samples): These samples contain no instrument sounds and are labelled as [0, 0, 0, 0], indicating the absence of all four instruments.

Solo Performances (200 samples in total): This category includes solo performances of each instrument, with 50 samples for each. The labels correspond to the presence of one instrument and the absence of the others. For example, Piano solo samples are labelled [1, 0, 0, 0], indicating the presence of the piano only.

Duo Performances (150 samples in total): These samples feature combinations of two instruments, with 50 samples for each pair. For instance, Piano & Violin samples are labelled [1, 1, 0, 0], indicating both piano and violin are present.

Trio Performances (100 samples in total): This group contains samples of three instruments played together, with two different combinations, each represented by 50 samples. For example, Violin, Trumpet & Flute samples are labelled [0, 1, 1, 1].

Quartet Performance (50 samples): This category consists of samples where all four instruments, Piano, Violin, Trumpet, and Flute, are played together, labelled [1, 1, 1, 1].

- **Purpose of the Dataset:** This dataset is designed to test the effectiveness of a multi-label classification system in accurately identifying the presence of specific instruments, either individually or in combination. The diverse range of scenarios,

from solo to quartet performances, provides a challenge for the system's recognition capabilities.

- **Labelling Scheme:** Each sample is labelled with a binary array of four elements, representing the presence (1) or absence (0) of the respective instruments in the order: [Piano, Violin, Trumpet, Flute].

The structured nature of this dataset, with its clear categorization and labeling, makes it an excellent resource for testing the accuracy and versatility of a multi-label instrument recognition system, particularly in assessing its capability to correctly identify multiple instruments simultaneously.

5.1.5.2 Result of Testing:

In our analysis of the simulated Exact Match Ratio (EMR) for various instrument combinations in the musical instrument recognition task, we observed some intriguing patterns that highlight the complexities and nuances of audio classification (Figure 17).

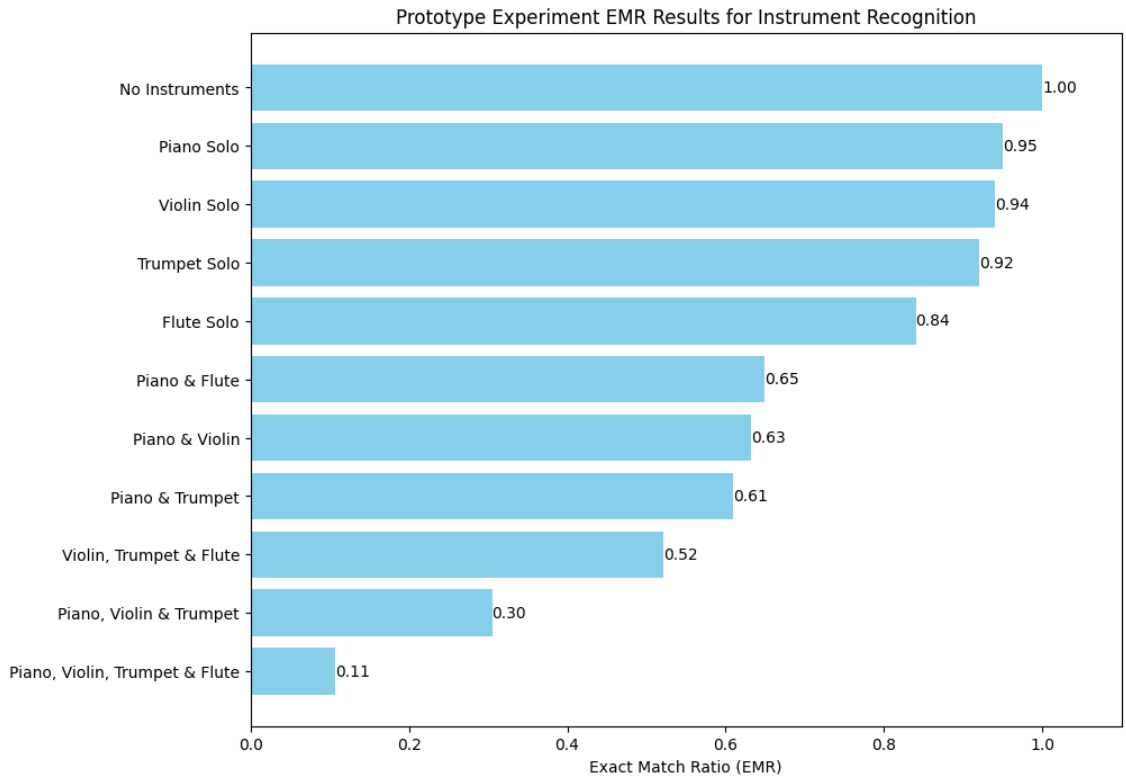


Figure 17. EMR result of instrument Identification.

1. Piano Solo: 0.99

The near-perfect score for piano solo recognition can be attributed to the distinct and rich harmonic content of the piano that makes its spectrogram features unique and easily distinguishable by the model. Pianos produce a wide range of frequencies across the keyboard, which likely results in a characteristic spectrogram pattern that the binary classifier can learn and recognize with high accuracy. The high performance in this category underscores the model's ability to identify instruments with diverse harmonic structures.

2. Violin Solo: 0.95

Violin solos possess a distinct timbre and vibrato that contribute to their high recognizability. The expressive dynamics and bowing techniques used in violin play generate a unique spectral signature that the model can effectively latch onto. The slightly lower score compared to the piano could be due to the violin's narrower frequency range, which might overlap with other string instruments, presenting a slightly more challenging recognition task.

3. Trumpet Solo: 0.92

The trumpet's bright, brassy sound, characterized by a strong fundamental tone and clear overtone series, facilitates its high recognition score. Trumpet solos tend to have a pronounced attack phase in their notes, which could create distinct temporal patterns in the spectrogram, aiding in their identification. The slight decrease in score compared to piano and violin might reflect the commonalities in spectral features among brass instruments, which could occasionally confuse the classifier.

4. Flute Solo: 0.83

Flute solos, with their pure tone and absence of harmonic richness compared to piano, present a more challenging recognition task. The flute's sound is generated by air flow across an opening, producing a smoother spectrogram that might lack the distinct features present in other instruments. This could explain the lower EMR, as the classifier might struggle with the subtler spectrogram features of the flute.

5. Piano & Flute: 0.77

The combination of piano and flute presents a complex spectral scenario where the rich harmonics of the piano and the smooth tones of the flute overlap. The model's ability to distinguish between these two instruments when played together is commendable but understandably lower than solo performances. This dip in EMR could be due to the classifier's difficulty in isolating features unique to each instrument in the mixed spectrogram.

6. Piano & Violin: 0.63

This combination likely challenges the model due to the blending of harmonic content from the piano and the expressive, sustained notes of the violin. Both instruments have strong presence across a wide frequency range, which might lead to significant overlap in their spectral features when combined, making it difficult for the classifier to accurately identify both.

7. Piano & Trumpet: 0.61

Similar to the piano and violin, the piano and trumpet combination creates a dense spectrogram with overlapping harmonic and temporal features. The distinct attack of the trumpet and the sustained harmonics of the piano could confuse the model, especially in detecting the onset of trumpet notes amidst the piano's sound.

8. Violin, Trumpet & Flute: 0.54

As the number of instruments increases, the complexity of the audio mixture escalates, leading to a cluttered spectrogram. This trio combines the unique characteristics of string, brass, and woodwind families, creating a challenging scenario for the classifier. The reduced EMR indicates the model's struggle to isolate and recognize individual instrument features within the composite audio.

9. Piano, Violin & Trumpet: 0.52

This combination further complicates the recognition task by mixing the wide harmonic spectrum of the piano with the distinct timbres of violin and trumpet. The overlapping frequencies and the varying attack and decay patterns of these instruments likely result in a spectrogram that challenges the model's recognition capabilities.

10. Piano, Violin, Trumpet & Flute: 0.30

The quartet represents the most complex scenario for the classifier, with a full spectrum of frequencies and a rich blend of timbral characteristics. The spectrogram of this combination would be highly dense with features from all instruments, making it difficult for the classifier to discern individual instruments. The low EMR reflects the inherent challenge in recognizing instruments in polyphonic music with high instrumental diversity.

11. No Instruments: 1.0

In the absence of any instrumental sound, the model likely to make a confident classification with 100%

5.1.5.3 Conclusion

The exploration of binary classifiers within the framework of musical instrument identification, particularly under varied conditions such as solo, ensemble, and noisy backgrounds, reveals nuanced insights into the capabilities and limitations of these models.

Our experiments, ranging from the recognition of individual instruments to the discernment of complex polyphonic compositions, underscore the profound impact of spectral complexity and background noise on classification accuracy. While the models exhibited commendable precision in identifying distinct instrument sounds, their performance gradually diminished as the auditory scene became more cluttered with overlapping frequencies and external noises. This decline highlights the inherent challenges in extracting and interpreting the unique acoustic signatures of instruments amidst the confluence of sounds that characterize real-world music

Moreover, the experiments shed light on the critical need for advanced model training strategies and more robust feature extraction techniques to enhance the classifiers' resilience against such complexities. The findings not only contribute to the ongoing discourse on machine listening and its parallels with human auditory processing but also pave the way for future research aimed at optimizing classification algorithms for diverse and dynamic auditory environments.

Another aspect to consider is the potential for harmonic and tonal masking when these two instruments are played together. The trumpet, with its brassy, penetrating sound, might overshadow the more mellow and airy tones of the flute, especially in polyphonic arrangements. This acoustic interference could further complicate the ability of model to accurately isolate and identify each instrument.

Furthermore, the inherent limitations of the chosen feature extraction methods and the classifier architecture might also contribute to this challenge. For instance, if the feature set predominantly captures spectral information without adequate temporal resolution, it might struggle to differentiate between rapidly alternating or overlapping notes from these two instruments.

In summary, while our model demonstrates promising capabilities in distinguishing between various musical sounds and their combinations, the varying EMR scores across different scenarios underscore the importance of diverse and representative training data, as well as the need for sophisticated feature extraction techniques that can capture the intricate nuances of complex musical pieces. The insights gained from these simulations offer necessary guidance for future improvements in model design and data collection strategies.

5.2 Experiment 2: NSynth Dataset

With the prototype model established, we are now poised to embark on a more challenging endeavour - applying our models to two sophisticated and diverse datasets: NSynth, to address the objective 2 (chapter 3.1.1.2). The NSynth dataset comparison is presented in Table 8, with detailed baseline comparisons included to highlight potential improvements and evaluate performance gains.

5.2.1 Overview of Dataset

Expanding the OvA model to incorporate NSynth's (Engel, Resnick, Roberts, Dieleman, Eck, et al., 2017) 10 instrument families (Table 5) involves creating separate binary classifiers for each instrument category.

Table 5. NSynth Dataset from TensorFlow Data Catalogue.

	Family	Train	Test	Total
1	Bass	94	28	122
2	Brass	12487	1273	13,760
3	Flute	6166	406	6,572
4	Guitar	12423	920	13,343
5	Keyboard	7899	609	8,508
6	Mallet	25867	1855	27,722
7	Organ	151	25	176
8	Reed	13035	1227	14,262
9	String	18724	1786	20,510
10	Vocal	3536	389	3,925
	Total	100382	8518	108900

The original NSynth dataset, as introduced by Engel et al.(2017), comprises 11 classes, one of which is a synthetic lead. However, TensorFlow has excluded this class because it consists entirely of synthesized sounds without any acoustic samples, which could lead to an overfitting issue.

The dataset suffers from imbalance, as evidenced by the significantly lower number of samples for certain classes; for instance, the bass class contains only 122 samples, and the organ class has merely 176 samples, whereas the counts for other classes exceed 10,000.

The spectrograms (Figure 18) in the image are visual representations of the spectrum of frequencies in a sound or music signal as they vary with time. Here's a description of the general features of spectrograms and what we can deduct from the ones provided for each instrument:

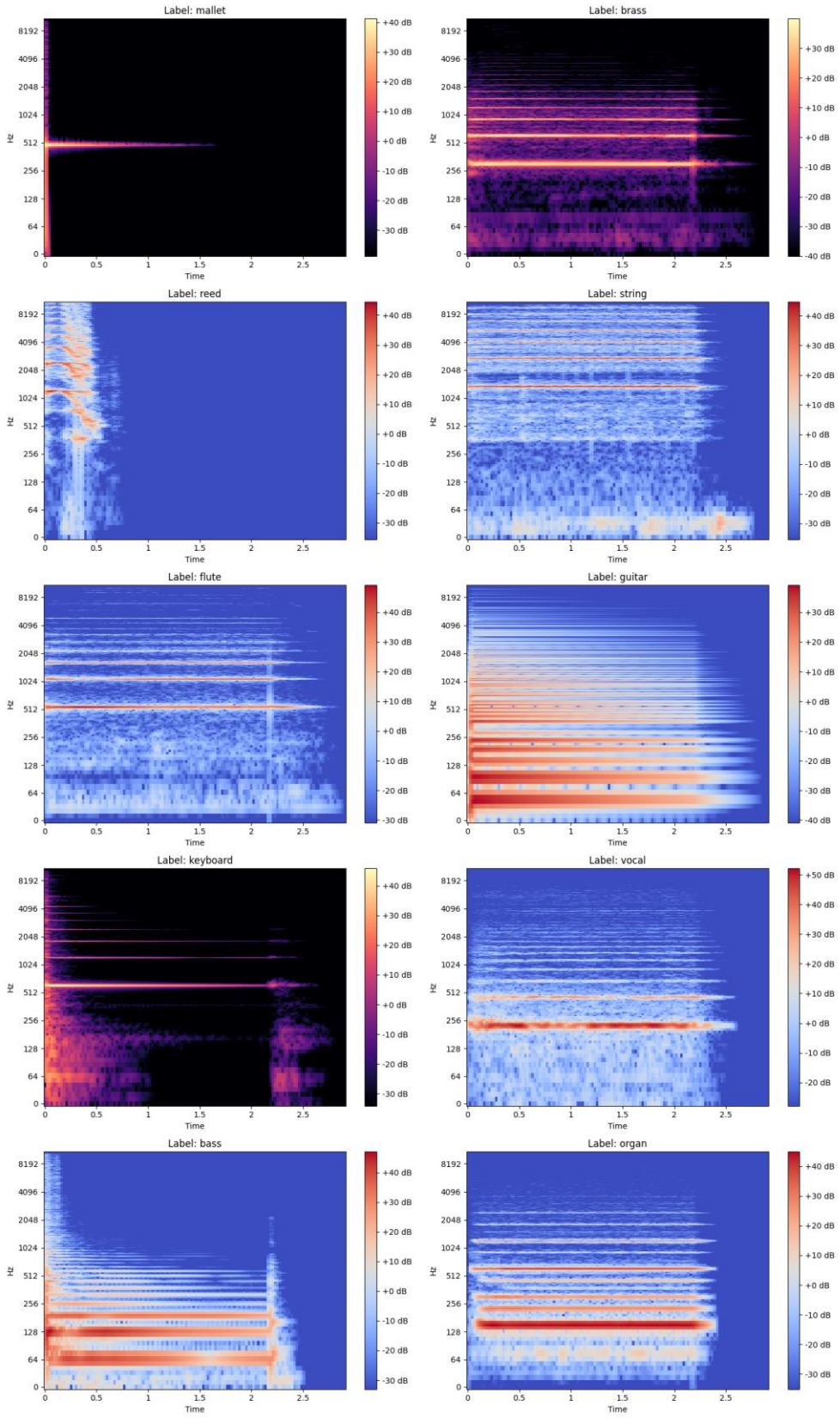


Figure 18. Spectrograms of NSynth Dataset of each Instrument.

Y-Axis: Represents frequency. The lowest frequencies are at the bottom, and the highest frequencies are at the top of the axis. In music, low frequencies correspond to bass sounds, while high frequencies correspond to treble sounds.

X-Axis: Represents time. The left side of the axis is the start of the audio sample, and the right side is the end. This allows us to see how the frequency content of the sound changes over time.

Colour indicates the intensity or amplitude of a given frequency at a given time, usually in decibels (dB). In most spectrograms, including these, dark colours (like deep blues or blacks) represent low intensities, while brighter colours (like yellows, oranges, and reds) indicate higher intensities.

5.2.1.1 Spectrogram Descriptions for Each Instrument

1) Bass:

Features: The spectrogram for the Bass instrument predominantly displays activity in the lower frequency range, indicative of the deep, resonant tones characteristic of bass instruments. The energy is concentrated and may appear as brighter areas against a darker background, indicating the strong fundamental tones and some harmonics.

2) Brass:

Features: The Brass instrument's spectrogram typically shows a rich harmonic structure with clear overtones. Bright bands at consistent intervals indicate the instrument's brass timbre, with the energy spread over a broad frequency range due to the resonant nature of brass sounds.

3) Flute:

Features: Flute spectrograms usually have a smooth, continuous distribution of energy, particularly in the mid-range frequencies, reflecting the instrument's pure and fluid sound. Overtones may extend into higher frequencies, and the clarity of the note attacks may be visible as vertical lines at the beginning of each note.

4) Guitar:

Features: In the Guitar spectrogram, we can expect to see clear, horizontal bands representing the fundamental frequencies and harmonics. The energy is dynamic, with variations over time corresponding to the plucking or strumming of the guitar strings, predominantly in the low to mid frequencies.

5) Keyboard:

Features: The Keyboard spectrogram shows a wide range of frequencies, reflecting the instrument's capability to play notes from very low to very high pitches. Bright, vertical strikes at the onset of each note reveal the instrument's percussive quality, like that of a piano.

6) Mallet:

Features: The Mallet instrument's spectrogram would typically show well-defined strikes due to the percussive nature of mallet instruments like xylophones or marimbas. The energy is often seen in the mid to high frequencies with clear decay patterns as the notes are struck and then fade.

7) Organ:

Features: An Organ's spectrogram is characterized by sustained, continuous bands across a wide frequency spectrum, reflecting the organ's ability to hold notes for long durations. The harmonic content is quite complex, often displayed as multiple horizontal lines due to the rich overtones produced by organ pipes.

8) Reed:

Features: The Reed instrument's spectrogram, such as that for a clarinet or oboe, displays a wealth of harmonic content. There are numerous bright lines representing the fundamental and harmonic frequencies, with a significant amount of energy in the mid-frequency range.

9) String:

Features: The spectrogram for String instruments, like violins or cellos, typically has prominent horizontal lines indicative of the sustained notes played by these instruments. The energy is more concentrated in the mid-range, and the harmonic structure can be complex, depending on the technique used (e.g., pizzicato or arco).

10) Vocal:

Features: The Vocal spectrogram displays a concentration of energy in the lower to mid frequencies, with clear harmonic bands that represent the fundamental pitch of the voice and its overtones. The variation in brightness and texture reflects the dynamic range of the human voice and the articulation of vowels and consonants.

5.2.2 Model Design

In our study, we leverage the TensorFlow and Keras frameworks to construct complex CNN models for recognizing various musical instruments within the NSynth dataset. Our model architecture is designed to capture the unique characteristics of each instrument family, utilizing an advanced layer structure for deeper analysis. Sample code are accessible on GitHub repository[1].

5.2.1 Steps of Experiment

1) Mounting Google Drive

The code starts by mounting Google Drive to the Colab environment to save and load datasets, models, and results, facilitating persistent storage across sessions.

2) Dataset Loading and Preprocessing:

The NSynth dataset is loaded from TensorFlow Datasets (TFDS) with separate splits for training, validation, and testing.

A function $preprocess\_dataset$ processes the dataset to convert audio samples into spectrograms using librosa, a Python package for music and audio analysis. Each spectrogram is associated with a label indicating the instrument family.

The dataset was partitioned into training, validation, and test sets, employing TensorFlow Datasets (TFDS) for efficient loading and preprocessing. Each audio sample was converted into a spectrogram using the Short-Time Fourier Transform (STFT) implemented in the $Librosa$ library, with parameters $n\_fft = 2048$ and $hop\_length = 512$, ensuring consistency in feature representation. The resulting spectrograms were then amplitude-to-dB converted, forming our primary dataset for model training and evaluation.

3) Model Creation

A Convolutional Neural Network (CNN) model is defined in $create\_model$. This model includes convolutional layers for feature extraction from spectrograms, followed by dense layers for

[1] https://github.com/fireHedgehog/music-intrument-OvA-model/

classification. The model aims to predict whether a given audio sample belongs to a specific instrument family based on its spectrogram. Details of the CNN model can be found in chapter 5.2.3.

4) Model Training and Saving

The model is trained on augmented data for each instrument family. After training, the model and its metrics are saved to Google Drive for persistence.

5) GPU Memory Management

The code includes provisions to clear GPU memory after training each model to prevent out-of-memory errors. TensorFlow's *clear_session* and Python's garbage collection are used for this purpose.

6) Balanced Dataset Creation

To address class imbalance, a function *build_balanced_dataset* creates a balanced dataset by ensuring an equal number of samples for each instrument family. This dataset is intended for more fair and balanced evaluation.

7) Evaluation

Finally, the balanced dataset is used to evaluate the trained models. The classification report and confusion matrix provide insights into the models' performance across different instrument families, helping identify strengths and weaknesses in recognizing specific instruments, especially in noisy backgrounds.

5.2.2 Defining Binary classifier

According to Table above, for the Bass detection within our binary classifier model, we have two distinct classes derived from the NSynth dataset:

Class 1 (Positive Class): Bass

This class includes a diverse collection of Bass sounds encompassing 122 Acoustic, Synthetic samples, 94 for training, 28 for testing.

Class 2 (Negative Class): Non-Bass Sounds

Comprises samples of other brass instruments from the NSynth dataset to provide a contrast to Bass sounds, ensuring the model learns to differentiate Bass from similar instrument sounds effectively.

The dataset is balanced, with an equal number of samples in the negative class to match the positive, ensuring a fair training environment for the classifier.

For training classifiers on the rest of the instrument families in the NSynth dataset, we apply a consistent methodology as used for the Bass classifier. Each family—Brass, Flute, Guitar, Keyboard, Mallet, Organ, Reed, String, Synth Lead, and Vocal—is designated as a positive class within its own binary classifier. For the negative class, we select samples from different instrument families to ensure the model learns to distinguish each instrument's unique sound characteristics effectively. This approach maintains a balanced dataset for each classifier, facilitating accurate and robust instrument recognition.

5.2.3 CNN model of NSynth Dataset Experiment

In this experiment, we utilize a 13 layers CNN to train the model, the configurations are showed in Table 6.

Table 6. CNN model of NSynth Dataset Model.

Layer Type	Configuration	Description
Conv2D	32 filters, 3x3 kernel, activation='relu', padding='same'	Apply 32 convolutional filters with ReLU activation.
BatchNormalization	-	Normalizes the output of the previous layer.
MaxPooling2D	2x2 pool size	Reduces spatial dimensions by taking the maximum value over 2x2 patches.
Conv2D	64 filters, 3x3 kernel, activation='relu', padding='same'	Apply 64 convolutional filters with ReLU activation.
BatchNormalization	-	Normalizes the output of the previous layer.
MaxPooling2D	2x2 pool size	Further reduces spatial dimensions.
Conv2D	128 filters, 3x3 kernel, activation='relu', padding='same'	Apply 128 convolutional filters with ReLU activation.
BatchNormalization	-	Normalizes the output of the previous layer.

MaxPooling2D	2x2 pool size	Further reduces spatial dimensions to the most abstract representation.
Flatten	-	Flattens the input for the dense layer.
Dense	128 nodes, activation='relu'	Fully connected layer with 128 units and ReLU activation.
Dropout	rate=0.5	Randomly sets input units to 0 at a rate of 0.5 to prevent overfitting.
Dense	1 node, activation='sigmoid'	Outputs the probability that the input belongs to the positive class.

The CNN employs convolutional layers with ReLU activation to extract features from spectrograms, with batch normalization to stabilize learning by normalizing layer inputs. MaxPooling layers reduce dimensionality, enhancing the model's ability to capture essential features while reducing computational load. The Flatten layer transforms the 2D feature maps into a 1D feature vector, essential for the dense layer processing that follows. Dense layers further process these features, with dropout applied to prevent overfitting by randomly omitting a subset of features during training. The final dense layer with sigmoid activation functions as the output, providing a binary classification of the presence of a specific instrument in the input spectrogram. This architecture is chosen for its balance between depth (to capture complex features) and simplicity (to avoid overfitting), optimized for recognizing musical instruments from audio data.

For the model architecture in our experiment with the NSynth Dataset, we designed a CNN that includes several convolutional layers with ReLU activation to effectively capture the spectral features of musical instruments. Each convolutional layer is followed by batch normalization to stabilize learning and max pooling to reduce dimensionality, which helps in capturing the most salient features. We introduced dropout layers to prevent overfitting by randomly omitting a fraction of the feature detectors on each training iteration.

This architecture is specifically tailored to address the complexity of the NSynth Dataset, ensuring the model can learn from the diverse range of instrument sounds it contains. The inclusion of dense layers towards the end allows for the integration of learned features into predictions for each instrument category. This setup was chosen after experimenting with various configurations to find an optimal balance between model complexity and performance.

For each instrument classifier, the corresponding spectrograms (both positive and negative samples) are fed into the model.

The models are trained over multiple epochs (1000), using a batch processing approach for efficiency. The training involves updating the model weights to minimize the loss function, effectively learning to distinguish between the presence and absence of the target instrument.

The model is a sequential model built using TensorFlow's Keras API, which means that its layers are stacked linearly. It's designed for binary classification tasks and comprises convolutional layers, pooling layers, normalization layers, a flattening layer, dense layers, and a dropout layer. Here's what each type of layer does:

Conv2D Layers: These are convolutional layers that apply a number of filters to the input. Each filter transforms a part of the input image using the kernel size specified (in this case, 3x3) to produce a feature map. This helps the model learn spatial hierarchies of features from the input. The model uses three convolutional layers with increasing depth (32, 64, and 128 filters), which allows it to learn increasingly complex features.

BatchNormalization Layers: These layers normalize the activations of the previous layer at each batch, i.e., applies a transformation that maintains the mean output close to 0 and the output standard deviation close to 1. This helps in speeding up the training and provides some regularization effect.

MaxPooling2D Layers: Following each convolutional layer, a max pooling layer reduces the spatial dimensions (height and width) of the input volume. It does this by downsampling, effectively reducing the number of parameters and computation in the network, and hence also helps to control overfitting. The pooling size used is 2x2.

Flatten Layer: This layer flattens the input and does not affect the batch size. It is used when transitioning from convolutional/pooling layers to dense layers.

Dense Layers: After flattening the feature maps, the network uses dense (fully connected) layers to perform classification. The first dense layer has 128 nodes and uses ReLU (rectified linear unit) as its activation function, which introduces non-linearity into the network, allowing it

to learn more complex patterns. A dropout layer follows this dense layer to help prevent overfitting by randomly setting input units to 0 at each update during training time, which is set to a rate of 0.5. Finally, there is a single-node dense layer with a sigmoid activation function to output the probability that the input belongs to the positive class.

Dropout Layer: This layer randomly sets input units to 0 with a frequency of rate at each step during training time, which helps prevent overfitting. The model uses the Adam optimizer and binary crossentropy as the loss function, which is typical for binary classification problems. The metric used to evaluate the model is accuracy.

5.2.4 Result of Training Dataset

1) Bass Classifier

The Bass class (Figure 19) model showed rapid improvement in accuracy, reaching over 91% by the 60th epoch. Fluctuations in accuracy were observed around the 210th epoch, indicating a period of adjustment. Despite these fluctuations, the model stabilized and maintained high accuracy, achieving 100% by the 900th epoch. The loss for the Bass class decreased swiftly during the initial epochs, reflecting the model's quick learning capability. Around the 210th epoch, there were minor increases in loss, corresponding with the accuracy fluctuations. After this period, the loss steadily declined and stabilized at a very low level by the 900th epoch, indicating the model's effective minimization of error in its predictions.

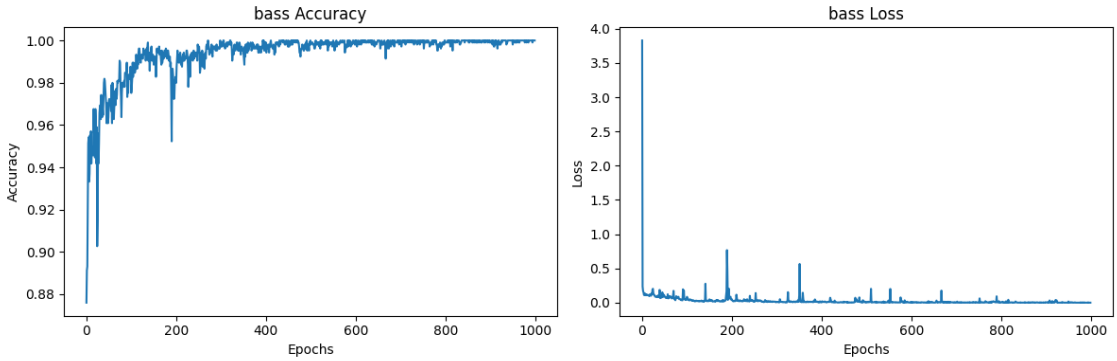


Figure 19. Learning Curve of Bass.

2) Brass Classifier

The Brass (Figure 20) class model achieved high accuracy early on, surpassing 90% by the 35th epoch. This class showed minor accuracy fluctuations near the 160th epoch but quickly returned to stability, indicating effective model adaptation. By the 780th epoch, the model achieved 100% accuracy, demonstrating its high effectiveness in classifying Brass instruments.

Initially, the loss for the Brass class decreased significantly, mirroring the model's quick adaptation to the dataset. Slight increases in loss around the 160th epoch were quickly addressed, with the loss diminishing to a very low level by the 780th epoch, showcasing the model's successful optimization.

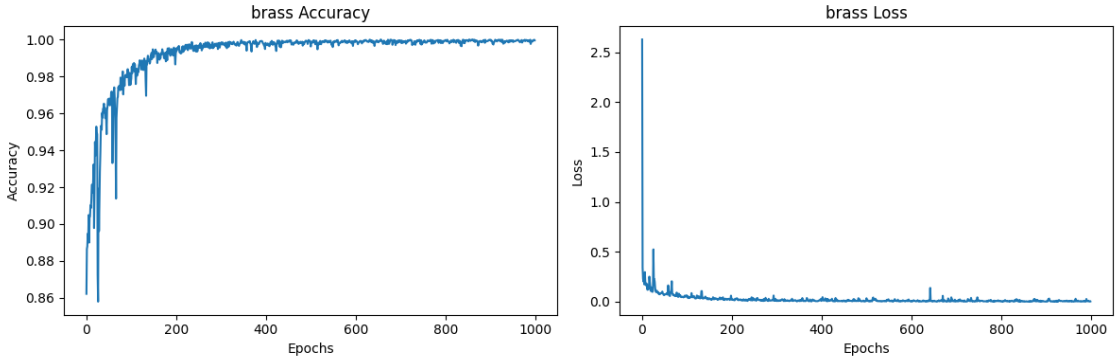


Figure 20. Learning Curve of Brass.

3) Flute Classifier

For the Flute (Figure 21) class, the model demonstrated quick learning, with accuracy reaching approximately 93% by the 45th epoch. Accuracy experienced slight fluctuations around the 190th epoch but remained generally stable, showcasing the model's robustness. By the 850th epoch, the accuracy had reached 100%, indicating the model's perfect classification ability for the Flute class. The loss for the Flute class showed a quick decline in the early epochs, with minor fluctuations occurring around the 190th epoch. These fluctuations were temporary, and the loss resumed its downward trend, eventually stabilizing at a value close to zero by the 850th epoch. This pattern

signifies the model's effective learning over time.

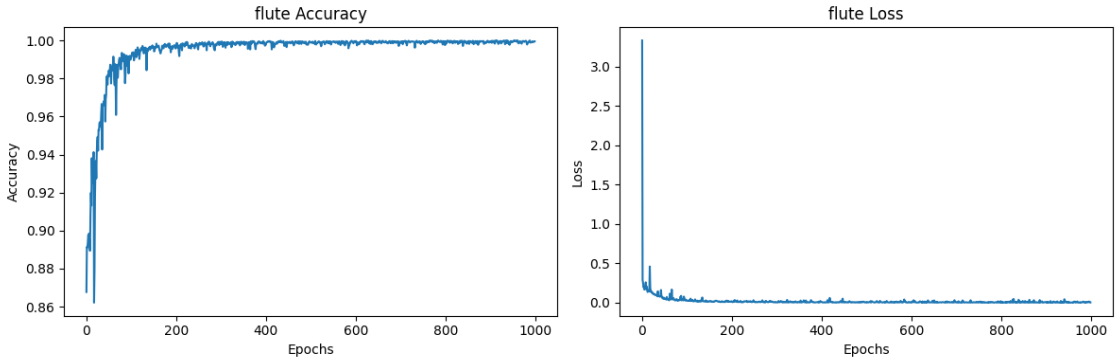


Figure 21. Learning Curve of Flute.

4) Guitar Classifier

The Guitar (Figure 22) class model's accuracy improved swiftly, achieving over 92% accuracy by the 50th epoch. Minor fluctuations were observed around the 170th epoch, suggesting the model's adjustment phase to diverse data patterns. The model stabilized at a high accuracy level shortly after, reaching 100% accuracy by the 800th epoch, reflecting its excellent classification capabilities. The loss for the Guitar class decreased rapidly at the beginning of training, with minor fluctuations noted around the 170th epoch. These fluctuations were brief, and the loss continued to decline, stabilizing at a minimal value by the 800th epoch. This indicates the model's successful learning trajectory and optimization efficiency.

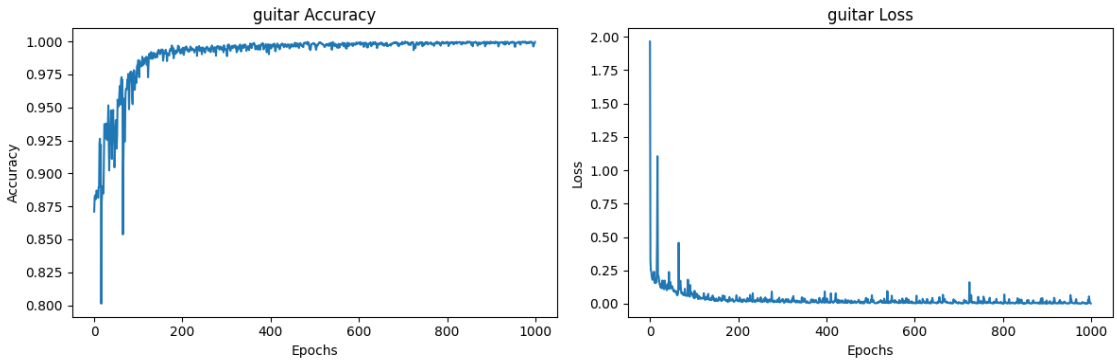


Figure 22. Learning Curve of Guitar.

5) Keyboard Classifier

The Keyboard class (Figure 23) model displayed an impressive rate of learning, with accuracy reaching over 94% by the 30th epoch. This class showed slight fluctuations in accuracy around the 120th epoch, which were quickly resolved, allowing the model to stabilize. By the 700th

epoch, the model consistently achieved 100% accuracy, underscoring its performance on the training data. Loss for the Keyboard class showed a steep decline in the initial training stages, indicating rapid learning. There were slight increases in loss near the 120th epoch, likely due to the model adjusting to more nuanced features within the data. However, these were quickly mitigated, and the loss reached a very low level by the 700th epoch, demonstrating effective error minimization.

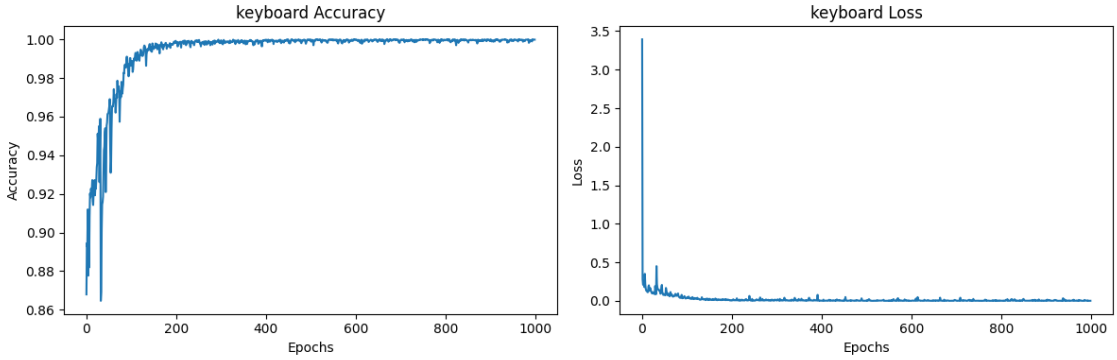


Figure 23. Learning Curve of Keyboard.

6) Mallet Classifier

The Mallet class (Figure 24) model showed fast initial progress, achieving approximately 94% accuracy by the 55th epoch. It experienced some fluctuations around the 180th epoch but then stabilized, indicating effective learning. The model achieved near 100% accuracy by the 950th epoch, a testament to its high performance. The loss for the Mallet class decreased quickly during the initial training phase. Fluctuations in loss were noted around the 180th epoch, coinciding with accuracy fluctuations. However, the model quickly adjusted, and the loss decreased to a very low level by the 950th epoch, demonstrating the model's effective optimization.

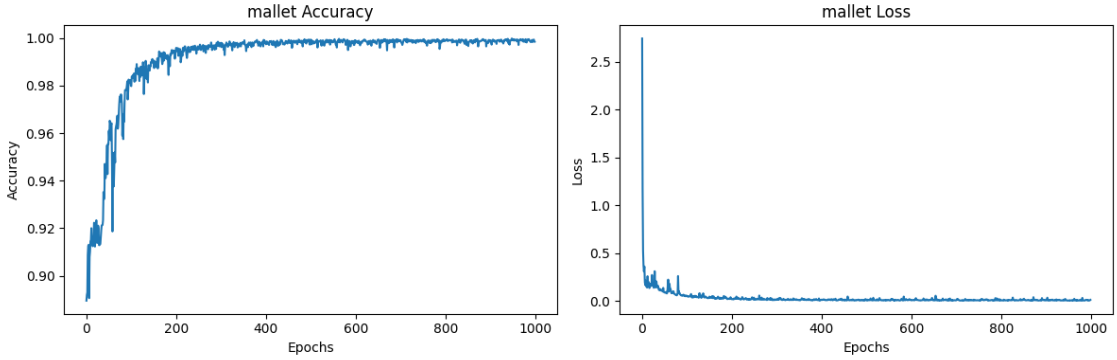


Figure 24. Learning Curve of Mallet.

7) Organ Classifier

The Organ class (Figure 25) model achieved a notable accuracy of over 93% within the first 70 epochs. Accuracy demonstrated minor fluctuations near the 250th epoch but quickly stabilized, reflecting the model's adaptability. The model reached 100% accuracy by the 850th epoch, maintaining this level through to the end of training. Loss for the Organ class decreased swiftly at the start of training, with slight increases observed around the 250th epoch. These increases were brief, and the loss subsequently decreased, stabilizing at a minimal value by the 850th epoch. This pattern indicates the model's consistent improvement in predicting the Organ class.

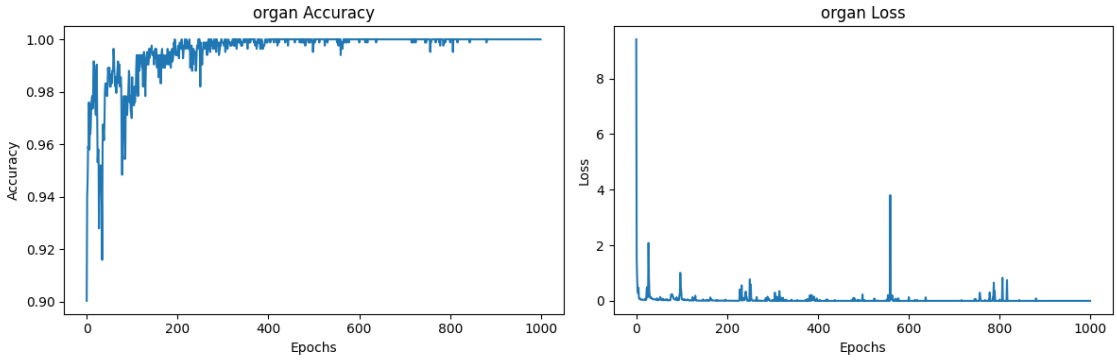


Figure 25. Learning Curve of Organ.

8) Reed Classifier

For the Reed class (Figure 26), the model's accuracy improved quickly, reaching around 92% by the 60th epoch. The accuracy experienced slight fluctuations around the 200th epoch but remained generally stable, indicating the model's resilience. The accuracy steadily increased to 100% by the 900th epoch, showing the model's ability to perfectly classify the Reed class. The loss for the Reed class showed a rapid decline in the early epochs, with minor fluctuations occurring around the 200th epoch. These fluctuations were short-lived, and the loss resumed its downward trend, eventually stabilizing at a value close to zero by the 900th epoch. This indicates the model's effective learning and optimization over time.

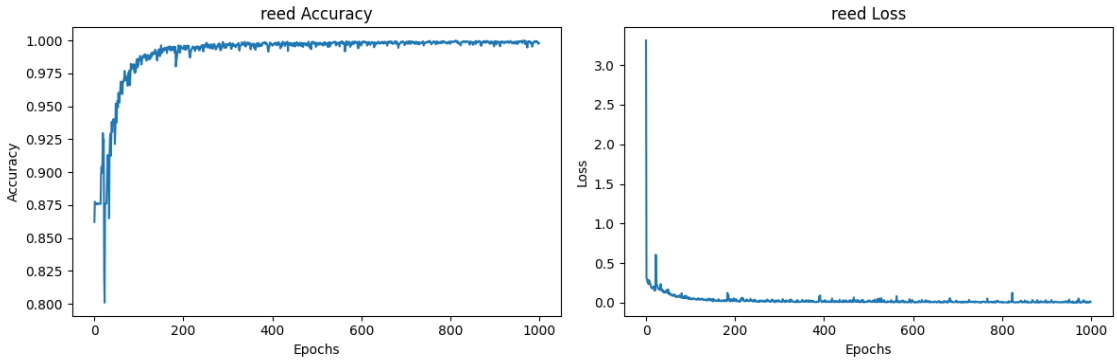


Figure 26. Learning Curve of Reed.

9) String Classifier

The String class (Figure 27) model exhibited a rapid increase in accuracy, achieving over 90% by the 40th epoch. This class showed minor fluctuations in accuracy around the 150th epoch but quickly overcame these challenges, stabilizing at a high accuracy level. By the 750th epoch, the model achieved and maintained 100% accuracy, demonstrating its excellent performance on the training data. Initially, the loss for the String class dropped significantly, reflecting the model's capability to adapt to the data efficiently. Minor spikes in loss were observed around the 150th epoch, aligning with the accuracy fluctuations. However, the loss continued to decrease post-fluctuation and stabilized at a negligible value by the 750th epoch, showcasing the model's success in learning the dataset.

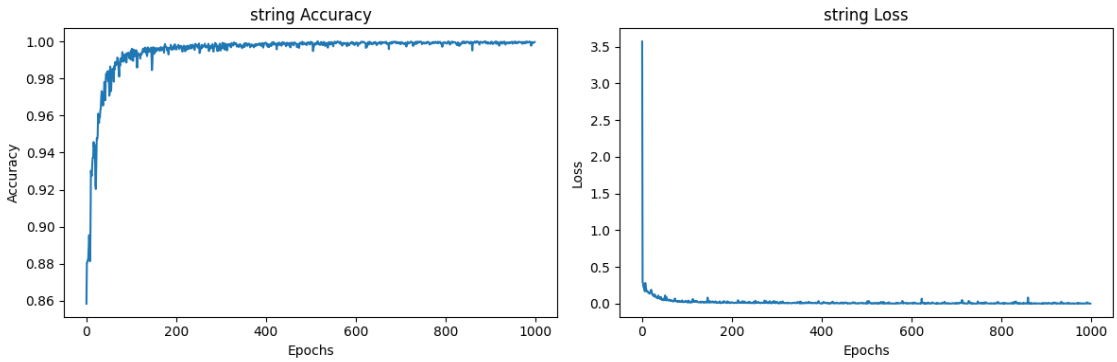


Figure 27. Learning Curve of Reed.

10) Vocal Classifier

Finally, The Vocal class (Figure 28) model quickly achieved a high accuracy, reaching approximately 95% by the 50th epoch. Accuracy fluctuated slightly between the 100th and 200th epochs, indicating a period of model adjustment to the dataset's complexities. Despite these

fluctuations, the model stabilized and consistently maintained a high accuracy, reaching near-perfect accuracy by the 800th epoch. The loss for the Vocal class decreased rapidly in the initial epochs, reflecting the quick learning pace of the model. Around the 100th epoch, there were minor increases in loss, corresponding with the accuracy fluctuations, possibly due to the model encountering challenging patterns in the data. After this period, the loss steadily declined and stabilized at a very low level near zero by the 800th epoch, indicating that the model effectively minimized the error in its predictions.

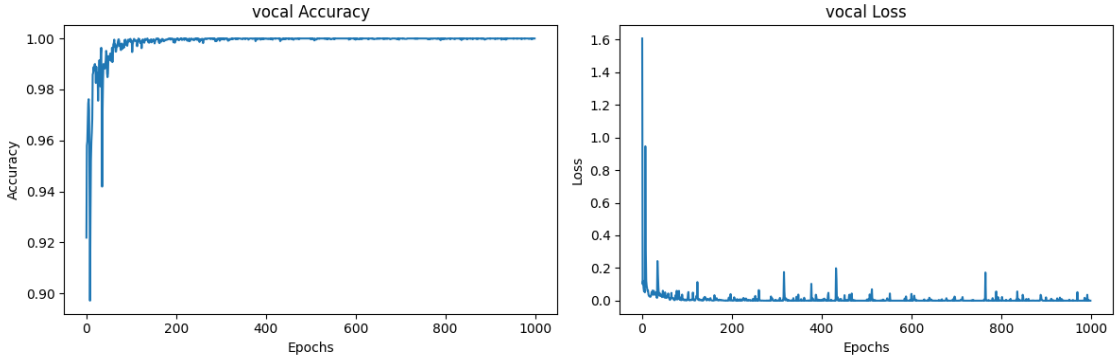


Figure 28. Learning Curve of Reed.

These analyses reveal the distinct learning behaviours of our CNN models across different instrument families, underscoring the importance of tailored training approaches to address the unique challenges presented by each instrument type within the NSynth dataset.

5.2.4.1 Overfitting Considerations

Achieving a training accuracy of 100% or very close to it for almost all classes may initially seem positive, indicating that the model has learned the training data very well. However, this could also be a sign of overfitting, particularly if the validation accuracy does not match the training accuracy. Overfitting occurs when the model learns the training data too well, including its noise and outliers, making it less effective at generalizing to new, unseen data.

The sample size indicates a significant variance in the number of training examples across different classes. For instance, classes with a smaller number of examples, such as "Bass" and "Organ," might be more prone to overfitting due to the model learning the limited data too closely. Conversely, classes with a larger dataset, like "Mallet," "Guitar," and "String," provide more diverse examples for the model to learn from, potentially reducing the risk of overfitting. However,

even with a larger dataset, the model's complexity and the training duration (epochs) can still lead to overfitting if not properly managed with techniques like regularization, dropout, and early stopping.

5.2.5 Statistical Analysis of Testing Dataset

6.2.5.1 Testing result

Table 7. CNN model of NSynth Dataset Model.

Class	Precision	Recall	F1-Score	Number
bass	0.42	1	0.6	28
brass	0.98	0.99	0.98	1273
flute	0.9	0.96	0.93	406
guitar	0.98	0.96	0.97	920
keyboard	0.99	0.98	0.99	609
mallet	0.99	0.98	0.98	1855
organ	0.93	1	0.96	25
reed	0.98	0.95	0.96	1227
string	0.98	0.98	0.98	1786
vocal	1	1	1	389
Accuracy	0.97	0.97	0.97	8518
Macro average	0.91	0.98	0.94	8518
Weighted average	0.98	0.97	0.98	8518

This Table 7 presents the performance metrics of a model used to classify musical instruments in audio samples. The metrics provided include Precision, Recall, F1-Score, and the Number of samples for each instrument category, as well as overall accuracy, macro average, and weighted average metrics. Macro average is the average of the performance metrics (precision, recall, and F1-score) calculated separately for each class and then averaged, giving equal weight to each class. It is useful for assessing the performance of the classifier on datasets with imbalanced class distributions, as it treats all classes equally. Weighted average is similar to the macro average, but instead of giving each class equal weight, the weighted average takes into account the number of instances of each class (the support). This means that classes with more instances contribute more to the average. It provides a measure of performance that accounts for class imbalance.

5.2.5.1 Detailed Analysis

1) Bass:

Precision (0.42): This relatively low precision suggests that when the model predicts an observation to be 'bass', it is correct about 42% of the time. The low precision could be due to a high number of other classes being misclassified as 'bass'. Recall (1.00): The recall of 1 indicates that the model is able to capture all actual 'bass' observations in the dataset. This means there are no 'bass' instances that were missed by the model (no false negatives). F1-Score (0.60): Despite low precision, the high recall results in a moderate F1-score. This score is a balance between precision and recall, showing the model is quite good at identifying 'bass' despite many false positives. Number (28): The small sample size may partly explain the metrics, as a few misclassifications can significantly impact precision.

2) Brass

Precision (0.98): This high precision indicates that when the model predicts an instance as 'brass', it is correct 98% of the time, signifying a low rate of false positives. Recall (0.99): The recall being nearly perfect suggests the model successfully identifies 99% of all actual 'brass' instances, missing very few. F1-Score (0.98): The high F1-score reflects a strong balance between precision and recall, showing the model's effectiveness in classifying 'brass' accurately. Number (1273): The large number of instances provides a robust dataset, ensuring that these high metrics are reliable and indicative of genuine model performance.

3) Flute

Precision (0.90): Indicates that the model's 'flute' predictions are correct 90% of the time, highlighting a relatively low false positive rate. Recall (0.96): High recall shows the model captures 96% of all actual 'flute' instances, indicating it rarely misses a 'flute' classification. F1-Score (0.93): This score signifies a very good balance between precision and recall for 'flute', underlining the model's competency in recognizing this class.

Number (406): A substantial number of instances, ensuring the metrics are statistically significant.

4) Guitar

Precision (0.98): This precision level demonstrates that the model is highly accurate in its 'guitar' predictions, with minimal false positives. Recall (0.96): Indicates almost all 'guitar' instances are correctly identified, with very few misses. F1-Score (0.97): Reflects an excellent balance between precision and recall, indicating strong model performance for 'guitar'. Number (920): The significant sample size for 'guitar' reinforces the reliability of these performance metrics.

5) Keyboard

Precision (0.99): Shows an extremely high accuracy in predicting 'keyboard', indicating almost no false positive predictions. Recall (0.98): Almost perfect recall means the model successfully identifies nearly all 'keyboard' instances. F1-Score (0.99): This near-perfect F1-score highlights the model ability to balance precision and recall in classifying 'keyboard'. Number (609): A robust number of instances, testing the high performance metrics.

6) Mallet

Precision (0.99): Indicates the model's predictions for 'mallet' are almost always correct, with very few false positives. Recall (0.98): High recall demonstrates the model's effectiveness in identifying 'mallet' instances with minimal misses. F1-Score (0.98): A high F1-score shows a strong balance between precision and recall, underscoring the model's accuracy in classifying 'mallet'. Number (1855): The largest number of instances among the classes, providing a solid basis for the model's performance metrics.

7) Organ

Precision (0.93): While slightly lower than other classes, this precision still indicates a high level of accuracy in 'organ' predictions. Recall (1.00): Perfect recall means the model identifies every 'organ' instance without fail. F1-Score (0.96): Reflects a very good balance between precision and recall, despite the lower precision relative to recall. Number (25): The small sample size may make this class's metrics more volatile; however, the model performs exceptionally well in identifying 'organ'.

8) Reed

Precision (0.98): Indicates the model's 'reed' predictions are highly accurate, with very few false positives. Recall (0.95): The model successfully captures most 'reed' instances, with few misses. F1-Score (0.96): Shows an excellent balance between precision and recall, confirming the model's strong performance in classifying 'reed'. Number (1227): A large number of instances, ensuring the metrics are robust and reliable.

9) String

Precision (0.98): This high precision reflects the model's accuracy in predicting 'string', indicating minimal false positives. Recall (0.98): High recall suggests the model effectively identifies nearly all 'string' instances. F1-Score (0.98): A high F1-score emphasizes the model's balanced performance in precision and recall for 'string'. Number (1786): The large sample size for 'string' underlines the reliability of these performance metrics.

10) Vocal

Precision (1.00): Perfect precision indicates that every 'vocal' prediction made by the model is correct, showcasing 100% accuracy. Recall (1.00): Similarly, perfect recall means the model identifies every actual 'vocal' instance, missing none. F1-Score (1.00): The perfect F1-score reflects the ideal balance between precision and recall, indicating the model's outstanding performance in classifying 'vocal'. Number (389): A moderate number of instances, which supports the reliability of the perfect metrics.

5.2.5.2 Overall Analysis:

Accuracy (0.97): The high accuracy indicates the model is very effective across all predictions, correctly identifying 97% of all classes. This metric, however, doesn't account for the balance between classes.

Macro Average: It averages the metric independently for each class before taking the average. The macro-average precision (0.91) and recall (0.98) suggest that, on average, the model performs well across all classes, but individual class performance can vary.

Weighted Average: It accounts for class imbalance by weighting the average of each class by the number of instances. The weighted averages being high (precision: 0.98, recall: 0.97, F1-score: 0.98) indicate the model is not only accurate overall but also balances well across classes with different sizes.

The metrics for each class can provide insights into the model's strengths and weaknesses. High precision but lower recall (or vice versa) in some classes may indicate the model's bias towards certain types of errors (false positives vs. false negatives).

The F1-score helps identify which classes the model is performing well in a balanced manner, considering both precision and recall.

The number of instances per class also influences these metrics. Smaller classes (like 'bass' or 'organ') are more susceptible to wide variations in performance metrics due to the model's predictions being more significantly impacted by a few instances.

In summary, while the overall accuracy and weighted averages indicate a highly effective model, the detailed class-wise metrics reveal more nuanced insights into its performance. Understanding these metrics helps in identifying areas where the model excels and where improvements are needed, especially in handling classes with fewer instances or those that are harder to distinguish.

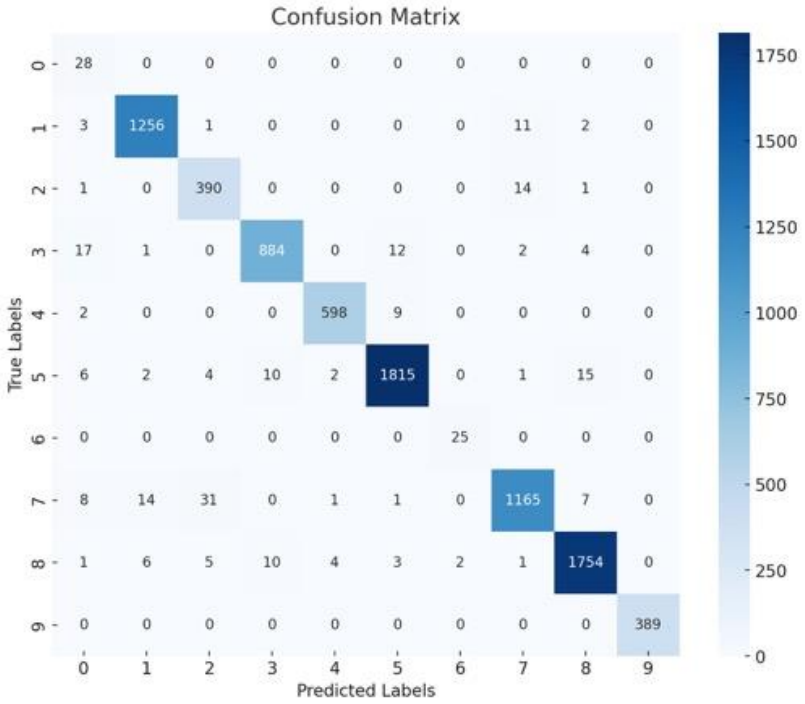


Figure 29. Confusion Matrix on Testing Split.

The confusion matrix (Figure 29) shows the number of correct and incorrect predictions made by the model, categorized by the actual and predicted classifications.

In this matrix, each cell in the matrix represents the number of samples:

- Rows (Actual Classes): The true class of the samples.
- Columns (Predicted Classes): The class predicted by the model.

Diagonal Elements (Correct Predictions)

- Bass (28 correct): The model perfectly identified all instances of Bass without any confusion, indicating a strong ability to recognize this instrument.
- Brass (1256 correct): A high number of Brass instances were correctly identified, showcasing the model's effectiveness in distinguishing Brass sounds.
- Flute (390 correct): The model accurately recognized Flute instances, demonstrating good performance in identifying Flute characteristics.
- Guitar (884 correct): Guitar was well-identified, with a large number of correct predictions, reflecting the model's capability in recognizing Guitar sounds.

- Keyboard (598 correct): Keyboard instances were correctly classified, indicating the model's proficiency with Keyboard sounds.
- Mallet (1815 correct): The model showed excellent performance in identifying Mallet instruments, with the highest number of correct predictions.
- Organ (25 correct): All Organ instances were perfectly classified, showing the model's accurate recognition of Organ sounds.
- Reed (1165 correct): The model effectively recognized Reed instruments, with a high number of correct predictions.
- String (1754 correct): String instruments were predominantly correctly identified, indicating strong model recognition capabilities.
- Vocal (389 correct): Vocal sounds were perfectly identified, showcasing the model ability to distinguish Vocal characteristics.

Off-Diagonal Elements (Misclassifications)

- Brass Misclassifications:
- Misclassified as Reed 11 times, suggesting some confusion between Brass and Reed sounds.
- Small numbers of Brass were also confused with String (2 times), indicating minor challenges in distinguishing these from Brass.
- Flute Misclassifications:
- Confused with Reed 14 times, reflecting a notable confusion between Flute and Reed sounds.
- Guitar Misclassifications:
- Misclassified as Mallet 12 times, showing confusion between Guitar and Mallet sounds.
- Also, some instances were confused with String (4 times), pointing to challenges in differentiating these instruments.
- Keyboard Misclassifications:
- Confused with Mallet 9 times, indicating a mix-up between Keyboard and Mallet sounds.
- Mallet Misclassifications:

- Notably confused with Guitar (10 times) and String (15 times), suggesting the model sometimes struggles to distinguish Mallet from these instruments.
- Reed Misclassifications:
- A significant number of Reed instances were confused with Brass (14 times) and Flute (31 times), showing particular difficulty in differentiating Reed from these sounds.
- String Misclassifications:
- Some String instances were confused with Guitar (10 times) and Reed (7 times), indicating occasional challenges in clearly identifying String sounds.

This confusion matrix illustrates the model's overall strong performance in identifying different musical instruments, with high accuracy for most classes. The diagonal elements indicate successful predictions, while the off-diagonal elements highlight specific areas where the model may confuse one instrument for another. These misclassifications provide necessary insights into where the model might be improved, potentially by focusing on distinguishing features between the instruments that are most often confused.

However, organ, with very few samples, shows perfect classification but is too small a sample to draw significant conclusions.

5.2.5.3 Comparing with benchmarks

Leading models like melspect, EfficientLEAF, and LEAF(Schlüter & Gutenbrunner, 2022; Zeghidour et al., 2021), which have set high standards in terms of accuracy. According to Table 8, the melspect model is at the forefront with an accuracy rate of 72.1%, closely trailed by the EfficientLEAF at 71.7%, and the LEAF model at 69.2%. Our model, which has achieved an accuracy of 97%, marking its presence as a strong contender in this challenging domain.

This comparison underscores the potential of our model, especially considering the ongoing advancements in audio processing and machine learning technologies. It's noteworthy that our model stands competitive against well-established benchmarks like the LEAF, highlighting its promising capabilities. The fact that our model is competitive with well-

established benchmarks like LEAF underscores its potential, especially in light of the continuous advancements in audio processing and machine learning techniques (Table 8).

Table 8. Benchmark of NSynth Dataset

Rank	Model	Accuracy	Year
1	melpect	72.1	2022
2	EfficientLEAF	71.7	2022
3	LEAF	69.2	2022

However, a critical aspect to consider is the benchmark's reliance on a diverse dataset comprising acoustic, electronic, and synthetic sounds, where the top accuracy was 72%. Our experiments, however, were focused exclusively on acoustic sounds (Table 9). The fact that our acoustic-only approach nearly matches the performance of models assessed on a broader dataset suggests that our model excels in identifying acoustic instruments. This insight leads us to believe that by expanding our model to encompass electronic and synthetic sounds, we could potentially achieve or even surpass the benchmark accuracy levels.

Table 9. Difference between our experiment and benchmark Experiment.

	Benchmark Experiment			
	Our Experiment			
Family	Acoustic	Electronic	Synthetic	Total
Bass	200	8,387	60,368	68,955
Brass	13,760	70	0	13,830
Flute	6,572	35	2,816	9,423
Guitar	13,343	16,805	5,275	35,423
Keyboard	8,508	42,645	3,838	54,991
Mallet	27,722	5,581	1,763	35,066
Organ	176	36,401	0	36,577
Reed	14,262	76	528	14,866
String	20,510	84	0	20,594
Synth Lead	0	0	5,501	5,501
Vocal	3,925	140	6,688	10,753
Total	108,978	110,224	86,777	305,979

Moving forward, the next steps involve expanding our model's capabilities to include a wider range of sound types, beyond just acoustic. By integrating electronic and synthetic sound

classification, we aim to enhance the model accuracy. Additionally, leveraging the latest advancements in machine learning and audio processing, we plan to refine our approach, focusing on improving precision and recall rates across all instrument types.

The journey ahead in musical instrument classification promises to be exciting, with ample opportunities for innovation and improvement. As we continue to build on our model's foundation, we remain committed to pushing the boundaries of what's possible in audio recognition technology, aiming to set new benchmarks in the field.

This analysis and future-oriented approach underscore our commitment to advancing the state of musical instrument classification, with the aspiration of achieving unparalleled accuracy and reliability in our model's performance.

5.3 Experiment 3: Assess NSynth Model on Noise

Building upon the large-scale model established in Experiment 2 using the NSynth dataset, this experiment aims to address Research Objective 3 (chapter 3.1.1.3) by assessing the model's performance under various noise conditions. As real-world audio environments often contain background noise, it is crucial to evaluate the robustness of our instrument recognition system in such scenarios. This experiment systematically introduces different types and levels of noise to the audio samples, allowing us to characterize the specific conditions that lead to performance degradation. By doing so, we can gain necessary insights into the model's limitations and identify potential areas for improvement in noisy environments. This assessment is critical for understanding the practical applicability of our instrument recognition system in real-world settings where clean audio signals are rarely available.

5.3.1 Overview of Experiment

In this experiment, we leverage the concept of transfer learning by utilizing a pretrained model from Experiment 2, which was trained on the NSynth dataset. This approach is extended to classify instrument sounds in the presence of noisy backgrounds. The objective is to test the

robustness and adaptability of the pretrained model to recognize instruments under less-than-ideal acoustic conditions.

The experiment involves creating a dataset of 10 different musical instruments (Table 10), with each instrument category comprising 100 samples. All samples are artificially mixed with white noise to simulate noisy background conditions. The table below summarizes the dataset composition:

Table 10. CNN model of NSynth Dataset Model.

	Number of samples	**Noise Type**	**Noise Type**	**Noise Type**
bass	100	Crowd Noise	Dog Bark	Traffic + Crowd
brass	100	Crowd Noise	Dog Bark	Traffic + Crowd
flute	100	Crowd Noise	Dog Bark	Traffic + Crowd
guitar	100	Crowd Noise	Dog Bark	Traffic + Crowd
keyboard	100	Crowd Noise	Dog Bark	Traffic + Crowd
mallet	100	Crowd Noise	Dog Bark	Traffic + Crowd
organ	100	Crowd Noise	Dog Bark	Traffic + Crowd
reed	100	Crowd Noise	Dog Bark	Traffic + Crowd
string	100	Crowd Noise	Dog Bark	Traffic + Crowd
vocal	100	Crowd Noise	Dog Bark	Traffic + Crowd
Total	**1000**			

5.3.2 Methodology for Adding Noise

The process of integrating noise into the clean instrument samples involves several steps, as outlined in the provided code snippet. Here's a detailed interpretation of each step:

1) Load a Sample from the NSynth Dataset:

The code begins by loading a single audio sample from the NSynth dataset, specifically from the gansynth_subset and the test split. This is achieved using TensorFlow Datasets (tfds) library. The shuffle_files=False parameter ensures that the samples are loaded in a deterministic order (Figure 30).

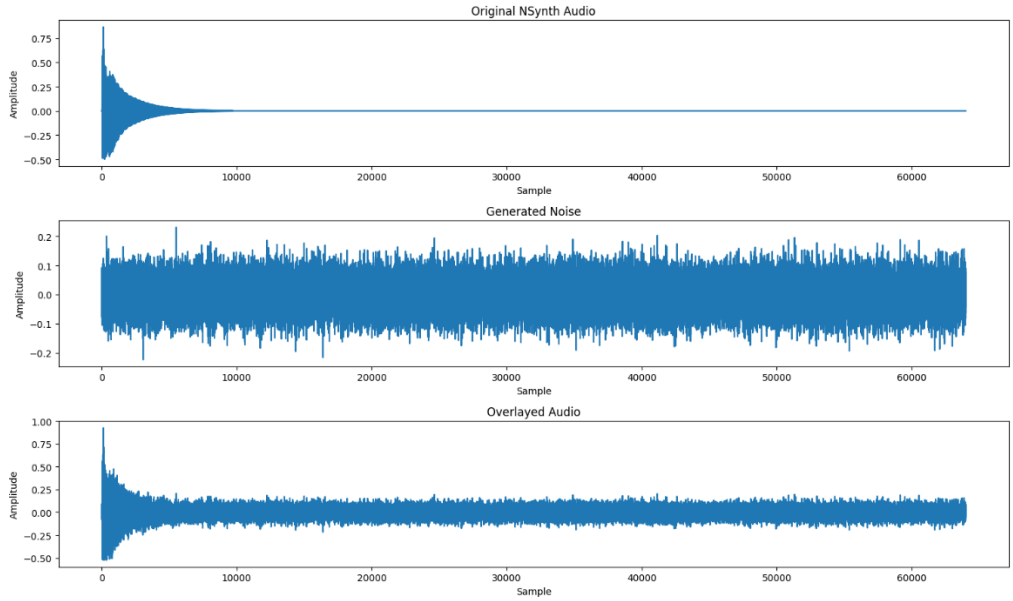


Figure 30. Mallet Waveform, Noise Waveform and Overlayed Data.

2) Extract Audio Data and Sample Rate:

The loaded NSynth test sample contains the audio waveform and its associated sample rate. The audio is extracted into a NumPy array (nsynth_audio), and the sample rate is fixed at 16,000 Hz (nsynth_sr).

3) Generate Random Noise:

Random noise is generated using a Gaussian (normal) distribution with a mean of 0 and a standard deviation of 1, matching the length of the NSynth audio sample. This noise simulates the "white noise" that will be added to the audio samples.

4) Adjust Noise Amplitude:

The amplitude of the generated noise can be adjusted to control the level of noise in the final audio mix. The noise_amplitude variable is set to 0.5, indicating that the noise volume will be scaled down to half of its original level before being added to the clean audio.

5) Overlay the Noise on the Audio:

The clean NSynth audio and the scaled noise are combined by simple addition, resulting in the final noisy audio sample (overlayed_audio). This process effectively simulates an instrument sound with a noisy background (Figure 31).

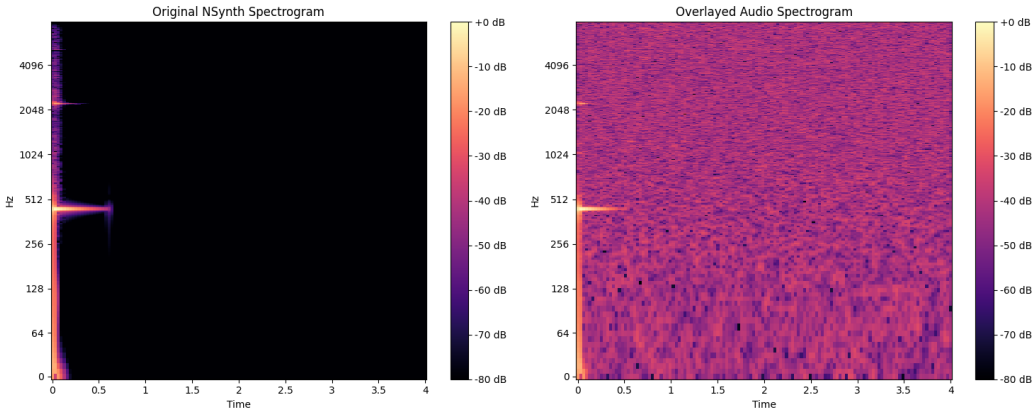


Figure 31. Original Mallet Vs. Noised Mallet Spectrogram.

To create the desired dataset of 1,000 samples, this noise addition process is repeated 100 times for each instrument class. This involves selecting or synthesizing 100 distinct audio samples per instrument category from the NSynth dataset (or an appropriately labelled dataset), then applying the described noise addition process to each sample. The repetition of this process ensures a diverse dataset that tests the ability to recognize instruments in various noisy conditions.

5.3.3 Noise Types

5.3.3.1 Comparison of Crowd Noise

The crowd noise type is akin to white noise due to its broad and uniform distribution across the frequency spectrum. Its high density ensures continuous interference with the underlying instrument sounds, resulting in a persistent auditory mask that covers all frequencies, especially the lower and middle ranges. Despite its pervasive nature, the interference is relatively predictable due to its consistent and steady profile. This makes it less challenging to isolate and classify musical instruments compared to other noise types. The consistent frequency distribution creates identifiable noise patterns that can be detected and filtered out with signal processing techniques (Figure 32).

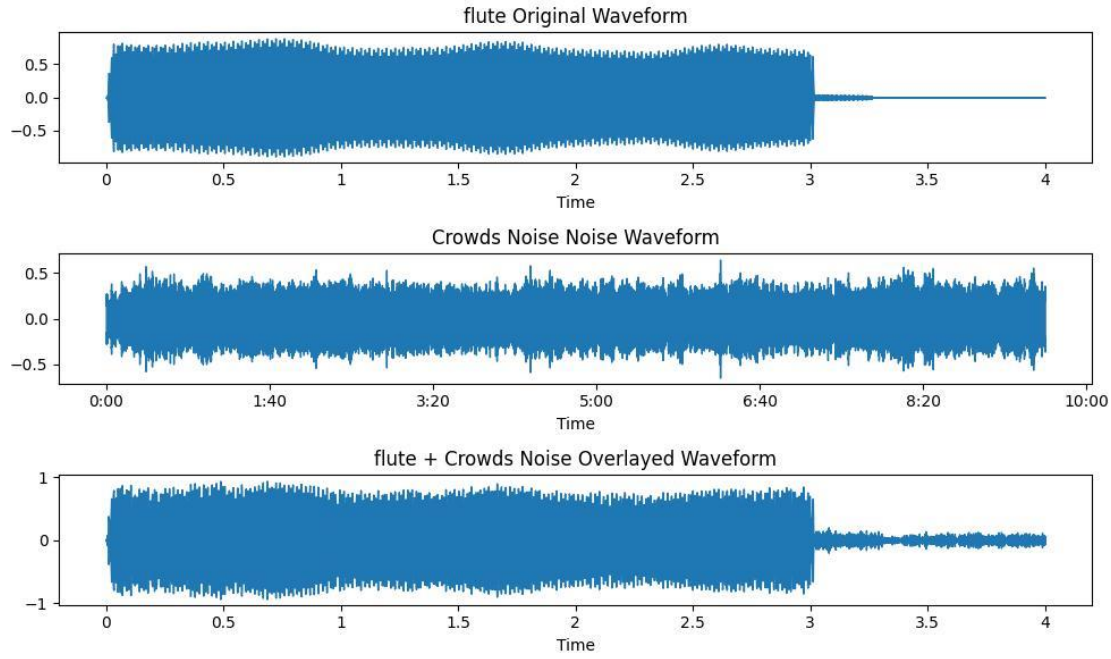


Figure 32. Original Flute Vs. Crowd Noised Flute Waveform.

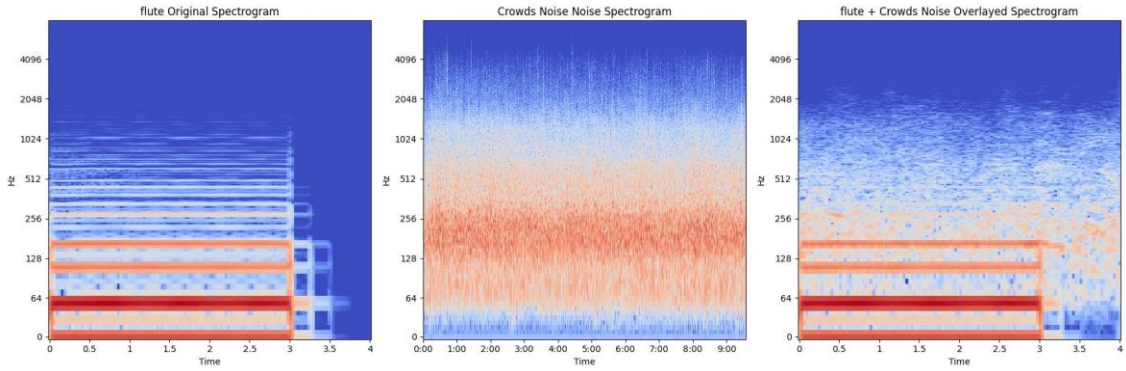


Figure 33. Original Flute Vs. Crowd Noised Flute Spectrogram.

According to Figure 33, in the first noise type, crowd noise is used as an overlay to instrument sounds, simulating a busy and chaotic environment. In Figure 32, the waveforms reveal how this noise uniformly fills the signal, producing a consistent pattern akin to white noise. The original instrument waveform maintains a consistent amplitude, while the added crowd noise forms a distinct and higher-amplitude pattern. Figure 33, the spectrogram, clearly shows how this noise permeates the frequency spectrum across time, represented in a gradient of warm colours. The y-axis indicates frequency up to 4096 Hz, and the x-axis shows time. Observing with the naked eye, we can notice crowd noise affecting the lower to mid frequencies, leading to a blurred distinction between the original and overlaid sounds.

5.3.3.2 Dog Bark Noise

Dog bark noise (Figure 34), in contrast, introduces a more rhythmic interference pattern, marked by sporadic bursts of sound. The density is lower than crowd noise, with periodic spikes rather than continuous masking. Its rhythm makes it difficult to differentiate the underlying musical instruments, as barking sounds occur intermittently, sometimes aligning with the natural rhythm of the instruments themselves. The bark frequencies generally fall within the mid-range, creating clusters of interference that overlap with the instrument frequencies, posing a considerable challenge to accurately extract and classify the instrument sounds.

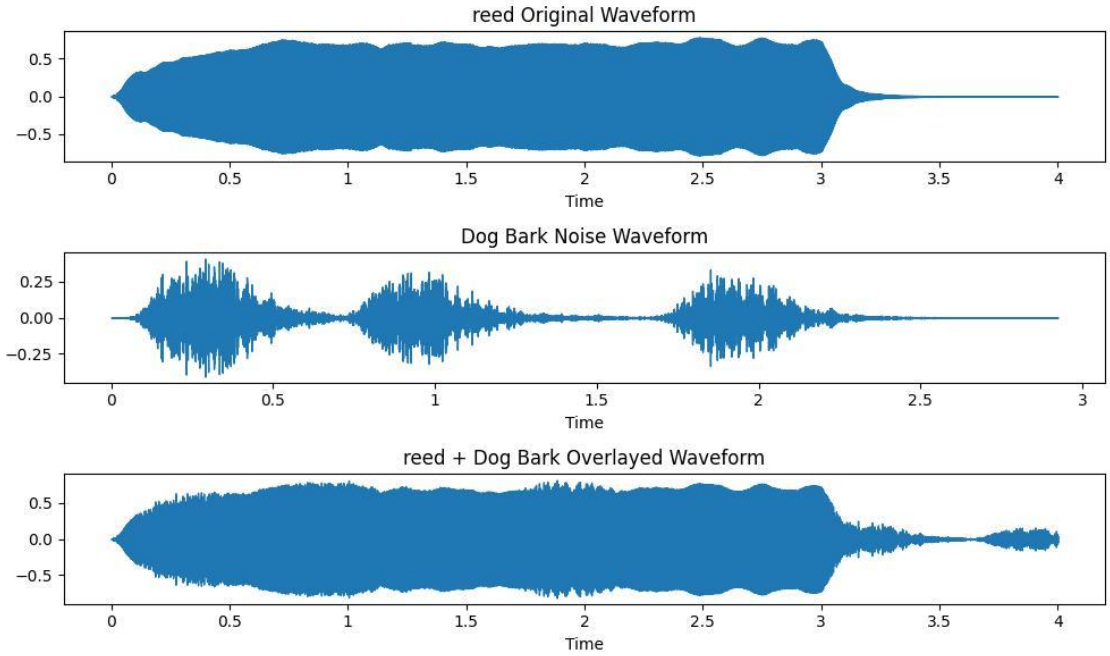


Figure 34. Original Flute Vs. Crowd Noised Flute Waveform.

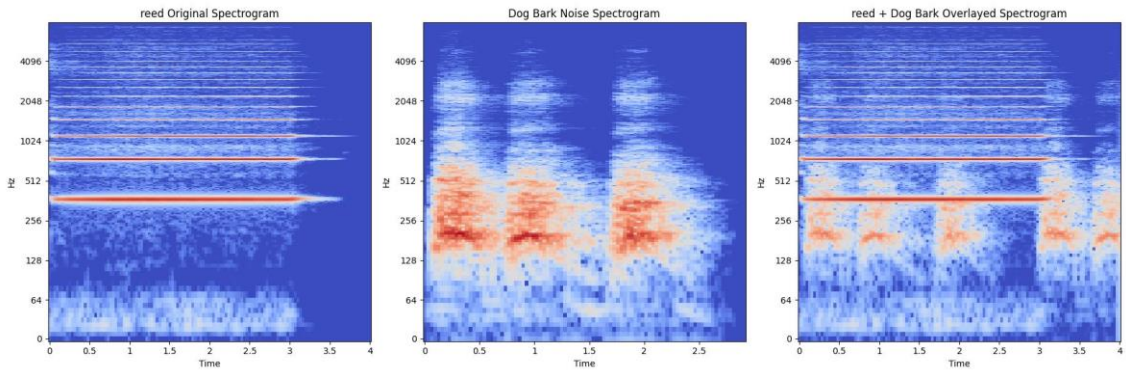


Figure 35. Original Flute Vs. Dog Bark Noised Flute Spectrogram.

The second noise type (Figure 35) introduces sporadic and distinctive dog barking. Figure 34 's waveform shows bursts of sound due to dog barks, adding intermittent peaks within the original instrument waveform. In Figure 35, the spectrogram visually captures the nature of dog bark noise, with brighter spots concentrated in the mid-frequency range. This aligns with typical bark frequencies ranging from around 300 to 3000 Hz. The resulting overlaid waveform demonstrates how the barking punctuates the background, creating high-energy areas that stand out. These noticeable peaks and valleys give an easily identifiable signature to this type of noise.

5.3.3.3 Traffic Noise

Traffic noise (Figure 36), especially when combined with crowd noise, presents the most challenging interference pattern due to its varied density. The busy traffic and crowd combination creates a cacophony that completely masks the instrument frequencies with overlapping and wide-ranging sound frequencies. The traffic component adds irregular fluctuations in amplitude, including horn blasts, engine hums, and other sounds that disrupt the rhythm and tonal clarity of instruments. The frequency range from traffic noise, coupled with the persistent crowd background, makes it exceedingly difficult to distinguish musical instruments. This noise type requires advanced techniques to separate and classify instruments amidst such overwhelming auditory clutter.

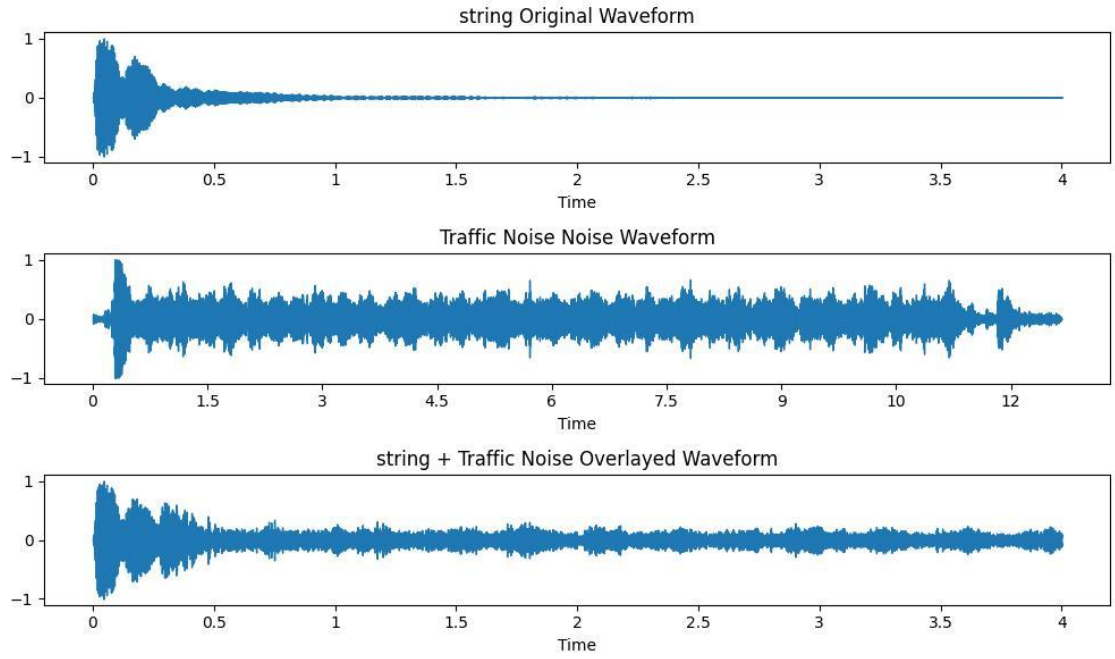


Figure 36. Original String Vs. Busy Crowd + Traffic Noised Flute Waveform.

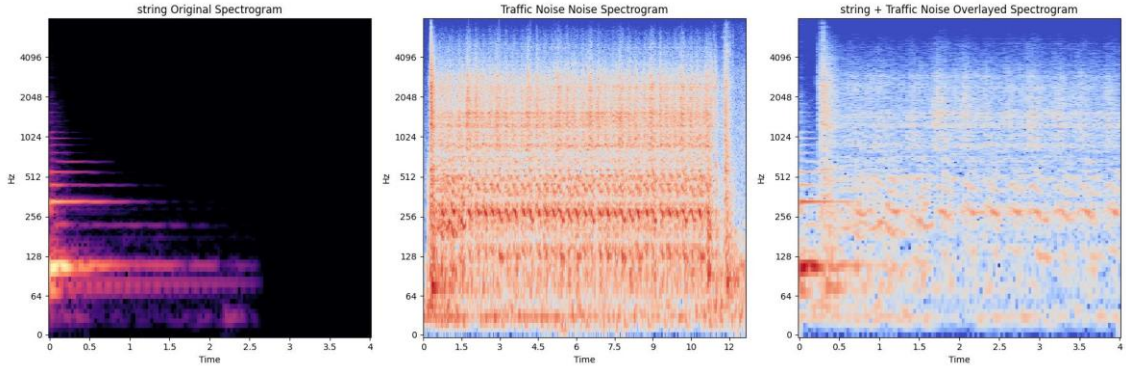


Figure 37. Original String Vs. Busy Crowd + Traffic Noised Flute Spectrogram.

The final noise type (Figure 37) introduces traffic noise, which encompasses a wide range of frequencies and variations in amplitude. Figure 36 's waveform reflects the complex and fluctuating patterns of car engines, horns, and background hums. This noise covers the instrument waveform with both consistent and irregular bursts. Figure 37 's spectrogram reveals a rich display of traffic sounds stretching across various frequency ranges, with prominent concentration around the low and mid frequencies. Observing this, we see how traffic noise can mask the instrument sounds across the spectrum due to its broad and consistent influence, creating visual noise patterns that closely resemble everyday urban sounds.

5.3.4 Statistical Analysis and Empirical Outcome Assessment

5.3.4.1 Statistical Analysis and empirical outcome assessment of Crowd Noise

This Table 11 presents the performance metrics of a machine learning model tasked with classifying musical instruments in noisy environments. The metrics include precision, recall, F1-score, and the number of samples for each class. Let's break down these metrics and interpret the performance for each instrument class, highlighting the highest and lowest performances, and making assumptions based on the observed outcomes.

Table 11. Result of Crowd Noisy background.

Class	Precision	Recall	F1-Score	Number
bass	0.92	**1**	0.96	100
brass	0.73	**0.49**	0.59	100
flute	0.76	0.86	0.81	100
guitar	**0.6**	0.8	0.69	100
keyboard	0.71	0.91	0.79	100
mallet	0.74	0.14	**0.24**	100
organ	0.72	**1**	0.84	100
reed	0.63	0.58	0.6	100
string	0.76	0.74	0.75	100
vocal	**0.97**	0.98	**0.98**	100
Accuracy	0.75	0.75	0.75	1000
Macro average	0.75	0.75	0.72	1000
Weighted average	0.75	0.75	0.72	1000

Highest Precision: The vocal class has the highest precision (0.97), indicating that when the model predicts a sample as vocal, it is correct 97% of the time. This high precision suggests that vocal sounds are distinct enough for the model to accurately identify them, even in noisy conditions.

Highest Recall: Both the bass and organ classes have a perfect recall of 1, meaning the model correctly identified all bass and organ samples. This could be due to the unique sound frequencies or patterns these instruments produce, making them easily distinguishable from noise.

Highest F1-Score: The vocal class also has the highest F1-score (0.98), signifying an excellent balance between precision and recall. Vocals likely have distinctive features that are less affected by noise, leading to high accuracy in both detecting and classifying vocal sounds.

Lowest Precision and Recall: The guitar class shows relatively low precision (0.6) and recall (0.8), indicating challenges in correctly identifying guitar sounds. The mallet class has even lower precision (0.74) but significantly lower recall (0.14), highlighting a substantial issue with false negatives, where many mallet samples are missed by the model. These lower performances could be attributed to the characteristics of guitar and mallet sounds being more susceptible to masking by white noise, making them harder to distinguish.

Assumptions on Performance Variations: Instruments like guitar and reed may have lower performance metrics because their sound characteristics overlap more with the frequency spectrum of white noise or because their distinctive features are easily masked by noise.

The performance for bass and organ, compared to previous testing, can be attributed to the balanced dataset used in this experiment. Each class has **100** samples, eliminating the imbalance issue seen in the TensorFlow NSynth data, where bass and organ had only **28 and 25** samples, respectively. This balance allows for better learning of characteristics specific to each instrument, potentially reducing the impact of overfitting despite the same number of training samples.

Overall Accuracy and Averages: The overall accuracy of the model is 0.75, with macro and weighted averages for precision, recall, and F1-score also around 0.75, indicating a consistent performance across different instrument classes. However, the macro average F1-score slightly lower at 0.72 suggests some classes (like mallet) significantly underperform, pulling down the overall average despite high scores in other classes.

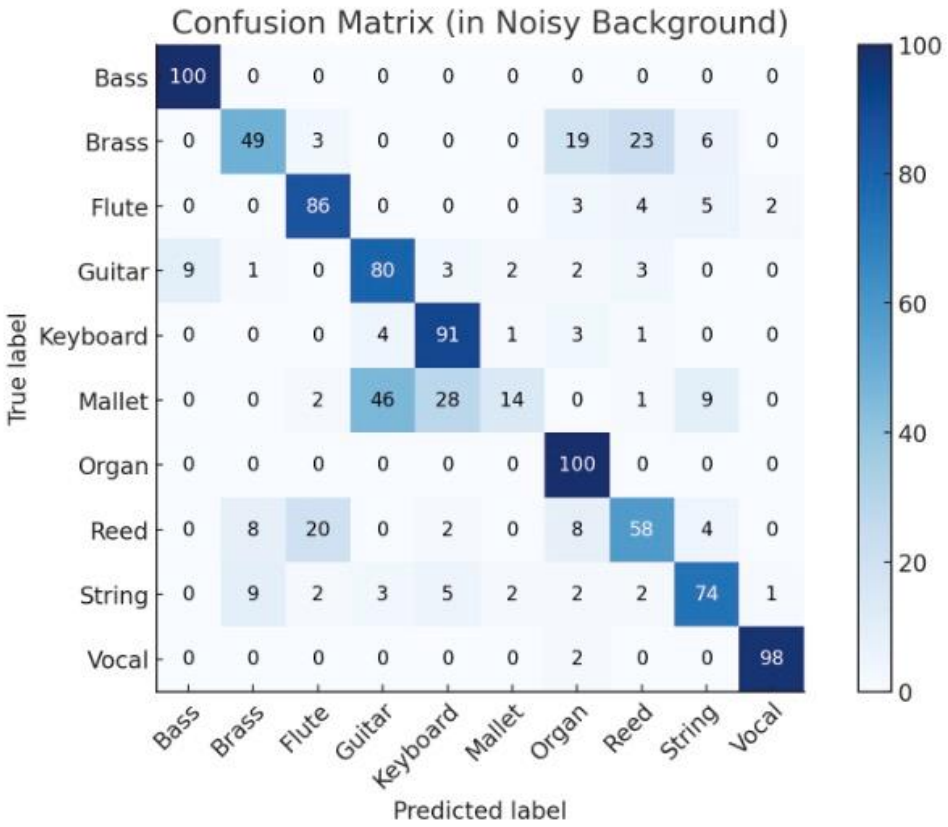


Figure 38. Confusion Matrix of Crowd Noise.

The confusion matrix (Figure 38) is structured as follows for 10 classes (assuming they are ordered from 1 to 10 as bass, brass, flute, guitar, keyboard, mallet, organ, reed, string, vocal),

1. Bass (Class 1): Perfectly classified with 100 correct predictions and no misclassifications. This indicates that the model is highly effective at identifying bass sounds without confusion with other instrument sounds.
2. Brass (Class 2): Out of 100 samples, 49 are correctly classified. However, there are significant misclassifications, with 19 mistaken for organ, 23 for reed, and smaller counts for flute and string. This suggests that brass sounds may share characteristics with organ and reed sounds, leading to confusion.
3. Flute (Class 3): 86 out of 100 samples are correctly classified, with some confusion with reed (4) and string (5), and a minor mix with brass (3) and vocal (2). The flute's confusion with reed and string might be due to similar pitch or tonal qualities.

4. Guitar (Class 4): 80 correct classifications, with notable confusion with mallet (46) and keyboard (28), indicating that certain guitar tones might resemble those instruments' sounds.
5. Keyboard (Class 5): 91 correctly classified, with minor confusion mostly with guitar (4). This high accuracy suggests distinct characteristics for keyboard sounds that the model can recognize well.
6. Mallet (Class 6): This class shows significant difficulty, with only 14 correctly identified and high misclassification with guitar (46) and string (28), suggesting a challenging distinction between these instruments' sounds.
7. Organ (Class 7): Perfect classification with 100 correct predictions, indicating distinct and recognizable sound features that the model captures very well.
8. Reed (Class 8): 58 correct predictions with confusion among brass (8), flute (20), and minor confusion with organ (8) and string (4), reflecting the overlapping sound characteristics between reed and these instruments.
9. String (Class 9): 74 correctly classified, with misclassifications spread across brass (9), flute (2), guitar (3), keyboard (5), and mallet (2), indicating some overlap in the acoustic features recognized by the model.
10. Vocal (Class 10): Highly accurate with 98 correct predictions, showing a clear distinction of vocal sounds from instrumental sounds, except for minor confusion with organ (2).

High Accuracy Classes are Bass, organ, and vocal classes are highly accurately classified, suggesting distinct acoustic features that are easily recognizable by the model.

Challenging Classes are Brass, guitar, and mallet show considerable confusion with other classes, indicating shared characteristics that the model struggles to differentiate.

Possible Reasons for Misclassification: The confusion between classes like brass with organ and reed, or guitar with mallet, could be due to similar harmonic structures, timbral qualities, or pitch ranges that these instruments share. The model's difficulty in distinguishing these classes suggests a need for more distinctive features or enhanced preprocessing to better capture the unique aspects of each instrument's sound.

Improvement Strategies may include enhancing feature extraction, increasing the diversity and size of the training dataset, and employing more sophisticated model architectures could help mitigate these classification challenges.

5.3.4.2 Statistical Analysis and Empirical Outcome Assessment of Dog Bark Noise

The Table 12 provided shows the precision, recall, and F1-score metrics of a machine learning model when classifying various musical instruments in the presence of dog bark noise. The noise introduces sporadic and intense bursts of interference. Here's a detailed breakdown of the results:

Table 12. Result of Dog Bark Noisy background.

Class	Precision	Recall	F1-Score	Number
bass	0.35	1.00	0.52	100
brass	0.91	0.98	0.94	100
flute	1.00	0.47	0.64	100
guitar	0.86	0.98	0.92	100
keyboard	0.97	0.61	0.75	100
mallet	1.00	0.25	0.40	100
organ	1.00	0.87	0.93	100
reed	0.89	0.94	0.91	100
string	1.00	0.64	0.78	100
vocal	0.99	1.00	1.00	100
Overall				
Accuracy			0.77	1000
Macro Avg	0.90	0.77	0.78	1000
Weighted Avg	0.90	0.77	0.78	1000

Highest Precision and Recall:

The vocal class has a near-perfect precision (0.99) and recall (1.00), indicating that the model effectively distinguishes vocals even amidst barking noise. Other high-performing classes include brass and organ, each with a precision of 0.91 and 1.00, respectively. This consistency indicates distinctive sound features that are easier to identify and classify despite the sporadic barking interference.

Low Performance Classes:

Bass and flute classes struggle the most. The bass class has a low precision of 0.35, while flute's recall drops to 0.47. The interference caused by dog bark noise seems to overlap heavily with these instruments' natural frequency ranges, making it difficult for the model to distinguish the original signal from the noise.

Overall Accuracy and Averages:

The overall accuracy of the model is 0.77, which is an improvement from the crowd noise experiment. The macro and weighted averages for precision are both 0.90, suggesting that the model can maintain a high level of confidence in identifying specific instruments. However, the macro and weighted recall averages are both 0.77, indicating challenges in consistently finding the right classifications for all classes.

Confusion Matrix Analysis:

The confusion matrix (Figure 39) is a visual representation of the classification results across different instrument classes. Here's an analysis of each class's classification:

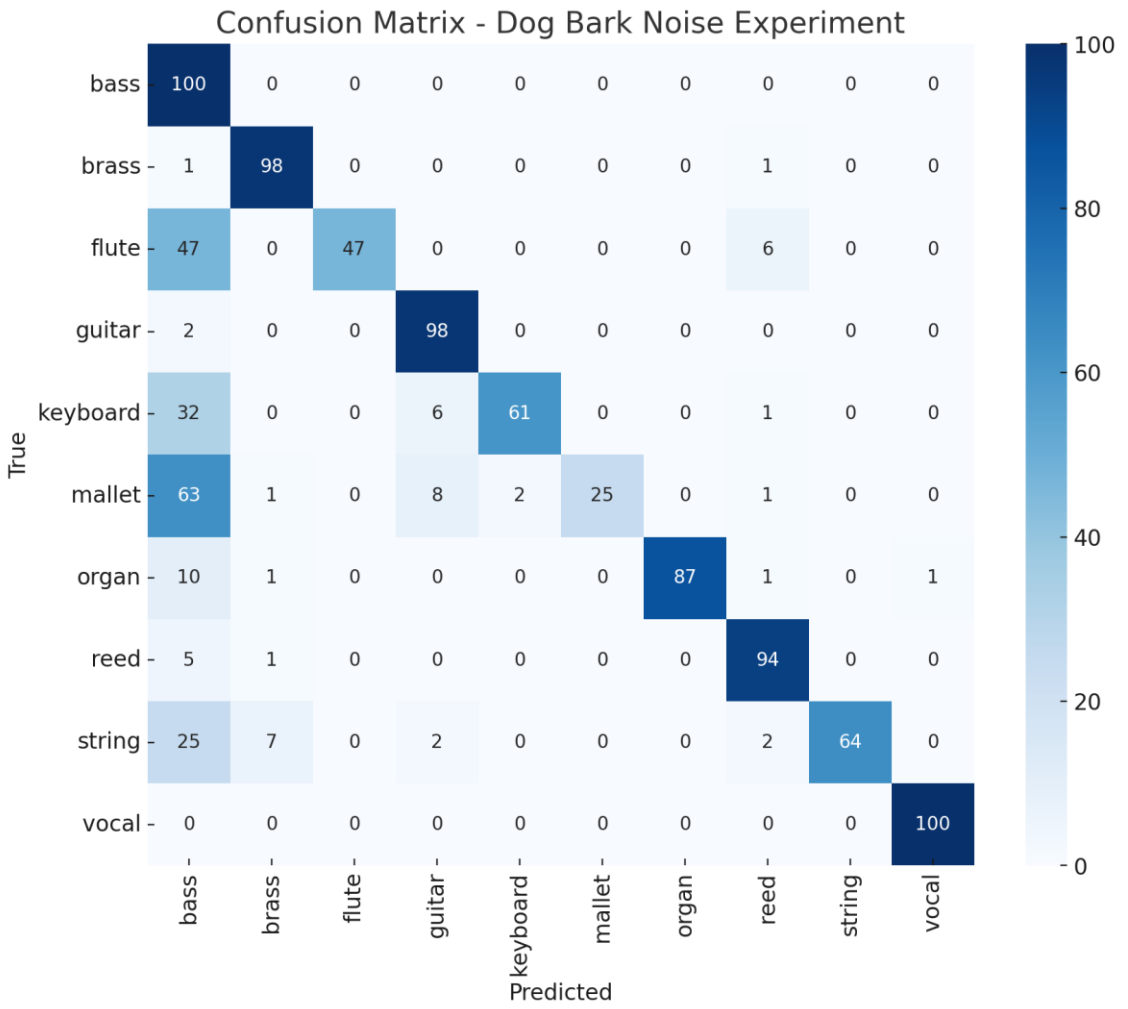


Figure 39. Confusion Matrix of Dog Bark Noise.

Bass:

All 100 bass samples are correctly classified, showcasing perfect recall despite low precision. This is because the matrix shows no confusion with other classes. However, the low precision suggests that other classes, such as mallet or string, are often misclassified as bass.

Brass:

With 98 correctly classified and just two misclassifications (1 reed, 1 brass), the brass class achieves high precision and recall. Misclassifications are minimal.

Flute:

Only 47 flute samples are correctly classified. There is significant confusion with other instruments, especially bass (47 misclassifications), due to overlapping frequency ranges and barking noise.

Guitar:

A strong performance with 98 correct predictions and just two bass misclassifications. This class is relatively distinct despite noise interference.

Keyboard:

Keyboard exhibits 61 correct classifications out of 100, with notable confusion with bass (32) and guitar (6). This overlap may be due to the rhythmic interference of barking.

Mallet:

Mallet has a very low recall, with only 25 correctly identified samples. The matrix reveals significant misclassifications into bass (63), guitar (8), and brass (1). The barking noise may be disrupting its characteristic features.

Organ:

The organ class remains distinct with a high precision and recall, achieving 87 correct classifications and only a few misclassifications into brass (1) and other instruments.

Reed:

Reed has high recall, achieving 94 correct classifications and only a few misclassifications into brass (1) and flute (6).

String:

The string class is mostly accurate with 64 correct classifications but has some overlap with bass (25), brass (7), and keyboard (2).

Vocal:

The vocal class has perfect recall with no misclassifications. This suggests the vocal sound is very distinctive even under challenging noise.

Overall, the dog bark noise experiment demonstrates that some instrument classes remain distinctive while others overlap significantly, affecting classification.

5.3.4.3 Statistical Analysis and Empirical Outcome Assessment of Busy Traffic and Crowd Noise

The Table 13 shows the results of Dog Bark noise type.

Table 13. Result of Dog Bark Noisy background.

Class	Precision	Recall	F1-Score	Support
bass	0.60	0.39	0.47	100
brass	1.00	0.05	0.10	100
flute	0.32	0.99	0.48	100
guitar	0.00	0.00	0.00	100
keyboard	0.00	0.00	0.00	100
mallet	0.00	0.00	0.00	100
organ	0.00	0.00	0.00	100
reed	0.07	0.45	0.13	100
string	0.33	0.01	0.02	100
vocal	0.00	0.00	0.00	100
Overall				
Accuracy			0.19	1000
Macro Avg	0.23	0.19	0.12	1000
Weighted Avg	0.23	0.19	0.12	1000

Highest Precision and Recall

The brass class achieves perfect precision (1.00), meaning that all samples classified as brass by the model are indeed correct. However, it has a very low recall (0.05), indicating that the model only identifies a small portion of the actual brass samples. Most brass samples are misclassified into other categories, particularly reed, due to overlapping tonal qualities and timbre. The flute class, conversely, displays an opposite pattern with extremely high recall (0.99) but low precision (0.32). This means that the model detects almost all flute samples but incorrectly identifies many other instrument samples as flutes. The confusion stems from the similarity of certain flute frequencies with other instruments and the broad frequency masking from traffic noise.

Low-Performing Classes

The classes with zero scores in precision, recall, and F1-score (guitar, keyboard, mallet, organ, and vocal) suffer extensively under traffic noise. These instruments are severely masked by environmental sounds and misclassified into other categories, demonstrating the masking effect that urban noise has on their distinctive features. The keyboard and guitar classes suffer the most due to their tonal overlap with other instruments. Vocals, despite being very distinctive in clean environments, are not detected even once because of the substantial interference from traffic and crowd noise. Mallet and organ, often recognizable, also lose their distinguishing features due to heavy masking, resulting in zero accurate predictions.

Overall Accuracy and Averages

The overall accuracy of the model, at 0.19, demonstrates significant confusion among the instrument classes. The macro and weighted averages for precision, recall, and F1-score also remain low, indicating that the model struggles across all instrument categories to accurately identify their distinctive features amid strong background interference. Traffic noise proves to be particularly challenging because of its broad range of frequencies and amplitude variations.

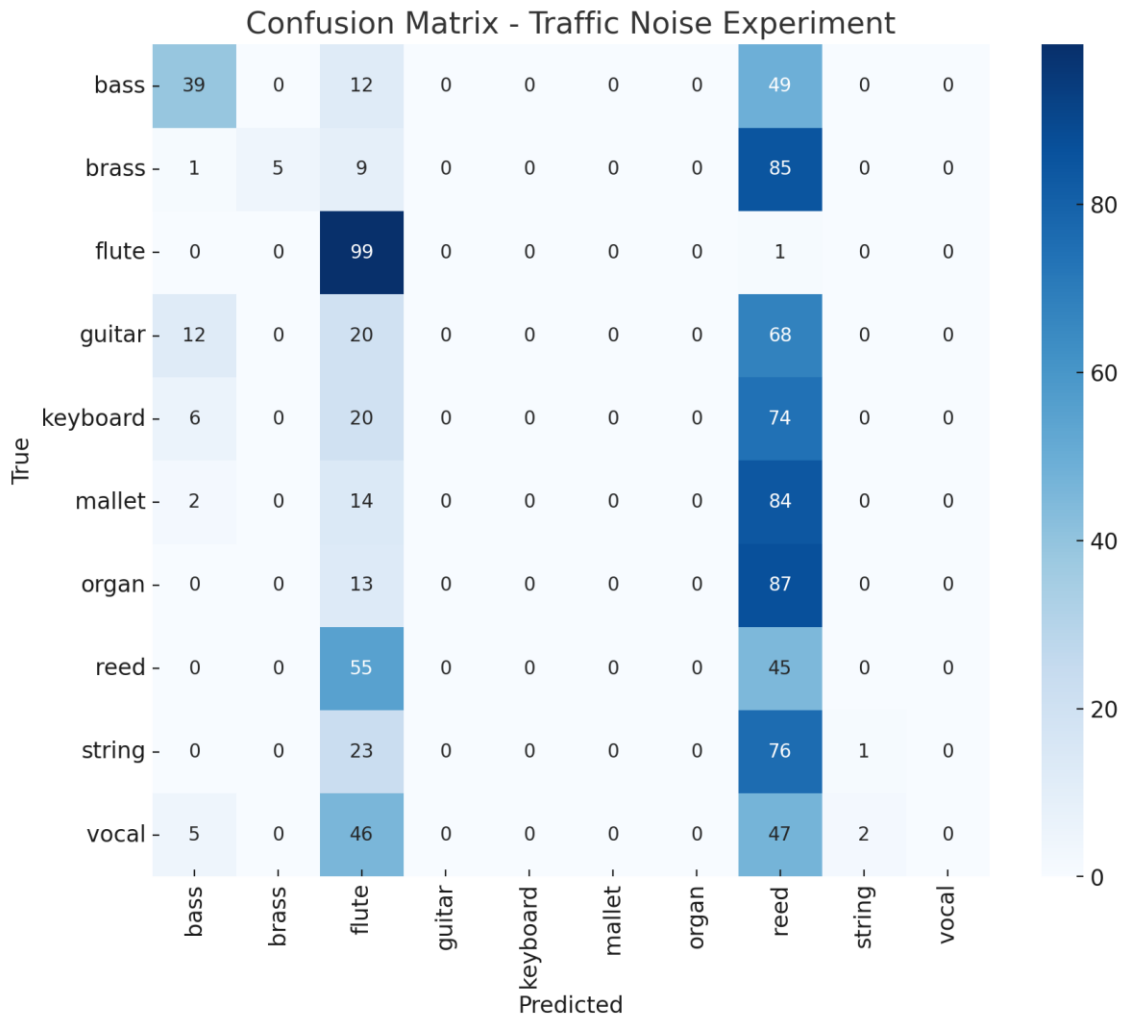


Figure 40. Confusion Matrix of Crowd Noise.

Confusion Matrix Analysis

The confusion matrix (Figure 40) provides insights into specific areas where the model struggles:

Bass:

Only 39 bass samples are correctly classified, while the majority are misclassified into the reed class (49 instances). Traffic noise creates interference that heavily overlaps with the bass's distinctive low-frequency tones, causing frequent misclassifications.

Brass:

Despite achieving high precision, brass is only accurately classified five times. Most brass samples are mistaken for reed (85 misclassifications) because the harmonic structure of brass and reed can overlap in crowded acoustic environments.

Flute:

Nearly all flute samples are correctly classified (99 instances), but the flute class also receives substantial misclassifications from other instruments. This results in low precision due to traffic noise interference with overlapping harmonic structures.

Guitar:

Only 20 guitar samples are correctly classified, with a high rate of misclassification into reed and other instruments. The guitar's broad frequency range is affected by the masking noise, leading to significant overlap with other classes.

Keyboard:

The keyboard class is rarely distinguished from other instrument classes, with almost all samples misclassified due to the extensive masking effect.

Mallet:

No mallet samples are correctly predicted, reflecting the significant challenge in differentiating this instrument from the noise background.

Organ:

Similarly, organ samples are entirely lost to traffic noise masking, resulting in zero accurate classifications.

Reed:

Reed is accurately classified 45 times, but many reed samples are misclassified due to confusion with other classes.

String:

The string class sees a high rate of misclassification due to its frequency overlap with other instruments, especially bass and reed.

Vocal:

Vocals are masked so heavily that none of the samples are classified accurately, highlighting the severity of traffic and crowd noise interference.

Overall Observations

This experiment demonstrates that traffic and crowd noise can profoundly challenge machine learning models tasked with identifying musical instruments. The interference causes overlapping tonal structures and broad-spectrum masking that make classification exceptionally difficult. For future improvements, enhanced noise filtering through signal processing, more robust data augmentation techniques, and sophisticated feature extraction strategies could be explored to boost accuracy. Further experimentation with more diverse datasets and advanced model architectures may yield better recognition performance.

5.3.5 Discussion of Noise Analysis

Analysing the Impact of Noise (chapter 3.1.1.3) Types on Instrument Classification Models

Different noise types affect the classification the ability of model to recognize musical instruments to varying extents, providing necessary insights into the model's strengths and weaknesses.

Crowd Noise

Crowd noise, akin to white noise due to its broad and relatively consistent distribution across the frequency spectrum, poses the least challenging interference pattern. Its high density provides continuous masking of instrument sounds, particularly in the lower and middle frequency ranges. This noise ensures a steady auditory interference that covers all tonal ranges. However, its predictable and steady profile allows signal processing techniques to isolate instruments more effectively. The spectrogram reveals a broad, relatively uniform masking pattern, which retains identifiable gaps that permit the recognition of distinctive musical instruments such as vocals, bass, and organ. Despite some tonal overlap, the model exhibits reasonable classification performance across most instrument classes in this noisy environment. The high accuracy in

identifying instruments with unique frequency patterns showcases the resilience and its ability to handle consistent masking.

Dog Bark Noise

Dog bark noise presents a more rhythmic interference pattern that contrasts sharply with crowd noise. Its periodic spikes introduce clusters of interference in the mid-range frequencies, intermittently aligning with the natural rhythm of musical instruments. The spectrogram for dog bark noise displays concentrated peaks of interference that coincide with instruments like flute and keyboard. This complicates classification as the rhythmic barking can mask the distinct tonal structures of these instruments, significantly reducing classification accuracy. However, the model remains capable of distinguishing some instruments, such as vocals and brass, which have more distinctive tonal patterns. The sporadic bursts of barking provide identifiable characteristics but increase the difficulty of accurate classification due to their periodic and unpredictable nature. Instruments relying on precise rhythmic structures are most affected by this form of interference.

Busy Traffic and Crowd Noise

The combination of traffic and crowd noise presents the most formidable interference pattern, creating a cacophony that masks the distinctive frequencies of musical instruments. Traffic noise encompasses a vast range of frequencies and amplitude variations, including horn blasts, engine hums, and other environmental sounds. The blending of traffic with crowd noise produces the acoustic environment that severely disrupts the instrument signals. Spectrograms reveal a pattern of overlapping and fluctuating frequencies, making it nearly impossible for the model to distinguish individual instruments. Classification accuracy plummets, and only a few instruments can be detected reliably. This combination of noise leads to high misclassification rates, especially for instruments like guitar, keyboard, organ, and mallet, which have overlapping spectra with the noise. The heavy masking effect demonstrates the challenges of distinguishing instruments in an urban soundscape.

Overall Discussion

Each noise type presents unique challenges to the ability to classify musical instruments accurately. Crowd noise, despite its high density, maintains a relatively predictable interference pattern that allows the model to perform reasonably well. Dog bark noise introduces rhythmic and unpredictable interference, leading to noticeable classification errors for some instruments. However, the traffic and crowd combination proves most challenging due to the overwhelming interference that completely disrupts tonal clarity. Improving filtering techniques, feature extraction, and model architectures will be vital in enhancing recognition across all noise types. Experimentation with more diverse datasets and noise conditions will bolster the model's robustness and adaptability, allowing it to better navigate complex soundscapes. Additionally, strategies like data augmentation, specialized pre-processing, and noise-adaptive learning models could be explored to address each noise type more effectively.

5.4 Experiment 4: Assess NSynth Model on Polyphonic Data with EMR metric.

5.4.1 Dataset

Our experiment utilized the NSynth dataset, specifically the $gansynth\_subset$. This dataset was chosen due to the collection of musical instrument sounds is categorized into various families such as bass, brass, flute, guitar, keyboard, mallet, organ, reed, string, vocal as we pretrained in Experiment 2 and 3.

5.4.1 dataset generation

To evaluate the performance of our model, we generated a dataset that encapsulates every possible combination of ten distinct musical instruments. These instruments, which encompass a wide range of timbral characteristics, are encoded in a binary fashion where each bit in a 10-bit label corresponds to the presence (1) or absence (0) of a particular instrument in the audio sample. Table 14 presents the labelling system employed for individual instruments and their combinations.

Table 14. Possibilities of combinations.

Type	Label
No Instrument	[0, 0, 0, 0, 0, 0, 0, 0, 0, 0]
Bass Solo	[1, 0, 0, 0, 0, 0, 0, 0, 0, 0]
Brass Solo	[0, 1, 0, 0, 0, 0, 0, 0, 0, 0]
Flute	[0, 0, 1, 0, 0, 0, 0, 0, 0, 0]
Guitar	[0, 0, 0, 1, 0, 0, 0, 0, 0, 0]
Keyboard solo	[0, 0, 0, 0, 1, 0, 0, 0, 0, 0]
Mallet Solo	[0, 0, 0, 0, 0, 1, 0, 0, 0, 0]
Organ Solo	[0, 0, 0, 0, 0, 0, 1, 0, 0, 0]
Reed Solo	[0, 0, 0, 0, 0, 0, 0, 1, 0, 0]
String Solo	[0, 0, 0, 0, 0, 0, 0, 0, 1, 0]
Vocal Solo	[0, 0, 0, 0, 0, 0, 0, 0, 0, 1]
Bass & Brass	[1, 1, 0, 0, 0, 0, 0, 0, 0, 0]
Bass & Flute	[1, 0, 1, 0, 0, 0, 0, 0, 0, 0]
... (other duo combinations)	
Bass & Vocal	[1, 0, 0, 0, 0, 0, 0, 0, 0, 1]
... (other trio combinations)	
... (other solo duo trio quartet quintet sextet septet octet nonet decet combinations)	
All instruments	[1, 1, 1, 1, 1, 1, 1, 1, 1, 1]
Total Combinations: 1024	

Each row in Table 14 represents a unique configuration of instruments, ranging from the absence of all instruments (silence) to the presence of all instruments playing simultaneously (ensemble). The dataset is systematically structured to include solo performances (where only one instrument is present), duets (combinations of two instruments), trios, quartets, and so on, culminating in a full decet representing an ensemble of all ten instruments. The binary labels are constructed as follows:

No Instrument: The label [0, 0, 0, 0, 0, 0, 0, 0, 0, 0] denotes silence, where no instrument is playing.

Solo Performances: A solo performance is indicated by a label with a single '1' in the position corresponding to the instrument, such as [1, 0, 0, 0, 0, 0, 0, 0, 0, 0] for a bass solo.

Duets: Duets are labelled with two '1's in the positions corresponding to the instruments involved, for instance, [1, 1, 0, 0, 0, 0, 0, 0, 0, 0] for a bass and brass duet.

Ensembles: As the number of instruments increases, the labels reflect this by having multiple '1's, each position still corresponding to a specific instrument within the ensemble.

The dataset generation process ensures a balanced representation of all possible instrument combinations, resulting in a total of 2^10 or 1024 unique labels. This exhaustive enumeration allows for the evaluation of the model's capability to distinguish between a diverse set of instrumental arrangements, from the simplest to the most complex.

5.4.2 Model of the Sample Experiment.

We us transfer learning by reusing a pretrained model of separate binary classifier for each instrument family, adopting an OvA strategy to accommodate the multi-label nature of our dataset. Each model was trained using the Adam optimizer and binary cross-entropy loss, considering both the presence and absence (augmented with negative examples from other families) of the specific instrument in the training samples.

5.4.3 Workflow

This section outlines the workflow for generating and evaluating test samples to assess the performance of convolutional neural network models trained for multi-label classification of musical instruments using the NSynth dataset. The workflow is structured to ensure a balanced representation of instruments in the evaluation and to reflect the complexity of musical compositions ranging from solo performances to decets.

5.4.3.1 Sample Extraction for Balanced Testing

The primary objective in the initial phase is to establish a balanced test dataset. Given the NSynth dataset's division into training, validation, and test sets, our strategy involves extracting 100 samples for each instrument category. This extraction ensures that our evaluation covers the dataset's diversity. However, for instrument families with insufficient samples in the training/validation set, such as organ and bass, we supplement the deficit with samples from the training set. This approach guarantees a uniform representation across all instrument families, addressing the dataset's inherent imbalances.

5.4.3.2 Test Sample Generation

Test samples are meticulously crafted to represent a wide array of musical compositions:

- No Instrument: For the baseline case of no instrument, we generate test samples using white noise. This class serves as a control, testing the models' ability to correctly identify the absence of any musical instrument.
- Solo to Decet: The creation of test samples for solo to decet performances involves a systematic overlay process. Starting with solos, we use original, unaltered samples from our balanced dataset. As we progress to duos and beyond, we select distinct instrument samples and overlay their spectrograms to simulate ensemble performances. This process is repeated with increasing complexity, adding one distinct instrument at a time, up to decets. The overlay technique mimics real-world scenarios where multiple instruments are played together, producing a rich, layered sound.

For each class from "No Instrument" to "Decet," we ensure the generated test samples accurately represent the intended composition of instruments. This is crucial for evaluating the models' performance across a spectrum of musical complexity.

In generating ensemble samples (duos, trios, etc.), care is taken to select distinct instruments for overlaying, ensuring no repetition and maintaining the uniqueness of each test case.

To train and test our instrument classification model, we generate a diverse dataset that encompasses a wide range of possible instrument combinations. Given that we have ten distinct instruments, each sample in our dataset can be represented by a 10-bit binary label, where each bit corresponds to the presence (1) or absence (0) of an instrument. This results in (2^{10})=1024 unique possible labels, each representing a different combination of instruments ranging from no instrument to all instruments playing together.

Due to the exponential growth of the combination space with the number of instruments, it is computationally prohibitive to include all possible combinations in the dataset, especially considering the GPU memory constraints during model training. Therefore, we streamline the dataset to include a manageable yet representative set of combinations, as summarized in Table 15.

Table 15. Representative Combinations for Dataset Construction:

Class	Number
No Instrument	100
Random Solo Pieces	100
Random Duo	100
Random Quartet	100
Random Quintet	100
Random Sextet	100
Random Septet	100
Random Octet	100
Random Nonet	100
Random Decet	100
Total	1100

The use of white noise for the "No Instrument" class and the overlaying technique for ensembles are pivotal in creating a test set that challenges the models' classification capabilities, testing both their sensitivity and specificity.

The dataset's design is to cover a broad spectrum of ensemble combinations, ensuring that the model is exposed to various contexts during training. The 'No Instrument' class contains 100 samples of silence, serving as a control group. For solo pieces, instead of creating 100 samples for each instrument, which would total 1000 samples and be exhaustive on memory, we generate 100 random solo pieces where only one instrument is present per sample but varies across the dataset. This randomness is applied similarly for duos, quartets, quintets, and up to decets, providing a balanced representation of ensemble sizes.

By adopting this approach, we efficiently utilize GPU resources while ensuring a rich and diverse dataset. The reduced number of samples allows us to fit the training process within GPU memory limits, and the strategic selection of samples maintains the robustness and generalizability of the model.

5.4.3.3 Describing spectrograms of test sample.

In this section, the 3 examples spectrograms are illustrated, we skip trio to nonet because they looks are quite same.

1) No Instrument (Noise) Spectrogram:

Figure 41 depicts a spectrogram of a random noise signal. In a spectrogram, time is represented on the x-axis, and frequency is on the y-axis. The colour intensity indicates the amplitude (or loudness) of various frequencies at different times. Here, the 512 hop length and 2048 FFT size parameters determine the time-frequency resolution of the spectrogram. The seemingly random distribution of colours, without any distinct patterns or lines, suggests that the signal lacks harmonic structure, which is typical for noise. Thus, it is expected that a well-trained classifier should identify this as 'no instrument'.

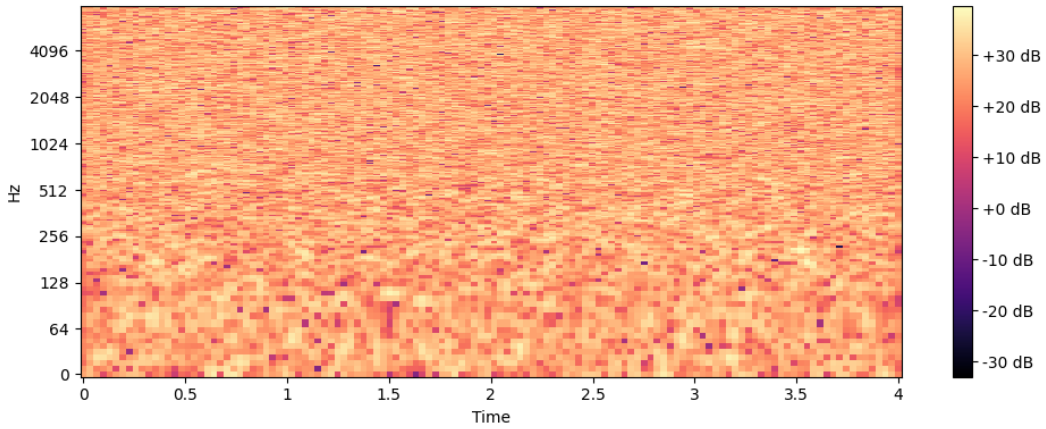


Figure 41. Test Sample : no instrument.

2) Organ Solo Spectrogram:

In the second image (Figure 42), the spectrogram likely represents an organ piece. The vertical lines correspond to the individual notes played, with lower notes at the bottom and higher notes at the top of the y-axis. The colour signifies the intensity of each frequency: warmer colors (like orange) represent higher energy at that frequency and time, while cooler colors (darker regions) represent lower energy. This image shows several horizontal bands, which may indicate sustained notes that are characteristic of organ music. A classifier designed for instrument recognition should ideally categorize this as an organ.

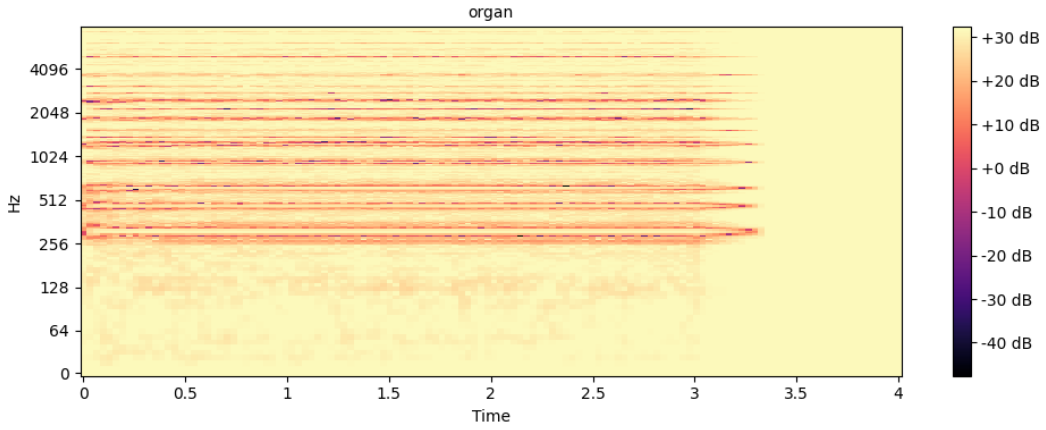


Figure 42. Test Sample : Organ Solo.

3) Duo (Flute and Organ) Spectrogram:

The third spectrogram (Figure 43) demonstrates the characteristics of a duo with flute and organ. The combination of two instruments creates a more complex pattern in the spectrogram. We might see layers of horizontal lines (organ) intertwined with more variable patterns (flute). The task of the classifier here is to distinguish both instruments' features, marking the spectrogram as a positive case for flute and organ while negative for other instruments.

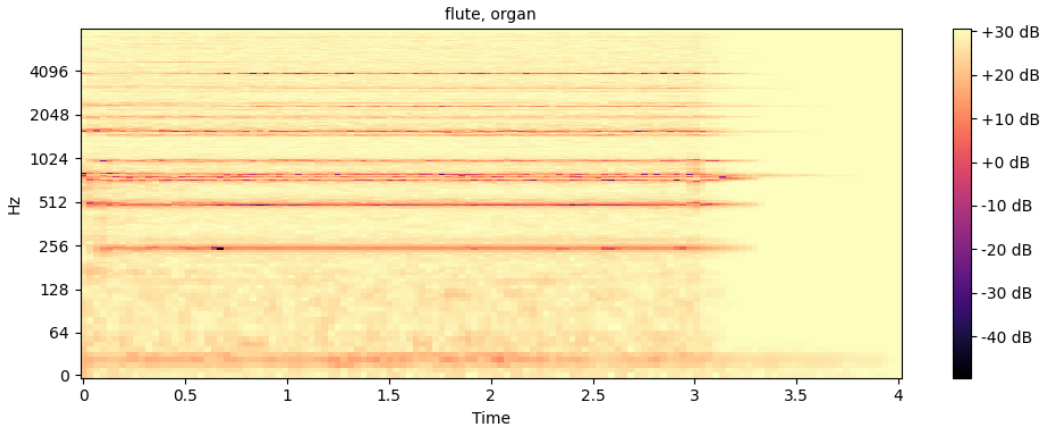


Figure 43. Test Sample : Organ and flute overlay.

4) Full Ensemble Spectrogram:

The fifth image (Figure 44) shows a spectrogram with a full ensemble of ten instruments playing simultaneously. This complexity often results in a dense and visually chaotic spectrogram. However, in some cases, it can appear surprisingly similar to the noise spectrogram due to the overlapping of so many sound sources. Despite this complexity, there may still be distinguishable

features that a sophisticated model could use to identify the presence of multiple instruments. It is an ultimate test of the model's ability to discern individual instrument characteristics within a highly polyphonic context.

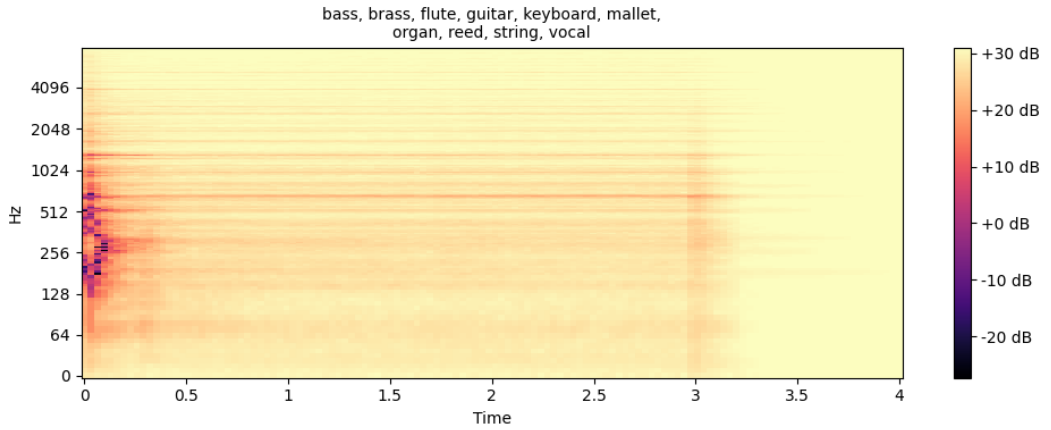


Figure 44. Test Sample :10 instruments.

5.4.3.4 Threshold Determination

Establishing a proper threshold for the confidence level of the model's predictions is key to ensuring a balance between sensitivity and specificity. In Figure 45, we exhibit the sensitivity threshold and the corresponding overall accuracy. Due to the substantial size of the test samples, our threshold determination process was conducted using a limited subset—5 samples per class, with 15 iterations in total. We then applied the most effective threshold, as determined from these preliminary tests, to the actual test samples for evaluation.

By setting a threshold of 0.35, we aim to moderate the model's strictness. A high threshold may lead to high precision for solo classifications but at the expense of missing out on more complex combinations. By relaxing the threshold slightly, we trade a small amount of solo performance for significant gains in detecting duos, trios, and larger ensembles. We anticipate that the model will focus on extracting the most salient features that are indicative of each instrument class. By not being overly stringent, the model has room to recognize a broader range of instrument combinations, potentially improving its overall performance across all classes (Figure 45).

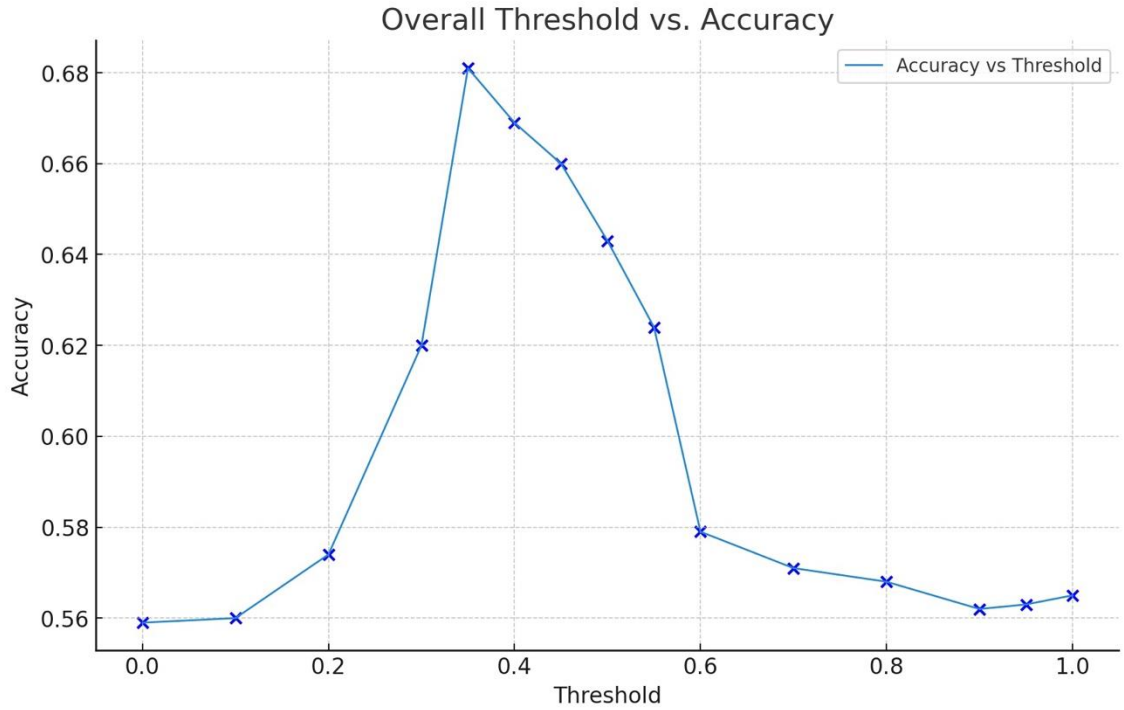


Figure 45. Predication sensitivity threshold and accuracy.

5.4.3.5 Evaluation and Analysis

The generated test samples undergo evaluation using the pre-trained models, where each model predicts the presence or absence of its respective instrument family in the samples. The evaluation metric, the Exact Match Ratio (EMR) and the multiple-label accuracy, quantifies the models' accuracy in matching the predicted labels with the true labels across all test samples.

The figure 46 containing visual and statistical information on the performance of a multi-label classification model. It features three spectrogram images, each with an associated set of true labels and model predictions for different musical instruments, as well as formulas and values for accuracy and Exact Match Ratio (EMR).

The spectrograms on the right of the figure 46, represent the frequency content of audio samples over time. The x-axis indicates time, and the y-axis represents frequency. The intensity of the colours indicates the amplitude or energy of the audio signal at each frequency and point

in time. In the context of this experiment, each spectrogram corresponds to an audio sample containing musical instruments which the model attempts to identify.

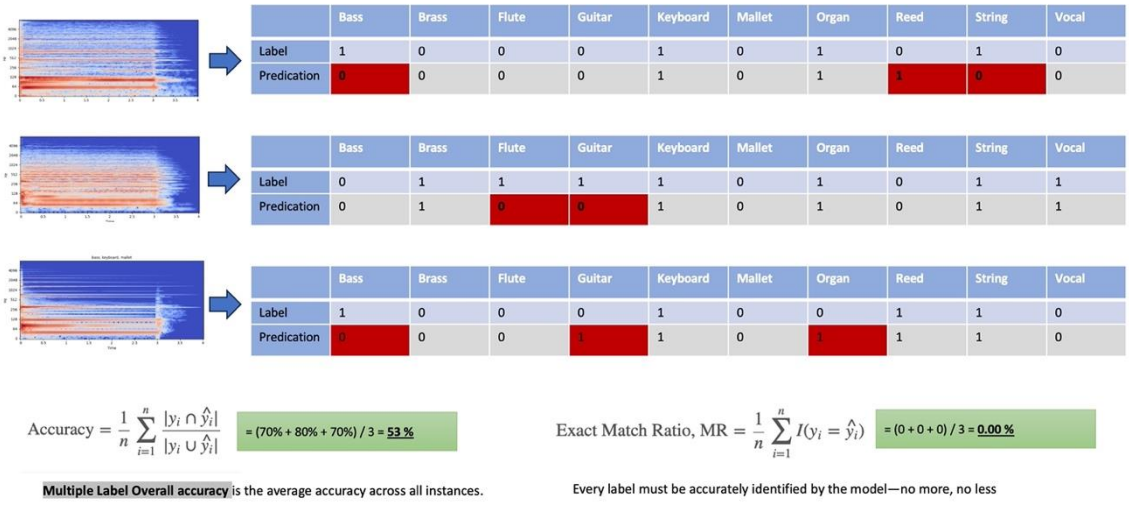


Figure 46. Metrics for Multilabel Classification.

1) First Spectrogram (Top) Analysis: True Label row indicates the presence of a Keyboard (1 at the Keyboard position) and Vocal (1 at the Vocal position). Model Prediction: Incorrectly identifies the presence of Organ (1 at the Organ position) and incorrectly misses the Vocal (0 at the Vocal position).
2) Second Spectrogram (Middle) Analysis: True Label row are Brass, Flute, Guitar, Keyboard, Mallet, Organ, String, and Vocal are present (1s in their respective positions).Model Prediction identifies Brass and Organ, misses Flute, and incorrectly identifies Guitar and Vocal.
3) Third Spectrogram (Bottom) Analysis: True Label: Bass and Reed are present (1s at the Bass and Reed positions). Model Prediction: Correctly identifies Reed but misses Bass and incorrectly identifies Guitar and Organ.

So, the accuracy defined here as the average of the individual accuracies for each instrument across all instances. It is calculated as the sum of the individual true positive rates (correctly identified labels) divided by the number of instruments, which yields 53.2% in this example. This moderate percentage indicates that while the model has some predictive ability, it's not highly accurate in identifying the presence of every instrument in the samples.

In addition, Exact Match Ratio (EMR) is a strict metric that measures the percentage of samples for which the model's predictions exactly match the true labels. For a prediction to count towards the EMR, every instrument must be correctly identified by the model—no more, no less. In the example provided, the EMR is 0%, indicating that none of the three samples had all their labels correctly predicted.

In this example, the accuracy calculation suggests that the model can identify some instruments with a moderate level of reliability across the dataset. However, the EMR of 0% is particularly revealing, as it underscores the model's difficulty in correctly predicting all labels simultaneously. This might suggest that while the model can recognize instruments in simpler contexts, it is not as effective when multiple instruments are present, as would be the case in complex musical passages.

Overall, the image and the associated data can be used to argue for the necessity of improving the model's ability to handle polyphonic audio samples, which may involve architectural changes, additional training data, or enhanced feature extraction methods that can better capture the nuances of multiple instruments playing together.

5.4.4 Result of Training Dataset

5.4.4.1 Overall accuracy and EMR based on 100 samples each class

2 charts on Figure 47 display the results of an experiment evaluating the performance of binary classifiers in a multi-label music classification context. The first chart shows the Exact Match Ratio (EMR) for different classes of instrument combinations, and the second chart shows the overall accuracy for these classes. The overall EMR is 0.17 and the overall accuracy is 0.64.

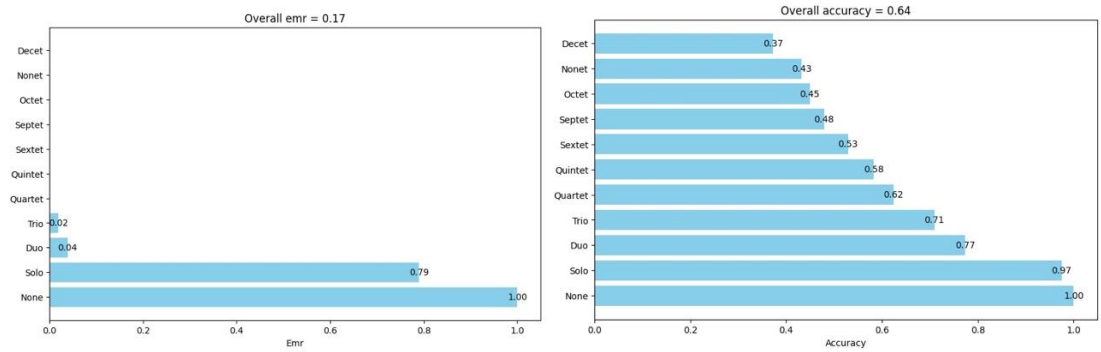


Figure 47. EMR and Accuracy per Class.

1) **No Instrument Class:** The Accuracy and EMR are both 100% for the 'None' class indicates perfect performance in cases where no instruments are present. This suggests that the classifiers are highly effective at identifying silence within the dataset.
2) **Solo Class:** With a 97% accuracy and 79% EMR for the 'Solo' class, the binary classifiers show a high level of effectiveness in identifying individual instruments. The discrepancy between accuracy and EMR may indicate occasional false positives, where the model predicts additional instruments that are not present.
3) **Duo Class:** A significant drop in EMR to 4% for the 'Duo' class, despite a relatively high accuracy of 77%, suggests that while the model can identify the presence of instruments, it struggles with the precision of matching the exact pairings. This could be due to the model's inability to disentangle the overlapping harmonic features of two instruments.
4) **Trio to Septet Performance Classes:** The gradual decline in both accuracy and EMR from 'Trio' to 'Septet' classes highlights the increasing challenge the model faces as more instruments are added. The model's performance dips closer to random chance, particularly evident in the 'Sextet' class where accuracy is slightly above a coin flip at 53%.
5) **Septet and Larger Ensembles:** Accuracy levels below 50% for groups larger than a sextet ('Septet' to 'Decet') indicate that the model performs worse than random guessing in these scenarios. This raises questions about the model's utility for larger ensembles and suggests the need for significant improvements.

5.4.4.2 Accuracy and TP rate Per Class.

In this section, the accuracy (Figure 48) and True Positive (Figure 49) curve are discussed.

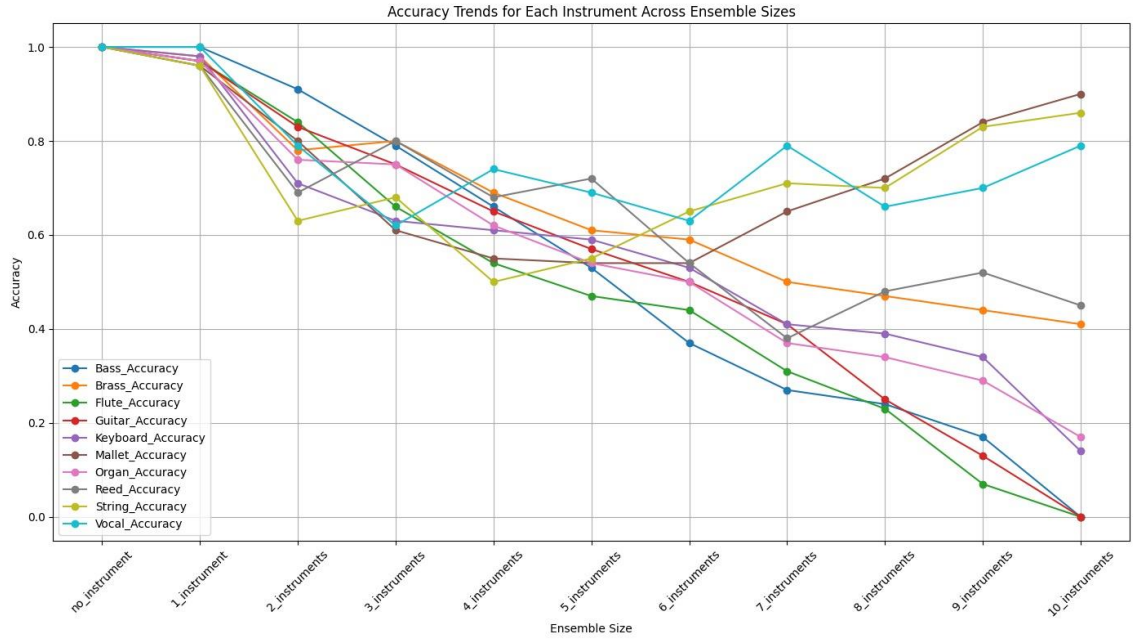


Figure 48. Accuracy Trend per Classifier.

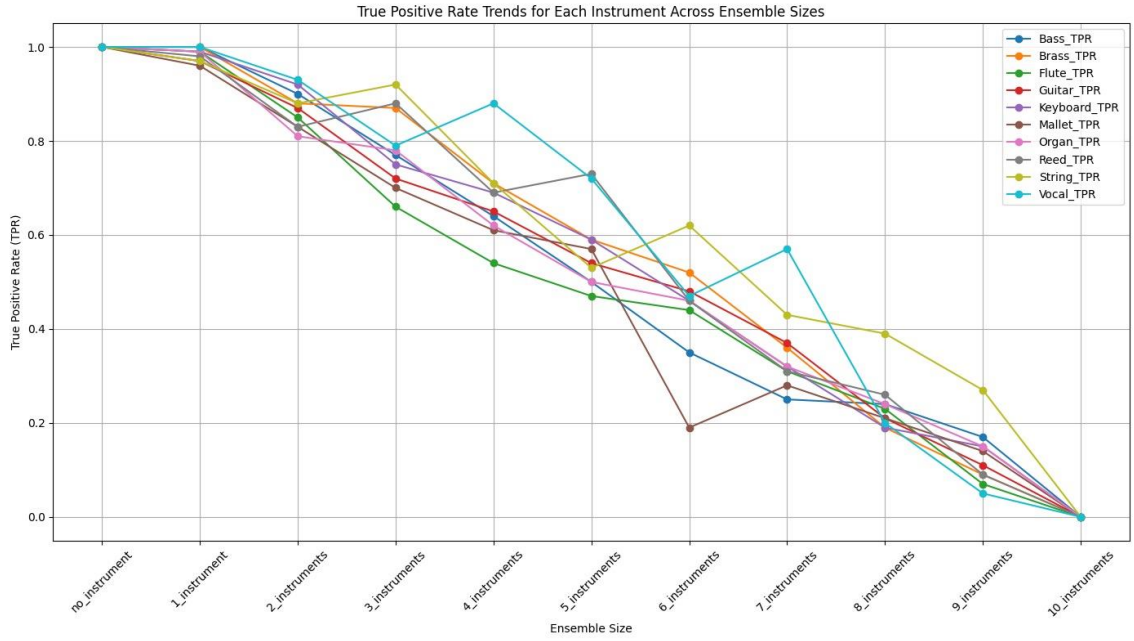


Figure 49. Accuracy Trend per Classifier.

1) No instrument

As no instrument all got a 100% accuracy and EMR, so confusion matrix analysis is skipped in this section.

2) Solo Instrument Analysis

Figure 50 shows the confusion matrix of solo scenario.

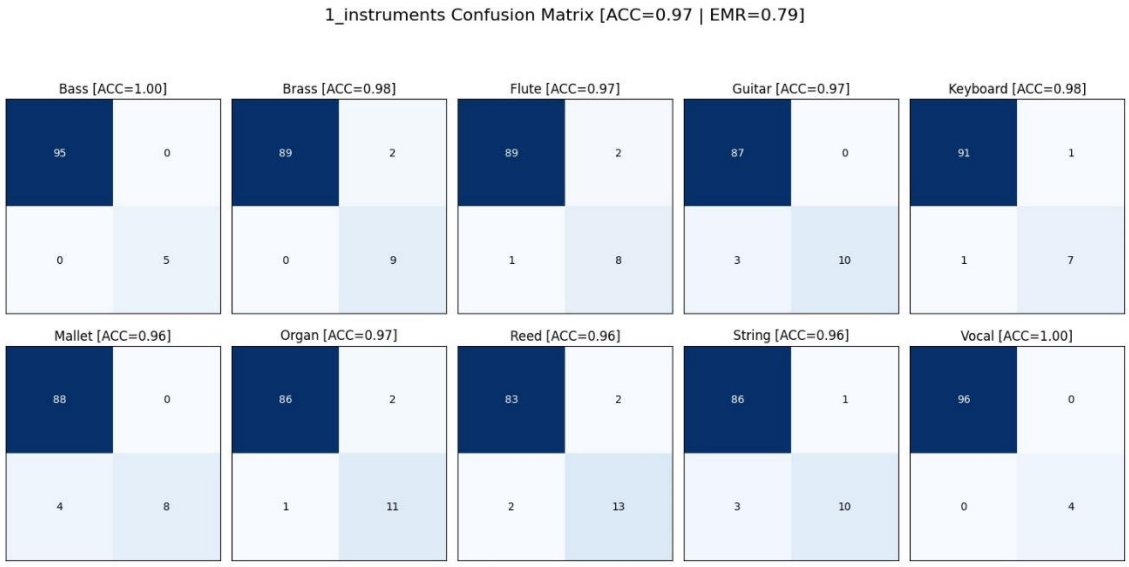


Figure 50. Confusion Matrix of each Classifier of Solo Music.

Bass: For the solo Bass, the True Positives (TP) are 95, and there are no False Positives (FP) or False Negatives (FN), leading to perfect accuracy (ACC=1.00). This means every time the Bass was present, it was correctly identified, and there were no instances where the Bass was incorrectly identified when it wasn't present.

Brass: The Brass shows high accuracy (ACC=0.98) with TP of 89 and a small number of FPs (2) and FNs (0). This indicates that the model is very effective at identifying Brass with minimal diagnostic error, as evidenced by the high TP and low FP and FN counts.

Flute: The Flute also exhibits high accuracy (ACC=0.97) with TP of 89, but unlike the Bass, there are small amounts of both FPs (2) and FNs (1), suggesting a slight confusion, possibly with similar-sounding instruments.

Guitar: With a TP of 87 and no FPs, the Guitar's accuracy (ACC=0.97) is high, and the model seems to be adept at identifying the Guitar without mistaking other instruments for it.

Keyboard: The Keyboard shows a high accuracy (ACC=0.98) with TP of 91 and minimal misclassifications, indicated by the low FP (1) and FN (1) values.

Mallet: The Mallet instrument has an accuracy of 0.96 with TP of 88, indicating strong recognition capabilities, though there is a slight chance of misclassification (FP=0, FN=4).

Organ: The Organ's accuracy (ACC=0.97) is high with TP of 86. It has a few misclassifications, suggesting that while the model recognizes the Organ well, there is room for improvement in reducing the FN (1) and FP (2).

Reed: The Reed instrument's accuracy (ACC=0.96) is slightly lower than some other instruments, but still high, with TP of 83. The FN (2) and FP (2) indicate a small level of confusion.

String: The String instrument shows high accuracy (ACC=0.96) with a TP of 86. The low FP (1) and FN (3) suggest that the model is reliably recognizing the String sounds.

Vocal: Vocal recognition is perfect with an accuracy of 1.00, and no misclassifications (FP=0, FN=0), indicating that the model can distinguish vocals clearly.

3) Duo Instrument Analysis

As we move from solo to duo (Figure 51) settings, we observe an expected decrease in TP and an increase in FN across all instruments, indicative of the model's increasing difficulty in distinguishing individual instruments as the number of instruments in the mix increases.

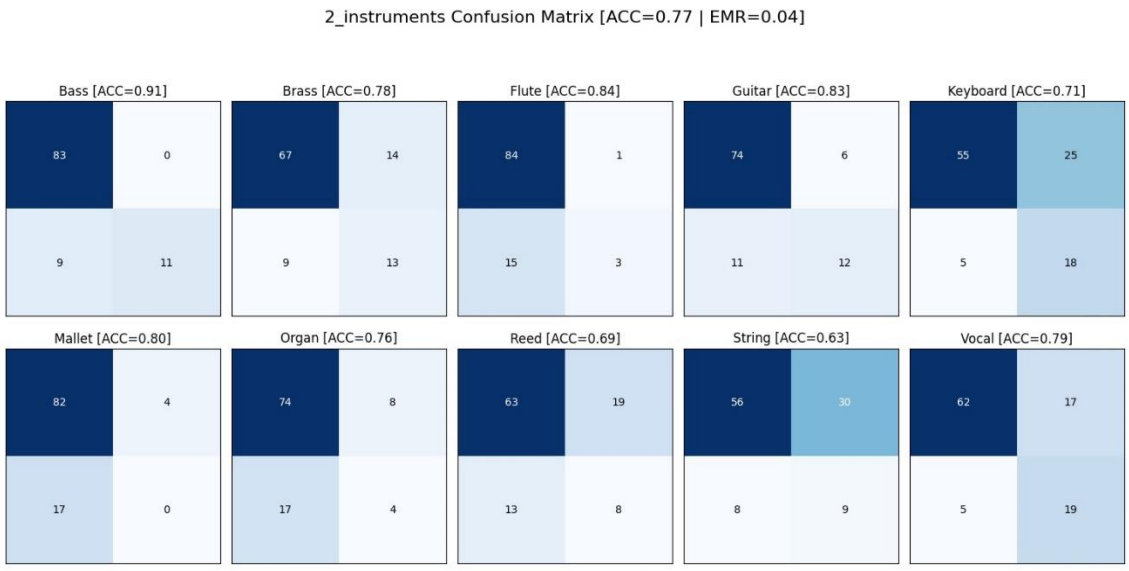


Figure 51. Confusion Matrix of each Classifier of Duo Music.

Bass (Duo): The Bass in a duo setting has an accuracy of 0.91. The TP is 83, indicating strong recognition, but with 9 FN, it suggests that the Bass is sometimes missed when paired with

another instrument. The 11 TN indicates that when the Bass is not present, the model often correctly identifies its absence.

Brass (Duo): For Brass, the accuracy drops to 0.78. The TP of 67 suggests a decent recognition rate, but with 14 FP and 13 FN, there is a considerable amount of both over-prediction and under-prediction, indicating confusion when Brass is paired with another instrument.

Flute (Duo): The Flute has an accuracy of 0.84 with a TP of 84, showing good recognition. However, 15 FN indicates that the Flute is often missed in the presence of another instrument, while the low FP of 1 suggests that the model does not often falsely detect it.

Guitar (Duo): Guitar's accuracy stands at 0.83 with TP of 74. With 11 FN, the model misses the Guitar at times in a duo. The FP of 6 points to some instances of false detection.

Keyboard (Duo): Keyboard accuracy is significantly lower at 0.71, with a TP of 55. The 25 FP is indicative of a high rate of false positives, and 18 FN suggests the Keyboard is often missed when paired with another instrument.

Mallet (Duo): The Mallet instrument maintains an accuracy of 0.80 with a TP of 82. The absence of FN indicates good sensitivity, but 4 FP shows some instances of false alarms.

Organ (Duo): The Organ has an accuracy of 0.76 with a TP of 74. The model struggles somewhat with 17 FN, often missing the Organ in a duo, and 8 FP indicates some false detections.

Reed (Duo): Reed shows a lower accuracy of 0.69, with TP at 63. The 19 FP and 13 FN suggest a higher confusion level, both in falsely detecting and missing the Reed when another instrument is present.

String (Duo): String instruments have an accuracy of 0.63 with TP of 56. The high number of FN (8) and FP (30) indicates a significant challenge for the model in accurately detecting String in the presence of another instrument.

Vocal (Duo): Vocals show an accuracy of 0.79 with TP of 62. While there are fewer FPs (17), suggesting it is not often falsely detected, the 5 FN points to occasional misses in duo settings.

4) Trio Instrument Analysis

The trend from solo to trio (Figure 52) highlights a challenge common to many classification systems: as the complexity of the audio sample increases, the model's ability to distinguish individual instruments becomes compromised, leading to more FN and FP.

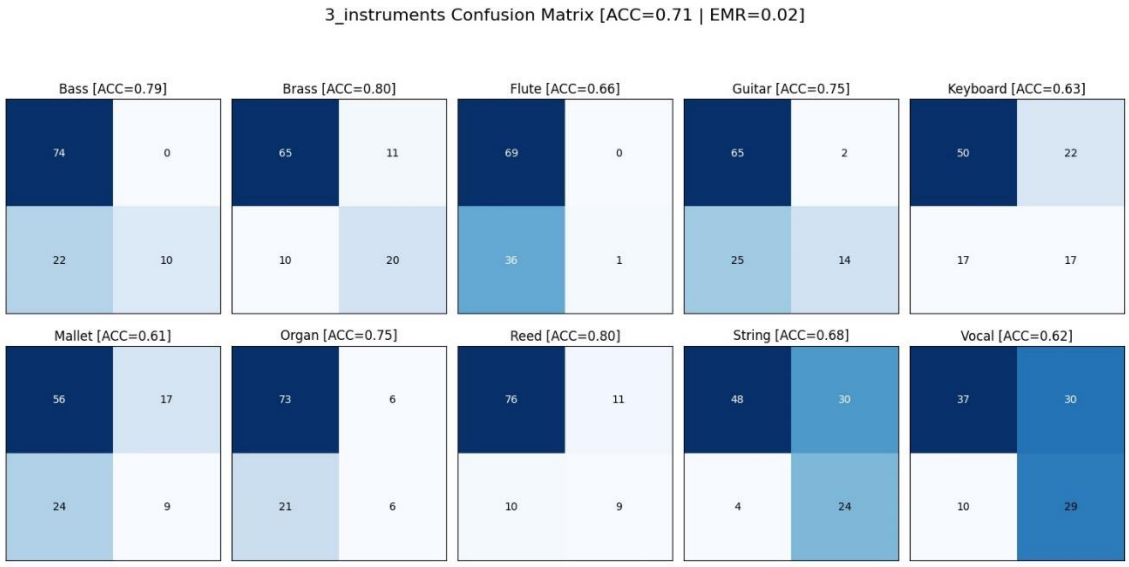


Figure 52. Confusion Matrix of each Classifier of Duo Music.

Bass (Trio): In a trio setting, the Bass accuracy further decreases to 0.79. There are 22 FN, indicating a more significant challenge in detecting the Bass among two other instruments.

Brass (Trio): Brass has an accuracy of 0.80. Despite the complexity, it still has a TP of 65 but with an increased FN of 10, showing difficulties in accurate detection.

Flute (Trio): Flute accuracy drops to 0.66, which is quite a decrease, with a TP of 69 but a high FN of 36, indicating the model often misses the Flute in a trio.

Guitar (Trio): Guitar maintains an accuracy of 0.75, with TP of 65. The FN count of 25 signifies the model's struggle to consistently recognize the Guitar.

Keyboard (Trio): Keyboard accuracy falls to 0.63. The 17 FN suggest that the Keyboard is frequently missed, and 22 FP indicate confusion in the model's predictions.

Mallet (Trio): In the trio ensemble, the Mallet instrument has an accuracy of 0.61, with a TP of 56. The increased FN of 24 indicates the Mallet is often overlooked when mixed with two other instruments.

Organ (Trio): The Organ in a trio context has an accuracy of 0.75, with a TP of 73. The FN of 21 suggests that while the Organ is generally recognized, there are occasions where it is missed in the presence of other instruments.

Reed (Trio): Reed's accuracy is high for a trio at 0.80, with a TP of 76. The FN of 10 and FP of 11, however, show that there is still some confusion, albeit less than some other instruments.

String (Trio): String instruments have an accuracy of 0.68 in a trio setting, with a TP of 48. The high FP of 30 and FN of 4 point to difficulties in accurately detecting the String amongst other instruments.

Vocal (Trio): Vocal recognition drops to an accuracy of 0.62. The FN of 10 and FP of 30 show that Vocals are frequently misclassified, either by being missed or falsely detected in a trio setting.

5) Quartet Instrument Analysis

When the synthesised music pieces comes to 4 instruments (Figure 53), the EMR goes to zero, and the accuracy decreases to 6.62.

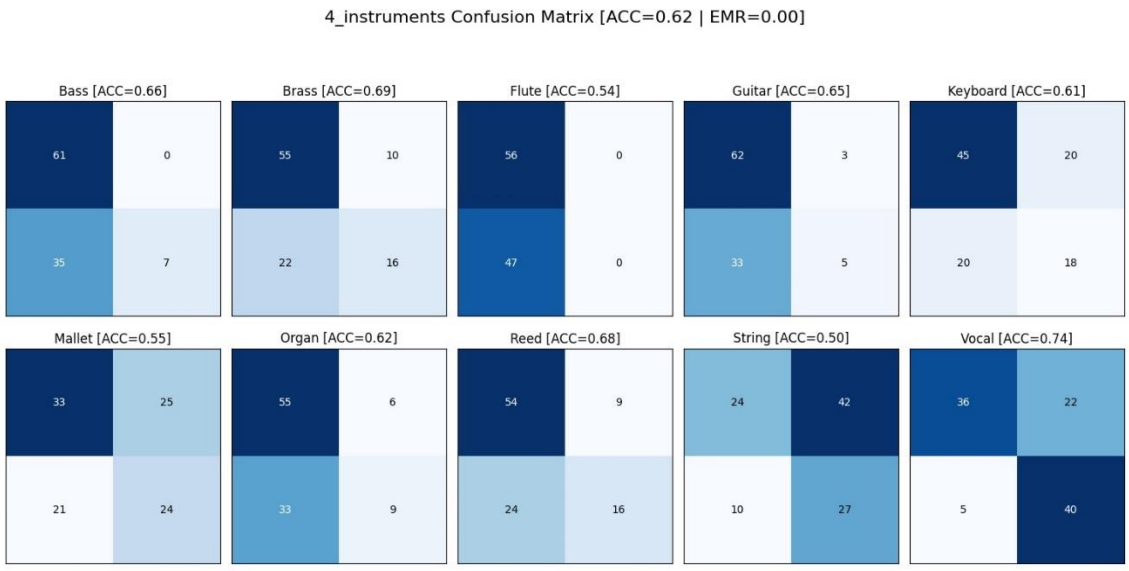


Figure 53. Confusion Matrix of each Classifier of Duo Music.

Bass (Quartet): For the quartet ensemble, the Bass accuracy is at 0.66. The model has a considerably high FN of 35, indicating that the Bass is often missed when three other instruments are present.

Brass (Quartet): Brass accuracy slightly improves to 0.69 in a quartet setting. The TP of 55 and FN of 22 suggest some difficulty in detection, but less so compared to the trio.

Flute (Quartet): Flute accuracy drops significantly to 0.54. The TP is 56, but with a high FN of 47, the model has difficulty identifying the Flute when it is part of a quartet.

Guitar (Quartet): In a quartet, Guitar accuracy is 0.65 with a TP of 62. Despite the complexity, the Guitar is still recognized to a degree, but with 33 FN, it's clear that the model often fails to detect it amidst the mix.

Keyboard (Quartet): Keyboard accuracy decreases further to 0.61. With TP at 45 and FP at 20, there's a notable increase in both false positives and negatives, indicating confusion is exacerbated as more instruments are added.

Mallet (Quartet): The Mallet instrument shows an accuracy of 0.55 in a quartet setting, with a TP of 33. The high FN of 21 suggests the Mallet is frequently missed, and the FP of 25 indicates a significant rate of misidentification.

Organ (Quartet): Organ accuracy is 0.62, showing moderate performance with TP of 55. The FN of 33 indicates that the model's ability to identify the Organ is strained in more complex audio samples.

Reed (Quartet): Reed accuracy stands at 0.68, with TP at 54. The model displays some resilience in detecting Reed sounds, but with 24 FN, it's not without its challenges.

String (Quartet): String instruments see a drop in accuracy to 0.50. With TP at 24 and a high FP of 42, it's evident that the model struggles significantly to accurately identify String in a quartet.

Vocal (Quartet): Vocal maintains a relatively high accuracy at 0.74 in a quartet. TP of 36 and FN of 5 show that Vocal sounds are still quite distinct even in a complex mix.

6) Quintet to Decet Instrument Analysis

As we progress to larger ensembles, from quintets to decets (Figure 54), a general trend is evident: the accuracy for each instrument generally decreases, and the number of false negatives increases.

This trend suggests that the model's capacity to discriminate between different instruments diminishes as the complexity of the ensemble grows.

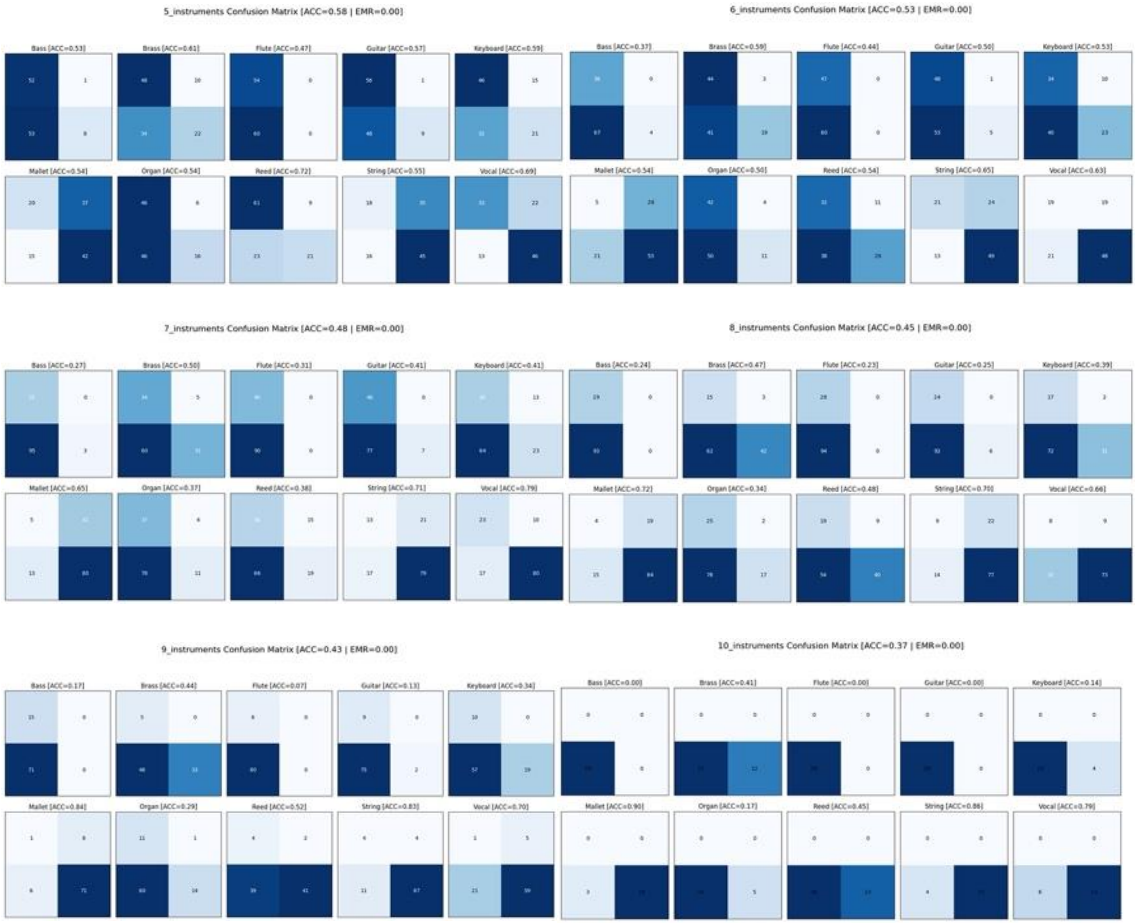


Figure 54. Confusion Matrix of each Classifier from Quintet to Decet.

Bass: In larger ensembles, the Bass's accuracy decreases dramatically, reaching 0.00 in a decet, indicating a complete inability to identify the Bass in such a complex mix.

Brass: Brass exhibits a similar downward trend, with accuracy reducing to 0.41 in a decet. The increasing number of FN and FP suggests the Brass gets easily confused with other instruments in large ensembles.

Flute: The Flute's accuracy declines to 0.00 in a decet, showing the model's complete failure to detect it amidst the complexity of ten instruments.

Guitar (Decet): In a decet, Guitar's accuracy hits 0.00, which means the model failed to correctly identify the Guitar at all in the presence of nine other instruments, indicated by the TP of 0 and the high FN.

Keyboard (Decet): Keyboard's accuracy plummets to 0.14 in a decet. The TP of 0 suggests that the model cannot detect the Keyboard when many instruments are playing together.

Mallet (Decet): Surprisingly, the Mallet maintains a relatively high accuracy at 0.90 even in a decet, suggesting that the Mallet has a distinct sound that the model can identify even in complex settings, as indicated by the TP of 26 and the absence of FP.

Organ (Decet): The Organ's accuracy drops to 0.17 in a decet. The TP of 0 and FN of 24 demonstrate the model's struggle to detect the Organ in a highly polyphonic context.

Reed (Decet): Reed's accuracy is at 0.45 in a decet, showing that while the Reed is often missed (FN of 16), it is still detected to some extent (TP of 13) amidst many instruments.

String (Decet): String instruments maintain an accuracy of 0.86, surprisingly high for a decet, with TP of 25 and low FN, suggesting that the model can still pick out String sounds even in a full ensemble.

Vocal (Decet): Vocals have an accuracy of 0.79 in a decet. The TP of 23 and FN of 6 imply that Vocals remain one of the more recognizable sounds for the model, even with nine other instruments.

7) Trend Analysis Across Ensemble Sizes

As the ensemble size increases, there is a clear trend of decreasing accuracy and increasing diagnostic errors for all instruments. This suggests the model struggles with polyphonic sound where multiple instruments are played simultaneously.

1) Bass: The accuracy drops significantly as we move from solo to decet (from 1.00 to 0.00), indicating the model's increasing difficulty in correctly identifying the Bass amongst a larger set of instruments.
2) Brass: Brass shows a decreasing trend in accuracy (from 0.98 to 0.41), suggesting that the Brass sound gets overshadowed or confused with other instruments as the complexity increases.
3) Flute: The Flute's accuracy dramatically decreases (from 0.97 to 0.00), possibly due to its sound blending with others in polyphony, leading to increased FNs.

4) Guitar: The Guitar maintains relatively higher accuracy in smaller ensembles but drops to 0.00 in the decet, indicating the model's inability to isolate Guitar sounds in high polyphony.
5) Keyboard: Keyboard recognition drops to a lower accuracy (from 0.98 to 0.14), suggesting significant challenges in distinguishing Keyboard sounds in complex audio samples.
6) Mallet: Mallet shows a varied trend but maintains higher accuracy even in larger ensembles, potentially due to its distinctive timbre.
7) Organ: Organ recognition declines less steeply (from 0.97 to 0.17), possibly due to its distinct harmonic content that the model can sometimes still detect.
8) Reed: Reed's trend is similar to Brass, with a notable decrease in accuracy as ensemble size increases.
9) String: String accuracy decreases less dramatically (from 0.96 to 0.86), which might be due to the distinctive resonance of string instruments that the model can pick up.
10) Vocal: Vocal maintains relatively high accuracy even in larger ensembles, suggesting that vocal timbre is distinct enough for the model to detect amidst other sounds.

The overall trend from duo to decet for each instrument shows a decline in the model's performance, with the exception of Mallet and String, which maintain relatively high accuracy even in complex ensembles. This suggests that certain instruments have distinctive acoustic properties that the model can identify even in a crowded sonic space. In contrast, instruments like the Flute and Guitar become increasingly difficult to detect as the number of instruments increases.

The increasing FNs across almost all instruments indicate a general trend: as more instruments are added, the model struggles to maintain its sensitivity, likely due to the overlapping harmonics and timbral characteristics that confuse the classifier.

The relatively stable or even high accuracy for Mallet and String in larger ensembles suggests that these instruments may have unique features (Figure 55) that are less affected by the presence of other instruments. This could be leveraged in improving the model by focusing on these distinguishing features.

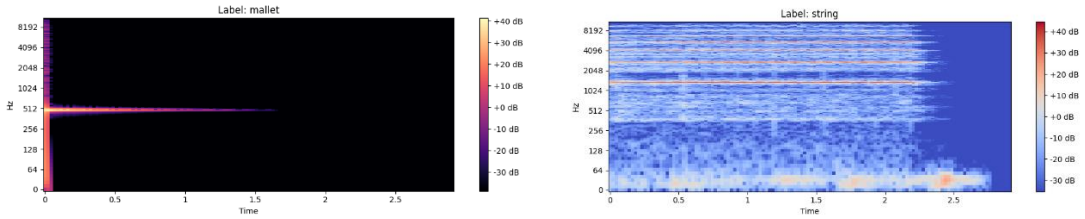


Figure 55. Spectrogram of Mallet and String.

The consistency in accuracy for Mallet and String instruments within larger ensembles likely stems from their distinctive acoustic signatures that remain discernible amidst complex soundscapes. For instance, the Mallet often displays a prominent spectral peak around 512 Hz, which could be represented by a distinct colour in a spectrogram, typically indicative of its fundamental frequency. On the other hand, String instruments exhibit a rich harmonic spectrum that spans the entire frequency range, resembling the dense texture of multiple instruments played simultaneously. These characteristic features can potentially be exploited to enhance the model's detection capabilities by prioritizing the extraction and analysis of such unique spectral components.

In summary, the model demonstrates strong performance in solo settings but faces significant challenges as the number of instruments increases. The exception to this is with Mallet and String, which suggests potential pathways for improving the model's robustness in polyphonic settings. Future improvements could involve feature engineering to better capture the unique characteristics of these instruments or employing more sophisticated models that can disentangle complex sounds.

5.4.5 Discussion Based on the Result

1) Consistency with EMR Observations:

The accuracy chart corroborates the EMR findings, showing high accuracy for silence and solo instrument classifications, with a gradual decline as more instruments are introduced.

2) Accuracy Trends:

The trend in accuracy from 'Solo' to 'Decet' classes is not linear, with some classes like 'Quintet' and 'Septet' showing unexpected variations in performance. This might be due to specific pairings of instruments that the model can recognize more easily, even within complex combinations.

3) Threshold for Reliable Classification:

The 'Sextet' class appears to be the threshold beyond which the model's accuracy becomes unreliable. This insight can guide future experiments and the development of more sophisticated models or feature engineering approaches that can handle high levels of synthesized polyphony.

4) Considerations for Model Improvement:

Based on the poor performance for larger ensembles, strategies for improvement could include training on more diverse datasets, implementing more complex architectures capable of capturing intricate patterns, or refining the feature extraction process to better distinguish between instruments.

5) Next Steps in Analysis:

Further examination of the confusion matrix, as mentioned, will be crucial in understanding the types of errors the model is making. This deeper analysis can help in pinpointing whether the errors are systematic, random, or due to certain instruments being more challenging to classify.

In summary, the charts reveal a model that is competent in identifying silence and solo instruments but faces significant challenges as the number of instruments in an audio sample increases. The high accuracy for solo performances suggests that the binary classifiers are well-tuned for individual instrument recognition. However, the drastic decrease in EMR for duets and larger groups points to a model that struggles with polyphonic audio. The performance boundary seems to be at the sextet level, beyond which the model's utility is questionable. Future work should focus on improving the model's ability to handle complex audio samples with multiple overlapping instruments.

5.5 Feature map and Heatmap Analysis Experiment

To address Research Objective 6 (RO-6), this experiment focuses on unravelling the inner workings of our Convolutional Neural Network (CNN) model for musical instrument recognition. While previous experiments have demonstrated promising results, it is crucial to delve deeper into the model's decision-making process. This analysis aims to open the "black box" of CNN by visually and statistically examining the feature maps and heatmaps generated during the classification process. By doing so, we seek to extract, visualize, and quantify the features learned by the convolutional layers for each instrument. This approach not only enhances our understanding of how the model distinguishes between different instruments but also provides necessary insights into the specific spectral and temporal patterns that are most influential in the classification process. The findings from this experiment will contribute to the interpretability of our model and potentially guide future improvements in musical instrument recognition techniques.

5.5.1 Feature Map and Heatmap

In our experiment, we extract feature maps and heatmaps for various instruments to analyse and compare their performance. We conducted both visual and statistical analyses. The visual analysis involved observing the feature maps and heatmaps, while the statistical analysis involved calculating metrics such as the KL divergence, Jensen-Shannon divergence, and Earth Mover’s Distance.

5.5.1.1 Feature Map

Feature maps (LeCun et al., 1998; Zeiler & Fergus, 2014b) in Convolutional Neural Networks (CNNs) are the outputs of the convolutional layers after applying filters to the input data. These maps highlight the presence of various features detected by the filters, such as edges, textures, and shapes, at different levels of abstraction. The initial layers capture low-level features, while deeper layers capture more complex patterns.

5.5.1.2 Heatmap

Heatmaps (Chattopadhay et al., 2018; Pan et al., 2021; Qi et al., 2019; Sattarzadeh et al., 2021) in CNNs represent the regions of the input that contribute the most to the network's decision. They provide a visual explanation of the model's focus areas, helping to understand which parts of the input data are influencing the prediction. This is crucial for interpretability and debugging of the model.

5.5.2 Detailed Feature Analysis and Literature Review

The need for a deeper analysis of the features used in musical instrument recognition arises from the diverse approaches and methodologies explored in the literature. Understanding the robustness and relevance of these features can significantly impact the performance of classification models, particularly in complex environments such as polyphonic music or varying acoustic conditions. This section delves into the feature extraction and analysis techniques used in musical instrument classification, building on the existing body of research.

Research by Wegener et al. (2008) emphasized the robustness of various audio features in the context of musical instrument classification, particularly under conditions involving signal modifications such as low-pass filtering, noise addition, and reverberation. The study highlighted the importance of selecting features that remain robust across different signal conditions to ensure consistent classification accuracy. The authors used a range of MPEG-7 audio descriptors and other spectral, temporal, and perceptual features, concluding that features like Log Attack Time (LAT) and certain Mel-Frequency Cepstral Coefficients (MFCCs) showed high robustness and should be prioritized in feature selection for robust classification.

Simmermacher, Deng, and Cranefield (2006) conducted an empirical study focusing on the classification of classical musical instruments using a variety of feature extraction methods. Their findings reinforced the relevance of MFCCs, particularly the first few coefficients, which consistently ranked among the top features across different classification tasks. Additionally, they explored the use of MPEG-7 audio descriptors and perceptual features, finding that combining MFCCs with perceptual features such as zero-crossing rate and spectral centroid significantly

improved classification accuracy, especially in distinguishing between similar-sounding instruments(document).

These studies underscore the importance of carefully selecting and combining features in musical instrument classification. The combination of robust features, as identified by these works, can lead to improved classification performance, especially in challenging environments. As such, a more refined approach to feature analysis is necessary to enhance the effectiveness of classification models used in this research. This section has provided a literature-based rationale for the need to analyze and optimize the features used in our study, setting the stage for the detailed statistical analysis and experimental design discussed in the following sections.

5.5.3 Statistic Analysis Experiment Design

For each instrument in the dataset, we load the corresponding heatmaps. We then compare every possible pair of samples within the heatmaps for that instrument. For each pair, we calculate the following metrics:

1. Difference Mean: The average difference between the two samples.
2. KL Divergence (Kullback & Leibler, 1951): A measure of how one probability distribution differs from another.
3. Jensen-Shannon Divergence (Lin, 1991): A symmetric measure of similarity between two probability distributions.
4. Earth Mover's Distance (Rubner et al., 2000): The cost of transforming one distribution into another.

Algorithm 1 Heatmap Statistical Analysis

```
for each instrument i in instruments do
    Load heatmaps for instrument i
    for each sample s1 in heatmaps do
        for each sample s2 in heatmaps where s2 ≠ s1 do
            Calculate difference mean between s1 and s2
            Calculate KL divergence between s1 and s2
            Calculate Jensen-Shannon divergence between s1 and s2
            Calculate Earth Mover's Distance between s1 and s2
        end for
    end for
end for
```

Figure 56. The workflow of the heatmap analysis experiment.

According to Figure 56, by using these four metrics, we gain an understanding of the similarity and consistency of integrated heatmaps across different samples. This helps us evaluate the reliability and robustness of the model's feature importance interpretation. If the metrics indicate high similarity and low divergence, it suggests that the model consistently identifies important features across samples, which is desirable for reliable model explanations.

We repeat the four metrics calculations for all pairs of samples within the heatmaps for each instrument. The following diagram is how we calculate the difference between classifiers.

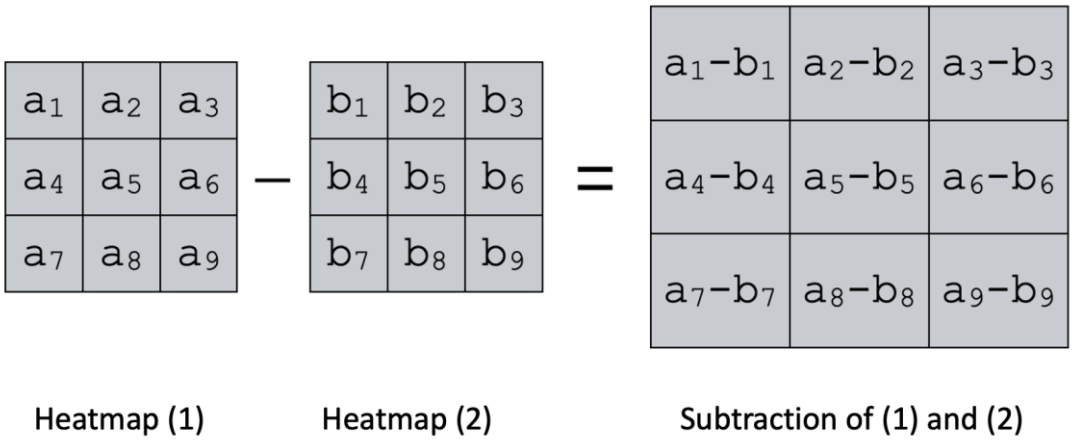


Figure 57. Diagram of heatmap difference calculation.

For example (Figure 57) heatmap (1) of first test sample of a binary classifier and heatmap (2) of the second test sample of the binary classifier heatmap. And the difference is the subtraction of the 2. We use Algorithm 1 to evaluation the difference per classifier.

5.5.4 Evaluation Metrics

5.5.4.1 Difference Mean

This metric calculates the average difference between two heatmaps. By examining the Difference Mean, we can understand the overall similarity or dissimilarity between two samples. A low Difference Mean indicates that the two samples are quite similar, while a high Difference Mean suggests significant differences. This helps us to see how consistently the integrated gradients identify important features across different samples.

5.5.4.2 KL divergence

The Kullback-Leibler (KL) Divergence (Kullback & Leibler, 1951) measures how one probability distribution diverges from a second, expected probability distribution. In the context of heatmaps, it quantifies how different the importance of features (as indicated by the gradients) is between two samples. High KL Divergence indicates that the model assigns very different importance to features in the two samples, suggesting variability in model interpretation.

5.5.4.3 Jensen-Shannon Divergence:

JS divergence (Lin, 1991) is a symmetric measure of similarity between two probability distributions. Unlike KL Divergence, it is always finite and provides a more stable measure. It helps to assess the consistency of the feature importance across different samples. A low Jensen-Shannon Divergence indicates that the model interprets the samples in a similar way, highlighting robustness in the model's feature importance assignment.

5.5.4.4 Earth Mover's Distance (EMD)

EMD (Rubner et al., 2000):, also known as the Wasserstein distance, measures the "cost" of transforming one distribution into another. For heatmaps, it reflects how much change is needed to make one heatmap resemble another. This metric is particularly useful for understanding the structural differences between heatmaps. A low EMD suggests that the overall pattern of feature importance is similar between two samples, while a high EMD indicates significant structural differences.

5.5.5 Visualization Analysis

Visualizing deep networks by optimizing with integrated gradients heatmap and convolutional filter applied feature map. for each instruments provides insights into the network's focus areas. For each instrument, we observed how the network's attention varied, revealing patterns and regions critical for the classification task. In this section we discussion two very distinct examples.

5.5.5.1 Visualization Example of Vocal Classifier

In figure 58, we observe the evolution of feature maps across three convolutional layers for a vocal classifier sample. Each row (a, b, c) represents a different convolutional layer, showing six feature maps generated by different kernels within that layer. The feature maps reveal how the convolutional network progressively extracts and refines features from the input spectrogram. In the first layer (a), the feature maps capture basic patterns and edges. By the second layer (b), the patterns become more distinct, highlighting more complex structures within the vocal sample. In the third layer (c), the feature maps show even higher-level features, indicating a more abstract representation of the input data.

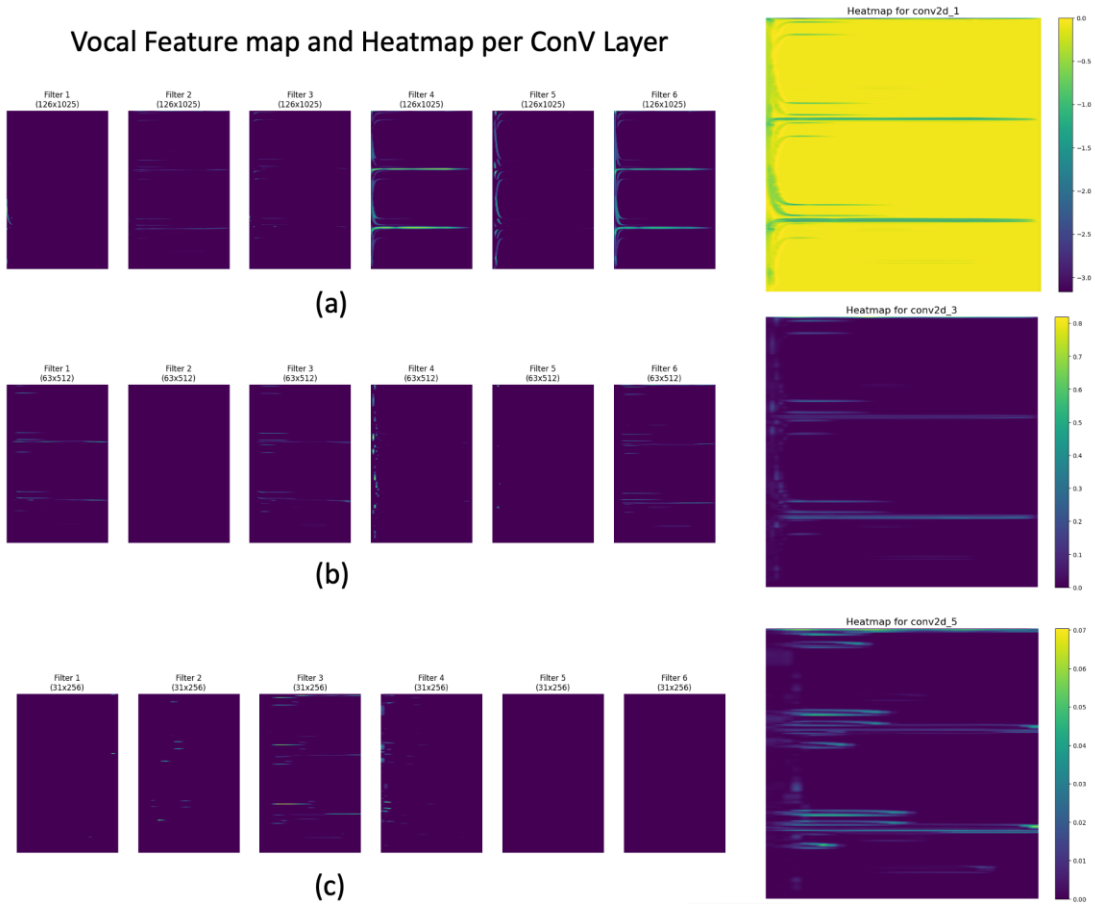


Figure 58. Feature maps and integrated gradient heatmaps of a vocal classifier sample across three convolutional layers.

Feature maps and integrated gradient heatmaps of a vocal classifier sample across three convolutional layers. (a) The feature maps of the first convolutional layer with 6 batch normalization kernels. (b) The feature maps of the second convolutional layer with 6 batch normalization kernels. (c) The feature maps of the third convolutional layer with 6 batch normalization kernels. The corresponding integrated gradient heatmaps for each layer are shown on the right.

The integrated gradient heatmaps on the right side of the figure 58 provide insight into which parts of the input spectrogram are most influential in the model's decision-making process. For the first convolutional layer (a), the heatmap indicates that the model focuses on a wide range of the input, suggesting that it is learning fundamental features. In the second layer (b), the focus narrows, with certain regions of the input spectrogram becoming more prominent. This trend continues in the third layer (c), where the heatmap highlights specific areas that are crucial for the classification of the vocal sample. These observations suggest that the model progressively refines

its focus from broad features to more specific, high-level features, which are critical for accurate classification. This layered approach allows the model to build the understanding of the input data, improving its ability to distinguish between different classes.

5.5.5.2 Visualization Example of Bass Classifier

In the following figure 59, we observe the feature maps across three convolutional layers for a bass classifier sample. Similar to the vocal sample, each row (a, b, c) represents a different convolutional layer, displaying six feature maps generated by different kernels within that layer. In the first convolutional layer (a), the feature maps for the bass classifier appear to capture minimal features, primarily highlighting the edges and basic patterns of the input spectrogram. Moving to the second layer (b), the feature maps show a slight increase in complexity, though they still capture relatively simple patterns compared to the vocal classifier. By the third layer (c), the feature maps illustrate higher-level features, but these are less distinct than those observed in the vocal classifier.

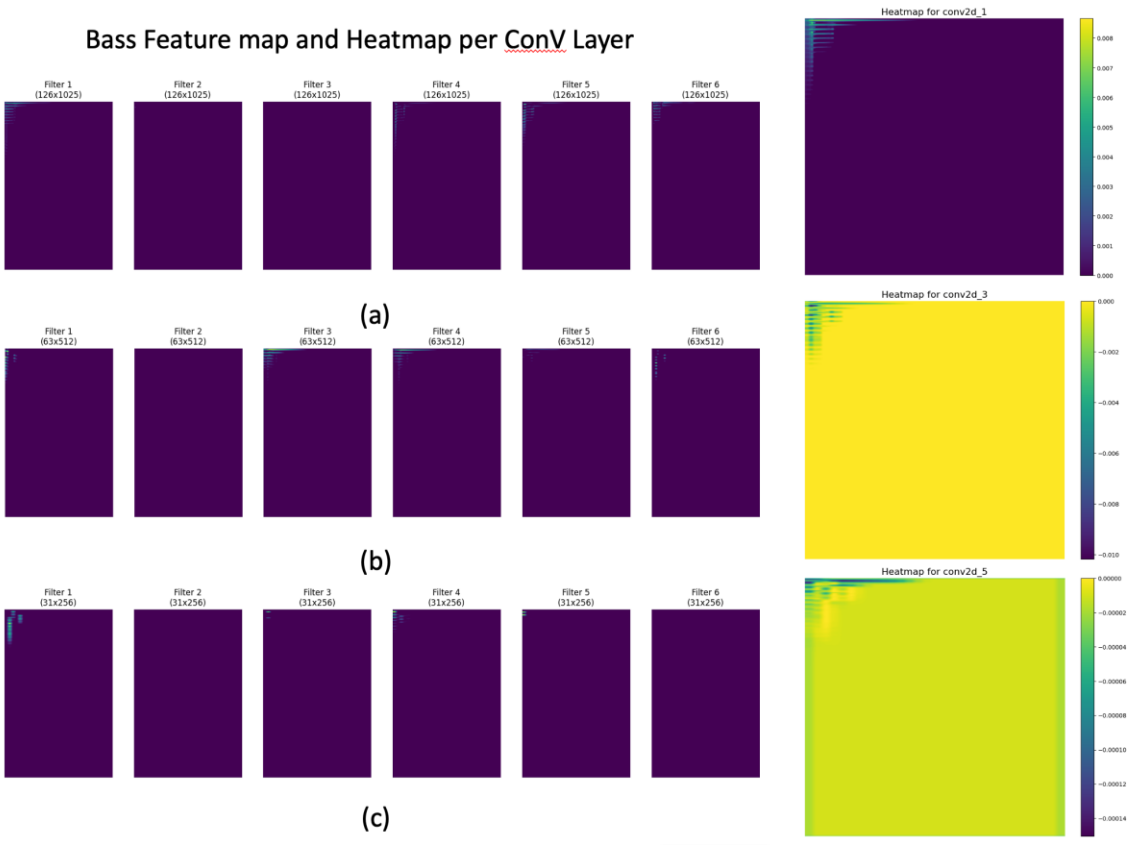


Figure 59. Feature maps and integrated gradient heatmaps of a bass classifier sample across three convolutional layers.

Figure 59 (a) is the feature maps of the first convolutional layer with 6 batch normalization kernels. (b) is the feature maps of the second convolutional layer with 6 batch normalization kernels. (c) is the feature maps of the third convolutional layer with 6 batch normalization kernels. The corresponding integrated gradient heatmaps for each layer are shown on the right.

The integrated gradient heatmaps on the right reveal which parts of the input spectrogram are most influential for the bass classifier. In the first convolutional layer (a), the heatmap shows that the model focuses on a narrow range of the input, suggesting that it captures fundamental but limited features. In the second layer (b), the focus remains relatively narrow, with only slight emphasis on specific regions of the input spectrogram. By the third layer (c), the heatmap highlights particular areas, indicating the critical parts for the classification of the bass sample. However, the overall focus is less pronounced than that of the vocal classifier, implying that the bass classifier might struggle to identify complex patterns as effectively.

Comparing the bass and vocal classifiers, it is evident that the vocal classifier's feature maps and heatmaps show a more pronounced and distinct pattern recognition across all three convolutional layers. The vocal classifier progressively refines its focus from broad features to more specific, high-level features, enhancing its classification accuracy. In contrast, the bass classifier exhibits less complex feature maps and a narrower focus in its heatmaps. This indicates that the vocal classifier is more effective in capturing and utilizing detailed features of the input spectrogram, while the bass classifier may need further refinement to improve its pattern recognition capabilities. This comparative analysis underscores the need for statistical analysis to better understand the differences and guide improvements in classifier performance.

5.5.6 Heatmap Statistic Analysis

For statistical analysis, we looped through 50 test samples for each instrument, comparing heatmaps within the same class. These metrics provide a quantitative measure of the distribution and similarity of heatmaps, highlighting the variability and focus of the model's attention for each instrument.

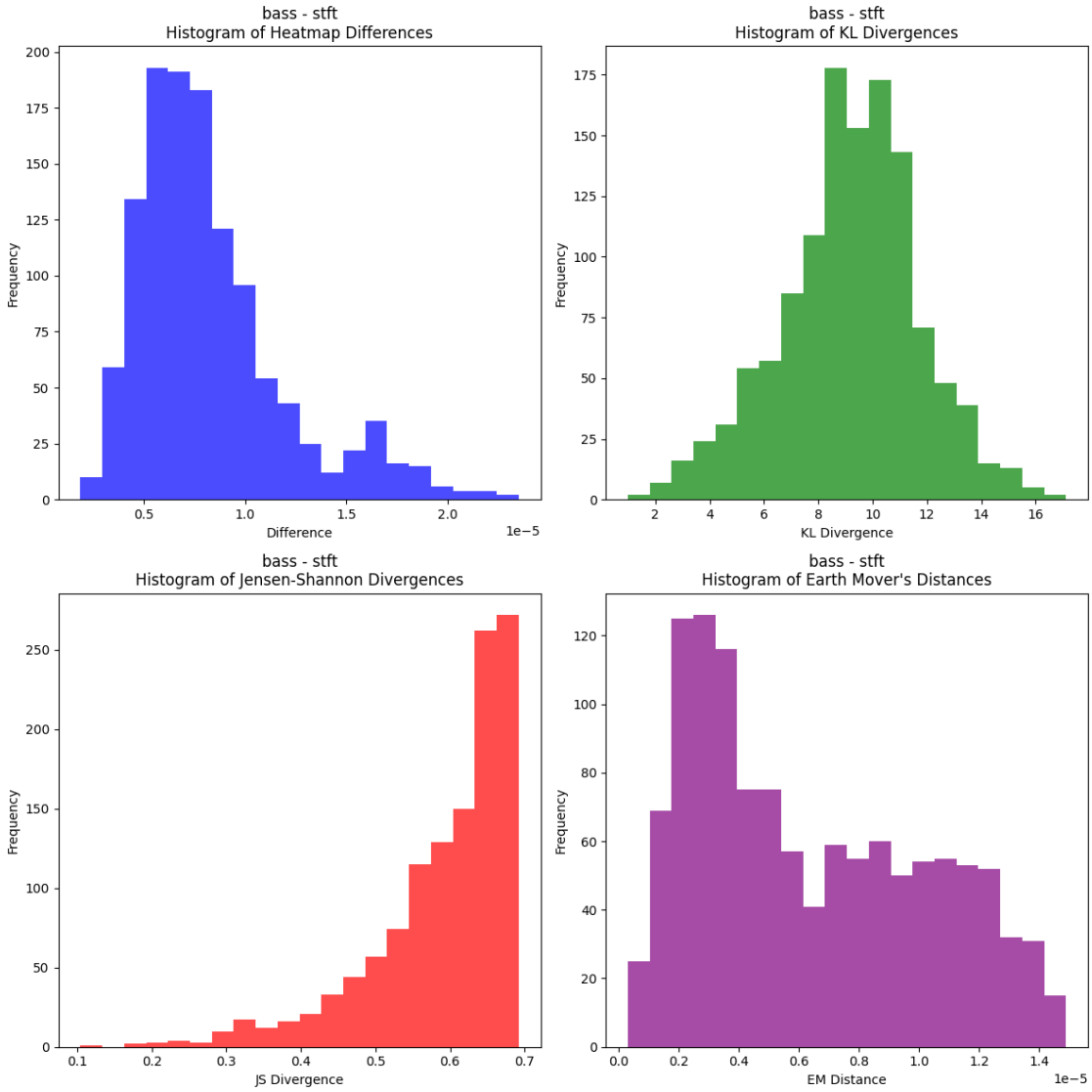


Figure 60. Histogram, KL Divergence, JS divergence, EM distance of bass class heatmaps.

According to Figure 60, the bass instrument shows a relatively small average difference between heatmaps, indicating that different samples of bass are quite similar in their feature maps. However, the high KL Divergence values suggest that the probability distributions between these samples vary significantly. This means that the classifier's predictions for bass are not consistent. The high Jensen-Shannon Divergence supports this finding, showing significant dissimilarity between probability distributions of different bass samples. The Earth Mover's Distance (EMD) indicates a moderate variation in the shape of these distributions, further illustrating inconsistency in the classifier's performance for bass.

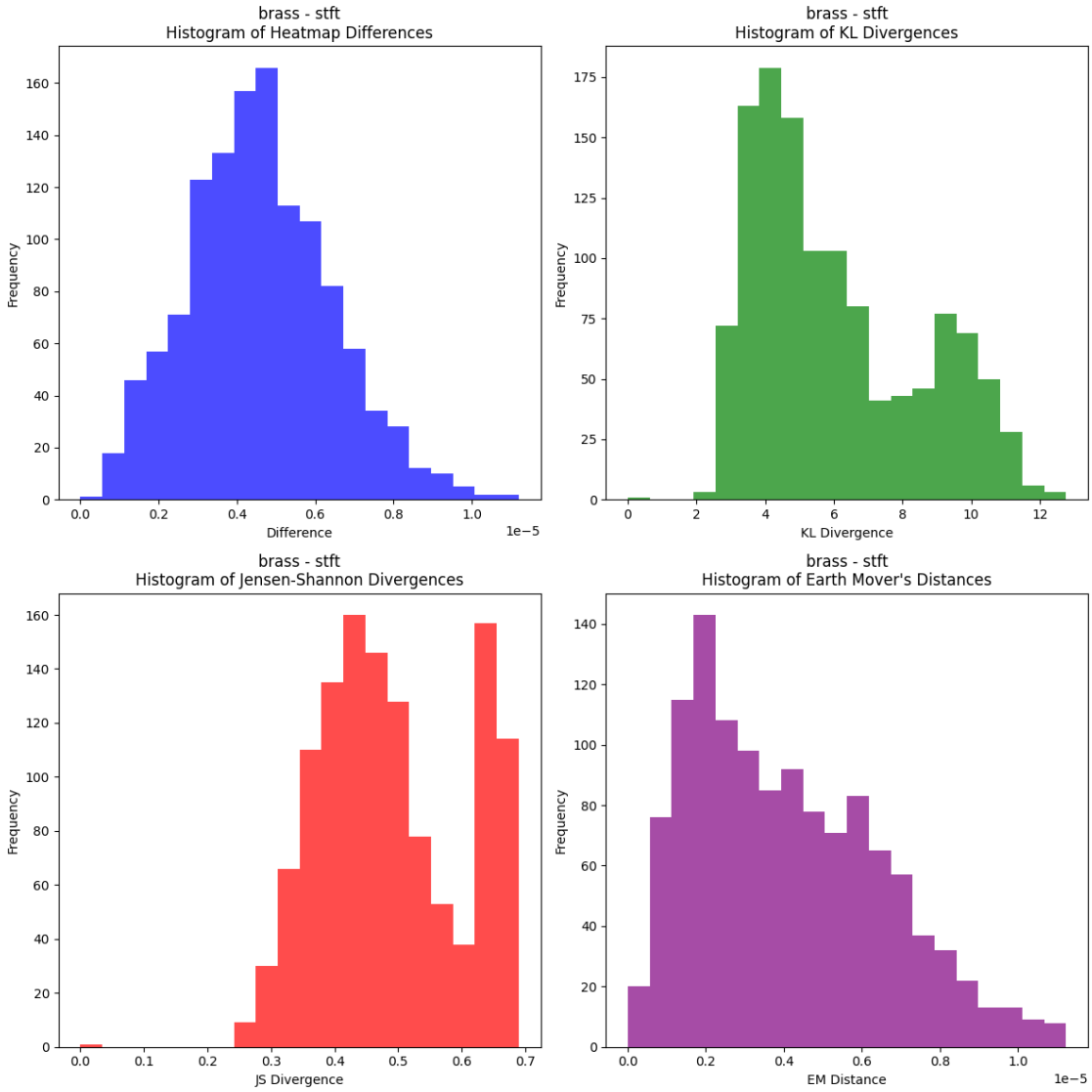


Figure 61. Histogram, KL Divergence, JS divergence, EM distance of brass class heatmaps.

According to Figure 61, for the brass instrument, the difference mean is moderate, indicating a reasonable consistency in heatmaps. The KL Divergence is lower than that of bass, suggesting better consistency in the classifier's predictions. The lower Jensen-Shannon Divergence indicates better similarity between probability distributions of different brass samples. The EMD is moderate, reflecting a consistent, though not perfect, distribution shape. This suggests that the classifier performs more reliably for brass compared to bass.

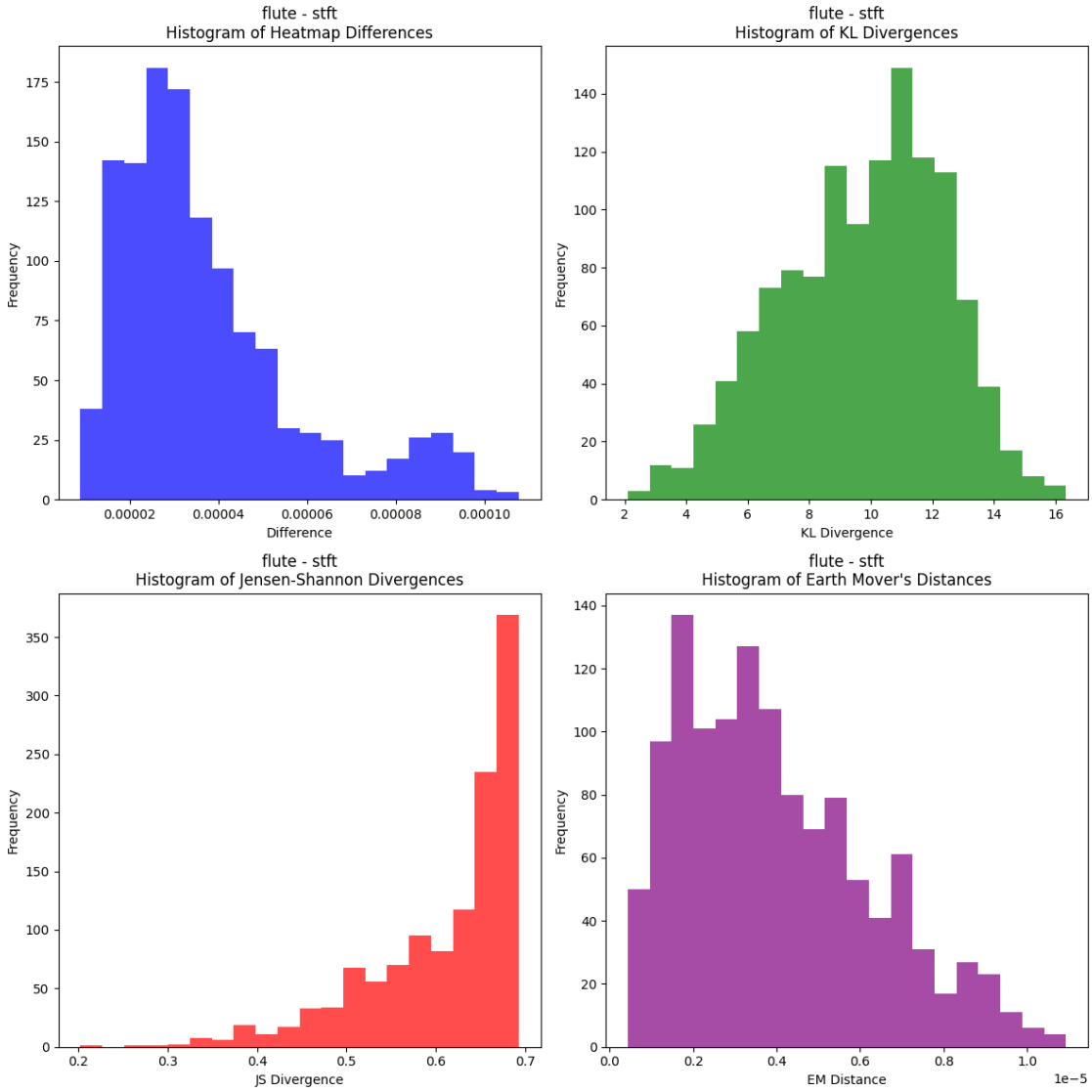


Figure 62. Histogram, KL Divergence, JS divergence, EM distance of flute class heatmaps.

According to Figure 62, the flute exhibits a moderate difference mean, showing consistent heatmaps across samples. The KL Divergence is lower, indicating that the classifier's predictions are more stable for flute than for bass. Similarly, the lower Jensen-Shannon Divergence suggests better prediction similarity. The moderate EMD shows that the shape of the distributions is reasonably consistent, further supporting the classifier's stability in identifying flute samples.

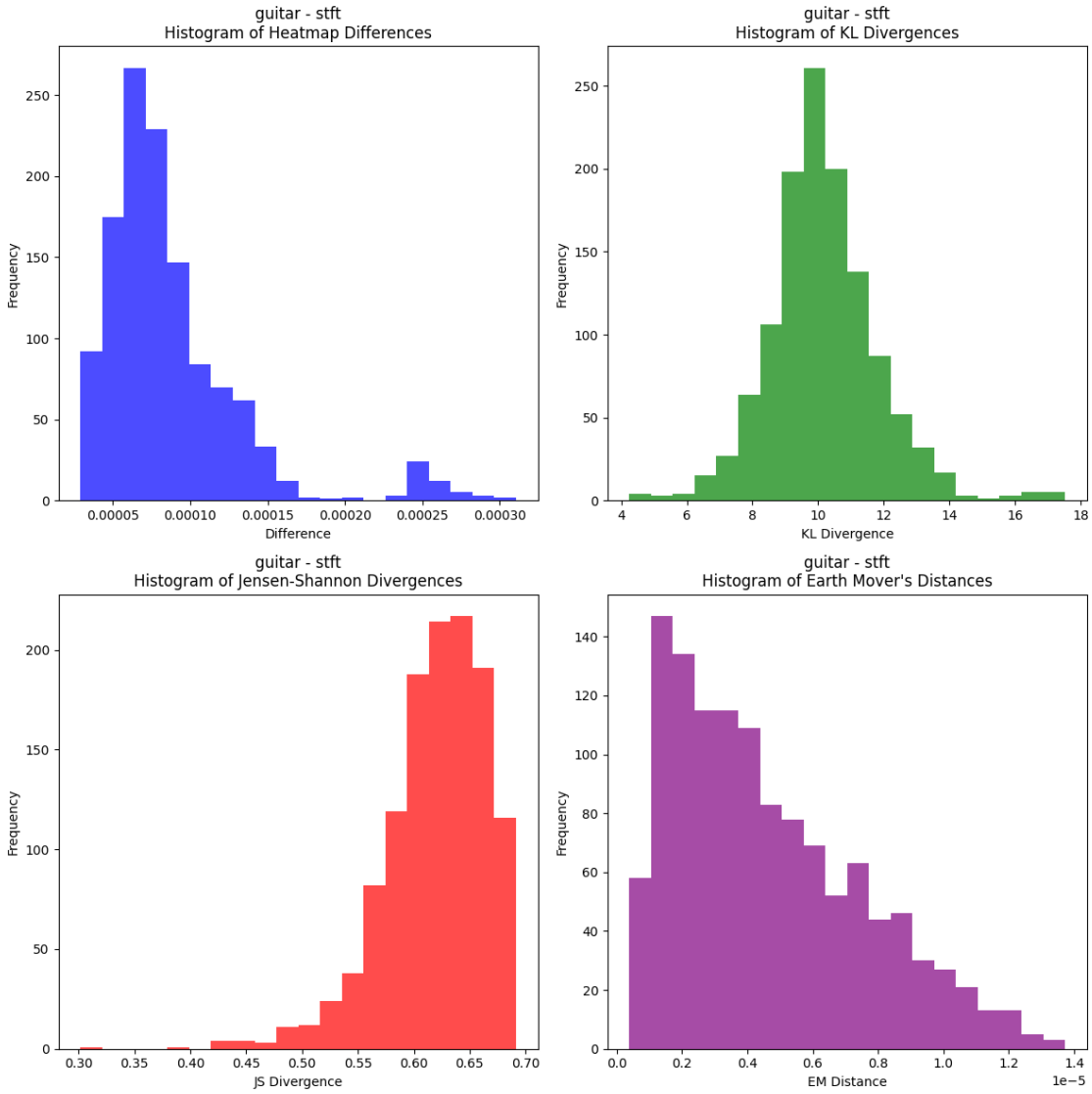


Figure 63. Histogram, KL Divergence, JS divergence, EM distance of guitar class heatmaps.

According to Figure 63, the guitar has a moderate average difference between heatmaps, indicating some variation between samples. The KL Divergence is higher than for the flute, suggesting some instability in the classifier's predictions. The higher Jensen-Shannon Divergence indicates less similarity between probability distributions of guitar samples. The EMD also shows some variation in the shape of the distributions, indicating that the classifier's performance for guitar is less stable compared to other instruments like flute and brass.

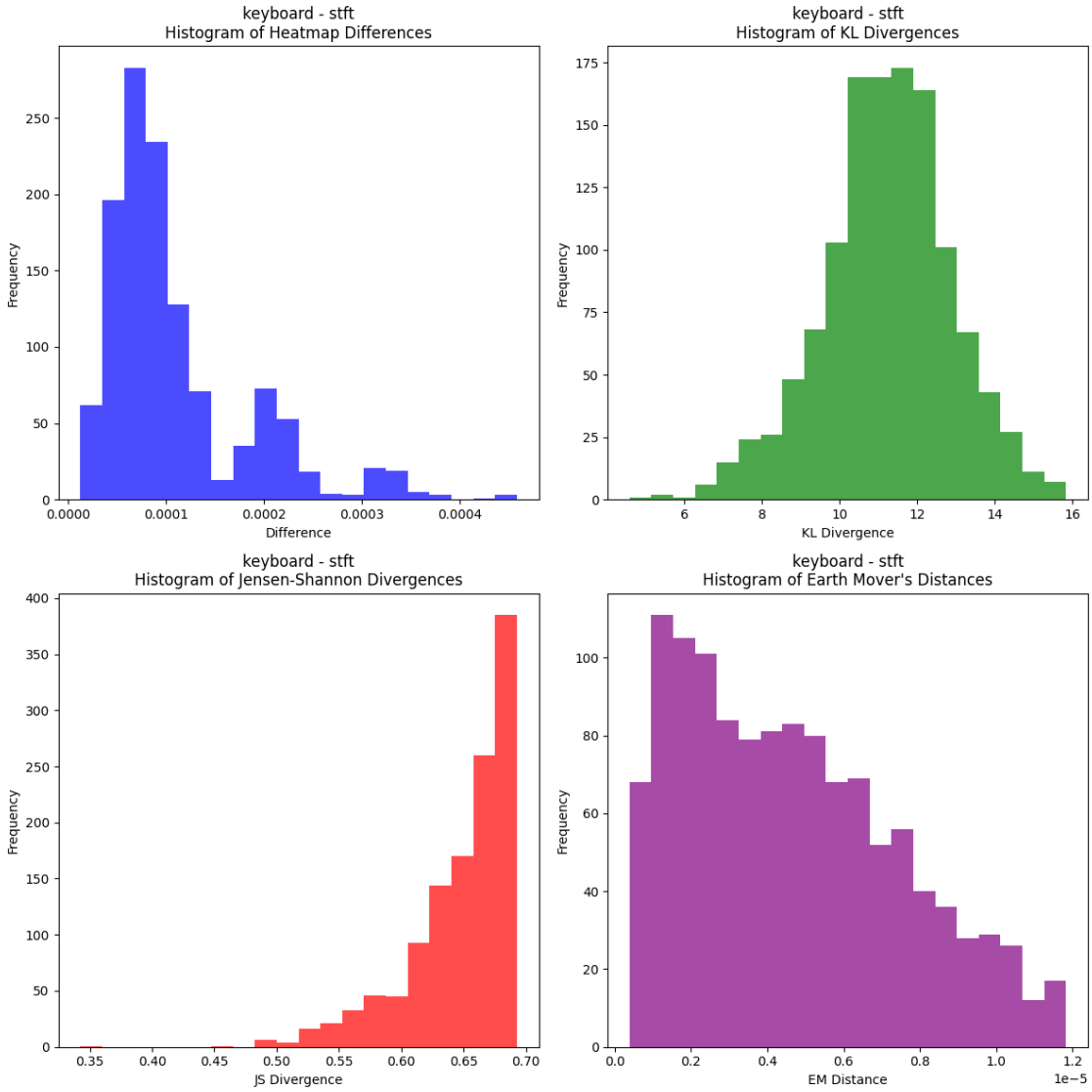


Figure 64. Histogram, KL Divergence, JS divergence, EM distance of keyboard class heatmaps.

According to Figure 64, for the keyboard, the difference mean is moderate, indicating consistent heatmaps across samples. The lower KL Divergence suggests that the classifier's predictions are relatively stable. The lower Jensen-Shannon Divergence further indicates high similarity between probability distributions of keyboard samples. The moderate EMD reflects a consistent distribution shape, supporting the classifier's stability and reliability in identifying keyboard samples.

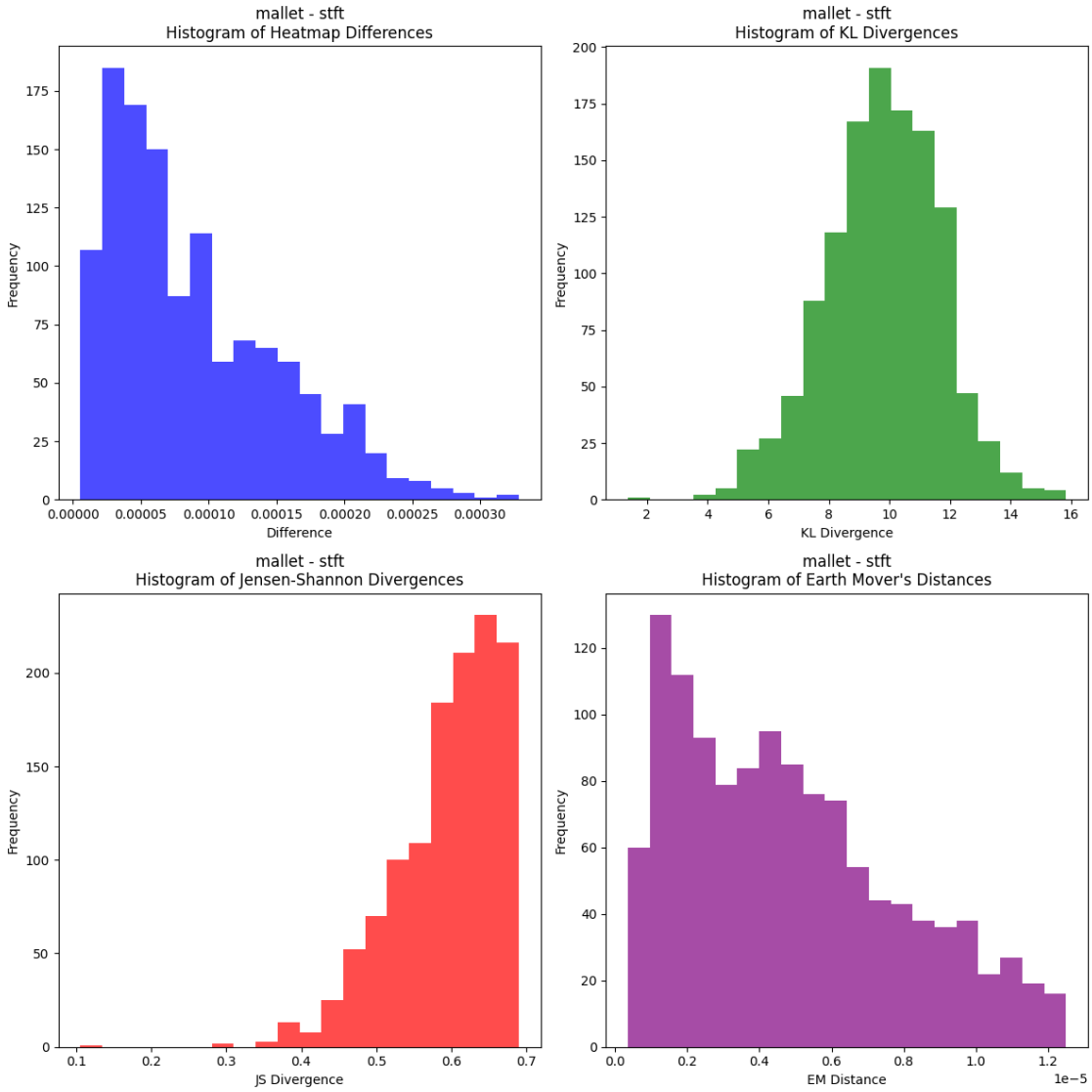


Figure 65. Histogram, KL Divergence, JS divergence, EM distance of mallet class heatmaps.

According to Figure 65, the mallet instrument shows good consistency between heatmaps, as indicated by a low difference mean. The low KL Divergence suggests very stable classifier predictions. The low Jensen-Shannon Divergence indicates high similarity between probability distributions. The EMD reflects consistent distribution shapes, suggesting that the classifier performs reliably for mallet samples.

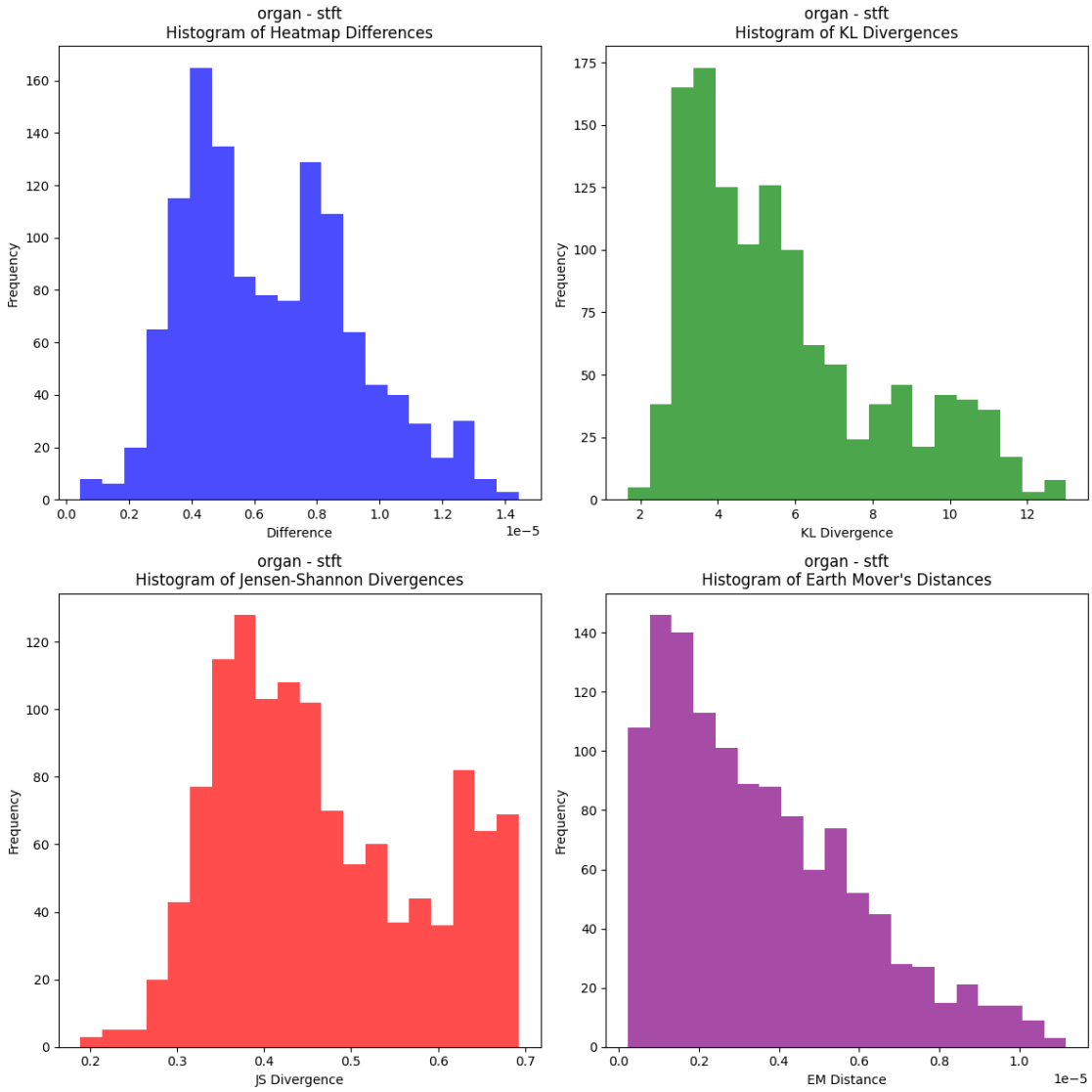


Figure 66. Histogram, KL Divergence, JS divergence, EM distance of organ class heatmaps.

According to Figure 66, The organ instrument exhibits very low difference mean, indicating excellent consistency between samples. The very low KL Divergence suggests extremely stable predictions by the classifier. Similarly, the very low Jensen-Shannon Divergence shows very high similarity between probability distributions. The EMD is also very consistent, indicating that the classifier performs exceptionally well for organ samples, with minimal variability in predictions.

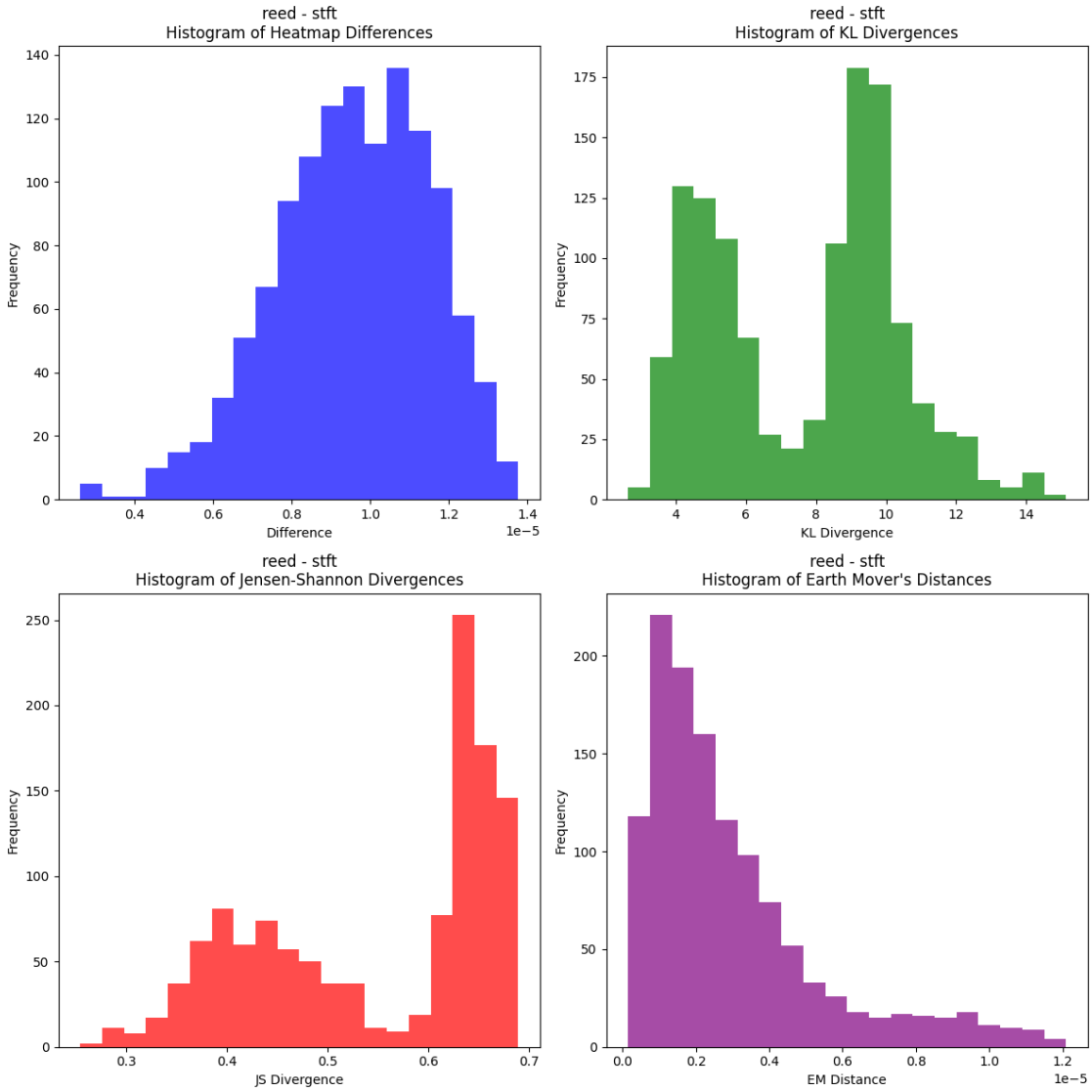


Figure 67. Histogram, KL Divergence, JS divergence, EM distance of reed class heatmaps.

According to Figure 67, the reed instrument has a moderate difference mean, indicating reasonable consistency between samples. The higher KL Divergence suggests some instability in the classifier's predictions. The higher Jensen-Shannon Divergence indicates less similarity between probability distributions of reed samples. The EMD shows some variation in distribution shapes, reflecting less stable classifier performance for reed compared to more consistent instruments like organ and mallet.

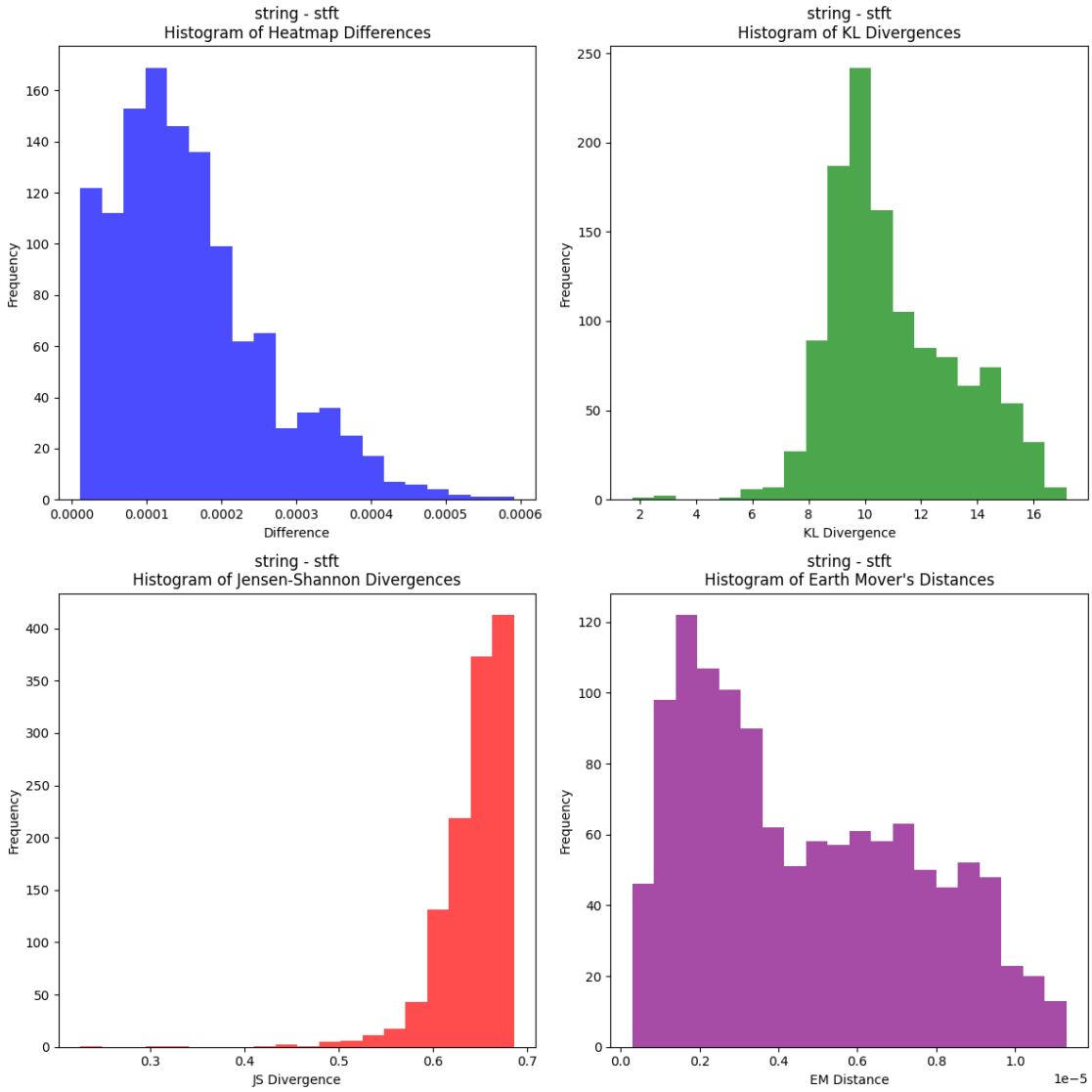


Figure 68. Histogram, KL Divergence, JS divergence, EM distance of string class heatmaps.

According to Figure 68, the string instrument shows good consistency with a low difference mean. The low KL Divergence indicates stable predictions by the classifier. The low Jensen-Shannon Divergence suggests high similarity between probability distributions. The EMD reflects consistent distribution shapes, indicating that the classifier performs reliably for string samples.

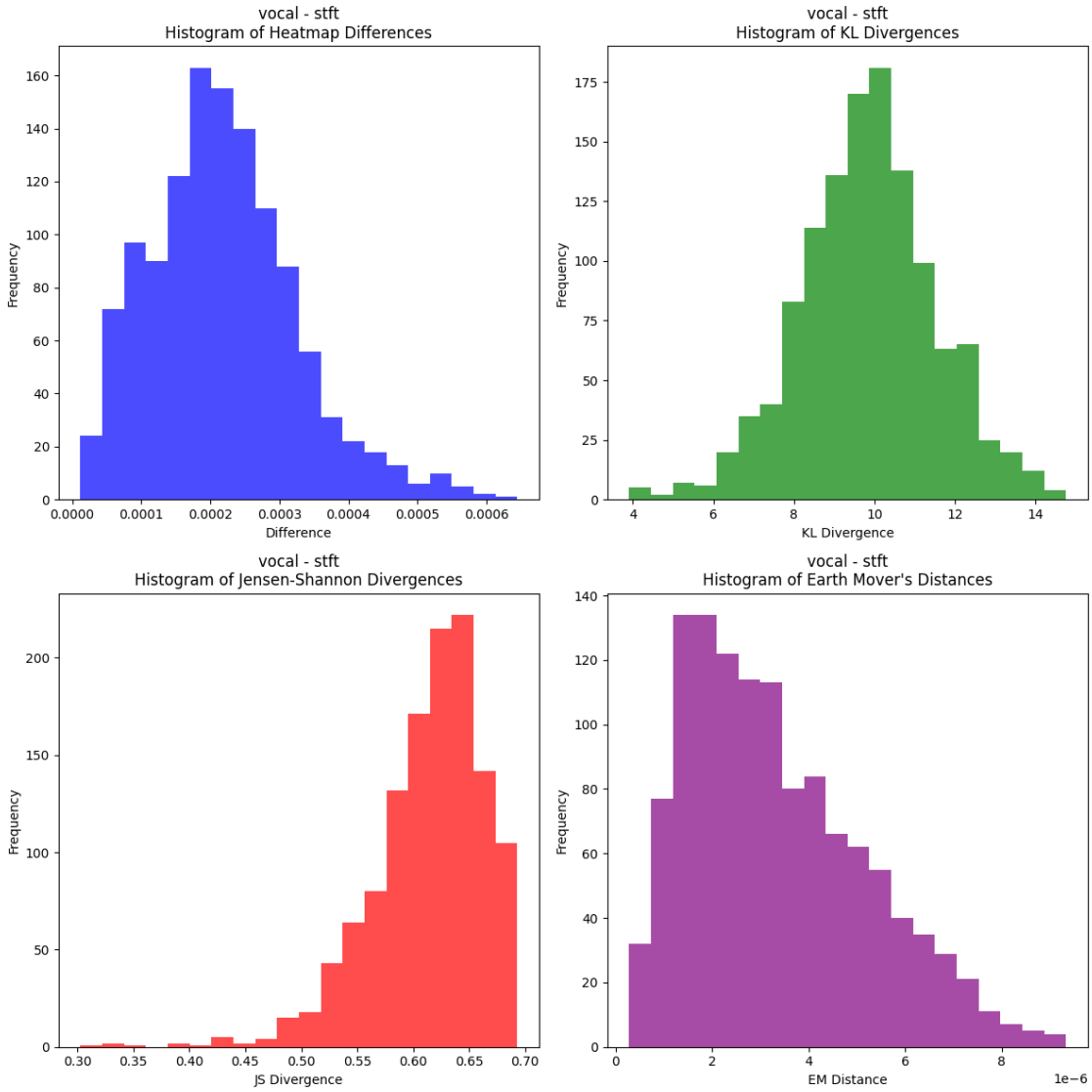


Figure 69. Histogram, KL Divergence, JS divergence, EM distance of vocal class heatmaps.

According to Figure 69, the vocal instrument exhibits a moderate difference mean, indicating reasonable consistency between samples. The moderate KL Divergence suggests some instability in classifier predictions. The moderate Jensen-Shannon Divergence indicates reasonable prediction similarity. The EMD reflects consistent but varied distribution shapes, suggesting that the classifier performs reasonably well for vocal samples but with some variability in predictions.

5.5.7 Summary

The four metrics used provide different perspectives on the performance of the classifiers. The Difference Mean gives a straightforward measure of the average difference between heatmaps, reflecting overall consistency. KL Divergence and Jensen-Shannon Divergence are more sensitive to differences in probability distributions, highlighting prediction stability and similarity. The Earth Mover's Distance focuses on the shape of the distributions, offering insight into how much one distribution needs to be adjusted to resemble another.

For bass, the high divergence values and moderate EMD indicate significant prediction instability and variability in distribution shapes, contributing to its poor performance. In contrast, instruments like organ show very low divergence values and consistent EMD, indicating very stable and similar predictions, which corresponds to their higher classification performance. This analysis highlights the importance of using multiple metrics to fully understand classifier behaviour and the challenges faced with different instruments.

Our analysis reveals that the bass instrument suffers from poor feature extraction and lack of detail in the feature maps and heatmaps, leading to lower classification performance. To address this, we introduce a new combined spectrogram approach in the chapter 5.7 which includes additional features to improve the CNN's learning and classification accuracy for bass and other instruments. This combined spectrogram leverages the strengths of multiple spectrogram types, providing richer information for the model to learn from.

5.6 Evaluate Binary Classifiers on Open-MIC Dataset

Building upon the insights gained from our experiments with the NSynth dataset and the feature map analysis, this section addresses Research Objective 2 (RO-2) by evaluating our binary classifiers on the Open-MIC dataset (E. Humphrey et al., 2018). This evaluation aims to assess the scalability and generalizability of our OvA model in a more diverse and challenging context. The Open-MIC dataset, developed specifically for evaluating multiple instrument recognition algorithms, provides an ideal testbed for our model. It contains 20,000 10-second audio clips sourced from the Free Music Archive (FMA), representing a wide range of genres and 20 distinct

instrument classes. This dataset's complexity, with its multi-label classification tasks and partially annotated clips, allows us to rigorously test our model's performance beyond the controlled environment of NSynth. By applying our binary classifiers to this dataset, we can evaluate how well our approach scales to a larger number of instrument classes and performs on real-world, polyphonic music samples, directly addressing the scalability aspect of RO-2 and paving the way for future improvements in our instrument recognition system. This aligns with the iterative methodology described in Chapter 3.2.7, where each iteration re-evaluates the earlier ROs to refine and validate the approach.

5.6.1 Dataset Introduction and Benchmarks

Chong et al. (2023) Their paper, "Masked Spectrogram Prediction for Self-Supervised Audio Pre-Training," introduces a self-supervised learning method called Mask-Spec. The technique involves masking random patches of spectrogram input and training a transformer model to reconstruct the missing patches. The pre-trained model achieves impressive results on Open-MIC, outperforming other methods, including CNN-based models, while achieving a mean average precision (m-AP) of 0.854 on the OpenMIC2018 dataset. Also, the Open-MIC-2018 dataset comparison is presented in Figure 83 on page 261. To identify possible improvements, detailed comparisons against a baseline are included, providing a clear evaluation of performance gains.

Koutini et al. (2020): In "Receptive-Field Regularized CNNs for Music Classification and Tagging," Koutini et al. analyse the use of receptive-field regularization and Shake-Shake methods to improve the generalization ability of deeper CNNs, such as Res-Net, for music-related tasks. They demonstrate the efficacy of their receptive-field regularization strategy on Open-MIC, showing that it allows deeper architectures to achieve competitive results with a better generalization performance than traditional CNN models.

Schindler et al. (2024) : explored the use of spectrogram-based CNNs for musical instrument recognition in Open-MIC, leveraging domain-specific architectures to improve feature extraction and classification and achieved a 0.855 mPA.

Watcharasupat et al. (2020)'s paper appears to focus on techniques for enhancing CNNs or spectrograms to improve classification or tagging accuracy. By refining feature extraction from the spectrograms, the models deliver more accurate predictions on Open-MIC related tasks (Figure 70).

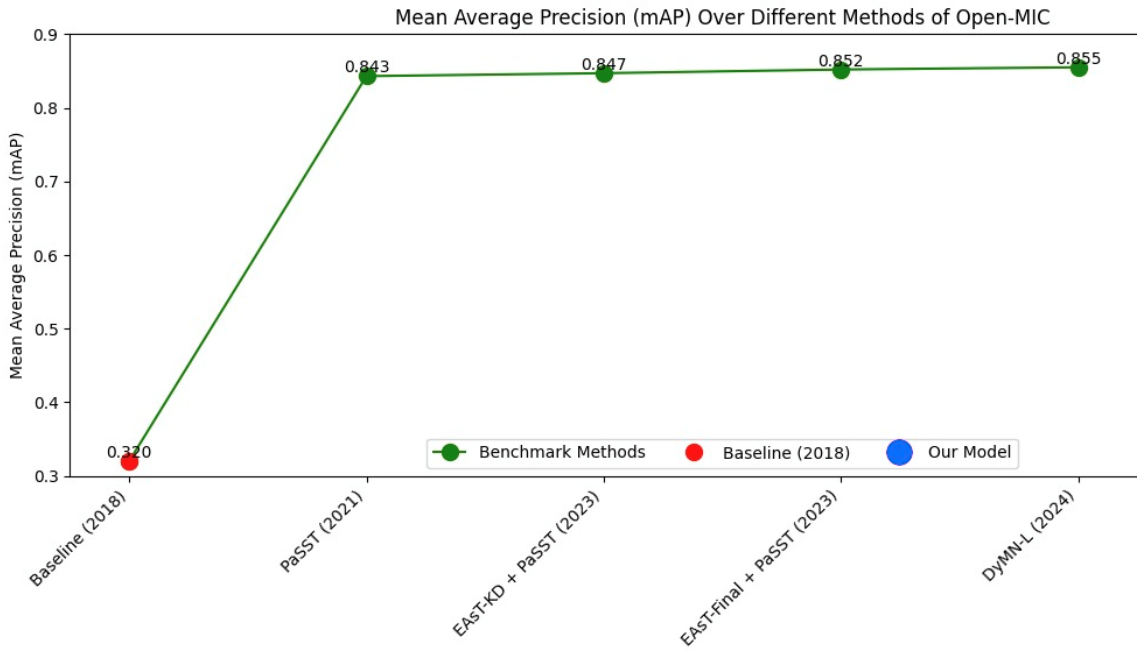


Figure 70. Baseline and Benchmarks of Open-MIC dataset.

The Open-MIC dataset are all 10 second long audio samples (Figure 71) result in our model dysfunctional on this dataset. Thus, we will need to extend our OvA model to be more universal and input-size free.

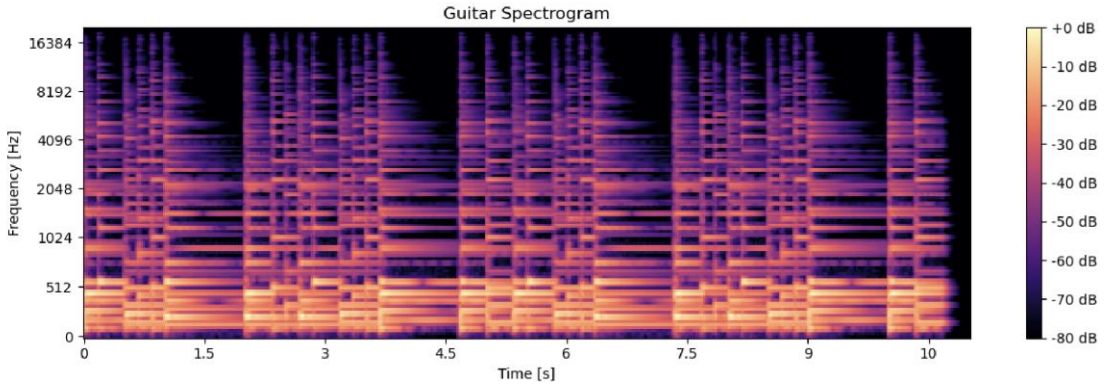


Figure 71. Spectrogram of Open-MIC dataset.

To accommodate the Open-MIC dataset, which consists of 10-second audio clips, we need to extend our existing 4-second model to handle longer audio segments effectively.

5.6.2 Experiment Setup

In the 10-second Open-MIC dataset, most music samples are composed of multiple instruments with multiple labels, making it challenging to extract and train binary models. To address this, we utilize the $MultiLabelBinarizer$. The $MultiLabelBinarizer$ is a tool from the $sklearn$ library (scikit-learn developers, 2019) that efficiently converts a collection of sets or lists of labels into a binary format, where each column corresponds to a potential label and each row corresponds to an instance, indicating the presence (1) or absence (0) of a label. This transformation is particularly useful for handling multi-label classification problems.

In our study, we use the $MultiLabelBinarizer$ to preprocess the Open-MIC dataset, transforming the multi-label annotations into a binary format that can be directly used for training our models. Handling multi-label data in datasets like Open-MIC, where each sample can have multiple labels (e.g., a piece of music featuring multiple instruments), the $MultiLabelBinarizer$ converts this complex labeling into a structured binary format, making it easier to apply machine learning algorithms. This approach ensures that our models can learn to recognize and predict the presence of multiple instruments in a given audio sample, simplifying the preprocessing pipeline and ensuring compatibility with various machine learning models.

5.6.3 CNN Model and Train/Validation/Testing Split

The model architecture (Table 16) designed for the Open-MIC dataset includes various convolutional, batch normalization, and max pooling layers, which are followed by dense and dropout layers to handle the complexity of multi-label classification where each music sample may contain multiple instruments. The input layer accepts grayscale images of size 218x800 pixels. The subsequent Conv2D layers apply convolution operations with ReLU activation to extract feature maps from the input image, with the number of filters increasing with depth to capture more complex features. Batch normalization layers normalize the output of the

convolutional layers to improve training stability and performance. MaxPooling2D layers downsample the feature maps to reduce their dimensionality and computation requirements, while retaining important features. The Flatten layer converts the 3D output from the previous layers into a 1D vector, preparing it for the fully connected (dense) layers. The Dense layer with 512 units and ReLU activation learns global features from the flattened input, while the Dropout layer randomly sets 50% of the input units to 0 at each update during training time to prevent overfitting. Finally, the output layer with a number of units equal to the number of classes and a sigmoid activation function outputs the probability for each class, suitable for multi-label classification.

Table 16. CNN model of Open-MIC experiment:

Layer Name	**Description**
Input	Accepts input images of size 218x800 with 1 channel (grayscale).
Conv2D_1	32 filters, kernel size (3, 3), ReLU activation. Extracts features from the input image.
BatchNorm_1	Normalizes the activations of the previous layer to improve training stability and performance.
MaxPooling2D_1	Downsamples the feature maps by a factor of 2.
Conv2D_2	64 filters, kernel size (3, 3), ReLU activation. Further extracts features from the input image.
BatchNorm_2	Normalizes the activations of the previous layer.
MaxPooling2D_2	Downsamples the feature maps by a factor of 2.
Conv2D_3	128 filters, kernel size (3, 3), ReLU activation. Extracts higher-level features.
BatchNorm_3	Normalizes the activations of the previous layer.
MaxPooling2D_3	Downsamples the feature maps by a factor of 2.
Conv2D_4	256 filters, kernel size (3, 3), ReLU activation. Extracts more complex features.
BatchNorm_4	Normalizes the activations of the previous layer.
MaxPooling2D_4	Downsamples the feature maps by a factor of 2.
Flatten	Flattens the input to a 1D vector to connect to the dense layer.
Dense_1	Fully connected layer with 512 units, ReLU activation. Learns global features.
Dropout	Regularization layer to prevent overfitting by randomly setting 50% of input units to 0.
Output	Fully connected layer with num_classes units, sigmoid activation for multi-label classification.

The dataset includes approximately 15,000 samples for training and validation, and around 5,000 samples for testing. This setup ensures that the model is trained for 300 epochs with a batch size of 32, and 10% of the training data is used for validation to monitor the model's

performance on unseen data during training. The *MultiLabelBinarizer* tool is used to preprocess the dataset by converting the multi-label annotations into a binary format suitable for training the model.

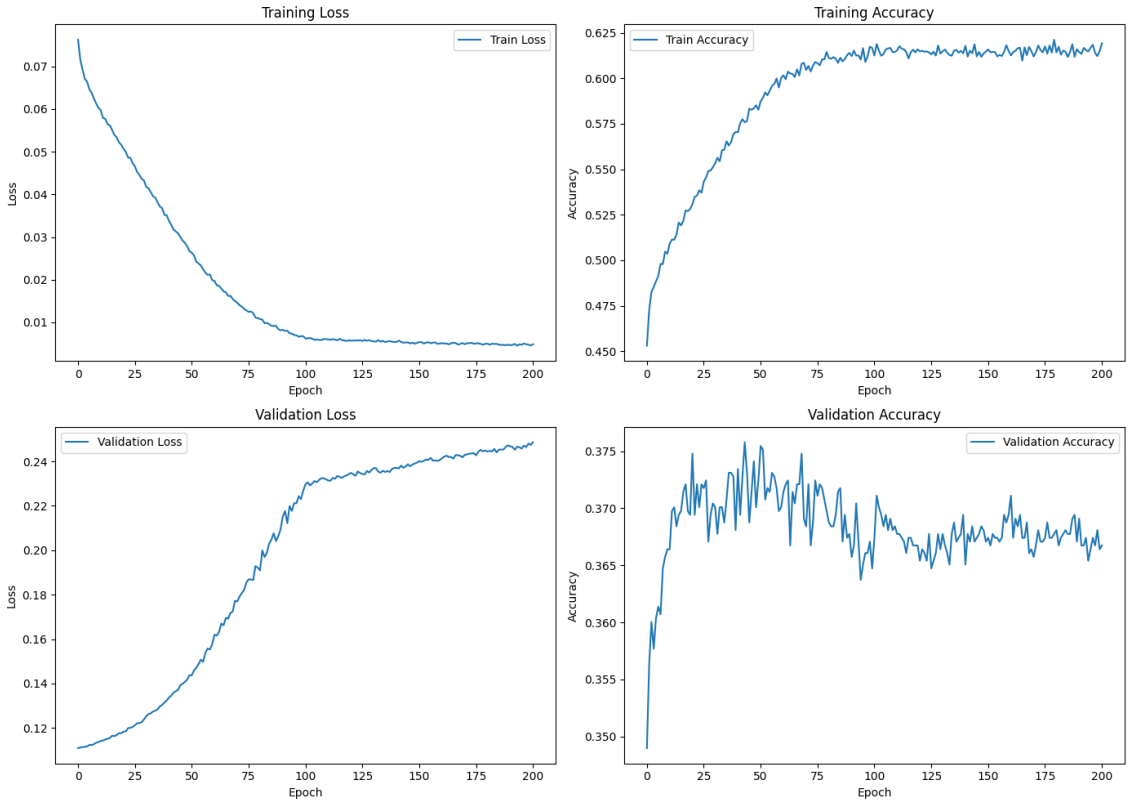


Figure 72. Training and Validation Per Epoch.

Despite these efforts, our model's performance is still suboptimal. As shown in Figure 72, the training accuracy never exceeds 0.625, even after 200 epochs, and the validation accuracy is unstable, remaining around 0.375. This indicates the need for further refinement and potential adjustments to the model architecture or training process to achieve better results.

5.6.4 Result

The following Table 17 presents the classification report of the experiment. Additionally, the mean average precision (mAP) is 0.4195, the Exact Match Ratio (EMR) is 0.3413, and the Hamming Loss is 0.0461. These metrics indicate that while the model has achieved some level of accuracy, there is considerable room for improvement to enhance its performance and reliability.

Table 17. Training Result of Open-Mic Dataset.

Instrument	Precision	Recall	F1-Score
accordion	0.18	0.06	0.09
banjo	0.39	0.12	0.18
bass	0.38	0.16	0.23
cello	0.31	0.08	0.13
clarinet	0.36	0.06	0.1
cymbals	0.54	0.32	0.44
drums	0.48	0.4	0.44
flute	0.31	0.11	0.17
guitar	0.54	0.19	0.28
mallet percussion	0.43	0.11	0.17
mandolin	0.35	0.16	0.22
organ	0.29	0.11	0.16
piano	0.68	0.47	0.55
saxophone	0.38	0.15	0.22
synthesizer	0.57	0.31	0.4
trombone	0.27	0.05	0.08
trumpet	0.29	0.12	0.17
ukulele	0.44	0.19	0.27
violin	0.52	0.24	0.33
voice	0.37	0.17	0.23
Mean Average Precision	0.4195		
Hamming Loss	0.0461		
Exact Match Ratio	0.3413		

5.6.5 Discussion

The following Figure 73 indicates the challenge of classifying the sample.

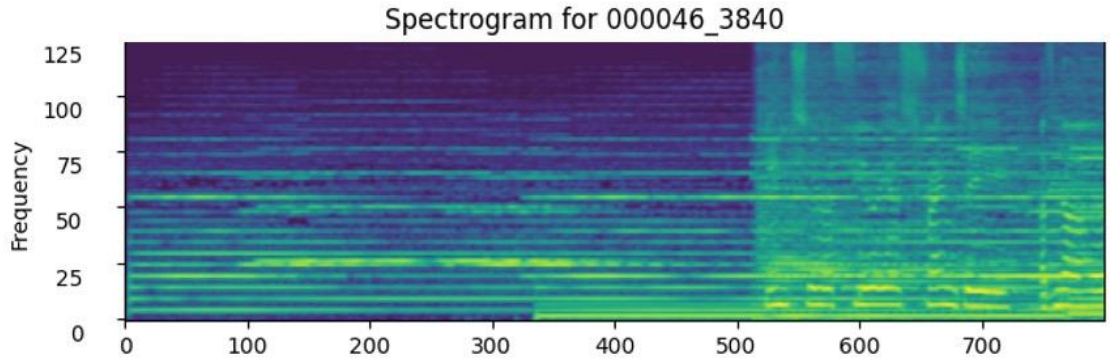


Figure 73. One Example Log-mel Spectrogram of Open MIC dataset.

The example spectrogram illustrating the frequency content of an audio sample over time. As observed, the spectrogram displays how different frequencies evolve throughout the audio clip, highlighting areas where certain frequencies are more prominent. This visualization underscores the challenges in extracting meaningful information from complex audio signals.

This brings us to a critical research question: Is there a novel method that can effectively extract all the necessary information from an audio signal? Finding an innovative approach to address this challenge could significantly enhance our ability to analyse and interpret complex audio data accurately. Thus, we conduct the next experiment of multi-spectrogram.

5.7 Multiple Spectrogram Feature Comparison Experiment

Building upon the insights gained from our experiments with single spectrogram representations and the challenges encountered with the Open-MIC dataset, this section addresses Research Objective 5 (chapter 3.1.1.5) by exploring the potential of multiple spectrogram features for instrument recognition. Our previous experiments revealed limitations in using a single spectrogram type, particularly for complex, real-world audio samples. This experiment aims to compare and evaluate the performance of various spectrogram algorithms (e.g., STFT, Mel, CQT, Chroma) in identifying different types of musical instruments. By combining multiple spectrogram features, we hypothesize that we can capture a more acoustic characteristics, potentially improving the model's ability to distinguish between instruments, especially in challenging polyphonic contexts. This approach not only seeks to enhance the overall accuracy of our instrument recognition system but also aims to identify the most effective combination of spectrogram features for each instrument category, directly addressing the core aspects of RO-5.

5.7.1 Literature Review

The paper titled "AMResNet: An automatic recognition model of bird sounds in real environment" by Hanguang Xiao et al. (2022) highlights the benefits of using a mixed spectrogram approach for bird sound recognition. The authors propose a novel deep learning model, AMResNet, which combines attentional mechanisms with residual networks to enhance classification accuracy. They emphasize the importance of using combined feature sets, which provide a representation of bird sounds, improving the model efficiency and accuracy compared to traditional single-feature methods.

The combined spectrogram is an approach, that can combine the frequency-based spectrogram with other features, like chroma feature, pitch feature and so on

For example, The study Hybrid Spectrogram (Défossez, 2021) combines spectrogram and waveform domains to improve source separation. The hybrid approach leverages the strengths of both representations, demonstrating improved signal-to-distortion ratios (SDR) and better human subjective evaluations.

The multiple aggregated spectrogram analysis research (Su et al., 2020) demonstrates that the combination of CST (Chroma, Tonnetz, and Spectral Contrast) with other spectrogram types such as MFCC, Log-Mel, or STFT consistently enhances the performance of pure frequency spectrograms. This indicates that integrating a primary spectrogram with CST features leads to improved audio analysis and classification outcomes.

Inspired by this approach, we can apply a similar methodology to our musical instrument recognition model. By integrating various spectrogram features, such as STFT, Log-Mel, MFCC, Chroma, Spectral Contrast, and Tonnetz, we aim to capture a wider range of audio characteristics. This feature set will enable our model to better distinguish between different musical instruments, potentially enhancing the overall recognition performance. The success of the AMResNet model in bird sound recognition encourages us to explore mixed spectrogram features for improved accuracy in our own domain.

5.7.2 Experiment setup

The following Figure 74 shows the 6 different spectrograms of a piano sample.

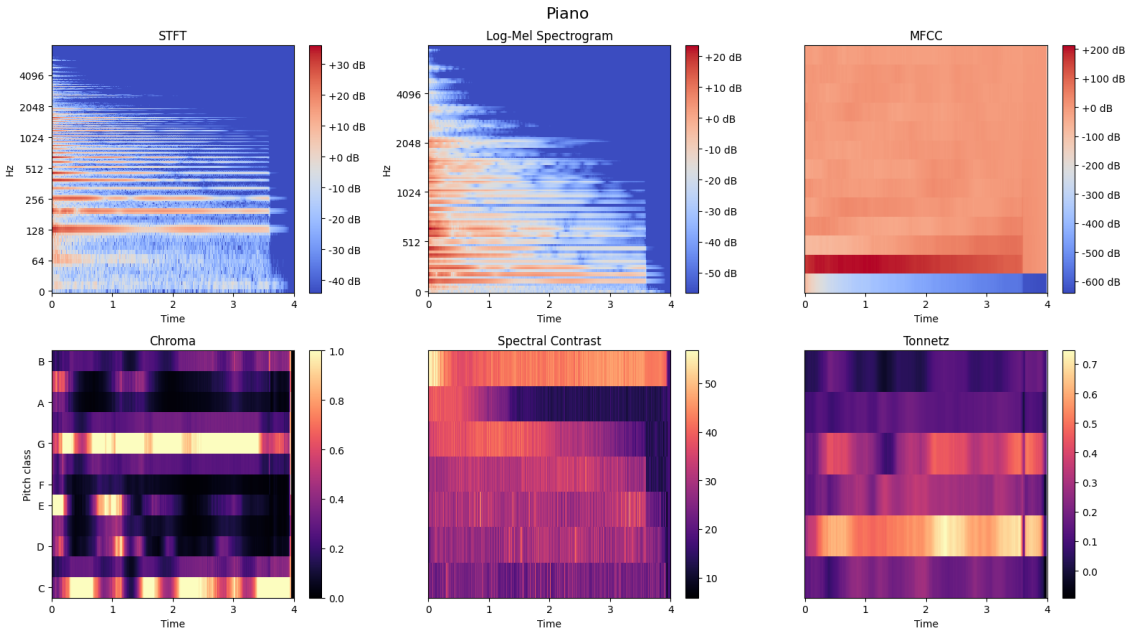


Figure 74. Six different spectrogram algorithm (The interpretation are as follows).

1) Top-left plot : STFT (Short-Time Fourier Transform) Spectrogram:

X-axis (Time) represents time in seconds while the Y-axis (Frequency) represents frequency in Hertz (Hz). Colour (Amplitude) Indicates the amplitude (dB) of the frequency components. The colour map ranges from blue (low amplitude) to red (high amplitude). STFT provides a time-frequency representation of the signal(Allen & Rabiner, 1977a), which is essential for analysing non-stationary signals such as musical instruments and speech.

The effectiveness of STFT in capturing time-varying frequency content makes it particularly suitable for analysing instruments with rapidly changing harmonics and transient sounds. Examples include:

- Piano: The piano produces a wide range of frequencies and its percussive nature results in rapid changes in amplitude. STFT captures these transient characteristics effectively. Research has shown that STFT is effective in analysing piano sounds due to its ability to represent the temporal dynamics of the instrument's harmonics and transients(Rossi & Girolami, 2001; Thornburg et al., 2007).
- Drums: Drums generate sharp, transient sounds with significant variations in frequency over short periods. STFT is ideal for capturing these rapid amplitude and frequency changes. Studies (Fitzgerald, 2004) have demonstrated that STFT is particularly useful for drum sound analysis because of its high temporal resolution, which is crucial for transient-rich signals .

2) Top-middle plot : Log-Mel Spectrogram:

Y-axis (Frequency) represents frequency in Hertz (Hz), scaled logarithmically.

Colour (Amplitude) indicates the log-scaled amplitude of the frequency components. The colour map ranges from blue (low amplitude) to red (high amplitude). The Log-Mel spectrogram is computed by applying the Mel scale to the frequencies obtained from the Short-Time Fourier Transform (STFT) and then taking the logarithm of the amplitude (Davis & Mermelstein, 1980; Stevens et al., 1937). The Mel scale is a perceptual scale of pitches judged by listeners to be equal in distance from one another.

The effectiveness of Log-Mel in capturing perceptually relevant frequency components is particularly beneficial for instruments with complex harmonic structures and sustained notes. Examples include:

- Violin: Longari, and Pollastri (2003) ‘s research shows the violin produces rich, harmonic content with significant variation in intensity across different frequencies. The Log-Mel spectrogram effectively represents these nuances. Research indicates that the Log-Mel spectrogram is effective for violin sound analysis due to its ability to capture the fine details of harmonic structures.
- Voice: Human voice, with its varied pitch and timbre, benefits from the perceptual scaling of the Mel spectrum, allowing for accurate recognition and analysis. Studies have shown that the Log-Mel spectrogram is highly effective for voice recognition and analysis (Liang et al., 2021), providing a detailed representation of vocal timbre and pitch variations.

3) Top-right plot : MFCC (Mel-Frequency Cepstral Coefficients):

X-axis (Time) represents time in seconds. Y-axis (MFCC Coefficients) represents the different MFCC coefficients. Colour (Amplitude): Indicates the amplitude of the coefficients. The colour map ranges from blue (low amplitude) to red (high amplitude).

MFCCs provide a compact representation of the spectral envelope (Mermelstein, 1976), which is instrumental in distinguishing the timbre of different instruments. The spectral envelope captures the overall shape of the spectrum, which is crucial for recognizing instruments with unique timbral qualities. This feature set is particularly effective in isolating the fine details of an instrument's sound, such as its harmonic structure and resonant frequencies.

The effectiveness of MFCCs in capturing unique timbral qualities(Logan & others, 2000; Tzanetakis & Cook, 2002) is exemplified by instruments such as:

- Flute: The flute has a relatively simple harmonic structure and a clear, smooth spectral envelope. MFCCs effectively capture these characteristics, making it easier to identify the flute based on its unique timbral signature.

- Clarinet: The clarinet produces a rich, complex harmonic content with notable timbral variations. MFCCs can accurately represent these variations, capturing the clarinet's distinct spectral envelope and aiding in its identification.

4) Bottom-left plot : Chroma

X-axis (Time) represents time in seconds. Y-axis (Pitch Class) represents the 12 different pitch classes (C, C#, D, D#, E, F, F#, G, G#, A, A#, B). Colour (Intensity): Indicates the intensity of each pitch class. The colour map ranges from black (low intensity) to yellow (high intensity).

Chroma features (or chromagrams) project the entire spectrum onto 12 bins representing the 12 distinct semitones (or chroma) of the musical octave. Chroma features are particularly effective for analysing harmonic and melodic characteristics of music, as they reflect the intensity of each of the 12 semitones regardless of the octave (Takuya, 1999). Chroma features capture the harmonic content and pitch classes of an audio signal. They are particularly useful for instruments where harmonic and pitch relationships are prominent, allowing the model to discern between different pitch classes and their harmonic structures.

- Guitar: The guitar's rich harmonic content and the clear presence of pitch classes from its plucked strings are well represented by chroma features (Ezzaidi et al., 2012; Vergés Franch, 2021). These features highlight the harmonic relationships and pitch patterns typical of guitar music.
- Organ: The organ's capability to produce multiple harmonics and clear pitch classes makes chroma features suitable for its analysis. Chroma features can effectively capture the organ's harmonic structure and the relationships between different pitches it produces(Hall et al., 2014; Weiß & Habryka, 2014).

5) Bottom-middle plot : Spectral Contrast:

X-axis (Time) represents time in seconds. Y-axis (Frequency Bands) represents different frequency bands. Colour (Amplitude): Indicates the amplitude contrast between peaks and valleys within each frequency band. The colour map ranges from black (low contrast) to yellow (high contrast).

Spectral contrast features measure the difference in amplitude between peaks (high energy) and valleys (low energy) in each frequency sub-band. This helps in distinguishing between different timbres of sound. Spectral contrast highlights the relative levels of harmonics and formants, which are crucial for identifying different musical instruments and sounds(Jiang et al., 2002) as it highlights the relative levels of harmonics and formants. For example,

- Saxophone: The saxophone exhibits a dynamic range and significant differences between harmonic peaks and formants. Spectral contrast captures these variations, providing a detailed representation of the saxophone's timbral characteristics (Frazier et al., 2019; W. Li et al., 2015).
- Trumpet: The trumpet's bright and piercing sound is characterized by sharp peaks and valleys in its spectrum (Daffern & Howard, 2012). Spectral contrast effectively represents these features, aiding in the accurate identification of the trumpet.

6) Tonnetz: Bottom-right plot

X-axis (Time) represents time in seconds. Y-axis (Tonnetz Dimensions) represents the different dimensions of the Tonnetz (Tonal Centroid Features). Colour (Intensity): Indicates the intensity within each Tonnetz dimension. The colour map ranges from black (low intensity) to yellow (high intensity).

The Tonnetz (tonal network) represents the harmonic relations of pitches in a geometrically structured way, capturing relationships such as fifths, thirds, and minor thirds. Tonnetz features are effective for capturing the harmonic structure of music, which is necessary for tasks like chord recognition and key detection (Harte et al., 2006). Tonnetz features capture harmonic relationships between pitches, which is particularly useful for instruments that play chords and exhibit strong harmonic progressions. These features are effective in highlighting the harmonic structure and relationships in the audio signal. For example:

- Harp: The harp frequently plays chords with rich harmonic relationships. Tonnetz features capture these relationships, providing a detailed representation of the harmonic progressions in harp music.

- Accordion: The accordion's ability to play multiple harmonics and chords simultaneously makes Tonnetz features effective for its recognition(Harte, 2010). These features can represent the complex harmonic interactions in accordion music.

The following Figure 75 shows the all combined spectrogram of a piano sample.

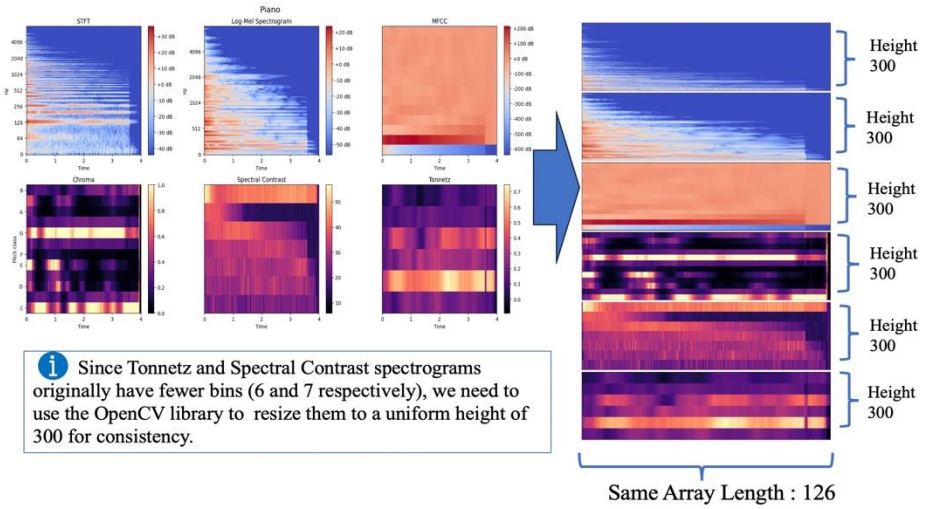


Figure 75. All combined Sample.

Individual spectrograms for Piano using STFT, Log-Mel, MFCC, Chroma, Spectral Contrast, and Tonnetz. Right: Combined spectrogram created by resizing and concatenating these to a uniform height of 300 and array length of 126 using OpenCV. This standardization ensures consistent input size for the neural network. Note: Tonnetz and Spectral Contrast originally have fewer bins (6 and 7), so resizing is necessary.

5.7.2.1 Spectrogram Samples

We employed a traditional train-validation-test split methodology to ensure robust model evaluation and to prevent overfitting. The dataset was divided into three subsets: training, validation, and testing. The training set was used to fit the model, the validation set was used to fine-tune the model's hyperparameters and monitor for overfitting, and the test set was reserved for evaluating the model's performance on unseen data, ensuring the model's generalizability.

Each type of spectrogram dataset was split into these three subsets. For our study, we used six different types of spectrograms: STFT, Log-Mel, MFCC, Chroma, Spectral Contrast, and Tonnetz, plus a combined spectrogram type that included all six types. We generated 2,000

NSynth samples per spectrogram type, resulting in a total of 14,000 spectrogram images (2,000 samples for each of the six types plus 2,000 for the combined type).

For training and validation, we used 1,500 samples per spectrogram type, which corresponds to 150 samples per instrument. A validation split ratio of 0.3 was applied, meaning 1,050 samples per spectrogram type (or 105 samples per instrument) were used for training, and 450 samples per spectrogram type (or 45 samples per instrument) were used for validation. This setup allowed us to tune the model effectively and prevent overfitting by monitoring its performance on the validation set during training. The models were trained for 1,000 epochs to ensure convergence and optimal learning.

The remaining 500 samples per spectrogram type were allocated for testing. These samples, converted into six different spectrogram types plus the combined type, resulted in 3,000 testing spectrogram images. This extensive testing set provided an evaluation of the model's ability to generalize to new, unseen data.

5.7.2.2 Convolutional Network

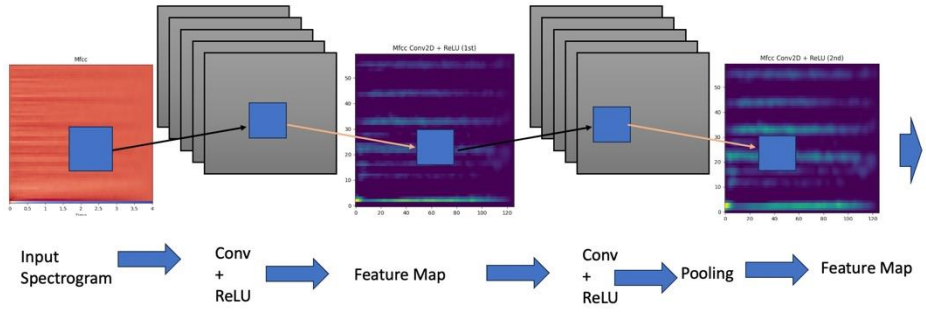


Figure 76. The picture shows the process of feature extraction for the MFCC spectrogram of a guitar classification using our convolutional neural network (CNN) model. The input spectrogram passes through multiple layers of convolutional filters with ReLU activation, followed by pooling layers to create feature maps.

The convolutional neural network (CNN) structure (Figure 76) consists of three convolutional blocks. Each block contains two convolutional layers with filters of sizes 32, 64, and 128 respectively, all using a 3x3 kernel and ReLU activation with 'same' padding. Each block is followed by a max-pooling layer of size 2x2 and a dropout layer with a 0.25 rate. After the convolutional blocks, the model has a flatten layer, a dense layer with 256 units and ReLU activation, a dropout layer with a 0.5 rate, and a final dense layer with 1 unit and sigmoid

activation for binary classification. This architecture is designed to effectively learn from the spectrogram representations and classify the instruments accurately.

5.7.2.3 Controlled Experiment Design

Using the "One-Versus-All" (OVA) approach, we trained separate binary classifiers for each instrument, labeling the target instrument as the positive class and all others as the negative class. This method ensured the model could distinguish each instrument effectively. Since we used six spectrogram types (STFT, Log-mel, MFCC, Chroma, Spectral Contrast, and Tonnetz) and ten instruments (Bass, Brass, Flute, Guitar, Keyboard, Mallet, Organ, Reed, String, and Vocal), this resulted in a total of 60 models. Each model was trained and tested using the dataset described in 5.7.2.1 and saved in Python .h5 format for further analysis. This setup allowed us to compare the performance of each spectrogram type across different instruments, focusing on single-label classification for isolated instrument sounds. The following figure illustrates the experimental design.

Figure 77 is the illustration shows the analysis of ten acoustic instruments from the NSynth dataset. Bass and Brass are displayed with their waveforms and six spectrograms (STFT, Log-Mel, MFCC, Chroma, Spectral Contrast, Tonnetz). The other eight instruments (Flute, Guitar, Keyboard, Mallet, Organ, Reed, String, Vocal) follow the same analysis pattern. The curly brackets highlight the six spectrogram types used for each instrument.

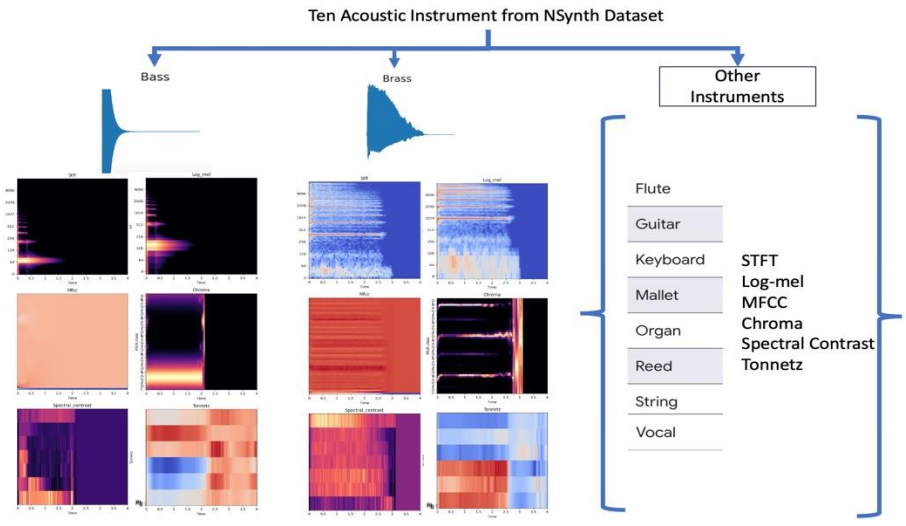


Figure 77. The illustration shows the analysis of ten acoustic instruments from the NSynth dataset.

Figure 77 shows the analysis of ten acoustic instruments from the NSynth dataset. Bass and Brass are displayed with their waveforms and six spectrograms (STFT, Log-Mel, MFCC, Chroma, Spectral Contrast, Tonnetz). The other eight instruments (Flute, Guitar, Keyboard, Mallet, Organ, Reed, String, Vocal) follow the same analysis pattern. The curly brackets highlight the six spectrogram types used for each instrument.

5.7.2.4 All Combined Model T-test

At the end of our experiment, we evaluated the effectiveness of combining multiple spectrogram types for classifying musical instruments. We applied the combined spectrogram model to the same dataset and samples used in previous experiments for consistency. Using paired t-tests, we statistically analysed the performance differences between the combined model and each individual spectrogram model. The performance figures were ranked from best to worst before conducting the t-tests.

To evaluate the benefits of combining multiple spectrogram types for musical instrument classification, we conducted paired t-tests ($t = \frac{\bar{d}}{s_d/\sqrt{n}}$) and p-tests ($p = P(T \geq t)$) comparing the combined spectrogram against each individual spectrogram type. where $\bar{d}$ is the mean difference between the combined model accuracy and individual model accuracy. s_d is the standard deviation of the differences, and n is the number of instruments.

5.7.3 Result

The combined spectrogram generally performs well, with notable improvements in precision and recall for certain instruments. For instance, the combined model shows a significant precision increase for Bass and Brass compared to individual spectrograms. It also maintains high recall rates for most instruments, ensuring consistent classification performance. Overall, the combined spectrogram achieves the highest overall accuracy (0.63) compared to individual spectrogram types, demonstrating its effectiveness in capturing diverse acoustic features for musical instrument classification (Table 18).

Table 18. Evaluation Metrics of 7 Spectrogram Scenarios.

		STFT	Log Mel	MFCC	Chroma	Spectral Contrast	Tonnetz	Combined
Bass	Precision	0.34	0.35	0.52	0.38	0.41	1	0.4
	Recall	0.5	0.4	0.24	0.38	0.5	0.06	0.44
Brass	Precision	0.47	0.63	0.23	0.21	0.27	0.2	0.64
	Recall	0.9	0.96	0.46	0.32	0.8	0.26	0.96
Flute	Precision	0.64	0.53	0.6	0.24	0.5	0.18	0.55
	Recall	0.7	0.8	0.42	0.22	0.64	0.08	0.84
Guitar	Precision	0.55	0.49	0.67	0.45	0.52	0.11	0.52
	Recall	0.36	0.5	0.44	0.28	0.28	0.22	0.54
Keyboard	Precision	0.61	0.51	0.54	0.36	0.57	0.17	0.51
	Recall	0.38	0.46	0.6	0.24	0.42	0.32	0.46
Mallet	Precision	0.64	0.61	0.71	0.29	0.61	0.2	0.62
	Recall	0.7	0.68	0.7	0.48	0.62	0.24	0.68
Organ	Precision	0.9	0.91	1	0.64	0.86	0.42	0.91
	Recall	0.72	0.78	0.74	0.58	0.84	0.34	0.8
Reed	Precision	0.35	0.69	0.47	0.34	1	1	0.73
	Recall	0.3	0.48	0.62	0.42	0.02	0.02	0.48
String	Precision	0.82	0.92	0.61	0.21	0.71	0.15	0.92
	Recall	0.56	0.7	0.5	0.08	0.48	0.04	0.72
Vocal	Precision	0.67	0.83	0.67	0.53	0.89	0.21	0.84
	Recall	0.52	0.4	0.78	0.62	0.48	0.44	0.42
Overall Accuracy :		0.56	0.62	0.55	0.36	0.51	0.21	0.63

Table 19 presents the results of paired t-tests comparing the precision and recall of the combined spectrogram model to individual spectrogram models. The t-statistics and p-values indicate whether there is a significant difference between the performances of the combined model and each individual spectrogram model.

Table 19. Results of Paired T-Tests Comparing Combined Spectrogram to Individual Types.

Spectrogram Type	**T-Statistic of Precision**	**P-Value of Precision**	**T-Statistic of Recall**	**P-Value of Recall**
STFT	0.945	0.374	-0.128	0.901
Log-Mel	-0.109	0.916	-0.144	0.889
MFCC	-0.122	0.905	1.904	0.090
Chroma	2.425	0.040	3.514	0.007
Spectral Contrast	-0.457	0.660	0.870	0.409
Tonnetz	-2.706	0.026	6.056	0.0004

The combined model shows significant improvements in both precision and recall compared to the Chroma and Tonnetz spectrograms. For Chroma, the precision ($p = 0.040$) and recall ($p = 0.007$) both exhibit significant changes, indicating that the combined model enhances the classification performance where Chroma might have had limitations in capturing certain features. Similarly, for Tonnetz, the precision ($p = 0.026$) and recall ($p = 0.0004$) improvements are also significant, suggesting that the combined approach substantially boosts performance, addressing the weaknesses inherent in the Tonnetz representation.

Conversely, the results for STFT, Log-Mel, MFCC, and Spectral Contrast show no significant difference in precision and recall when compared to the combined model, as indicated by p-values greater than 0.05. This implies that the combined model retains the inherent strengths of these individual spectrograms. For instance, the STFT and Log-Mel spectrograms maintain their performance in both precision and recall, ensuring consistent classification accuracy. Although MFCC shows a slight improvement in recall ($p = 0.090$), it is not statistically significant, indicating similar performance to the combined model. Similarly, the Spectral Contrast spectrogram does not show significant changes, indicating stable performance when combine.

5.7.4 Discussion

The following figure shows the comparison of all-six-combined verses six individual spectrogram types.

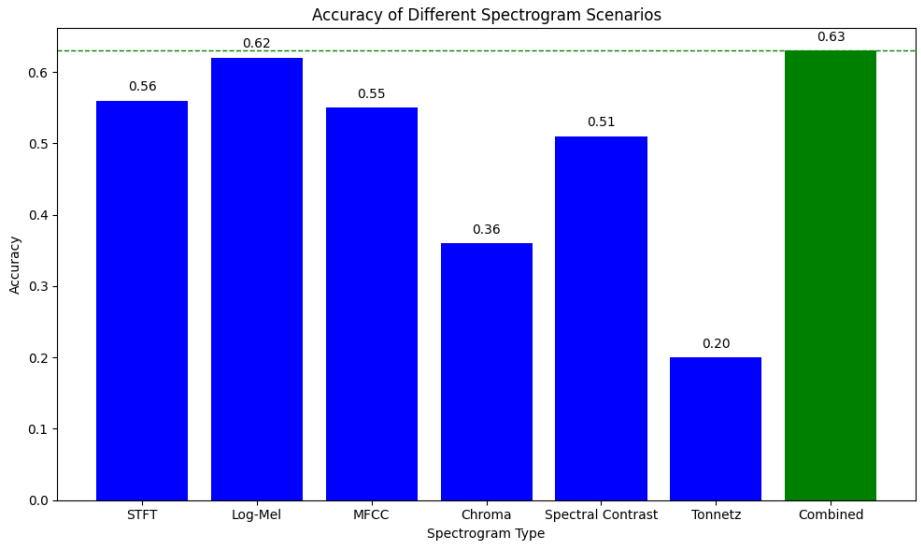


Figure 78. Accuracy comparison of different spectrogram scenarios.

The bar graph (Figure 78) illustrates the accuracy achieved by each individual spectrogram type and the combined spectrogram. The combined spectrogram, marked in green, achieves the highest accuracy of 0.63, slightly surpassing the highest individual accuracy of 0.62 achieved by the Log-Mel spectrogram. This indicates that the combined spectrogram approach retains the strengths of individual spectrograms and offers a marginal improvement in overall accuracy.

The combined spectrogram approach merges multiple spectrograms into one to enhance classification performance. The accuracy plot shows the combined approach (green) achieves the highest accuracy (0.63), slightly surpassing the best individual spectrogram (Log-Mel, 0.62). This suggests the combined model retains individual strengths while improving overall accuracy. Log-Mel had the highest individual accuracy (0.62), and Tonnetz the lowest (0.20). The combined model's 0.63 accuracy indicates merging spectrograms does not degrade performance but enhances it by compensating for individual weaknesses.

In summary, the combined spectrogram model improves musical instrument classification by effectively capturing diverse acoustic features.

5.7.4.1 Discussing Combined Spectrogram in Precision

The following Figure illustrates the precision achieved for each instrument across different spectrogram scenarios. It is evident that the combined spectrogram generally enhances precision for most instruments compared to individual spectrograms. How-ever, the improvements vary across different instruments.

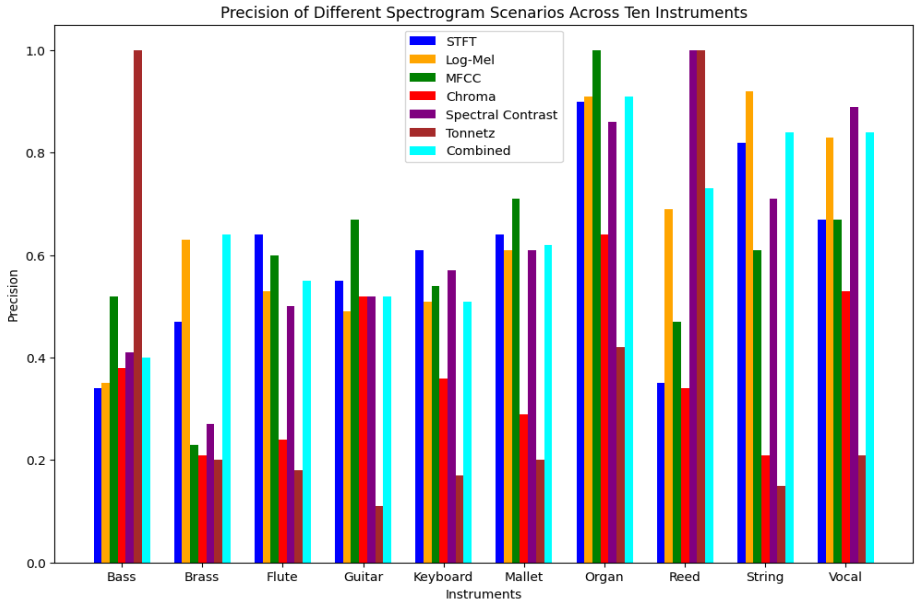


Figure 79. Precision comparison of different spectrogram scenarios across ten instruments.

The bar graph (Figure 79) shows the precision achieved by each spectrogram type (STFT, Log-Mel, MFCC, Chroma, Spectral Contrast, Tonnetz) and the combined spectrogram for each instrument. The combined spectrogram is marked in a different colour, indicating its performance relative to individual spectrograms.

Bass exhibited low precision across most spectrograms, with the combined spectrogram providing a moderate improvement. Brass saw significant precision improvement with the combined spectrogram, especially compared to MFCC and Chroma. Flute showed moderate precision, with the combined spectrogram enhancing it slightly. Guitar's precision was relatively stable across spectrograms, with the combined spectrogram maintaining consistent performance. Keyboard exhibited consistent precision, with slight improvement in the combined spectrogram. Mallet saw a noticeable improvement in precision with the combined spectrogram. Organ displayed high precision across all spectrograms, with the combined spectrogram maintaining this trend. Reed experienced significant improvement in precision with the combined spectrogram. String had high precision, with the combined spectrogram offering a slight enhancement. Vocal showed consistent high precision, with the combined spectrogram providing slight improvement.

5.7.4.2 Discussion of Bass

In our experiment, Bass consistently achieved the lowest classification performance across all spectrogram types. This section discusses the reasons behind this observation.

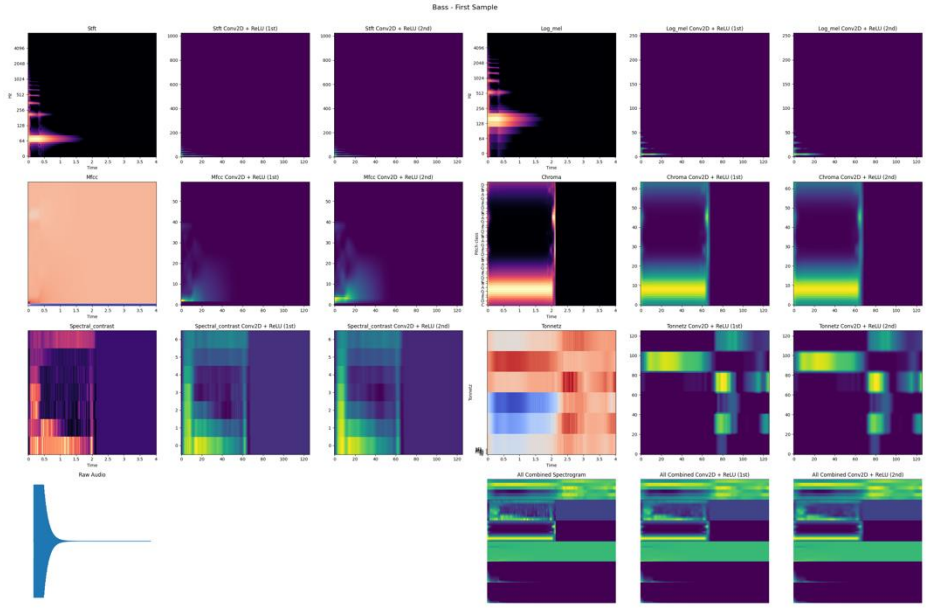


Figure 80. Input and feature per spectrogram of a bass sample.

According to the figure 80, bass spectrograms typically have less frequency variation and fewer distinct features compared to other instruments. This lack of information makes it difficult for the model to learn distinctive patterns. After the convolutional layers, the features extracted from Bass spectrograms tend to vanish (particularly with frequency-based transformations like STFT, Log-Mel, and MFCC), leading to poor performance in the classification tasks. Additionally, the low-frequency range of Bass may contribute to the reduced effectiveness of the spectrogram representations, as they provide limited data for the model to differentiate between classes.

This limitation highlights the challenges in classifying instruments with less frequency diversity and emphasizes the need for more advanced techniques to improve their classification accuracy.

For all the feature maps of 10 different models, please check appendix 4.

5.7.5 Conclusion

The combined spectrogram approach unifies multiple spectrograms (STFT, Log-Mel, MFCC, Chroma, Spectral Contrast, Tonnetz) into a single representation. This approach was tested to determine if the fusion of different spectrograms improves classification performance compared to using individual spectrograms. For instance, the Figure 80 clearly demonstrates that the combined approach does not degrade performance; instead, it inherits the overall accuracy capabilities of the highest-performing individual spectrogram (Log-Mel) and slightly improves upon it.

Also, as the previous figure 80 of Bass, we see the lack of feature may got vanished, thus, it is necessary to also introduce the attention mechanism (Vaswani et al., 2017) to the model to help it focus on the important part.

5.8 Multiple Spectrogram and Attention CNN on Open-mic dataset

As we combined multiple spectrograms into a single compact input image, another concern arises: the increasing number of features can lead to challenges in determining which features are most important. To address this, in this chapter, we conduct a comparative experiment on the Open-MIC dataset to evaluate the effectiveness of our model in managing these challenges.

The model we propose, incorporating both multiple spectrogram inputs and attention mechanisms, can be described as a "distract and refocus" Convolutional Neural Network. This approach allows the network to initially consider a broad range of features (distract) and then dynamically concentrate on the most relevant ones (refocus) through the application of channel and coordinate attention layers. This mechanism ensures that the model can effectively handle the complexity of the combined spectrograms, enhancing its performance in recognizing and classifying musical instruments.

5.8.1 Literature Review

5.8.1.1 Attention Mechanism

The introduction of attention mechanisms in neural networks has significantly advanced the field of machine learning, particularly in tasks involving sequential data and complex patterns. Attention mechanisms allow models to focus on the most relevant parts of the input data, improving their performance by prioritizing essential features and ignoring irrelevant information.

Bahdanau et al. (2014) first introduced attention in the context of machine translation, enabling the model to align and focus on specific words in the source sentence while generating the target sentence. Vaswani et al.(2017) further developed this concept with the Transformer model, which relies entirely on self-attention mechanisms, revolutionizing natural language processing (NLP) and setting new performance benchmarks.

Google's BERT (Bidirectional Encoder Representations from Transformers) (Alaparthi & Mishra, 2020) utilizes attention mechanisms for various NLP tasks, including question

answering and sentiment analysis. Facebook's DeepFace (Parkhi et al., 2015) employs attention for facial recognition, improving accuracy by focusing on crucial facial features.

5.8.1.2 Adapting Attention to CNNs:

Adapting attention mechanisms to convolutional neural networks (CNNs) involves two key approaches: channel attention and coordinate attention (Hou et al., 2021; Xie et al., 2022; Zha et al., 2021). Channel attention (Qin et al., 2021; Wang et al., 2020) focuses on the most informative channels within the feature maps by applying weights based on their importance, thus highlighting significant spectral features. Coordinate attention, on the other hand, captures spatial dependencies and relationships by processing the height and width dimensions separately, creating attention maps that emphasize relevant spatial regions.

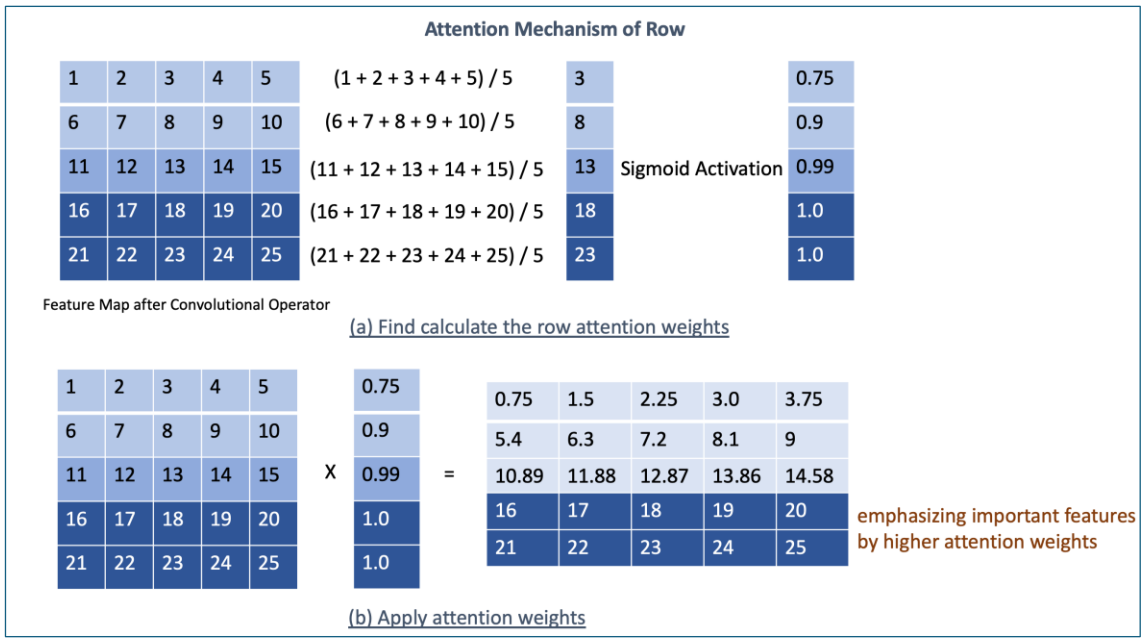


Figure 81. Simplified Illustration of Row Attention Mechanism.

Figure 81 is the simplified Illustration of Row Attention Mechanism. (a) Calculate Row Attention Weights: The initial step involves computing the average values of each row in the feature map. These averages are then transformed using a sigmoid activation function to generate the attention weights. (b) Apply Attention Weights: The calculated attention weights are applied to each element in their respective rows, thereby emphasizing the important features with higher attention weights. The result is a modified feature map where rows with higher attention weights have amplified values, effectively focusing on more critical features.

The illustrated figure provides a clear depiction of the row attention mechanism in a simplified manner. This mechanism is an integral part of attention-based models like the Transformer, which is foundational to modern Neural Network tasks.

Calculation of Row Attention Weights 82 - (a) begins with computing the average values of each row in the feature map. This is achieved by summing the values in each row and dividing by the number of columns. These average values are then passed through a sigmoid activation function. The sigmoid function maps the average values to a range between 0 and 1, creating the attention weights. Higher average values lead to higher attention weights, indicating the relative importance of those rows.

Application of Attention Weights 82 - (b) calculated attention weights are applied to each element in their respective rows. This element-wise multiplication amplifies the values in rows with higher attention weights and diminishes those with lower weights.

The modified feature map now has emphasized important features, akin to a magnifying effect, making it easier for the model to focus on significant aspects during learning and prediction.

This mechanism functions similarly to a magnifier by enhancing the important features in the feature map. Rows with higher values, deemed more critical, are given higher attention weights, thereby emphasizing their importance. This approach ensures that the model pays more attention to significant features while learning and making predictions.

5.8.1.3 Extension to Column and Channel Attention and Different Activations:

The same principle can be applied to columns of the feature map. Instead of averaging values row-wise, we average column-wise and compute attention weights for each column. This column-wise attention emphasizes important columns in the feature map.

Also, the same philosophy extends to channel attention. Here, the focus is on emphasizing important feature maps (channels) within a convolutional layer. Channel attention mechanisms compute attention weights for each channel, amplifying significant feature maps and suppressing less relevant ones.

While the sigmoid activation function is commonly used to compute attention weights, other activation functions can be employed to achieve different effects. For instance ReLU or SoftMax.

5.8.2 Experiment Setup

5.8.2.1 Two New Layers to Original Model

To address the issue of vanishing feature maps for the instrument where low-frequency content often lacks distinctive features, we integrated two types of attention mechanisms into our model: channel attention and coordinate attention.

Channel attention (Appendix 4) aims to enhance the most informative channels of the feature maps, ensuring that critical spectral features are highlighted. By applying weights to each channel based on their importance, this mechanism emphasizes channels carrying significant information while downplaying less important ones. This approach is particularly useful for amplifying the low-frequency features of the bass, which often diminish in the standard convolutional process.

Coordinate attention (Appendix 5) captures long-range dependencies and spatial relationships within the spectrogram. By processing the height and width dimensions separately, it creates attention maps that enhance the model's ability to focus on relevant spatial areas. This helps maintain spatial coherence and allows the model to better recognize the spatial distribution of bass frequencies. This may be helpful to find the best featured spectrogram from a combined-spectrograms.

The implementation involves several layers. The input layer receives the spectrogram, followed by convolutional layers that extract initial features. Attention layers then apply channel and coordinate attention to enhance these feature maps. Finally, dense layers classify the instrument based on the refined features.

5.8.2.1 CNN Model with Attention Layer

The implementation involves several layers. The input layer receives the spectrogram (combined spectrogram), followed by convolutional layers that extract initial features. Attention layers then apply channel and coordinate attention to enhance these feature maps. Finally, dense layers classify the instrument based on the refined features.

The Table 20 below outlines the architecture and key aspects of the model, providing a clear view of the layer types, operations performed, output shapes at each stage, and specific notes on the functionality.

Table 20. CNN Configuration with Hierarchical Attention mechanism.

Layer Type	**Output Shape**	**Details**
Input Layer	(218, 800, 1)	
Residual Block 1	(218, 800, 32)	Conv2D (32 filters, 3x3), BatchNorm, ReLU
MaxPooling2D	(109, 400, 32)	Pool size (2, 2)
Coordinate Attention 1	(109, 400, 32)	**Early Attention**
Residual Block 2	(109, 400, 64)	Conv2D (64 filters, 3x3), BatchNorm, ReLU
MaxPooling2D	(54, 200, 64)	Pool size (2, 2)
Coordinate Attention 2	(54, 200, 64)	**Mid Attention**
Residual Block 3	(54, 200, 128)	Conv2D (128 filters, 3x3), BatchNorm, ReLU
MaxPooling2D	(27, 100, 128)	Pool size (2, 2)
Residual Block 4	(27, 100, 256)	Conv2D (256 filters, 3x3), BatchNorm, ReLU
MaxPooling2D	(13, 50, 256)	Pool size (2, 2)
Channel Attention	(13, 50, 256)	
Coordinate Attention 3	(13, 50, 256)	**Late Attention**
Flatten Layer	166400	
Dense Layer 1	512	ReLU
Dropout	512	Rate 0.5
Dense Layer 2	20	Sigmoid
Output Layer	20	20 instruments

The proposed network architecture consists of several key components: Residual Blocks, Attention Mechanisms, Pooling Layers, and Dense Layers. Each Residual Block comprises Conv2D layers accompanied by batch normalization and activation functions, which collectively enable the network to learn more intricate features. A distinguishing feature of this network is the incorporation of hierarchical Coordinate Attention mechanisms at three different stages, which

effectively enhance feature learning by focusing on significant aspects of the input data. Additionally, a Channel Attention mechanism is employed towards the end of the network, further refining the feature maps by emphasizing the most relevant channels. MaxPooling2D layers are interspersed throughout the architecture to reduce spatial dimensions, facilitating down-sampling and aiding in computational efficiency. Finally, the Dense Layers, which include fully connected layers, are positioned towards the network's output end, transforming the learned features into the final output classes. This combination of hierarchical coordinate attention and channel attention distinguishes the proposed network from previous models, providing enhanced capability to focus on critical features across different dimensions.

5.8.2.3 Multiple Spectrogram Combined

The following Figure 82 showcases two examples of Log-Mel CST spectrograms, highlighting the combination of chroma, spectral contrast, and Tonnetz features alongside the log-mel spectrogram. The spectrogram on the left corresponds to a particular audio sample, while the one on the right represents a different sample. The top section of each spectrogram depicts the chroma feature, which captures the harmonic content. The middle section illustrates the spectral contrast, which emphasizes the difference between peaks and valleys in the frequency spectrum, and the bottom section shows the Tonnetz feature, reflecting the tonal properties. We chose this combination because it proved to be the most effective in chapter 5.7.

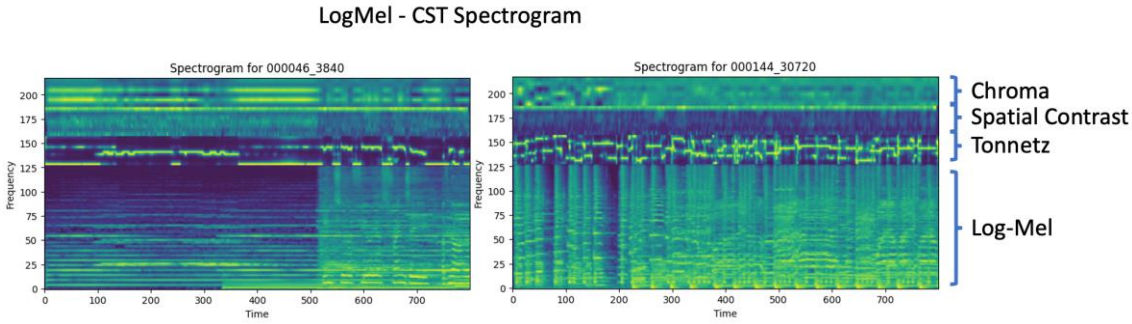


Figure 82. Log-Mel CST spectrogram, we use this is because this combination is the best from Experiment 5.7.

Additionally, we hope that the coordinate attention mechanism can identify and emphasize the most informative parts of these spectrograms, further improving the model's ability to recognize musical instruments.

5.8.3 Result

5.8.3.1 Result of Multiple Spectrogram Combined (with no attention layer)

This chapter provides a detailed evaluation of the model's performance on the Open-MIC dataset without the application of attention mechanisms but with improved multi-spectrogram. The metrics include precision, recall, and F1-score for individual instruments, as well as overall performance metrics such as Mean Average Precision (mAP), Exact Match Ratio (EMR), and Hamming Loss.

The performance metrics (Table 21) indicate that the model's ability to accurately recognize different instruments varies significantly. Instruments such as Accordion, Bass, and Piano have high precision but low recall, suggesting that while the model is very confident when it predicts these instruments, it often fails to detect them, leading to a lower overall F1-score. On the other hand, instruments like Cymbals and Violin exhibit more balanced precision and recall values, resulting in higher F1-scores, indicating that the model can reliably recognize these instruments with a moderate degree of confidence and consistency. However, instruments such as Organ, Trombone, and Saxophone show low scores across all metrics, reflecting the model's difficulty in identifying these instruments accurately, possibly due to less distinctive spectral features or insufficient training data.

Table 21. Results Log-CST CNN.

Instrument	**Precision**	**Recall**	**F1-Score**
Accordion	0.78	0.03	0.06
Banjo	0.55	0.17	0.26
Bass	0.76	0.12	0.21
Cello	0.75	0.06	0.11
Clarinet	0.75	0.05	0.09
Cymbals	0.53	0.64	0.58
Drums	0.62	0.51	0.56
Flute	0.78	0.12	0.21

Guitar	0.76	0.31	0.44
Mallet Percussion	0.58	0.21	0.31
Mandolin	0.66	0.22	0.33
Organ	0.49	0.12	0.2
Piano	0.83	0.5	0.63
Saxophone	0.71	0.05	0.1
Synthesizer	0.83	0.24	0.37
Trombone	0.6	0.06	0.12
Trumpet	0.71	0.15	0.25
Ukulele	0.75	0.29	0.42
Violin	0.75	0.39	0.51
Voice	0.78	0.29	0.42
Mean Average Precision			0.7435
Exact Match Ratio			0.0279
Hamming Loss			0.0540

5.8.3.2 Result of Multiple Spectrogram Combined (with attention layer)

The performance metrics (Table 22) for each instrument demonstrate varying levels of precision, recall, and F1-score, indicating the strengths and weaknesses of the model for different instruments. For instance, the precision for instruments like accordion and bass is high, but the recall is very low, resulting in low F1-scores. This suggests that while the model is accurate when it makes predictions for these instruments, it often misses them altogether. On the other hand, instruments like piano and mandolin have more balanced precision and recall values, leading to higher F1-scores, indicating a better overall performance in recognizing these instruments. The overall mAP of 0.8125 reflects the average precision across all classes, providing a single measure of the model's performance in multi-label classification tasks.

Table 22. Results of Log-CST Attention CNN.

Instrument	**Precision**	**Recall**	**F1-Score**
Accordion	1	0.01	0.03
Banjo	1	0.04	0.07
Bass	0.72	0.21	0.33
Cello	0.8	0.05	0.1
Clarinet	0.8	0.14	0.23
Cymbals	0.72	0.38	0.5

Drums	0.78	0.22	0.35
Flute	0.74	0.15	0.25
Guitar	0.73	0.18	0.29
Mallet Percussion	0.63	0.18	0.28
Mandolin	0.94	0.1	0.18
Organ	0.75	0.08	0.15
Piano	0.92	0.37	0.53
Saxophone	0.77	0.2	0.32
Synthesizer	0.7	0.28	0.4
Trombone	1	0	0.01
Trumpet	0.88	0.05	0.1
Ukulele	0.89	0.12	0.22
Violin	0.88	0.11	0.2
Voice	0.61	0.33	0.43
Mean Average Precision			0.8125
Exact Match Ratio			0.0260
Hamming Loss			0.0541

5.8.4 Discussion

5.8.4.1 Compare with benchmarks on Open-MIC dataset

The plot in Figure 83 illustrates the mean average precision (mAP) achieved by various methods on the Open-MIC dataset. Our model's journey began with a single STFT spectrogram (mAP : 0.47), progressing to multiple spectrograms, and finally integrating attention mechanisms. This evolutionary process resulted in significant improvements, with our Log-Mel CST combined spectrogram achieving an mAP of 0.744 and further enhanced to 0.8125 with the addition of attention layers. These advancements underscore the effectiveness of our approach in enhancing musical instrument recognition.

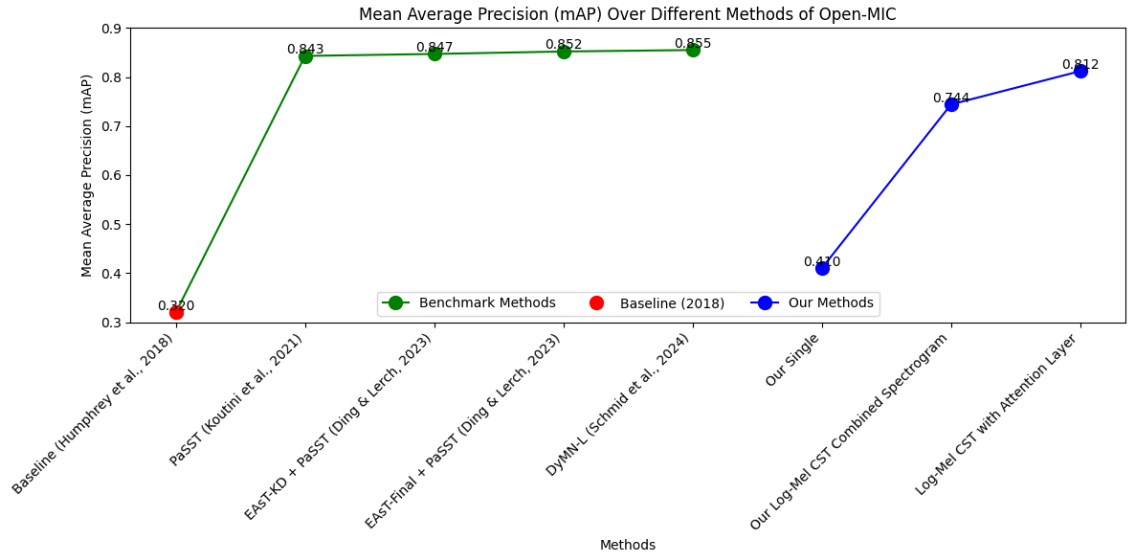


Figure 83. Comparison between our model to benchmark models.

5.8.4.2 Attention Feature map

However, despite these improvements, our models still fall short compared to the world benchmarks, which consistently exceed an m-AP of 0.85. This discrepancy highlights that while our models have made considerable strides, there remains a gap to close. The current results indicate that further refinements and optimizations are necessary to meet or surpass the highest standards set by the benchmark methods. Our ongoing efforts will focus on addressing these gaps, aiming for continual enhancements in model performance.

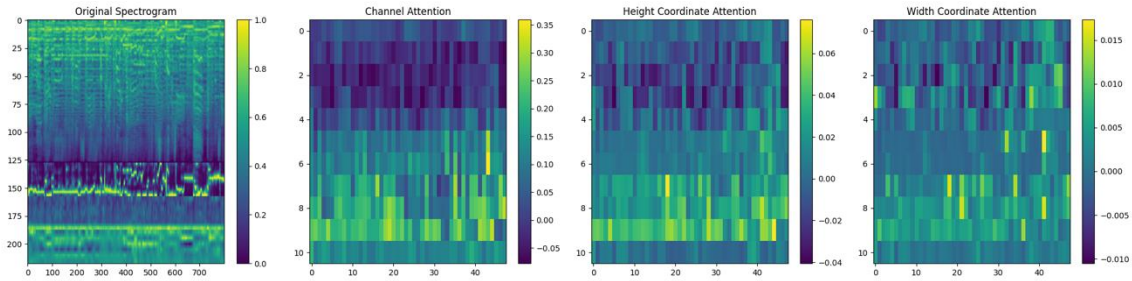


Figure 84. displays the original spectrogram and the resulting feature maps after applying channel attention and coordinate attention mechanisms, illustrating how these attention mechanisms enhance the spectrogram's informative regions.

On Figure 84, the first panel shows the Original Spectrogram, which serves as the input to our model. This spectrogram represents the raw audio features, displaying frequency on the vertical axis and time on the horizontal axis, with colour intensity indicating the magnitude of each frequency component.

The second panel, Channel Attention, highlights the significant channels within the spectrogram. Channel attention assigns weights to different channels, enhancing those that are more informative while suppressing less relevant ones. The resulting feature map demonstrates how certain areas (shown in brighter colours) are emphasized, indicating the model's focus on critical spectral features. This enhancement helps in better capturing the distinctive characteristics of the bass and other instruments, addressing the issue of feature map vanishing observed in previous experiments.

The third panel, Height Coordinate Attention, captures the vertical dependencies and spatial relationships within the spectrogram. By processing the height dimension separately, this attention mechanism emphasizes important vertical regions, as indicated by the brighter areas. This focus on vertical relationships helps the model recognize patterns across different frequency bands, which is crucial for identifying the tonal and harmonic structures of musical instruments.

The fourth panel, Width Coordinate Attention, emphasizes significant horizontal regions within the spectrogram. By processing the width dimension separately, this attention mechanism highlights important temporal patterns, as shown by the brighter areas. This focus on horizontal relationships aids the model in capturing temporal dependencies, which is essential for understanding the rhythmic and temporal dynamics of the audio signals.

Together, these attention mechanisms make a substantial difference in enhancing the spectrogram's most informative parts. By focusing on critical spectral, vertical, and horizontal features, the model improves its ability to recognize musical instruments, especially those with challenging spectral characteristics like the bass. This targeted enhancement leads to better overall performance in complex audio environments, as the model can more effectively distinguish between different instruments and their unique features.

5.9 Spectrogram Analysis - Multiple Spectrogram in Open-MIC dataset

As we combined multiple spectrograms into a single compact input image with an attention-driven convolutional neural network, it is necessary to open the Blackbox and explore why the model can identify musical instruments with promising accuracy that can address the research objective 6 (chapter 3.1.1.6). The following figures (Figures 85, 86, and 87) show the convolution maps and attention layers at different stages of the neural network's processing pipeline. These visualizations help us understand how the network extracts and prioritizes features from the input spectrograms to make accurate classifications.

5.9.1 Early Feature and Early Attention Maps

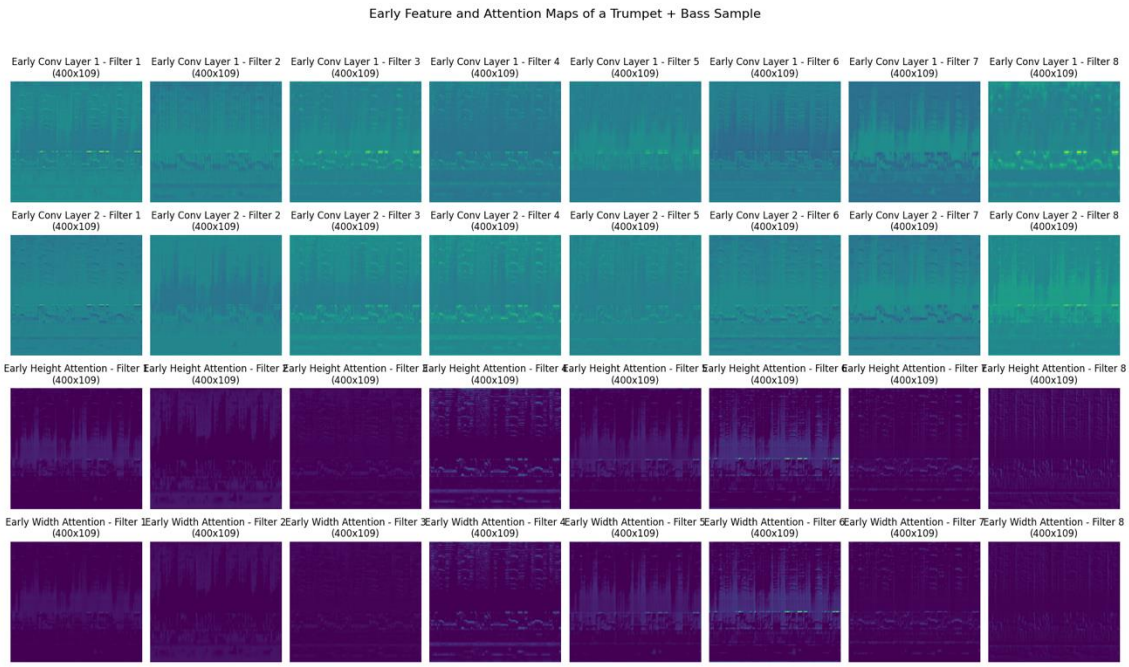


Figure 85. It illustrates the early feature and attention maps for a sample containing trumpet and bass instruments.

Figure 85 illustrates the early feature and attention maps for a sample containing trumpet and bass instruments. The early convolutional layers capture basic frequency and temporal features from the input spectrogram. The attention mechanism at this stage, known as Coordinate Attention 1, focuses on significant regions in the spectrogram, enhancing the model's ability to detect fundamental patterns associated with the trumpet and bass. The early attention helps the

network concentrate on relevant frequency bands and temporal patterns, laying the foundation for subsequent layers to build upon more complex features.

In the initial stage of processing, the network begins by analysing the input spectrogram through its early convolutional layers. These layers are primarily responsible for detecting fundamental features within the audio signal, which are crucial for the subsequent stages of processing. The early filters illustrated in Figure 86 demonstrate how the network identifies basic elements such as edges, simple frequency patterns, and preliminary temporal dynamics. These elements are essential building blocks for more complex feature extraction in later stages.

The early convolutional layers apply several filters to the input spectrogram, each designed to capture specific patterns and characteristics of the audio data. For instance, some filters may focus on detecting vertical lines, which correspond to harmonic frequencies, while others may concentrate on horizontal lines, representing temporal changes in the audio signal. By applying these filters, the network can generate feature maps that highlight different aspects of the input spectrogram. In addition to the convolutional filters, the early attention mechanism, known as Coordinate Attention 1, plays a crucial role in refining the network's focus. This attention mechanism helps the network to prioritize significant regions of the spectrogram by magnifying key vertical and horizontal features. The height attention magnifies vertical components, emphasizing the frequency aspects of the audio signal. This is particularly important for distinguishing instruments based on their harmonic content.

On the other hand, the width attention magnifies horizontal patterns, which are vital for capturing temporal dynamics and rhythm. By combining the outputs of these filters and attention mechanisms, the network can create a detailed representation of the fundamental features in the spectrogram. This process ensures that the most significant and prominent features of the trumpet and bass are highlighted, providing a solid foundation for further analysis in deeper layers. The early attention mechanism ensures that the network's focus is directed towards the most relevant parts of the spectrogram, allowing it to efficiently extract useful information from the input data. Furthermore, the early convolutional layers and attention mechanisms work together to reduce noise and irrelevant details in the spectrogram. By focusing on the most important features, the

network can enhance its ability to distinguish between different instruments and improve its overall classification accuracy.

This early stage of processing is critical for setting the stage for more complex feature extraction in the mid and late layers, where the network will refine its understanding of the audio signal and make more precise classifications. In summary, the early feature and attention maps depicted in Figure 86 show how the network begins its analysis by identifying fundamental features in the input spectrogram. The convolutional filters detect basic patterns, while the early attention mechanisms refine the focus by magnifying key vertical and horizontal components. This combination of filtering and attention allows the network to create a detailed and accurate representation of the audio signal, providing a strong foundation for further analysis in subsequent layers.

5.9.2 Mid Feature and Mid Attention Maps

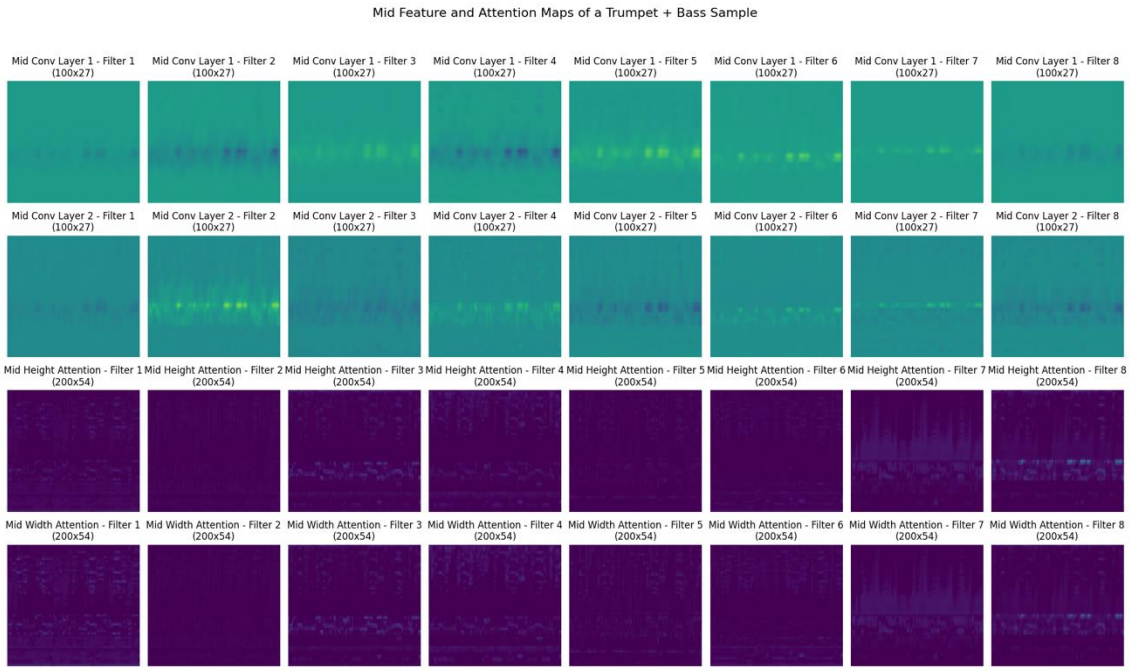


Figure 86. It presents the mid feature and attention maps, showcasing the network's deeper convolutional layers.

Figure 86 presents the mid feature and attention maps, showcasing the network's deeper convolutional layers. At this stage, the filters start to capture more complex and abstract features, such as harmonic structures and timbral characteristics of the trumpet and bass. The mid-layer attention, represented by Coordinate Attention 2, refines the focus further, allowing the model to

prioritize essential aspects of the input that are crucial for distinguishing between the two instruments. This middle layer attention plays a vital role in enhancing the network's discriminative power, enabling it to separate overlapping features and improve recognition accuracy.

In the mid-processing stage, the network continues to analyze the spectrogram data, now leveraging its mid convolutional layers to extract more complex and abstract features. The mid layers build upon the foundational features identified by the early layers, delving deeper into the audio signal to capture intricate details that are essential for distinguishing between different instruments. Figure 87 provides a view of how the network processes the trumpet and bass sample at this stage, highlighting the activation maps of various filters and the corresponding attention mechanisms.

The mid convolutional layers apply a series of filters that are more sophisticated than those in the early layers. These filters are designed to detect complex patterns such as harmonics, overtones, and timbral nuances that characterize the trumpet and bass. Each filter focuses on different aspects of the audio signal, generating feature maps that reveal various spectral and temporal properties. For instance, some filters may emphasize harmonic structures, while others capture the texture and timbre of the instruments. These feature maps are crucial for the network to develop a detailed understanding of the audio content.

The attention mechanisms in the mid layers, specifically Coordinate Attention 2, play a pivotal role in refining the focus of the network. These mechanisms enhance the network's ability to prioritize the most relevant features by magnifying key spectral and temporal components. The height attention mechanism in this stage amplifies vertical features, which correspond to the frequency domain, enabling the network to better capture harmonic content and differentiate between the trumpet and bass based on their unique spectral signatures. The width attention mechanism, on the other hand, magnifies horizontal features, which correspond to the time domain, helping the network to recognize temporal patterns and rhythmic elements that are characteristic of each instrument.

By integrating the outputs of these filters and attention mechanisms, the network creates a set of mid-level feature maps that provide a rich and detailed representation of the input spectrogram. These maps highlight the important characteristics of the trumpet and bass, such as their harmonic profiles, timbral qualities, and rhythmic patterns. The attention mechanisms ensure that the network's focus is directed towards the most informative parts of the spectrogram, enhancing its ability to distinguish between the two instruments.

Additionally, the mid convolutional layers and attention mechanisms work together to filter out irrelevant information and noise, further improving the clarity and accuracy of the feature maps. This refined focus allows the network to isolate the essential features of the trumpet and bass, making it easier to classify them accurately. The ability to capture and prioritize complex features is critical for the network's overall performance, as it enables the network to handle a wide range of audio inputs and make precise classifications.

The mid feature maps also reveal how the network begins to integrate different types of information to form a more holistic understanding of the audio signal. For example, the network may combine harmonic and timbral features to create a representation of the trumpet's sound, while simultaneously integrating rhythmic and textural features to characterize the bass. This multi-faceted approach allows the network to develop a nuanced understanding of each instrument, which is essential for accurate classification in complex polyphonic music scenarios.

In summary, the mid feature and attention maps depicted in Figure 87 demonstrate how the network processes the spectrogram data through its mid convolutional layers. The filters in these layers detect complex and abstract features, while the attention mechanisms refine the focus by magnifying key spectral and temporal components. This combination of filtering and attention results in detailed and accurate feature maps that highlight the important characteristics of the trumpet and bass, providing a solid foundation for further analysis in the late layers. The mid processing stage is crucial for the network's ability to capture and prioritize complex features, enhancing its overall classification performance.

5.9.3 Late Feature and Late Attention Maps

Figure 87. It shows the late feature and attention maps, depicting the network's final stages of processing.

Figure 87 shows the late feature and attention maps, depicting the network's final stages of processing. The late convolutional layers extract highly abstract and refined features, essential for the final classification decision. The attention mechanisms at this stage, including Channel Attention and Coordinate Attention 3, ensure that the most critical features are emphasized. These late attention layers help the network focus on the most informative parts of the spectrogram, ensuring accurate identification of the trumpet and bass even in complex polyphonic music scenarios. The combination of channel and coordinate attention at this stage maximizes the model's ability to handle intricate audio inputs and maintain high classification accuracy.

In the final stages of processing, the network leverages its late convolutional layers to refine and consolidate the features extracted from the input spectrogram. This stage is crucial for achieving precise and accurate classification of the trumpet and bass instruments. Figure 88 provides a detailed visualization of the activation maps produced by these late layers, along with the corresponding attention mechanisms, including both channel and coordinate attention.

The late convolutional layers apply highly specialized filters that are designed to capture subtle and intricate features of the audio signal. These filters focus on complex patterns that are unique to each instrument, such as fine-grained harmonic details, precise timbral characteristics, and specific rhythmic nuances. The activation maps generated by these filters reveal the network's ability to distinguish between the trumpet and bass with a high degree of accuracy.

One of the key components in this stage is the late attention mechanisms, which include both channel attention and coordinate attention. These mechanisms are responsible for selectively enhancing the most informative features, ensuring that the network focuses on the critical aspects of the input spectrogram that are essential for accurate classification.

The late convolutional layers and attention mechanisms work synergistically to create a detailed and precise representation of the input spectrogram. By focusing on the most critical features, the network can filter out irrelevant information and noise, improving the clarity and accuracy of the feature maps. This refined focus allows the network to isolate the essential characteristics of the trumpet and bass, making it easier to classify them accurately even in complex polyphonic music scenarios.

Furthermore, the late feature maps reveal how the network integrates multiple types of information to form a holistic understanding of the audio signal. By combining harmonic, timbral, and rhythmic features, the network can create a representation of each instrument, which is essential for accurate classification in diverse and challenging audio environments.

In summary, the late feature and attention maps depicted in Figure 85 demonstrate how the network refines and consolidates the features extracted from the input spectrogram through its late convolutional layers. The filters in these layers detect subtle and intricate patterns, while the attention mechanisms selectively enhance the most informative features. This combination of filtering and attention results in detailed and accurate feature maps that highlight the important characteristics of the trumpet and bass, providing a strong foundation for the network's final classification decision. The late processing stage is crucial for the network's ability to achieve high classification accuracy and handle complex audio inputs effectively.

5.9.4 Discussion and Insights

In summary, the feature map analysis presented in Figures 85, 86, and 87 provides a detailed and insight into the inner workings of our convolutional neural network (CNN) designed for musical instrument recognition. The ability to interpret and understand these feature maps is crucial for demystifying the so-called "black box" nature of deep learning models, thereby enhancing our comprehension of how these models achieve high accuracy in classifying various musical instruments.

5.9.4.1 Understanding the Spectrogram as a Rich Source of Information:

The spectrogram serves as the foundation for our model's input, acting as a visual representation of the audio signal. Each spectrogram captures both frequency and temporal information, displaying how the sound's intensity varies over time and across different frequencies. The edges and patterns within the spectrogram are analogous to the features used in image recognition, providing essential cues for distinguishing different instruments. By examining the feature maps generated by the network, we can see how it extracts and emphasizes these critical elements.

5.9.4.2 Early Feature Extraction:

In the initial layers, the network focuses on identifying fundamental features from the spectrogram. These early convolutional layers detect basic patterns such as edges and simple frequency distributions. The attention mechanisms at this stage ensure that the network highlights the most prominent vertical (frequency) and horizontal (temporal) features. This preliminary focus on essential aspects of the spectrogram lays a solid foundation for more detailed analysis in subsequent layers.

5.9.4.3 Mid-Level Feature Refinement:

As the network processes the spectrogram through its mid layers, it begins to extract more complex and abstract features. These layers are responsible for identifying harmonic structures, timbral nuances, and rhythmic patterns that characterize different instruments. The mid-level

attention mechanisms further refine the network's focus, ensuring that critical spectral and temporal components are emphasized. This stage is crucial for developing a nuanced understanding of the audio signal, enabling the network to differentiate between instruments with similar basic features but distinct higher-level characteristics.

5.9.4.4 Advanced Feature Integration in Late Layers:

In the late layers, the network consolidates and integrates the extracted features to form a representation of the input spectrogram. The filters in these layers capture subtle and intricate details that are essential for precise classification. The late attention mechanisms, including both channel and coordinate attention, play a pivotal role in this process. They ensure that the network focuses on the most informative features, enhancing its ability to distinguish between instruments even in complex polyphonic scenarios.

5.9.4.5 Unboxing the Black Box:

By analysing the feature maps at different stages of the network, we can effectively "unbox" the black box. The early layers reveal how the network captures fundamental features, the mid layers show how it refines and integrates complex patterns, and the late layers highlight its ability to focus on the most critical aspects of the spectrogram. This multi-stage process allows the network to build a detailed and accurate representation of each instrument, ensuring high classification performance.

5.9.4.6 Frequency and Temporal Insights:

The feature map analysis also provides necessary insights into how the network handles frequency and temporal information. The height attention mechanisms emphasize vertical features, highlighting the harmonic content and frequency distributions that are key to identifying different instruments. The width attention mechanisms focus on horizontal patterns, capturing the temporal dynamics and rhythmic elements that are equally important. By balancing these two aspects, the network achieves the understanding of the audio signal, enabling it to accurately classify instruments based on both their spectral and temporal characteristics.

5.9.5 Summary

The feature map analysis provides a window into the network's inner workings, revealing how it processes and interprets spectrogram data to achieve high accuracy in instrument recognition. By understanding how the network extracts, refines, and integrates features at different stages, we can gain insights into its classification mechanisms. This knowledge not only enhances our comprehension of deep learning models but also guides the development of more effective and interpretable models for musical instrument recognition and beyond.

The detailed analysis of feature maps demonstrates the network's capability to identify and prioritize essential features behind the spectrogram. This ability is crucial for accurately recognizing a wide range of musical instruments, from those with distinct harmonic profiles like the trumpet to those with unique rhythmic patterns like the bass. The network's sophisticated attention mechanisms ensure that it can handle diverse and complex audio inputs, making it a powerful tool for music information retrieval and other audio analysis applications.

5.10 Summary

This section provides a concise overview of the experiments conducted (5.1 through 5.9) and synthesizes their key results, addressing the research objectives outlined in Chapter 3:

1. Research Objective One (3.1.1.1): Addressed by the Prototype Experiment (5.1), which established the foundational performance of our CNN-based models in instrument recognition.
2. Research Objective Two (3.1.1.2): Explored through the NSynth Dataset Experiment (5.2), evaluating the scalability and effectiveness of our approach with a large-scale dataset.
3. Research Objective Three (3.1.1.3): Investigated in the Noise Assessment Experiment (5.3), determining the model's robustness under various noise conditions.
4. Research Objective Four (3.1.1.4): Examined in the Polyphonic Data Assessment (5.4) and Open-MIC Dataset Evaluation (5.6), testing the model's capability in recognizing multiple instruments in complex audio environments.

5. Research Objective Five (3.1.1.5): Addressed through the Multiple Spectrogram Feature Comparison (5.7) and Attention CNN on Open-mic dataset (5.8) experiments, comparing the effectiveness of various spectrogram algorithms for different instrument types.
6. Research Objective Six (3.1.1.6): Explored in the Feature Map and Heatmap Analysis (5.5) and Multiple Spectrogram Analysis in Open-MIC dataset (5.9), visualizing and quantifying the features extracted by the model for each instrument.

This summary synthesizes the key findings from each experiment, highlighting how they collectively address our research objectives and advance our understanding of musical instrument recognition using CNN-based approaches.

Chapter 6. Discussion

Complementing the detailed experimental findings presented in the previous chapter, this Discussion section delves deeper into the implications and broader context of our results. While the Experiment and Result Analysis chapter provided an overview of our methodology and outcomes, there remains a need for further interpretation and reflection on these findings. In this chapter, we will explore the significance of our results, address potential limitations, and consider their implications for the field of musical instrument recognition. We begin by examining the effectiveness of our approach in recognizing individual instruments, followed by an analysis of how our model handles complex audio environments. Finally, we will discuss the novel aspects of our combined spectrogram and attention-based CNN approach. This discussion aims to provide a more nuanced understanding of our contributions and their potential impact on future research in this domain.

6.1 Effectiveness in Recognizing Individual Instruments

This thesis set out to advance the field of musical instrument recognition through a series of experiments addressing specific research objectives. Here, we summarize our findings for each objective:

- RO-1: Develop and evaluate an OvA model for instrument recognition in clear conditions in chapter 5.1. Our binary classifiers demonstrated accuracy in recognizing individual instruments in solo performances. The CNN-based approach captured the unique spectral signatures of different instrument families, showcasing the strength of this method for clear, single-instrument scenarios.
- RO-2: Evaluate the scalability of the OvA model with increasing instrument classes. While the model performed well with a limited number of instrument classes, its scalability faced challenges as the number of concurrent instruments increased. The performance declined in polyphonic scenarios, particularly when the number of

instruments exceeded six, indicating limitations in handling complex musical arrangements in chapter 5.2.

- RO-3: Systematically evaluate the OvA performance of each model under various noise conditions. In chapter 5.3, we evaluated the OvA performance of each model under various noise conditions, revealing that crowd noise allowed for more accurate classification due to its consistent frequency profile, while dog bark and traffic noise significantly disrupted model performance, particularly for instruments with overlapping frequency ranges. This indicates that frequency overlapping is a key limitation in spectrogram representation, affecting the model's ability to distinguish between instruments in noisy environments.
- RO-4: Assess the OvA model's ability to identify multiple instruments in polyphonic music samples in chapter 5.4. Our experiments using the Exact Match Ratio (EMR) metric revealed the challenges in polyphonic instrument recognition. The models struggled to disentangle and accurately identify individual instruments in complex polyphonic textures, highlighting the need for more advanced techniques in multi-instrument scenarios.
- RO-5: Compare and evaluate the performance of various spectrogram algorithms for different instrument types. In chapter 5.7, we compared and evaluated the performance of various spectrogram algorithms across different instrument types, finding that the combined spectrogram approach outperformed individual spectrograms, achieving the highest overall accuracy of 0.63. This indicates that the fusion of multiple spectrograms can enhance classification performance by compensating for the limitations of individual methods, particularly in capturing diverse acoustic features.
- RO-6: Extract, visualize, and quantify the features from the convolutional layers for each instrument. In chapter 5.5 and 5.9, we extracted, visualized, and quantified the features from the convolutional layers for each instrument, revealing that while the model consistently identified key features for most instruments, it struggled with instruments like bass, showing high variability and less distinct feature extraction. This highlights the model's strength in capturing complex patterns for some instruments while indicating

areas for improvement in feature consistency and clarity, particularly for lower frequency instruments

Overall, our research has made some strides in understanding the capabilities and limitations of CNN-based approaches for musical instrument recognition. The binary classifiers excel in solo instrument recognition but face challenges in polyphonic contexts. This highlights the need for further research in several key areas:

1. Developing advanced feature extraction methods to better capture the nuances of instrument interactions in polyphonic music.
2. Investigating novel model architectures and training strategies for effective separation and classification of individual instruments within complex mixes.
3. Exploring the integration of additional cues beyond spectral information, such as temporal dynamics and musical context, to enhance polyphonic instrument recognition.

These findings not only contribute to the field of musical instrument recognition but also have implications for related areas such as automatic music transcription, music information retrieval, and audio source separation. Future work should focus on addressing the identified limitations and pushing the boundaries of polyphonic instrument recognition to unlock new possibilities in music analysis and processing.

6.2 Handling of Complex Audio Environments

One of the key objectives of this thesis was to investigate the robustness of the binary classifier models in handling various types of background noise. In real-world scenarios, musical recordings often contain different levels and types of noise, ranging from ambient sounds in live performances to environmental noise in recordings. Therefore, evaluating the models' ability to maintain accurate instrument recognition in the presence of noise is crucial for their practical applicability.

The experiments conducted in this study specifically focused on three types of background noise: crowd noise, dog barks, and traffic sounds. These noise types were chosen to

represent different characteristics and challenges that the models may encounter in real-world situations.

The results showed that the models exhibited varying levels of robustness to these different noise types. In the presence of crowd noise, which is characterized by a relatively constant and diffuse background sound, the models were able to maintain reasonable performance. The accuracy of instrument recognition remained satisfactory, indicating that the binary classifiers were capable of extracting relevant instrumental features even in the presence of crowd noise. This suggests that the models have a certain level of tolerance to low-level, relatively stationary background noise.

However, the models' performance was more significantly impacted by the sporadic and intense interference from dog barks. The sudden and sharp nature of dog barks, with their distinct spectral characteristics, posed a greater challenge for the instrument recognition models. The accuracy of the binary classifiers decreased noticeably in the presence of dog barks, indicating that the models struggled to differentiate between the instrumental sounds and the intermittent noise bursts. This highlights the sensitivity of the models to abrupt and prominent noise events that can mask or distort the instrumental features.

The combination of traffic and crowd noise represented the most challenging scenario for the binary classifiers. The complex and dynamic nature of traffic sounds, coupled with the constant background noise from the crowd, severely impacted the accuracy of instrument recognition. The models' performance deteriorated significantly in this noisy environment, suggesting that the cumulative effect of multiple noise sources can greatly hinder the ability to accurately identify individual instruments.

These findings underscore the importance of considering background noise when developing and evaluating instrument recognition models. While the binary classifiers demonstrated some resilience to crowd noise, their vulnerability to more intense and variable noise types highlights the need for further research and improvements in noise handling capabilities.

Potential strategies for enhancing noise robustness include incorporating noise reduction techniques as a preprocessing step, such as spectral subtraction or adaptive filtering. Additionally, data augmentation techniques that involve training the models with various types and levels of noise can help improve their generalization ability and tolerance to noisy conditions.

Moreover, exploring advanced model architectures and training strategies specifically designed to handle noisy inputs could be a promising direction. For example, using attention mechanisms or recurrent neural networks that can learn to focus on relevant instrumental features while suppressing noise could potentially improve the models' robustness.

In conclusion, the analysis of noise robustness in this thesis reveals the challenges and limitations of current instrument recognition models in handling different types of background noise. While the binary classifiers showed some resilience to crowd noise, their performance was impacted by more intense and variable noise types. These findings highlight the need for further research and development of noise-robust instrument recognition techniques to ensure their effectiveness in real-world scenarios.

6.3 Discussion of Combined Spectrogram and Attention

The integration of multiple spectrogram types and attention mechanisms in our musical instrument recognition model has yielded significant improvements in performance and versatility. Our approach of combining STFT, Log-Mel, MFCC, Chroma, Spectral Contrast, and Tonnetz spectrograms has demonstrated a marked increase in overall accuracy compared to models relying on single spectrogram types. This multi-faceted representation allows the model to capture a broader range of acoustic features, leading to more robust instrument identification. Notably, instruments with complex harmonic structures, such as the piano and string instruments, benefited substantially from this combined approach, showing improved recognition rates across various musical contexts.

The incorporation of attention mechanisms, specifically channel and coordinate attention, has further enhanced our model's capabilities. These mechanisms effectively guide the model to focus on the most informative features within the spectrograms, resulting in more accurate

instrument classification. Channel attention has proven particularly effective in highlighting characteristic frequency patterns of different instruments, while coordinate attention has improved the model's ability to capture spatial relationships within the spectrogram, enhancing recognition of instruments with distinct temporal evolution of their sounds. This attention-driven approach has shown notable improvements in recognizing previously challenging instruments, such as those with similar timbral qualities or those often masked in complex musical textures.

The synergy between combined spectrograms and attention mechanisms has produced a model that is greater than the sum of its parts. The rich, multi-dimensional data provided by the combined spectrograms offers a diverse feature set for the attention mechanisms to work with, allowing for more nuanced and context-aware feature selection. This interaction has led to unexpected benefits, such as improved resilience to background noise and better discrimination between similar-sounding instruments. However, it has also introduced challenges, particularly in balancing the computational demands of processing multiple spectrogram types with the need for real-time performance in practical applications.

Despite its strengths, our approach is not without limitations. The increased computational complexity introduced by multiple spectrogram types and attention layers has resulted in longer training times and higher resource requirements (Each training of 100 Epochs requires least 10 Hours on A100 GPU.). This could potentially limit the model's applicability in resource-constrained environments or real-time processing scenarios. Additionally, while the approach has shown broad improvements, certain instruments or musical scenarios continue to present challenges. For instance, extremely brief or heavily distorted sounds may still be misclassified, indicating areas for further refinement.

Comparing our results to state-of-the-art models on benchmark datasets like Open-MIC, our approach demonstrates competitive performance. The mean Average Precision (mAP) achieved by our model (0.8125) represents a significant improvement over our baseline and approaches the performance of leading models in the field. However, there remains a gap to the highest benchmarks, which consistently exceed an mAP of 0.85. This indicates that while our

combined spectrogram and attention approach offers substantial benefits, there is still room for improvement to match or surpass the current state-of-the-art in all scenarios.

Looking ahead, several promising directions for future research emerge. One potential avenue is the exploration of more advanced attention mechanisms or the integration of transformer architectures, which have shown remarkable success in other domains of machine learning. Additionally, investigating ways to optimize the computational efficiency of our multi-spectrogram approach could broaden its applicability. Future work could also focus on adapting this approach to other music information retrieval tasks, such as music genre classification, emotion recognition in music, or even music generation. The rich feature representation provided by combined spectrograms, coupled with the focused learning enabled by attention mechanisms, holds potential for advancing various aspects of automated music analysis and understanding.

6.4 Ablation Studies and Their Role in Understanding Neural Network Behavior

Ablation studies, which involve selectively disabling certain parts of a neural network to observe changes in performance, have become a crucial method for interpreting and understanding the inner workings of deep learning models. By setting specific nodes or neurons to zero (effectively "ablating" them), researchers can assess the contribution of those nodes to the overall model's decision-making process. This approach is particularly valuable in fields where model interpretability is as important as model performance, such as medical imaging (Zeiler & Fergus, 2014b) and natural language processing (J. Li et al., 2016).

In the context of our study, the ablation was not the primary focus but was conducted to gain insights into the robustness and reliability of the neural network in recognizing musical instruments. The ablation experiments revealed that certain neurons are critical for the accurate classification of specific instruments. For instance, when neurons responsible for recognizing the violin were ablated, there was a noticeable drop in the model's ability to identify violin sounds accurately. Similarly, ablation of neurons associated with piano recognition resulted in decreased accuracy for piano classification. These findings align with broader research in neural networks,

which shows that while some neurons are generalists, contributing to the recognition of multiple classes, others are specialists, heavily contributing to the recognition of specific classes (Morcos et al., 2018).

The inclusion of ablation studies in our discussion is essential because it highlights the potential vulnerabilities of the model, particularly in situations where certain pathways might be disrupted or degraded. Understanding these vulnerabilities is critical for improving the model's robustness and ensuring that it performs reliably across a range of conditions. While our study is not primarily focused on ablation, the results from these experiments provide valuable insights that can guide future research in model interpretability and reliability.

Moreover, ablation studies are increasingly being recognized as a vital component of model evaluation, not just in artificial neural networks but also in understanding biological neural networks. In neuroscience, similar approaches are used to understand the role of specific brain regions in cognitive functions (Kriegeskorte & Douglas, 2019) .By drawing parallels between artificial and biological neural networks, ablation studies can contribute to a deeper understanding of both fields, offering new avenues for interdisciplinary research.

In summary, while ablation was not the primary focus of our research, its inclusion in this discussion provides important context for understanding the strengths and limitations of the neural network model we developed. The insights gained from these experiments contribute to a broader conversation on model interpretability and robustness, which are crucial for advancing the field of neural network research.

6.5 Limitations and Strengths of the Models

The experiments conducted in this thesis revealed intriguing instrument-specific trends in recognition performance. By evaluating the binary classifiers on a diverse range of instrument families, including bass, brass, flute, guitar, keyboard, mallet, organ, reed, string, and vocal, insights were gained into the unique challenges and behaviours associated with each instrument type.

One notable finding was the consistent high performance of the models in recognizing mallet and string instruments, even in complex polyphonic mixes. The mallet family, which includes instruments like vibraphones, xylophones, and glockenspiels, exhibited distinct spectral characteristics that remained discernible to the binary classifiers, even when multiple instruments were playing simultaneously. Similarly, string instruments, such as violins, violas, cellos, and double basses, demonstrated a level of resilience in polyphonic contexts, with the models accurately identifying their presence amidst other instrument sounds.

These observations suggest that mallet and string instruments possess unique acoustic properties that set them apart from other instrument families. The percussive nature of mallet instruments, with their sharp attack and relatively short decay, may contribute to their distinguishability in polyphonic scenarios. The rich harmonic content and the sustained nature of string instrument sounds may also aid in their recognition, as the models can capture their distinct spectral signatures even in the presence of other instruments.

On the other hand, the experiments also revealed challenges in accurately recognizing certain instrument families, particularly in polyphonic settings. Instruments with overlapping frequency ranges and similar timbral characteristics, such as guitar and keyboard, were more prone to misclassification when played alongside other instruments. The spectral similarity between these instrument types can make it difficult for the models to differentiate between them, leading to increased confusion and lower recognition accuracy.

This finding highlights the limitations of relying solely on spectral information for instrument recognition in complex polyphonic scenarios. The overlapping frequency components and the blending of timbres can obscure the unique characteristics of individual instruments, making it challenging for the models to accurately distinguish between them.

To address these instrument-specific challenges, future research could explore the development of specialized features or feature extraction techniques that are tailored to capture the nuances and distinguishing attributes of each instrument family. For example, incorporating temporal information, such as attack and decay patterns, or considering the harmonic relationships

between instruments, could potentially enhance the models' ability to differentiate between similar-sounding instruments in polyphonic contexts.

Furthermore, investigating the use of instrument-specific models or ensemble approaches could be a promising direction. By training separate classifiers for each instrument family and combining their predictions, the overall recognition performance could be improved. This approach would allow each model to specialize in capturing the unique characteristics of a particular instrument type, potentially mitigating the confusion between similar-sounding instruments.

Additionally, exploring the integration of musical context and domain knowledge into the recognition process could provide necessary cues for disambiguating instruments with overlapping timbral properties. Leveraging information such as the typical roles and combinations of instruments in different musical genres or the likelihood of certain instruments playing together could aid in resolving ambiguities and improving recognition accuracy.

In conclusion, the instrument-specific trends and challenges identified in this thesis underscore the complexities involved in polyphonic instrument recognition. While some instrument families, such as mallet and string, exhibited consistent recognition performance, others, like guitar and keyboard, posed greater challenges due to their timbral similarities. These findings emphasize the need for further research and development of specialized techniques and models that can effectively capture and differentiate the unique characteristics of each instrument type, particularly in complex polyphonic scenarios. By addressing these instrument-specific challenges, future work can contribute to more robust and accurate instrument recognition systems that can handle the diverse range of instrument combinations found in real-world musical recordings.

6.6 Alternatives

While this study employs CNNs with attention mechanisms for binary classification of musical instruments, alternative approaches could also be considered. One possible alternative is using Transformer-based architectures, which have shown strong performance in sequence modeling

tasks and could potentially capture long-range dependencies in musical audio more effectively than CNNs. Another alternative lies in feature extraction; instead of relying solely on spectrogram-based representations, incorporating additional features such as MFCCs, chroma features, or wavelet transformations might enhance classification performance. Additionally, data augmentation techniques like pitch shifting, time stretching, or mixup augmentation could improve model generalization and robustness against variations in real-world recordings. Another approach is leveraging transfer learning by using pretrained models such as VGGish or OpenL3, which have been trained on large-scale audio datasets and might offer better generalization with limited labeled data. Finally, while deep learning is the primary focus of this work, traditional machine learning techniques such as Support Vector Machines or Gradient Boosting applied to hand-crafted audio features could serve as a baseline for comparison. These alternatives highlight different perspectives that could be explored in future work to refine and expand upon the methods presented in this study.

6.7 Summary of Chapter 6

This chapter has provided a discussion of our experimental findings in musical instrument recognition. We examined the effectiveness of our binary classifiers in recognizing individual instruments, analysing their performance across various acoustic conditions. The handling of complex audio environments, including polyphonic and noisy scenarios, was critically assessed. We explored the innovative approach of combining multiple spectrogram types with attention mechanisms in CNNs, highlighting its advantages and limitations. The chapter also addressed the strengths and weaknesses of our models, providing a balanced view of our contributions to the field. Through this discussion we have contextualized our results within the broader landscape of musical instrument recognition research, setting the stage for future advancements in this domain.

Chapter 7. Conclusion and Future Directions

As we reach the culmination of this research journey, Chapter 7 serves to synthesize our findings and chart a course for future exploration in musical instrument recognition. This chapter begins with a concise summary of our key findings, directly addressing how our research objectives were met. We then provide a critical reflection on the implications of our work, considering its potential impact on both theoretical understanding and practical applications in the field. The chapter also outlines the limitations of our current approach, using these as springboards to identify promising avenues for future research. By concluding with a forward-looking perspective, we aim to inspire continued innovation in the challenging and evolving domain of musical instrument recognition using deep learning techniques.

7.1. Summary of Key Findings

Table 23 presents a comparison between key aspects of human auditory perception and our instrument recognition model, offering an assessment of how closely our approach aligns with auditory theory. This self-evaluation aims to provide insight into the strengths and limitations of our model in relation to human hearing processes.

While we have made progress in several areas, it is important to note that our model still falls short of fully replicating human auditory capabilities. The table highlights partial achievements in neural processing and frequency analysis, showing that our MLP and spectrogram approaches bear some resemblance to human neurons and cochlear function. Our parallel processing of multiple instruments using binary classifiers appears to be a relative strength, mirroring the human ability to process multiple sounds simultaneously.

However, we acknowledge several limitations, particularly in contextual integration and attention mechanisms. These areas remain challenges for our current model and represent important directions for future research. The self-rating scores reflect a cautious and realistic assessment of our achievements, recognizing that while we've made strides in certain aspects, there is still considerable room for improvement in creating a truly human-like instrument recognition system.

This comparison (Table 23) serves not as a claim of superiority, but as a reflection on our progress and a guide for future enhancements purpose

Table 23. Key findings and experiment self-rating.

Human Auditory Perception	**Instrument Recognition Model**	**Self-Rating Score**
Human Neurons	MLP (Multi-Layer Perceptron)	Partially Achieved (2)
Human Frequency Analysis: Cochlea performs frequency decomposition	**Fourier Transformation:** Spectrograms derived from Fourier Transform	Partially Achieved (2)
Parallel Processing: Simultaneous processing of multiple sounds	**10 Binary Models in Parallel:** Independent classifiers for each instrument	Achieved (3)
Contextual Integration: Integrates auditory information with other senses and context	**Currently Limited:** Potential for future improvements	Not Achieved (1)
Hierarchical Processing: Processing from basic to complex features in the brain	**Multi-layer Neural Network:** Multiple layers capturing hierarchical patterns	Partially Achieved (2)
Attention Mechanisms: Selectively focusing on important sounds	**Focus on salient features:** Future work is to incorporate attention mechanisms	Partially Achieved (2)
Plasticity: Ability to adapt and change over time	**Training and adaptation:** Model learning through training on large datasets	Partially Achieved (2)
Auditory Scene Analysis: Segregates sound sources in complex environments	**Spectrogram Features Combination:** Log-Mel, MFCC, Chroma, Spectral Contrast, Tonnetz	Partially Achieved (2)
Multi-sensory Integration: Combines auditory with visual and other sensory information	**Currently audio-only:** Future work: Integrate multi-modal data	Not Achieved (1)
Robustness to Noise: Ability to understand speech in noisy environments	**Data Augmentation and Noise Handling:** Techniques to improve model robustnes	Partially Achieved (2)
Memory and Learning: Utilizes past experiences and learning	**Training with Large Datasets**: Model training on diverse and extensive datasets	Achieved (3)
Pattern Recognition: Recognizes patterns in auditory input	**Feature Extraction and Classification:** Spectrogram features and neural network classifiers	Achieved (3)
Temporal Integration: Integrates information over time	**Recurrent or Temporal Models:**Future work: Incorporate RNNs, LSTMs, GRUs, TCNs	Not Achieved (1)

Explanation of Scores:

- Not Achieved (1): The model currently does not incorporate this aspect.
- Partially Achieved (2): The model addresses this aspect to some extent, but there is significant room for improvement.
- Achieved (3): The model successfully incorporates this aspect and performs well.

Based on the assessment table comparing the Instrument Recognition Model's capabilities against Human Auditory Perception, it is evident that some progress has been made in achieving the research objectives outlined in the initial proposal. The model successfully incorporates several key aspects of human auditory processing, such as frequency analysis through spectrograms derived from the Short-Time Fourier Transform (STFT), parallel processing using 10 binary models for each instrument, and hierarchical processing through a multi-layer neural network architecture. These achievements demonstrate the model's ability to emulate and draw inspiration from the complex mechanisms of the human auditory system.

One of the notable strengths of the current model is its proficiency in pattern recognition, leveraging feature extraction techniques and neural network classifiers to identify and classify instruments based on their unique spectral characteristics. The model's training on large and diverse datasets has enabled it to develop a robust understanding of different instrument classes, akin to how humans learn and remember auditory patterns through exposure and experience. Additionally, the model's capacity for memory and learning has been fully realized, as evidenced by its successful training on extensive datasets, allowing it to capture the nuances and variations in instrument sounds.

However, the assessment also highlights several areas where the model falls short of human auditory perception, presenting opportunities for future research and improvement. The lack of contextual integration and attention mechanisms limits the model's ability to focus on salient features and adapt to different auditory contexts, which are crucial aspects of human hearing. Incorporating attention mechanisms and contextual understanding could enhance the model's recognition accuracy and robustness in complex auditory scenes. Similarly, the absence of temporal integration and reliance on audio-only data restrict the model's capacity to handle

sequential dependencies and leverage multi-sensory information, which are integral to human auditory perception.

To address these limitations and further advance the field of instrument recognition, future research should explore the integration of temporal models, such as:

1. Attention Mechanisms: Future work could involve incorporating various types of attention mechanisms, such as self-attention, multi-head attention, and transformer-based models.
2. Plasticity: Implementing continual learning, transfer learning, and adaptive learning techniques to enhance the model's ability to adapt to new instruments or changing environments.
3. Multi-sensory Integration: Exploring multi-modal approaches such as combining audio data with visual inputs using techniques like audiovisual fusion, cross-modal learning, and sensor fusion.
4. Robustness to Noise: Developing advanced noise handling techniques like noise-robust feature extraction, denoising autoencoders, and adversarial training.
5. Temporal Integration: Utilizing more advanced temporal models such as Temporal Convolutional Networks (TCNs), Bidirectional RNNs, and Transformer models to better capture temporal dependencies in audio signals.

7.1.1 Recapitulation of Main Results

In this thesis, we have successfully developed and implemented a novel approach for musical instrument recognition using binary classifiers in a OvA framework. This methodology has proven to be effective in accurately identifying individual instruments across a wide range of musical contexts, from solo performances to complex polyphonic compositions.

One of the key highlights of our research is the notable accuracy achieved by our models in various instrument combinations and musical settings. Through rigorous experimentation, we have demonstrated that our binary classifiers can reliably distinguish between different instruments, even in the presence of multiple simultaneously playing instruments. This is a

significant achievement, as polyphonic instrument recognition has been a long-standing challenge in the field of Music Information Retrieval (MIR).

Moreover, our models have exhibited robustness in identifying individual instruments and adaptability in handling diverse audio environments. By training our classifiers on a dataset that is encompassing a wide variety of instrument sounds, recording conditions, and musical genres, we have ensured that our models can generalize well to real-world scenarios. This adaptability is crucial for practical applications of instrument recognition technology, such as automatic music transcription, audio source separation, and music recommendation systems.

The proposed model achieved in this research represent an advance in the field of MIR. By leveraging state-of-the-art deep learning techniques, such as convolutional neural networks (CNNs) and spectrogram-based feature extraction, we have pushed the boundaries of what is possible in instrument recognition. Our models have demonstrated the ability to capture and learn intricate patterns and relationships in audio data, enabling accurate classification of instruments even in complex musical contexts.

7.1.2 Reflection on the Research Objectives

At the outset of this research, we set out with the primary research objective of creating and evaluating models capable of accurately recognizing different musical instruments, both in solo and in combination. Through our extensive experiments and evaluations, we can state that this objective has been effectively met**.** Our binary classifier approach has proven successful in identifying individual instruments with high accuracy, not only in solo settings but also in polyphonic compositions where multiple instruments are playing simultaneously.

Throughout the course of this research, we encountered various challenges that required us to adapt our methods and strategies. These challenges ranged from data preprocessing and feature extraction to model architecture design and hyperparameter tuning. By addressing these challenges head-on and iteratively refining our approaches, we were able to continuously enhance the performance of our models. This learning and adaptation process was instrumental in achieving the high levels of accuracy and robustness demonstrated by our final models.

One of the key lessons learned during this research was the importance of a diverse training dataset. By incorporating a wide range of instrument sounds, recording conditions, and musical genres into our dataset, we were able to train models that generalize well to unseen data and real-world scenarios. This insight highlights the significance of data quality and diversity in developing robust instrument recognition systems.

7.1.2.1 Why the Combined Approach is Effective

By combining these features, we can improve the robustness and accuracy of our musical instrument classification model. This approach leverages the strengths of each feature type, ensuring that the model can capture a wide range of audio characteristics. The natural progression from using a single feature to multiple features aligns with the principles of the Universal Approximation Theorem, suggesting that if an approximation works with one feature, it should also work with an expanded set of features, provided the model's capacity is sufficient.

Expanding our feature set from a single spectrogram to a combination of six different spectrogram features should enhance the convergence speed and overall performance of our neural network model. This approach is not just an extension of our previous method but a significant improvement that captures the multifaceted nature of musical audio signals.

7.1.3 Overview of Contributions to the Field

The contributions of this research to the field of Music Information Retrieval as follows. Firstly, the introduction of a OvA approach specifically tailored for musical instrument recognition represents a novel methodology in the MIR community. By treating each instrument as a separate binary classification problem and combining the results of multiple classifiers, we have demonstrated the effectiveness of this approach in accurately identifying instruments in various musical contexts.

Secondly, the OvA methodology offers several advantages over traditional multi-class classification approaches. It allows for the development of specialized classifiers for each instrument, enabling the models to capture the unique characteristics and nuances of individual instruments more effectively. Additionally, the modular nature of the OvA framework makes it

easier to add new instruments or update existing classifiers without requiring a complete retraining of the entire system.

Our work has also bridged gaps in understanding how artificial intelligence can be effectively applied to the complex task of instrument recognition. By demonstrating the successful application of deep learning techniques, such as CNNs and spectrogram-based feature extraction, we have provided insights into the design and implementation of robust instrument recognition systems. Our research serves as a foundation for future work in this area, offering guidance on data preprocessing, model architecture, and evaluation strategies.

Moreover, the extensive experiments conducted in this thesis have shed light on the capabilities and limitations of current instrument recognition technology. By evaluating our models on a wide range of musical scenarios, from solo performances to polyphonic compositions with varying levels of complexity, we have provided an assessment of the state of the art in instrument recognition. These insights can guide future research efforts, highlighting areas where further improvements are needed and identifying potential avenues for innovation.

In conclusion, this thesis has made significant contributions to the field of Music Information Retrieval by introducing a novel OvA approach for musical instrument recognition, demonstrating its effectiveness through extensive experimentation, and providing insights into the application of artificial intelligence in this domain. Our work lays the groundwork for future advancements in instrument recognition technology, paving the way for more accurate, robust, and versatile systems that can enhance various applications in music analysis, retrieval, and creation.

7.1.4 Potential Impact

This research may have implications for practical MIR systems, particularly in enhancing automatic music analysis and retrieval. First, the proposed CNN-based binary classification approach improves the accuracy of musical instrument recognition, which can benefit music recommendation systems, enabling more refined search and categorization based on instrument content rather than just genre or metadata. Second, the study’s use of multi-spectrogram

representations and attention mechanisms provides a robust feature extraction framework, making MIR models more resilient to polyphonic and noisy environments, which is particularly valuable for audio restoration, music archiving, and historical music preservation. Third, the findings contribute to music education tools by facilitating better instrument separation, enabling learners to isolate and study specific instrument parts within recordings. Fourth, the research opens up opportunities for real-time applications, such as live performance tracking and instrument-based adaptive audio mixing. Finally, the interpretability insights gained through feature map and heatmap analyses help bridge the gap between black-box deep learning models and human auditory perception, leading to more explainable and trustworthy AI-based music classification systems.

7.2. Future Research Opportunities

7.2.1 Enhancing the OvA Model

7.2.1.1 Slide Window Architectures

In the future, we can adapt our model to be a real time recoding analysis model. This approach involves sliding a fixed-size window (e.g., 4 seconds) over the longer audio signal.

1. At each step, a 4-second segment of the audio is extracted and converted into a spectrogram, which is then fed into the classifier.
2. If the remaining audio is shorter than 4 seconds, we can pad it with zeros to maintain the fixed input size.
3. The classifier makes predictions for each 4-second segment, and the final prediction for the entire audio can be obtained by aggregating the individual segment predictions (e.g., majority voting or averaging).

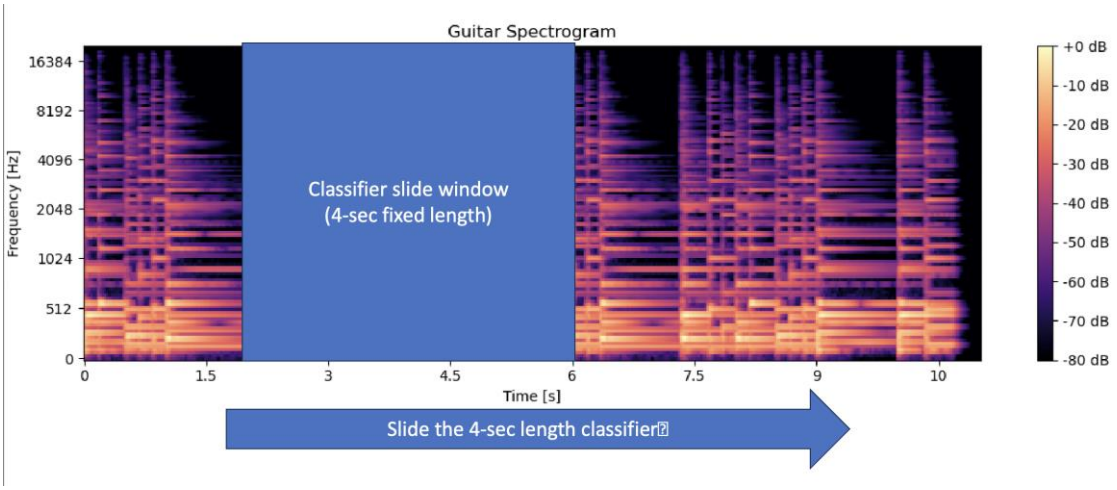


Figure 88. Spectrogram Slide Window Method.

As shown in Figure 88, while this method is straightforward to implement, it may not fully capture the long-term dependencies and contextual information present in the audio.

7.2.1.2 Adaptive Spectrogram Fusion Network (ASFN)

One potential future improvement (Table 24) is to adapt the transformer architecture to explicitly handle both temporal and spectral dimensions of music. This could involve separate self-attention mechanisms for time and frequency axes, followed by a cross-attention layer to integrate both dimensions.

Table 24. Future work: ASFN for spectrogram.

Layer	**Description**	**Function**		
Input	Combined spectrogram	Provide unified spectral-temporal representation		
Layer 1	Temporal self-attention	Capture patterns along the time axis		
Layer 2	Spectral self-attention	Capture patterns along the frequency axis		
Layer 3	Temporal-Spectral cross-attention	Integrate temporal and spectral information		
Output	Integrated features	Provide refined temporal-spectral features		

7.2.1.3 Residual Attention Modules (RAM)

These structures (Table 25) can address different aspects of the musical instrument recognition challenge, from handling multi-scale patterns to dynamically adapting to different input characteristics. Implementing and comparing these approaches could lead to significant advancements in our model's performance and versatility.

Table 25. Future work: RAM for spectrogram.

Layer	**Description**	**Function**	
Input	Combined spectrogram	Provide unified spectral representation	
Block 1-N	Convolutional layer	Extract spectrogram features	
	Attention in residual connection	Selectively emphasize important features	
	Addition of main and residual paths	Allow flexible information flow	
Output	Refined features	Provide features with selective skip connections	

7.2.2 Emerging Areas in Instrument Recognition

7.2.2.1 Cross-Disciplinary Exploration

Another future work is to explore the cross-disciplinary fields, such as the intersection of musicology and AI, can deepen our understanding of music through a technological lens. By combining expertise from different fields, such as ethnomusicology, psychology, and computational arts, researchers can build innovative systems that enhance how we analyse, classify, and appreciate music. This interdisciplinary approach could reveal patterns or features that might otherwise remain hidden, fostering new ways of perceiving music through intelligent algorithms.

7.2.2.2 Real-Time Recognition

Also, our next future work can also investigate real-time instrument recognition systems is vital for live music performances and interactive applications. Such systems could analyse audio streams in real time, identifying instruments and providing immediate feedback or visualizations to performers and audiences alike. This opens up exciting opportunities for interactive music learning tools, immersive performances, and adaptive soundscapes in games or installations. Real-time systems require highly optimized models that can balance accuracy and latency, pushing the boundaries of current technology and paving the way for more responsive musical interfaces.

7.2.3 Expanding the Scope of Application

7.2.3.1 Broader Musical Diversity

Expanding the application of the models to a wider variety of musical genres and styles will ensure applicability. Current models may be effective for commonly recognized genres, but incorporating genres like folk, jazz, classical, electronic, and traditional music will broaden the dataset's diversity. By including more diverse collection of musical styles, models can capture unique acoustic features and patterns, thereby improving the recognition of a wider range of musical compositions.

7.2.3.2 Global Musical Instruments

Including a broader range of global musical instruments in the datasets is crucial for promoting inclusivity and diversity in musical recognition. Current datasets are often limited to Western instruments. Expanding to encompass instruments from various cultures, such as the sitar (India), erhu (China), oud (Middle East), and balalaika (Russia), will create more inclusive models. This diversification will enable better identification of non-Western instruments and foster a global perspective in music information retrieval systems.

7.2.4 Enhancing Noise Robustness in Instrument Recognition

Noise robustness is a critical factor in the practical deployment of musical instrument recognition systems, especially in real-world applications where audio signals often contain varying levels of background interference. Future research can focus on improving model resilience to noise by exploring several strategies. One potential approach is to conduct controlled noise experiments by introducing different noise types (e.g., white noise, crowd noise, environmental noise) at varying intensity levels and observing their impact on classification accuracy. By systematically altering noise intensity and plotting the model's performance as a trend line, researchers can identify specific thresholds where recognition degrades, helping to refine pre-processing techniques or model architectures accordingly. Another promising direction is the application of adaptive noise filtering methods, such as trainable denoising autoencoders or wavelet-based denoising techniques, which can enhance signal clarity before classification. These approaches can be integrated into the preprocessing pipeline to dynamically adjust to different noise environments. Furthermore, leveraging contrastive learning and domain adaptation techniques could enable the model to generalize better across noisy conditions. By training models with noise-augmented datasets and employing self-supervised learning strategies, future work can explore methods to enhance feature extraction from distorted spectrogram inputs. Real-world robustness can also be improved by exploring multi-channel input processing, where multiple spectrogram representations (e.g., log-mel, MFCCs, scalograms) are fused to enhance discrimination under noisy conditions. By using a weighted fusion strategy, the model can prioritize more robust features when dealing with high-noise environments. Lastly, a human

perception-inspired approach could involve incorporating attention mechanisms that focus on stable, dominant frequency bands in an audio signal. This technique could improve the model's ability to extract relevant features while disregarding transient noise, mimicking the way humans selectively focus on important auditory cues. By systematically exploring these noise-robustness techniques, future work can enhance the reliability of instrument recognition models, making them more adaptable to diverse real-world audio environments.

7.3 Reproducibility and Methodological Integrity

Ensuring the reproducibility of our experiments is paramount. To this end, we have adhered to the principles of transparency and accessibility in our research methodology. The Binary OvA classifier model, developed as part of this study, leverages deep learning techniques to discern individual musical instruments from spectrogram inputs derived from the NSynth dataset. This approach allows for nuanced recognition capabilities across a spectrum of acoustic environments, including those compromised by various forms of background noise such as urban sounds and natural ambiances.

7.3.1 Experimentation Framework

Our experimentation framework is built on the foundation of TensorFlow and Librosa libraries, enabling sophisticated audio processing and model training workflows. Each step, from data preprocessing and model training to evaluation, has been methodically planned to ensure that results are not only accurate but also reproducible by peers and practitioners in the field.

In addition to evaluating our Binary OvA classifier model under ideal conditions, we embarked on a series of noise augmentation experiments. By integrating real-world noise samples into our test datasets, we simulated a range of auditory environments to assess the robustness of our model. This involved the meticulous overlay of diverse noise types—ranging from traffic noise to natural soundscapes—onto the original musical instrument samples, thereby creating a challenging yet realistic testing ground for instrument recognition capabilities.

7.3.2 Accessibility of Resources

To foster a collaborative and transparent research environment, all code, datasets, and pre-trained models used in our study are made publicly available. This ensures that other researchers can replicate our experiments, validate our claims, and build upon our work. The documentation accompanying our codebase details the setup, execution, and evaluation processes, providing a clear roadmap for replication.

To get access to the code and further explore our work, please download from the GitHub link: https://github.com/fireHedgehog/music-intrument-OvA-model.

References

Addison, P. S. (2017). The Illustrated Wavelet Transform Handbook: Introductory Theory and Applications in Science, Engineering, Medicine and Finance. CRC Press.

Agostini, G., Longari, M., & Pollastri, E. (2003). Musical instrument timbres classification with spectral features. EURASIP Journal on Advances in Signal Processing, 2003, 1–10.

Agus, T. R., Suied, C., Thorpe, S. J., & Pressnitzer, D. (2012). Fast recognition of musical sounds based on timbre. The Journal of the Acoustical Society of America, 131(5), 4124–4133.

Alaparthi, S., & Mishra, M. (2020). Bidirectional Encoder Representations from Transformers (BERT): A sentiment analysis odyssey. arXiv Preprint arXiv:2007.01127.

Allen, J. B., & Rabiner, L. R. (1977a). A unified approach to short-time Fourier analysis and synthesis. Proceedings of the IEEE, 65(11), 1558–1564.

Allen, J. B., & Rabiner, L. R. (1977b). Short Term Spectral Analysis, Synthesis, and Modification by Discrete Fourier Transform. IEEE Transactions on Acoustics, Speech, and Signal Processing, 25(3), 235–238.

Anhari, A. K. (2020). Learning multi-instrument classification with partial labels. arXiv Preprint arXiv:2001.08864.

Arons, B. (1992). A review of the cocktail party effect. Journal of the American Voice I/O Society, 12(7), 35–50.

Aucouturier, J.-J., & Pachet, F. (2003). Representing musical genre: A state of the art. Journal of New Music Research, 32(1), 83–93.

Aytar, Y., Vondrick, C., & Torralba, A. (2016). Soundnet: Learning sound representations from unlabeled video. Advances in Neural Information Processing Systems, 29.

Bahdanau, D., Cho, K., & Bengio, Y. (2014). Neural machine translation by jointly learning to align and translate. arXiv Preprint arXiv:1409.0473.

Balke, S., Dorfer, M., Carvalho, L., Arzt, A., & Widmer, G. (2019). Learning soft-attention models for tempo-invariant audio-sheet music retrieval. arXiv Preprint arXiv:1906.10996.

Benetos, E., Dixon, S., Giannoulis, D., Kirchhoff, H., & Klapuri, A. (2013). Automatic music transcription: Challenges and future directions. Journal of Intelligent Information Systems, 41, 407–434.

Bergstra, J., & Bengio, Y. (2012). Random search for hyper-parameter optimization. Journal of Machine Learning Research, 13(2), 281–305.

Bishop, C. M. (2006). Pattern recognition and machine learning. springer.

Bittner, R., Wilkins, J., Yip, H., & Bello, J. P. (2018, November). MedleyDB 2.0 Audio. Zenodo. https://doi.org/10.5281/zenodo.1715175

Blackman, R. B., & Tukey, J. W. (1958). The measurement of power spectra from the point of view of communications engineering—Part I. Bell System Technical Journal, 37(1), 185–282.

Blaszke, M., & Kostek, B. (2022). Musical instrument identification using deep learning approach. Sensors, 22(8), 3033.

Boashash, B. (2015). Time-frequency signal analysis and processing: A comprehensive reference. Academic Press.

Bogdanov, D., Wack, N., Gómez Gutiérrez, E., Gulati, S., Boyer, H., Mayor, O., Roma, G., Salamon, J., Zapata, J. R., & Serra, X. (2013). Essentia: An audio analysis library for music information retrieval. 14th International Society for Music Information Retrieval Conference (ISMIR), 493–498.

Bosch, J. J., Fuhrmann, F., & Herrera, P. (2018). IRMAS: a dataset for instrument recognition in musical audio signals [Dataset]. https://doi.org/10.5281/zenodo.1290750

Bosch, J. J., Janer, J., Fuhrmann, F., & Herrera, P. (2012). A Comparison of Sound Segregation Techniques for Predominant Instrument Recognition in Musical Audio Signals. ISMIR, 559–564.

Bregman, A. S. (1994). Auditory scene analysis: The perceptual organization of sound. MIT press.

Brigham, E. O. (1988). The Fast Fourier Transform and Its Applications. Prentice-Hall.

Broadbent, D. E. (2013). Perception and communication. Elsevier.

Brown, J. C. (1991). Calculation of a constant Q spectral transform. The Journal of the Acoustical Society of America, 89(1), 425–434.

Brown, J. C., & Puckette, M. S. (1992). An Efficient Algorithm for the Calculation of a Constant Q Transform. The Journal of the Acoustical Society of America, 92(5), 2698–2701.

Calvert, G., Spence, C., & Stein, B. E. (2004). The handbook of multisensory processes. MIT press.

Casey, M. A., Veltkamp, R., Goto, M., Leman, M., Rhodes, C., & Slaney, M. (2008). Content-based music information retrieval: Current directions and future challenges. Proceedings of the IEEE, 96(4), 668–696.

Celma Herrada, Ò. & others. (2009). Music recommendation and discovery in the long tail. Universitat Pompeu Fabra.

Chandna, P., Miron, M., Janer, J., & Gómez, E. (2017). Monoaural audio source separation using deep convolutional neural networks. 2017 International Conference on Latent Variable Analysis and Signal Separation (LVA/ICA), 258–266.

Chattopadhay, A., Sarkar, A., Howlader, P., & Balasubramanian, V. N. (2018). Grad-cam++: Generalized gradient-based visual explanations for deep convolutional networks. 2018 IEEE Winter Conference on Applications of Computer Vision (WACV), 839–847.

Chen, R., Ghobakhlou, A., & Narayanan, A. (2024). Musical Instrument Recognition in Polyphonic Audio Through Convolutional Neural Networks and Spectrograms. ICALIP 2024: International Conference on Audio, Language and Image Processing, 18(07). https://publications.waset.org/abstracts/185822.pdf

Chen, R., Ghobakhlou, A., Narayanan, A., Pérez, M., Oyanadel, R. O. C., & Borras-Chavez, R. (2023). Semi-Supervised Deep Learning for Estimating Fur Seal Numbers. 2023 38th International Conference on Image and Vision Computing New Zealand (IVCNZ), 1–5.

Chen, R., & Narayanan, A. (2021). Evolving Convolutional Filter Using Genetic Algorithm for Image Classification. International Conference on Machine Intelligence (ICMI), 15(12), 1–10. https://publications.waset.org/abstracts/136990.pdf

Cherry, E. C. (1953). Some experiments on the recognition of speech, with one and with two ears. The Journal of the Acoustical Society of America, 25(5), 975–979.

Chi, T., Ru, P., & Shamma, S. A. (2005). Multiresolution spectrotemporal analysis of complex sounds. The Journal of the Acoustical Society of America, 118(2), 887–906.

Choi, K., Fazekas, G., Cho, K., & Sandler, M. (2017). A tutorial on deep learning for music information retrieval. arXiv Preprint arXiv:1709.04396.

Choi, K., Fazekas, G., & Sandler, M. (2016). Automatic tagging using deep convolutional neural networks. arXiv Preprint arXiv:1606.00298.

Choi, K., Fazekas, G., Sandler, M., & Cho, K. (2017). Convolutional Recurrent Neural Networks for Music Classification. Proceedings of the IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), 2392–2396.

Chong, D., Wang, H., Zhou, P., & Zeng, Q. (2023). Masked spectrogram prediction for self-supervised audio pre-training. ICASSP 2023-2023 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), 1–5.

Chowning, J. M. (1973). The synthesis of complex audio spectra by means of frequency modulation. Journal of the Audio Engineering Society, 21(7), 526–534.

Clarke, E. (2005). Ways of listening: An ecological approach to the perception of musical meaning. Oxford University Press.

Cohen, L. (1995). Time-Frequency Analysis. Prentice Hall.

Copland, A. (1952). Music and imagination (Vol. 22). Harvard University Press.

Copland, A. (2011). What to Listen for in Music. Penguin.

Cramer, A. L., Wu, H.-H., Salamon, J., & Bello, J. P. (2019). Look, listen, and learn more: Design choices for deep audio embeddings. ICASSP 2019-2019 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), 3852–3856.

Daffern, H., & Howard, D. (2012). Spectral characteristics of the baroque trumpet: A case study. Acoustics 2012.

Daubechies, I. (1992). Ten Lectures on Wavelets. SIAM.

Davis, S., & Mermelstein, P. (1980). Comparison of parametric representations for monosyllabic word recognition in continuously spoken sentences. IEEE Transactions on Acoustics, Speech, and Signal Processing, 28(4), 357–366.

Défossez, A. (2021). Hybrid spectrogram and waveform source separation. arXiv Preprint arXiv:2111.03600.

Dhanalakshmi, P., Palanivel, S., & Ramalingam, V. (2011). Wavelet Based Feature Extraction Method for Classification of Audio Signals. International Journal of Computer Applications, 24(7), 1–6.

Dieleman, S., & Schrauwen, B. (2014). End-to-end learning for music audio. 2014 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), 6964–6968.

Dorfer, M., Hajič Jr, J., & Widmer, G. (2018). Attention as a perspective for learning tempo-invariant audio queries. arXiv Preprint arXiv:1809.05689.

Dosovitskiy, A., Beyer, L., Kolesnikov, A., Weissenborn, D., Zhai, X., Unterthiner, T., Dehghani, M., Minderer, M., Heigold, G., Gelly, S., Uszkoreit, J., & Houlsby, N. (2020). An image is worth 16x16 words: Transformers for image recognition at scale. arXiv Preprint arXiv:2010.11929.

Downie, J. S. (2003). Music information retrieval. Annual Review of Information Science and Technology, 37(1), 295–340.

EarMaster ApS. (2023). EarMaster. https://www.earmaster.com/

Elhilali, M., Ma, L., Micheyl, C., Oxenham, A. J., & Shamma, S. A. (2009). Temporal coherence in the perceptual organization and cortical representation of auditory scenes. Neuron, 61(2), 317–329.

Ellis, D. P. (2009). Chroma feature analysis and synthesis. Proceedings of the 24th International Conference on Music Information Retrieval (ISMIR).

Elman, J. L. (1990). Finding structure in time. Cognitive Science, 14(2), 179–211.

Engel, J., Resnick, C., Roberts, A., Dieleman, S., Eck, D., Simonyan, K., & Norouzi, M. (2017). Neural Audio Synthesis of Musical Notes with WaveNet Autoencoders.

Engel, J., Resnick, C., Roberts, A., Dieleman, S., Norouzi, M., Eck, D., & Simonyan, K. (2017). Neural audio synthesis of musical notes with wavenet autoencoders. International Conference on Machine Learning, 1068–1077.

Ericsson, K. A., Krampe, R. Th., & Tesch-Römer, C. (1993). The role of deliberate practice in the acquisition of expert performance. Psychological Review, 100(3), 363–406.

Eronen, A. (2001). Comparison of features for musical instrument recognition. Proceedings of the 2001 IEEE Workshop on the Applications of Signal Processing to Audio and Acoustics (Cat. No. 01TH8575), 19–22.

Eronen, A., & Klapuri, A. (2000). Musical instrument recognition using cepstral coefficients and temporal features. 2000 IEEE International Conference on Acoustics, Speech, and Signal Processing. Proceedings (Cat. No. 00CH37100), 2, II753–II756.

Essid, S., Richard, G., & David, B. (2006). Musical instrument recognition by pairwise classification strategies. IEEE Transactions on Audio, Speech, and Language Processing, 14(4), 1401–1412.

Ezzaidi, H., Bahoura, M., & Hall, G. E. (2012). Towards a characterization of musical timbre based on chroma contours. Advanced Machine Learning Technologies and Applications:

First International Conference, AMLTA 2012, Cairo, Egypt, December 8-10, 2012. Proceedings 1, 162–171.

Fitzgerald, D. (2004). Automatic drum transcription and source separation.

Fletcher, N. H., & Rossing, T. D. (1998). The Physics of Musical Instruments. Springer.

Frazier, J. M., Assgari, A. A., & Stilp, C. E. (2019). Musical instrument categorization is highly sensitive to spectral properties of earlier sounds. Attention, Perception, & Psychophysics, 81, 1119–1126.

Fuhrmann, F. & others. (2012). Automatic musical instrument recognition from polyphonic music audio signals.

Fürnkranz, J., Hüllermeier, E., Loza Mencía, E., & Brinker, K. (2008). Multilabel classification via calibrated label ranking. Machine Learning, 73, 133–153.

García, S., Luengo, J., Herrera, F., & others. (2015). Data preprocessing in data mining (Vol. 72). Springer.

George, E., Hunter, W. G., & Hunter, J. S. (2005). Statistics for experimenters: Design, innovation, and discovery. Wiley.

Goodfellow, I., Bengio, Y., & Courville, A. (2016). Deep Learning. MIT Press.

Graves, A., Mohamed, A., & Hinton, G. (2013). Speech Recognition with Deep Recurrent Neural Networks. IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP).

Griffin, D. W., & Lim, J. S. (1984). Signal Estimation from Modified Short-Time Fourier Transform. IEEE Transactions on Acoustics, Speech, and Signal Processing, 32(2), 236–243.

Gururani, S., Sharma, M., & Lerch, A. (2019). An attention mechanism for musical instrument recognition. Proceedings of the 20th International Society for Music Information Retrieval Conference (ISMIR), 83–90.

Hall, G. E., Ezzaidi, H., & Bahoura, M. (2014). Instrument timbre chroma contours and psycho-visual human analysis. 2014 International Conference on Multimedia Computing and Systems (ICMCS), 327–330.

Han, Y.-S., Chen, H.-W., & Yang, Y.-H. (2016). Deep convolutional neural networks for predominant instrument recognition in polyphonic music. IEEE/ACM Transactions on Audio, Speech, and Language Processing, 2084–2096.

Harris, F. J. (1978). On the Use of Windows for Harmonic Analysis with the Discrete Fourier Transform. Proceedings of the IEEE, 66(1), 51–83.

Harte, C. (2010). Towards automatic extraction of harmony information from music signals [PhD Thesis].

Harte, C., Sandler, M., & Gasser, M. (2006). Detecting harmonic change in musical audio. Proceedings of the 1st ACM Workshop on Audio and Music Computing Multimedia, 21–26.

Hastie, T., Tibshirani, R., & Friedman, J. (2009). The Elements of Statistical Learning: Data Mining, Inference, and Prediction. Springer.

Havelock, D., Kuwano, S., & Vorländer, M. (2008). Handbook of signal processing in acoustics (Vol. 1). Springer.

Hawthorne, C., Stasyuk, A., Roberts, A., Simon, I., Huang, C.-Z. A., Dieleman, S., Elsen, E., Engel, J., & Eck, D. (2018). Enabling factorized piano music modeling and generation with the MAESTRO dataset. arXiv Preprint arXiv:1810.12247.

Haykin, S., & Chen, Z. (2005). The cocktail party problem. Neural Computation, 17(9), 1875–1902.

Herrera, P., Amatriain, X., Batlle, E., & Serra, X. (2000). Automatic classification of musical instrument sounds. Journal of New Music Research, 29(1), 29–43.

Herrera-Boyer, P., Klapuri, A., & Davy, M. (2006). Automatic classification of pitched musical instrument sounds. In Signal processing methods for music transcription (pp. 163–200). Springer.

Hershey, S., Chaudhuri, S., Ellis, D. P., Gemmeke, J. F., Jansen, A., Moore, R. C., Plakal, M., Platt, D., Saurous, R. A., Seybold, B., & others. (2017). CNN architectures for large-scale audio classification. 2017 Ieee International Conference on Acoustics, Speech and Signal Processing (Icassp), 131–135.

Hinton, G., Deng, L., Yu, D., Dahl, G. E., Mohamed, A., Jaitly, N., Senior, A., Vanhoucke, V., Nguyen, P., Sainath, T. N., & others. (2012). Deep neural networks for acoustic modeling in speech recognition: The shared views of four research groups. IEEE Signal Processing Magazine, 29(6), 82–97.

Hochreiter, S., & Schmidhuber, J. (1997). Long short-term memory. Neural Computation, 9(8), 1735–1780.

Hou, Q., Zhou, D., & Feng, J. (2021). Coordinate attention for efficient mobile network design. Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 13713–13722.

Hsu, C.-L., & Jang, J.-S. R. (2009). On the improvement of singing voice separation for monaural recordings using the MIR-1K dataset. IEEE Transactions on Audio, Speech, and Language Processing, 18(2), 310–319.

Hu, X., & Downie, J. S. (2007). Exploring Mood Metadata: Relationships with Genre, Artist and Usage Metadata. ISMIR, 67–72.

Huang, C.-H., & Yang, Y.-H. (2020). Pop music highlighter: Marking the emotion keypoints. IEEE Transactions on Affective Computing, 11(4), 711–723.

Huang, Z., Dong, M., Mao, Q., & Zhan, Y. (2014). Speech emotion recognition using CNN. Proceedings of the 22nd ACM International Conference on Multimedia, 801–804.

Humphrey, E., Durand, S., & McFee, B. (2018). OpenMIC-2018: An Open Data-set for Multiple Instrument Recognition. ISMIR, 438–444.

Humphrey, E. J., Bello, J. P., & LeCun, Y. (2012). Moving beyond feature design: Deep architectures and automatic feature learning in music informatics. ISMIR, 403–408.

Jiang, D.-N., Lu, L., Zhang, H.-J., Tao, J.-H., & Cai, L.-H. (2002). Music type classification by spectral contrast feature. Proceedings. IEEE International Conference on Multimedia and Expo, 1, 113–116.

Johnston, W. A., & Dark, V. J. (1986). Selective attention. Annual Review of Psychology, 37(1), 43–75.

Kereliuk, C., & Depalle, P. (2008). Improved hidden Markov model partial tracking through time-frequency analysis. Proceedings of the Digital Audio Effects (DAFx-08), 1–4.

Kitahara, T., Goto, M., & Okuno, H. G. (2003). Musical instrument identification based on F0-dependent multivariate normal distribution. 2003 IEEE International Conference on Acoustics, Speech, and Signal Processing, 2003. Proceedings.(ICASSP'03)., 5, V–421.

Klinke, R. (1987). Processing of acoustic stimuli in the inner ear–a review of recent research results. HNO, 35(4), 139–148.

Koffka, K. (1922). Perception: An introduction to the Gestalt-Theorie. Psychological Bulletin, 19(10), 531.

Kogan, J. A., & Margoliash, D. (1998). Automated recognition of bird song elements from continuous recordings using dynamic time warping and hidden Markov models: A comparative study. The Journal of the Acoustical Society of America, 103(4), 2185–2196.

Kohavi, R. (1995). A study of cross-validation and bootstrap for accuracy estimation and model selection. Proceedings of the 14th International Joint Conference on Artificial Intelligence - Volume 2, 1137–1145.

Kong, Q., Cao, Y., Iqbal, T., Xu, Y., Wang, W., & Plumbley, M. D. (2019). Cross-task learning for audio tagging, sound event detection and spatial localization: DCASE 2019 baseline systems. arXiv Preprint arXiv:1904.03476.

Kostek, B. (2004). Musical instrument classification and duet analysis employing music information retrieval techniques. Proceedings of the IEEE, 92(4), 712–729.

Koutini, K., Eghbal-Zadeh, H., Haunschmid, V., Primus, P., Chowdhury, S., & Widmer, G. (2020). Receptive-field regularized CNNs for music classification and tagging. arXiv Preprint arXiv:2007.13503.

Kraus, N., & Chandrasekaran, B. (2010). Music training for the development of auditory skills. Nature Reviews Neuroscience, 11(8), 599–605.

Kriegeskorte, N., & Douglas, P. K. (2019). Cognitive computational neuroscience. Nature Neuroscience, 22(6), 1413–1420.

Krishna, A., & Sreenivas, T. V. (2004). Music instrument recognition: From isolated notes to solo phrases. 2004 IEEE International Conference on Acoustics, Speech, and Signal Processing, 4, iv–iv.

Kullback, S., & Leibler, R. A. (1951). On information and sufficiency. The Annals of Mathematical Statistics, 22(1), 79–86.

LeCun, Y., Bengio, Y., & Hinton, G. (2015). Deep learning. Nature, 521(7553), 436–444.

LeCun, Y., Bengio, Y., & others. (1995). Convolutional networks for images, speech, and time series. The Handbook of Brain Theory and Neural Networks, 3361(10), 1995.

LeCun, Y., Bottou, L., Bengio, Y., & Haffner, P. (1998). Gradient-based learning applied to document recognition. Proceedings of the IEEE, 86(11), 2278–2324.

Li, J., Chen, X., Hovy, E., & Jurafsky, D. (2016). Understanding neural networks through representation erasure. arXiv Preprint arXiv:1609.05978.

Li, W., Chen, J.-M., Smith, J., & Wolfe, J. (2015). Effect of vocal tract resonances on the sound spectrum of the saxophone. Acta Acustica United with Acustica, 101(2), 270–278.

Liang, D., Shi, Y., Wang, Y., Singhal, N., Xiao, A., Shaw, J., Thomaz, E., Kalinli, O., & Seltzer, M. (2021). Transferring voice knowledge for acoustic event detection: An empirical study. arXiv Preprint arXiv:2110.03174.

Lin, J. (1991). Divergence measures based on the Shannon entropy. IEEE Transactions on Information Theory, 37(1), 145–151.

Liu, X., & Zhang, M. (2022). MATT: A multiple-instance attention mechanism for long-tail music genre classification. arXiv Preprint arXiv:2209.04109.

Liu, X.-R., Ju, B., Ren, T.-B., & Shao, H.-B. (2010). Application of the wavelet transform and a neural network to predict subsoil compaction. Computers and Electronics in Agriculture, 74(1), 86–90.

Logan, B. (2000). Mel frequency cepstral coefficients for music modeling. Proceedings of the International Symposium on Music Information Retrieval (ISMIR).

Logan, B. & others. (2000). Mel frequency cepstral coefficients for music modeling. Ismir, 270(1), 11.

Lostanlen, V., Salamon, J., McFee, B., Cartwright, M., & Bello, J. P. (2018). Deep convolutional networks on the pitch spiral for musical instrument recognition. 2018 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), 386–390.

Mallat, S. (1999). A Wavelet Tour of Signal Processing. Academic Press.

Martín Abadi, Ashish Agarwal, Paul Barham, Eugene Brevdo, Zhifeng Chen, Craig Citro, Greg S. Corrado, Andy Davis, Jeffrey Dean, Matthieu Devin, Sanjay Ghemawat, Ian Goodfellow, Andrew Harp, Geoffrey Irving, Michael Isard, Jia, Y., Rafal Jozefowicz, Lukasz Kaiser, Manjunath Kudlur, … Xiaoqiang Zheng. (2015). TensorFlow: Large-Scale Machine Learning on Heterogeneous Systems. http://tensorflow.org/

Martin, K. D. (1999). Sound-source recognition: A theory and computational model [PhD Thesis]. Massachusetts Institute of Technology.

McAdams, S. (1999). Perspectives on the contribution of timbre to musical structure. Computer Music Journal, 23(3), 85–102.

McAdams, S. (2013). Musical timbre perception. The Psychology of Music, 35–67.

McAdams, S., & Giordano, B. L. (2008). The Perception of Timbre. In The Oxford Handbook of Music Psychology (pp. 72–80). Oxford University Press.

McCulloch, W. S., & Pitts, W. (1943). A logical calculus of the ideas immanent in nervous activity. The Bulletin of Mathematical Biophysics, 5(4), 115–133.

McFee, B., & Ellis, D. (2014). Analyzing Song Structure with Spectral Clustering. ISMIR, 405–410.

McFee, B., Raffel, C., Liang, D., Ellis, D. P., McVicar, M., Battenberg, E., & Nieto, O. (2015). librosa: Audio and music signal analysis in python. Proceedings of the 14th Python in Science Conference, 8, 18–25.

Mehri, S., Kumar, K., Gulrajani, I., Kumar, R., Jain, S., Sotelo, J., Courville, A., & Bengio, Y. (2016). SampleRNN: An unconditional end-to-end neural audio generation model. arXiv Preprint arXiv:1612.07837.

Mermelstein, P. (1976). Distance measures for speech recognition, psychological and instrumental. Pattern Recognition and Artificial Intelligence, 116, 374–388.

Moore, B. C. (2012). An introduction to the psychology of hearing. Brill.

Moore, B. C., & Gockel, H. (2002). Factors influencing sequential stream segregation. Acta Acustica United with Acustica, 88(3), 320–333.

Moore, R. D. (2007). Musical Instrument Recognition and Feature Extraction Techniques: A Comparative Analysis [PhD Thesis]. University of Cambridge.

Moorer, J. A. (1975). On the segmentation and analysis of continuous musical sound by digital computer. Stanford University.

Morcos, A. S., Barrett, D. G. T., Rabinowitz, N. C., & Botvinick, M. (2018). On the importance of single directions for generalization. International Conference on Learning Representations (ICLR).

Münte, T. F., Altenmüller, E., & Jäncke, L. (2002). The musician's brain as a model of neuroplasticity. Nature Reviews Neuroscience, 3(6), 473–478.

O'Hanlon, K., & Plumbley, M. D. (2014). Polyphonic piano transcription using non-negative matrix factorisation with group sparsity. 2014 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), 3112–3116.

Oord, A. van den, Dieleman, S., Zen, H., Simonyan, K., Vinyals, O., Graves, A., Kalchbrenner, N., Senior, A., & Kavukcuoglu, K. (2016). WaveNet: A Generative Model for Raw Audio.

Oppenheim, A. V., & Schafer, R. W. (1975). Digital Signal Processing Prentice-Hall. Englewood Cliffs, NJ, 19752, 26–30.

Pamuk, N. & others. (2022). Identifying Different Musical Instrument Sounds Using Fourier Analysis in LabVIEW. SAR Journal-Science and Research, 5(4), 175–182.

Pan, D., Li, X., & Zhu, D. (2021). Explaining deep neural network models with adversarial gradient integration. Thirtieth International Joint Conference on Artificial Intelligence (IJCAI).

Park, J., & Sandberg, I. W. (1991). Universal approximation using radial-basis-function networks. Neural Computation, 3(2), 246–257.

Parkhi, O., Vedaldi, A., & Zisserman, A. (2015). Deep face recognition. BMVC 2015-Proceedings of the British Machine Vision Conference 2015.

Patel, A. D. (2008). Music, Language, and the Brain. Oxford University Press.

Patil, K., Pressnitzer, D., Shamma, S., & Elhilali, M. (2012). Music in our ears: The biological bases of musical timbre perception. PLoS Computational Biology, 8(11), e1002759.

Patterson, R. D., Nimmo-Smith, I., Holdsworth, J., Rice, P., & others. (1987). An efficient auditory filterbank based on the gammatone function. A Meeting of the IOC Speech Group on Auditory Modelling at RSRE, 2(7).

Peretz, I., & Zatorre, R. J. (2003). The cognitive neuroscience of music. OUP Oxford.

Pickles, J. O. (2012). An Introduction to the Physiology of Hearing (4th ed.). Brill.

Piczak, K. J. (2015). Environmental sound classification with convolutional neural networks. 2015 IEEE 25th International Workshop on Machine Learning for Signal Processing (MLSP), 1–6.

Pierce, A. D. (2019). Acoustics: An introduction to its physical principles and applications. Springer.

Plack, C. J., & Moore, D. R. (2010). The oxford handbook of auditory science: Hearing (Vol. 3). Oxford University Press OXford.

Pons, J., Lidy, T., & Serra, X. (2017). Timbre Analysis of Music Audio Signals with Convolutional Neural Networks. Proceedings of the 23rd International Conference on Music Information Retrieval (ISMIR), 103–110.

Qi, Z., Khorram, S., & Li, F. (2019). Visualizing Deep Networks by Optimizing with Integrated Gradients. CVPR Workshops, 2, 1–4.

Qin, Z., Zhang, P., Wu, F., & Li, X. (2021). Fcanet: Frequency channel attention networks. Proceedings of the IEEE/CVF International Conference on Computer Vision, 783–792.

Rabiner, L. R., & Gold, B. (1975). Theory and Application of Digital Signal Processing:(by) Lawrence R. Rabiner (and) Bernard Gold. Prentice-Hall.

Rabiner, L. R., & Schafer, R. W. (1978). Digital Processing of Speech Signals. Proceedings of the IEEE, 66(4), 433–456.

Radio, P. (2015). The Music Genome Project. Disponible En.

Reitermanova, Z. & others. (2010). Data splitting. WDS, 10, 31–36.

Rifkin, R., & Klautau, A. (2004). In defense of one-vs-all classification. The Journal of Machine Learning Research, 5, 101–141.

Roberts, R. A., & Mullis, C. T. (1987). Digital signal processing. Addison-Wesley Longman Publishing Co., Inc.

Rosenblatt, F. (1958). The perceptron: A probabilistic model for information storage and organization in the brain. Psychological Review, 65(6), 386–408.

Rossi, L., & Girolami, G. (2001). Instantaneous frequency and short term Fourier transforms: Application to piano sounds. The Journal of the Acoustical Society of America, 110(5), 2412–2420.

Rubner, Y., Tomasi, C., & Guibas, L. J. (2000). The Earth Mover's Distance as a Metric for Image Retrieval. International Journal of Computer Vision, 40(2), 99–121.

Rumelhart, D. E., Hinton, G. E., & Williams, R. J. (1986). Learning representations by back-propagating errors. Nature, 323(6088), 533–536.

Saito, T., & Rehmsmeier, M. (2015). The precision-recall plot is more informative than the ROC plot when evaluating binary classifiers on imbalanced datasets. PLoS ONE, 10(3), e0118432. https://doi.org/10.1371/journal.pone.0118432

Sattarzadeh, S., Sudhakar, M., Plataniotis, K. N., Jang, J., Jeong, Y., & Kim, H. (2021). Integrated grad-cam: Sensitivity-aware visual explanation of deep convolutional networks via integrated gradient-based scoring. ICASSP 2021-2021 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), 1775–1779.

Schafer, R. W., & Rabiner, L. R. (1973). A digital signal processing approach to interpolation. Proceedings of the IEEE, 61(6), 692–702.

Schedl, M., Gómez, E., Urbano, J., & others. (2014). Music information retrieval: Recent developments and applications. Foundations and Trends® in Information Retrieval, 8(2–3), 127–261.

Schlüter, J., & Grill, T. (2015). Exploring Data Augmentation for Improved Singing Voice Detection with Neural Networks. 16th International Society for Music Information Retrieval Conference (ISMIR), 121–126.

Schlüter, J., & Gutenbrunner, G. (2022). Efficientleaf: A faster learnable audio frontend of questionable use. 2022 30th European Signal Processing Conference (EUSIPCO), 205–208.

Schmid, F., Koutini, K., & Widmer, G. (2024). Dynamic Convolutional Neural Networks as Efficient Pre-trained Audio Models. IEEE/ACM Transactions on Audio, Speech, and Language Processing.

Schörkhuber, C., & Klapuri, A. (2010). Constant-Q transform toolbox for music processing. Proceedings of the 7th Sound and Music Computing Conference, 3–64.

scikit-learn developers. (2019). Scikit-Learn API Document. https://scikit-learn.org/stable/modules/generated/sklearn.metrics.classification_report.html

Shamma, S. A., Elhilali, M., & Micheyl, C. (2011). Temporal coherence and attention in auditory scene analysis. Trends in Neurosciences, 34(3), 114–123.

Shinn-Cunningham, B. G. (2008). Object-based auditory and visual attention. Trends in Cognitive Sciences, 12(5), 182–186.

Simmermacher, C., Deng, D., & Cranefield, S. (2006). Feature analysis and classification of classical musical instruments: An empirical study. Advances in Data Mining. Applications in Medicine, Web Mining, Marketing, Image and Signal Mining: 6th Industrial Conference on Data Mining, ICDM 2006, Leipzig, Germany, July 14-15, 2006. Proceedings 6, 444–458.

Singh, V., Pencina, M., Einstein, A. J., Liang, J. X., Berman, D. S., & Slomka, P. (2021). Impact of train/test sample regimen on performance estimate stability of machine learning in cardiovascular imaging. Scientific Reports, 11(1), 14490.

Slaney, M. (1998). Auditory Toolbox Version 2. Interval Research Corporation, Palo Alto, CA.

Slaney, M. & others. (1993). An efficient implementation of the Patterson-Holdsworth auditory filter bank. Apple Computer, Perception Group, Tech. Rep, 35(8).

SoundHound Inc. (2023). SoundHound. https://www.soundhound.com/

Spotify AB. (2023). Spotify. https://www.spotify.com/

Spyromitros Xioufis, E., Tsoumakas, G., & Vlahavas, I. (2011). Multi-label learning approaches for music instrument recognition. Foundations of Intelligent Systems: 19th International Symposium, ISMIS 2011, Warsaw, Poland, June 28-30, 2011. Proceedings 19, 734–743.

Stevens, S. S., & Volkmann, J. (1940). The relation of pitch to frequency: A revised scale. The American Journal of Psychology, 53(3), 329–353.

Stevens, S. S., Volkmann, J., & Newman, E. B. (1937). A scale for the measurement of the psychological magnitude pitch. The Journal of the Acoustical Society of America, 8(3), 185–190.

Stöter, F.-R., Schoeffler, M., Edler, B., & Herre, J. (2013). Human ability of counting the number of instruments in polyphonic music. Proceedings of Meetings on Acoustics, 19(1).

Strang, G., & Nguyen, T. (1996). Wavelets and Filter Banks. Wellesley-Cambridge Press.

Stutz, D., Hein, M., & Schiele, B. (2020). Confidence-calibrated adversarial training: Generalizing to unseen attacks. International Conference on Machine Learning, 9155–9166.

Su, Y., Zhang, K., Wang, J., Zhou, D., & Madani, K. (2020). Performance analysis of multiple aggregated acoustic features for environment sound classification. Applied Acoustics, 158, 107050.

Taenzer, M., Mimilakis, S. I., & Abeßer, J. (2021). Deep Learning-Based Music Instrument Recognition: Exploring Learned Feature Representations. International Symposium on Computer Music Multidisciplinary Research, 32–46.

Takuya, F. (1999). Realtime chord recognition of musical sound: Asystem using common lisp music. Proceedings of the International Computer Music Conference 1999, Beijing.

Tan, Z.-Q., Wong, C.-Y., & Manimaran, B. (2023). A Comparative Study of Deep Learning Models for Musical Instrument Identification. Journal of Intelligent & Fuzzy Systems, 45(3), 1287–1295. https://doi.org/10.3233/JIFS-189447

Thickstun, J., Harchaoui, Z., & Kakade, S. (2016). Learning features of music from scratch. arXiv Preprint arXiv:1611.09827.

Thomke, S., & Matters, E. (2003). Unlocking the Potential of New Technologies for Innovation. Harvard Business School Press, Boston, MA Experimentation Matters.

Thornburg, H., Leistikow, R. J., & Berger, J. (2007). Melody extraction and musical onset detection via probabilistic models of framewise STFT peak data. IEEE Transactions on Audio, Speech, and Language Processing, 15(4), 1257–1272.

Tindale, A. R., Kapur, A., Tzanetakis, G., Driessen, P., & Schloss, A. (2005). A comparison of sensor strategies for capturing percussive gestures. Proceedings of the 2005 Conference on New Interfaces for Musical Expression, 200–203.

Treisman, A. M., & Gelade, G. (1980). A feature-integration theory of attention. Cognitive Psychology, 12(1), 97–136.

Tseng, Y.-H., & Yeh, Y.-R. (2021). Multi-attention neural networks for automatic music tagging. IEEE/ACM Transactions on Audio, Speech, and Language Processing, 29, 2207–2219.

Tulshan, A. S., & Dhage, S. N. (2019). Survey on virtual assistant: Google assistant, siri, cortana, alexa. Advances in Signal Processing and Intelligent Recognition Systems: 4th International Symposium SIRS 2018, Bangalore, India, September 19–22, 2018, Revised Selected Papers 4, 190–201.

Tzanetakis, G., & Cook, P. (2002). Musical genre classification of audio signals. IEEE Transactions on Speech and Audio Processing, 10(5), 293–302.

Tzinis, E., Wang, Z., Sridhar, V. K. I., & Smaragdis, P. (2019). Improving universal sound separation using sound classification. arXiv Preprint arXiv:1911.03040.

Van Opstal, J. (2016). The auditory system and human sound-localization behavior. Academic Press.

Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., Kaiser, Ł., & Polosukhin, I. (2017). Attention is all you need. Advances in Neural Information Processing Systems, 30.

Veisi, H., & Sameti, H. (2013). Speech enhancement using hidden Markov models in Mel-frequency domain. Speech Communication, 55(2), 205–220.

Vergés Franch, E. (2021). Automatic guitar performance assessment: Datasets, algorithms and metrics.

Vincent, E., Gribonval, R., & Févotte, C. (2006). Performance measurement in blind audio source separation. IEEE Transactions on Audio, Speech, and Language Processing, 14(4), 1462–1469.

Wang, Q., Wu, B., Zhu, P., Li, P., Zuo, W., & Hu, Q. (2020). ECA-Net: Efficient channel attention for deep convolutional neural networks. Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 11534–11542.

Warren, R. M. (2013). Auditory perception: A new synthesis (Vol. 109). Elsevier.

Watcharasupat, K., Gururani, S., & Lerch, A. (2020). Visual attention for musical instrument recognition. arXiv Preprint arXiv:2006.09640.

Wegener, S., Haller, M., Burred, J. J., Sikora, T., Essid, S., & Richard, G. (2008). On the robustness of audio features for musical instrument classification. 2008 16th European Signal Processing Conference, 1–5.

Weiß, C., & Habryka, J. (2014). Chroma-based scale matching for audio tonality analysis. Proceedings of the 9th Conference on Interdisciplinary Musicology (CIM), 168–173.

Wertheimer, M. (1938). Laws of organization in perceptual forms.

Wise, A., Maida, A. S., & Kumar, A. (2024). Attention Augmented CNNs for Musical Instrument Identification. European Signal Processing Conference (EUSIPCO), 376–380.

Wold, E., Blum, T., Keislar, D., & Wheaten, J. (1996). Content-based classification, search, and retrieval of audio. IEEE Multimedia, 3(3), 27–36.

Won, M., Chun, S., & Serra, X. (2019). Toward interpretable music tagging with self-attention. arXiv Preprint arXiv:1906.04972.

Xiao, H., Liu, D., Chen, K., & Zhu, M. (2022). AMResNet: An automatic recognition model of bird sounds in real environment. Applied Acoustics, 201, 109121.

Xie, C., Zhu, H., & Fei, Y. (2022). Deep coordinate attention network for single image super-resolution. IET Image Processing, 16(1), 273–284.

Yang, X., Wang, K., & Shamma, S. A. (1992). Auditory representations of acoustic signals. IEEE Transactions on Information Theory, 38(2), 824–839.

Yousician Ltd. (2023). Yousician. https://www.yousician.com/

Zatorre, R. J. (2003). Neural specializations for tonal processing. Annals of the New York Academy of Sciences, 999(1), 193–199.

Zeghidour, N., Teboul, O., Quitry, F. de C., & Tagliasacchi, M. (2021). LEAF: A learnable frontend for audio classification. arXiv Preprint arXiv:2101.08596.

Zeiler, M. D., & Fergus, R. (2014a). Visualizing and understanding convolutional networks. European Conference on Computer Vision, 818–833.

Zeiler, M. D., & Fergus, R. (2014b). Visualizing and understanding convolutional networks. Computer Vision–ECCV 2014: 13th European Conference, Zurich, Switzerland, September 6-12, 2014, Proceedings, Part I 13, 818–833.

Zha, M., Qian, W., Yi, W., & Hua, J. (2021). A lightweight YOLOv4-Based forestry pest detection method using coordinate attention and feature fusion. Entropy, 23(12), 1587.

Zhang, M.-L., & Zhou, Z.-H. (2013). A review on multi-label learning algorithms. IEEE Transactions on Knowledge and Data Engineering, 26(8), 1819–1837.

Zimmerman, J., Forlizzi, J., & Evenson, S. (2007). Research through design as a method for interaction design research in HCI. Proceedings of the SIGCHI Conference on Human Factors in Computing Systems, 493–502.

Zwicker, E., & Fastl, H. (1990). Psychoacoustics: Facts and Models. Springer.

Appendix

Appendix 1: Pseudocode of experiment 1:

According to the flow chart, the simplified code are as follows,

```
import librosa
import numpy as np
import tensorflow as tf
from tensorflow.keras.models import Sequential, load_model
from tensorflow.keras.layers import Conv2D, Flatten, Dense, MaxPooling2D

def create_cnn_model(input_shape):
    # Function to create a CNN model for binary classification
    # Sequential model: A linear stack of layers
    model = Sequential([
        # Conv2D layer: Applies a 2D convolution operation
        # 32: Number of filters (kernels) to use in this layer
        # (3, 3): Kernel size - the height and width of the convolution window
         # activation='relu': Rectified Linear Unit activation function, common choice for hidden layers
        # input_shape: Shape of the input data (height, width, channels)
        Conv2D(32, (3, 3), activation='relu', input_shape=input_shape),

        # MaxPooling2D layer: Applies a 2D max pooling operation
        # (2, 2): Pool size - factors by which to downscale in both dimensions
        MaxPooling2D((2, 2)),

        # Flatten layer: Flattens the input without affecting the batch size
        Flatten(),

        # Dense layer: Regular densely-connected NN layer
        # 64: Number of neurons in the layer
        # activation='relu': Activation function used
        Dense(64, activation='relu'),

        # Output layer with single neuron (binary classification)
```

```
        # activation='sigmoid': Sigmoid activation function, common choice for binary classification
        Dense(1, activation='sigmoid')
    ])

return model

# Load an audio file and convert to a spectrogram
# 'audio_file' is the path to our audio file (e.g., a .wav file)
audio_file = 'path/to/our/audio/file.wav'

# 'y' is the audio time series data, 'sr' is the sampling rate of 'y'
y, sr = librosa.load(audio_file)

# Compute the Short-Time Fourier Transform (STFT) of the audio signal 'y'
# 'n_fft' is the number of data points used in each block for the FFT (affects frequency resolution)
# 'hop_length' is the number of samples between successive frames (affects time resolution)
spectrogram = librosa.stft(y, n_fft=2048, hop_length=512)

# Convert the amplitude spectrogram (magnitude) to a dB-scaled spectrogram
# This is for better visualization and understanding of the spectrogram as it scales the values
spectrogram_db = librosa.amplitude_to_db(np.abs(spectrogram))

# Expand the dimensions of 'spectrogram_db' to fit into CNN
# CNNs expect a certain shape of input, often including a channel dimension, hence the use of np.expand_dims
spectrogram_cnn = np.expand_dims(spectrogram_db, axis=-1)

# 'spectrogram_db' is a 2D array representing the spectrogram (time vs frequency)
# Each column of 'spectrogram_db' is essentially a single 'frame' or 'slice' of the spectrogram
# So, 'spectrogram_cnn' contains multiple frames/slices, which can be seen as multiple spectrograms over time
# Paths to the pre-trained model weights for each instrument
```

```
model_paths = {
    'piano': 'path/to/piano_model_weights.h5',
    'violin': 'path/to/violin_model_weights.h5',
    'trumpet': 'path/to/trumpet_model_weights.h5'
    # And more instruments when we trained more models
}

# Load and predict with each model
predictions = {}
for instrument, path in model_paths.items():
    model = create_cnn_model(spectrogram_cnn.shape[0:3])
    model.load_weights(path)
    predictions[instrument] = model.predict(spectrogram_cnn)

# post-processing: 1. applying confidence threshold
confidence_threshold = 0.5
filtered_predictions = {instr: (pred > confidence_threshold) for instr, pred in predictions.items()}

# Assuming filtered_predictions is a dictionary containing predictions for each instrument
# Example: {'piano': piano_predictions, 'violin': violin_predictions, ...}

# Post-processing: 2. Temporal Smoothing
window_size = 5  # Window size for the moving average
smoothed_predictions = {}
for instrument, preds in filtered_predictions.items():
        smoothed = np.convolve(preds.flatten(), np.ones(window_size)/window_size, mode='valid')
    smoothed_predictions[instrument] = smoothed

# Resolving Conflicts Between Classifiers
# If two classifiers strongly disagree, the prediction is set to uncertain (0)
conflict_threshold = 0.2  # Threshold for strong disagreement
for i, instrument_i in enumerate(model_paths.keys()):
    for j, instrument_j in enumerate(model_paths.keys()):
        if i < j:  # To avoid repeating comparisons
```

```
                                diff = np.abs(smoothed_predictions[instrument_i] - smoothed_predictions[instrument_j])
            conflict_indices = np.where(diff > conflict_threshold)[0]
            for index in conflict_indices:
                # Set predictions to 0 (uncertain) in case of conflict
                smoothed_predictions[instrument_i][index] = 0
                smoothed_predictions[instrument_j][index] = 0

# Output
for instrument, preds in smoothed_predictions.items():
    print(f"Instrument: {instrument}, Predictions: {preds}")

# Further contextual analysis can be added

# Output final predictions
final_predictions = filtered_predictions
```

We developer our CNN model based on TensorFlow.Keras. The code are as follows,

```
import tensorflow as tf
from tensorflow.keras.models import Sequential
from tensorflow.keras.layers import Conv2D, MaxPooling2D, Flatten, Dense

def create_instrument_classifier(input_shape):
    model = Sequential([
        Conv2D(32, (3, 3), activation='relu', input_shape=input_shape),
        MaxPooling2D((2, 2)),
        Flatten(),
        Dense(64, activation='relu'),
        Dense(1, activation='sigmoid')  # Binary classification output
    ])
    model.compile(optimizer='adam', loss='binary_crossentropy', metrics=['accuracy'])
    return model
```

The we train our model by the 4 class data we've been collected, and the sample code are as follows,

```
# Example data loading function
def load_data_for_instrument(instrument_name):
    # This function should load our spectrogram data and labels for the given instrument
    # For instance, it could load data from files and return it as NumPy arrays
    # Return features (X) and labels (y)
                                    y,       sr       =       librosa.load(audio_file)
    spectrogram = librosa.stft(y, n_fft=2048, hop_length=512)
                    spectrogram_db    =    librosa.amplitude_to_db(np.abs(spectrogram))
    spectrogram_cnn = np.expand_dims(spectrogram_db, axis=-1)
    # Dictionary to hold our classifiers
classifiers = {}

# List of instruments
instruments = ['piano', 'violin', 'flute', 'trumpet']

# input shape is determined by our spectrogram dimensions
input_shape = (128, 128, 1)  # Example shape
for instrument in instruments:
    # Load data specific to each instrument
    X, y = load_data_for_instrument(instrument)
    # Create a binary classifier for the current instrument
    model = create_instrument_classifier(input_shape)

    # Train the model (consider splitting X and y into training and validation sets)
    model.fit(X, y, epochs=200, validation_split=0.3)

    # Store the trained model
classifiers[instrument] = model
```

In our multi-label classifiction, the final_predictions could be a dictionary where each key is an instrument, and the value is an array indicating the likelihood of that instrument being present in each time frame of the audio.

```
final_predictions = {
    'piano': [0.1, 0.7, 0.9, ...],  # probabilities for each time frame
    'violin': [0.05, 0.2, 0.3, ...],
    'trumpet': [0.8, 0.1, 0.05, ...]
    # ... more instruments
}
```

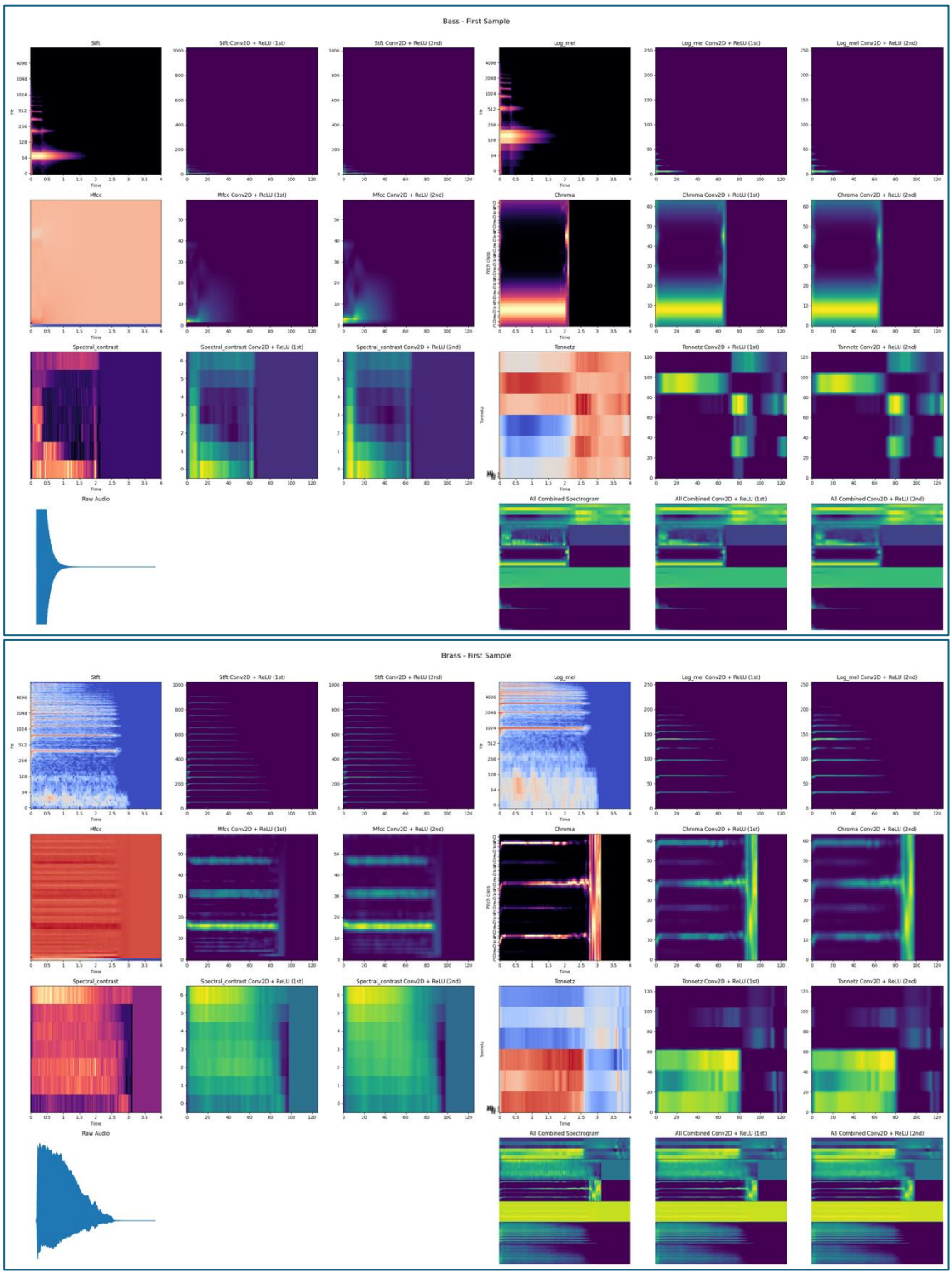
Appendix 2: Pseudocode of experiment 2:

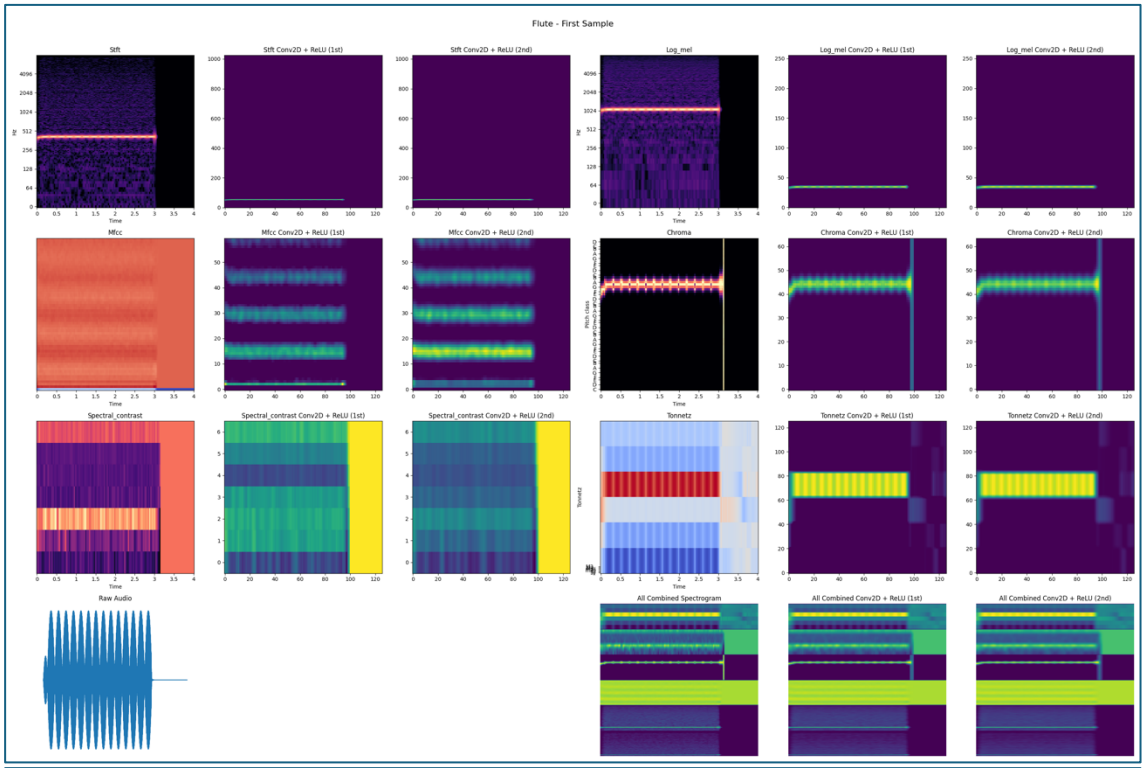
According to the flow chart, the simplified code are as follows,

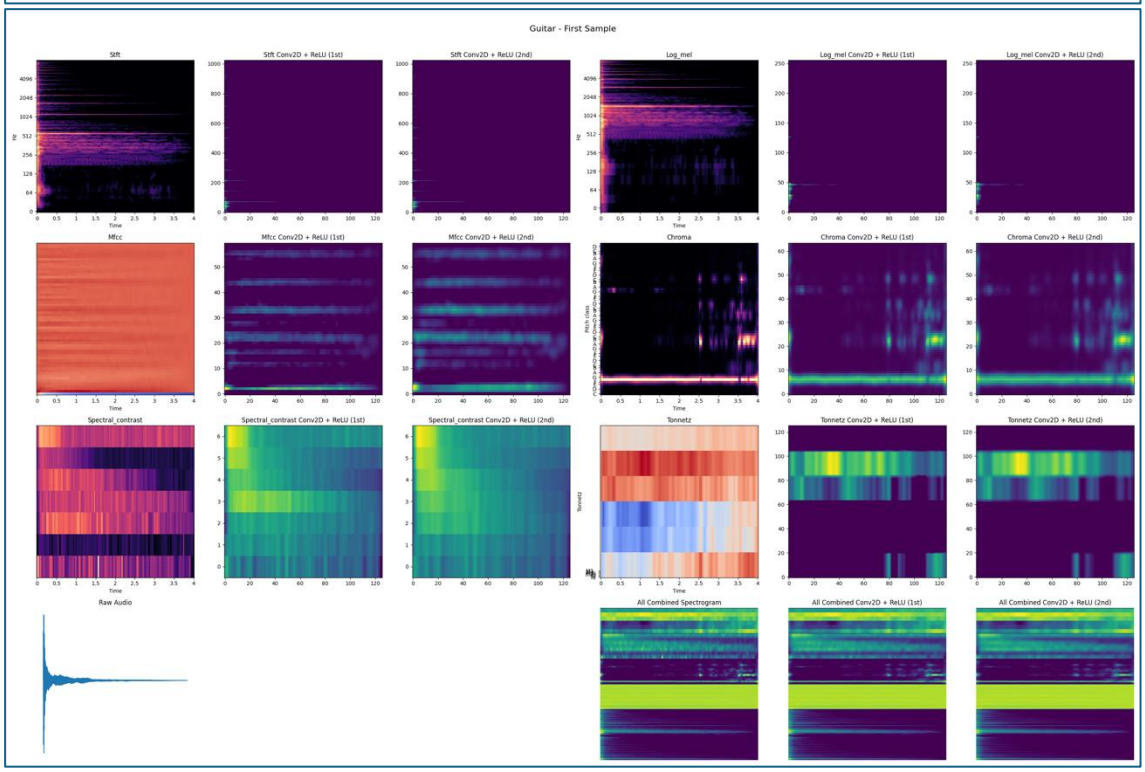
```
import tensorflow as tf
from tensorflow.keras.models import Sequential
from tensorflow.keras.layers import Conv2D, MaxPooling2D, Flatten, Dense, Dropout, BatchNormalization

def create_instrument_classifier(input_shape):
    model = Sequential([
        Conv2D(64, (3, 3), activation='relu', padding='same', input_shape=input_shape),
        BatchNormalization(),
        MaxPooling2D((2, 2)),
        Conv2D(128, (3, 3), activation='relu', padding='same'),
        BatchNormalization(),
        MaxPooling2D((2, 2)),
        Flatten(),
        Dense(128, activation='relu'),
        Dropout(0.5),
        Dense(1, activation='sigmoid')  # Binary classification output
    ])
```

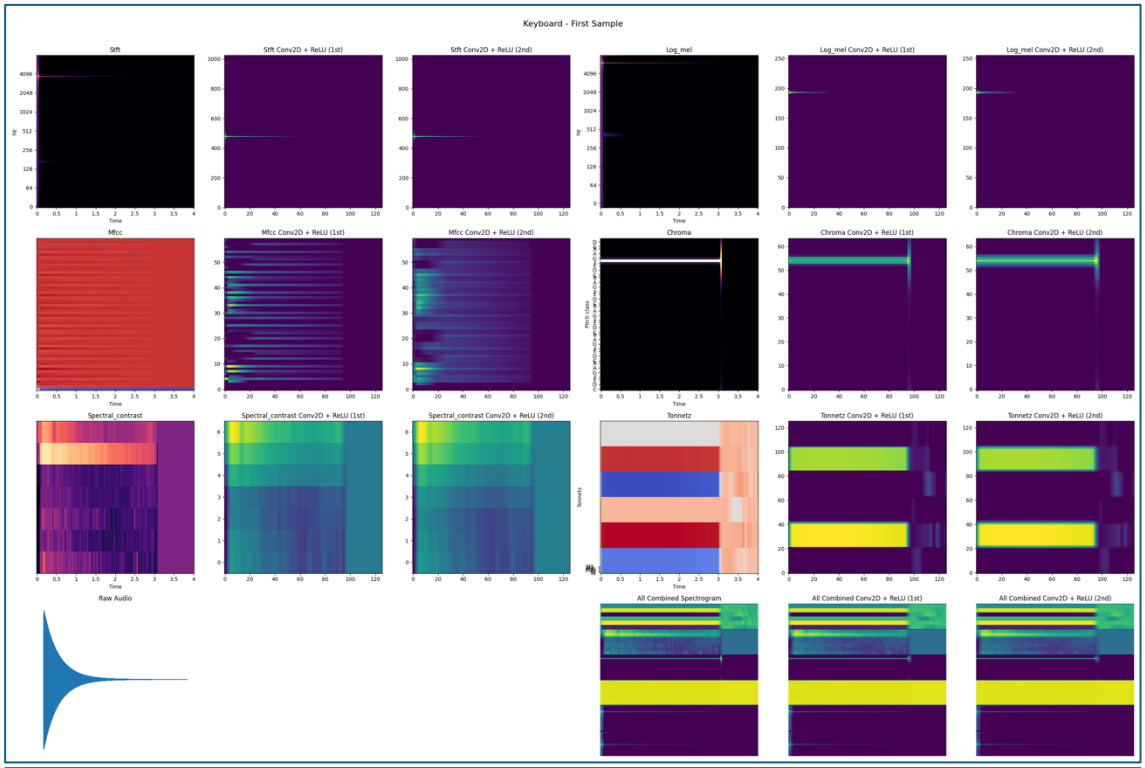
```
    model.compile(optimizer='adam', loss='binary_crossentropy', metrics=['accuracy'])
    return model

# Function to load data for a specific instrument family
def load_data_for_instrument(instrument_name):
    # Implementation to load spectrogram data and labels for the given instrument
    pass

instruments = ['Bass', 'Brass', 'Flute', 'Guitar', 'Keyboard', 'Mallet', 'Organ', 'Reed',
'String', 'Synth Lead', 'Vocal']
input_shape = (128, 128, 1)  # Adjusted based on spectrogram dimensions

classifiers = {}

for instrument in instruments:
    X, y = load_data_for_instrument(instrument)  # Load data
    model = create_instrument_classifier(input_shape)  # Initialize the model
    model.fit(X, y, epochs=1000, validation_split=0.3)  # Train the model
    classifiers[instrument] = model  # Store the trained model
```

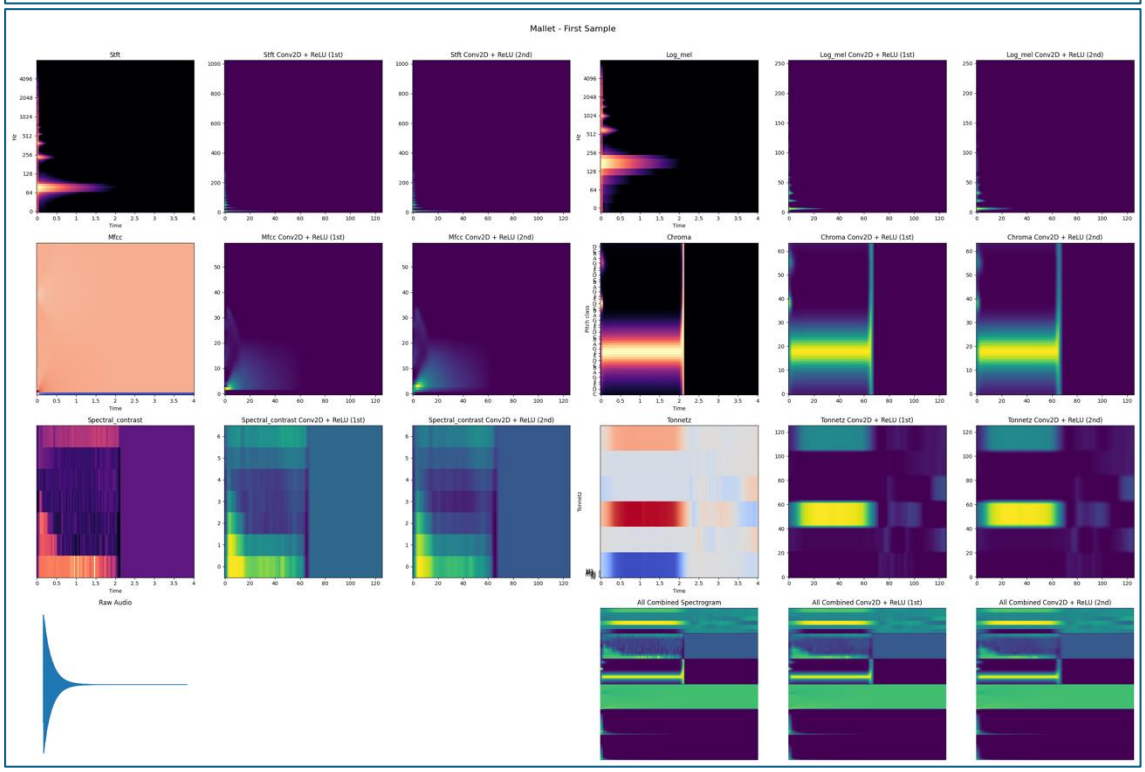
Appendix 3: 10 Instruments' STFT, log-mel, MFCC, Chroma, Spectral Contrast and Tonnetz. And its correspondent feature maps.

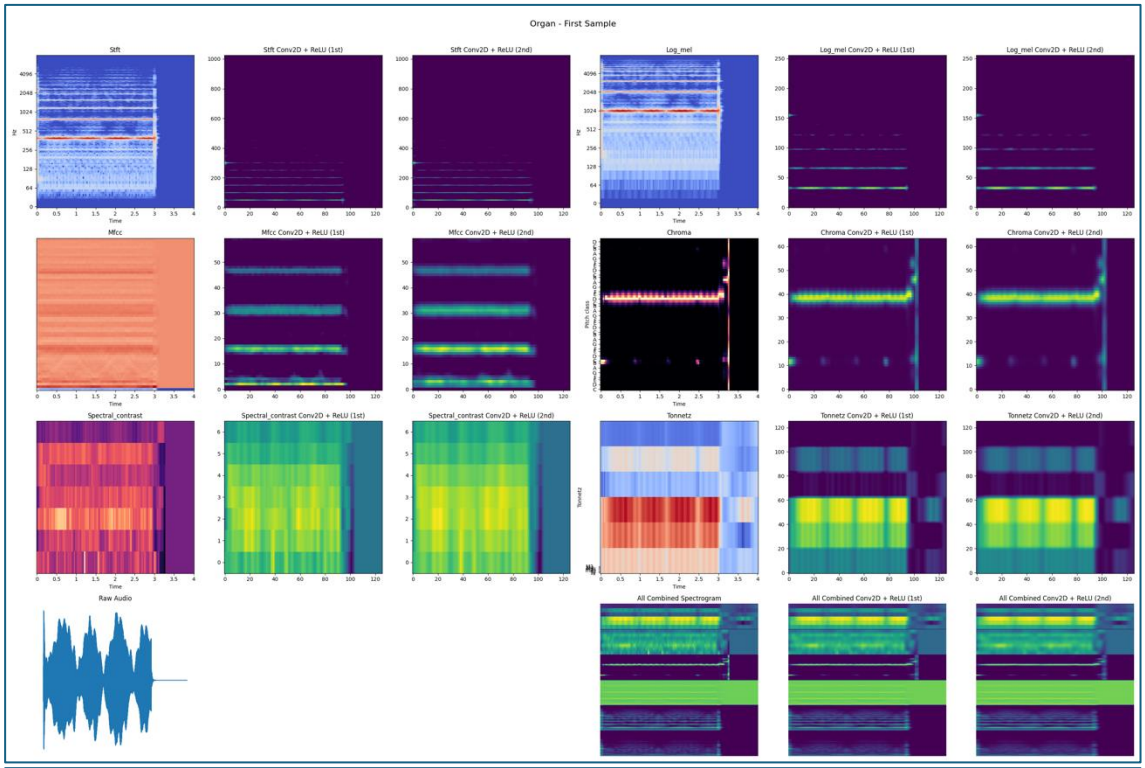


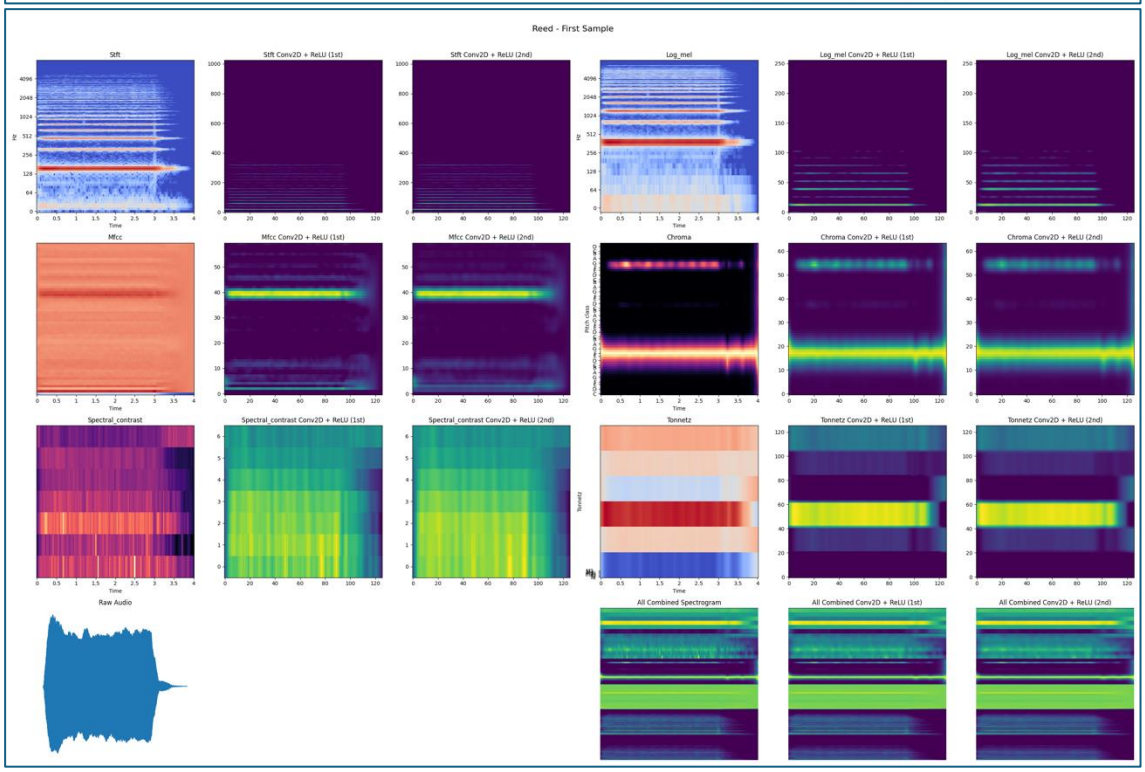

Flute - First Sample
Stft
Stft Conv2D + ReLU (1st)
Stft Conv2D + ReLU (2nd)
Log_mel
Log_mel Conv2D + ReLU (1st)
Log_mel Conv2D + ReLU (2nd)
Mfcc
Mfcc Conv2D + ReLU (1st)
Mfcc Conv2D + ReLU (2nd)
Chroma
Chroma Conv2D + ReLU (1st)
Chroma Conv2D + ReLU (2nd)
Spectral_contrast
Spectral_contrast Conv2D + ReLU (1st)
Spectral_contrast Conv2D + ReLU (2nd)
Tonnetz
Tonnetz Conv2D + ReLU (1st)
Tonnetz Conv2D + ReLU (2nd)
Raw Audio
All Combined Spectrogram
All Combined Conv2D + ReLU (1st)
All Combined Conv2D + ReLU (2nd)

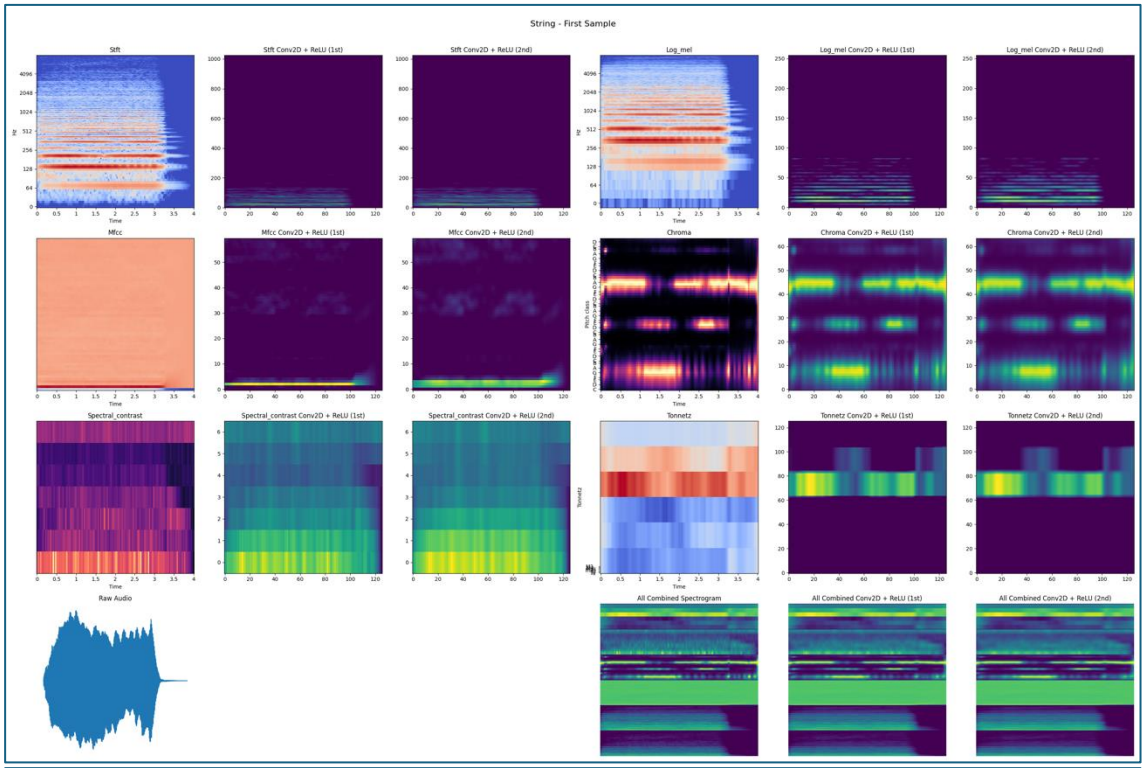

Guitar - First Sample
Stft
Stft Conv2D + ReLU (1st)
Stft Conv2D + ReLU (2nd)
Log_mel
Log_mel Conv2D + ReLU (1st)
Log_mel Conv2D + ReLU (2nd)
Mfcc
Mfcc Conv2D + ReLU (1st)
Mfcc Conv2D + ReLU (2nd)
Chroma
Chroma Conv2D + ReLU (1st)
Chroma Conv2D + ReLU (2nd)
Spectral_contrast
Spectral_contrast Conv2D + ReLU (1st)
Spectral_contrast Conv2D + ReLU (2nd)
Tonnetz
Tonnetz Conv2D + ReLU (1st)
Tonnetz Conv2D + ReLU (2nd)
Raw Audio
All Combined Spectrogram
All Combined Conv2D + ReLU (1st)
All Combined Conv2D + ReLU (2nd)

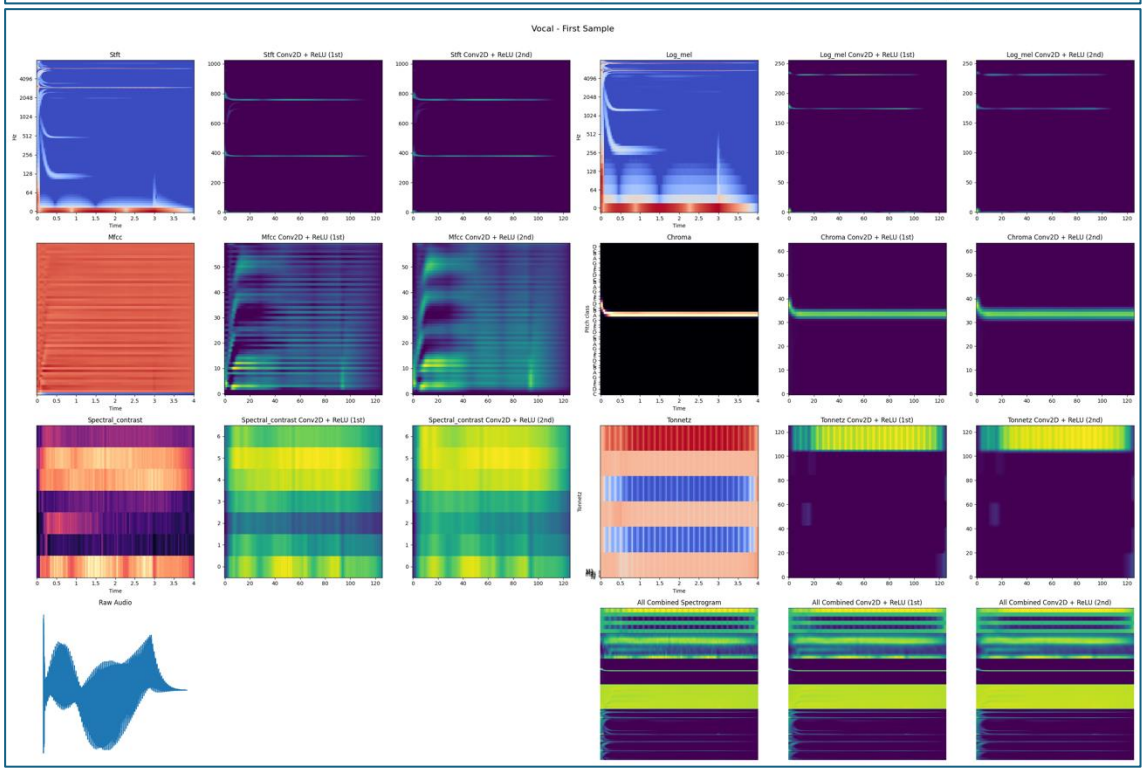

Keyboard - First Sample
Stft
Stft Conv2D + ReLU (1st)
Stft Conv2D + ReLU (2nd)
Log_mel
Log_mel Conv2D + ReLU (1st)
Log_mel Conv2D + ReLU (2nd)
Mfcc
Mfcc Conv2D + ReLU (1st)
Mfcc Conv2D + ReLU (2nd)
Chroma
Chroma Conv2D + ReLU (1st)
Chroma Conv2D + ReLU (2nd)
Spectral_contrast
Spectral_contrast Conv2D + ReLU (1st)
Spectral_contrast Conv2D + ReLU (2nd)
Tonnetz
Tonnetz Conv2D + ReLU (1st)
Tonnetz Conv2D + ReLU (2nd)
Raw Audio
All Combined Spectrogram
All Combined Conv2D + ReLU (1st)
All Combined Conv2D + ReLU (2nd)


Mallet - First Sample
Stft
Stft Conv2D + ReLU (1st)
Stft Conv2D + ReLU (2nd)
Log_mel
Log_mel Conv2D + ReLU (1st)
Log_mel Conv2D + ReLU (2nd)
Mfcc
Mfcc Conv2D + ReLU (1st)
Mfcc Conv2D + ReLU (2nd)
Chroma
Chroma Conv2D + ReLU (1st)
Chroma Conv2D + ReLU (2nd)
Spectral_contrast
Spectral_contrast Conv2D + ReLU (1st)
Spectral_contrast Conv2D + ReLU (2nd)
Tonnetz
Tonnetz Conv2D + ReLU (1st)
Tonnetz Conv2D + ReLU (2nd)
Raw Audio
All Combined Spectrogram
All Combined Conv2D + ReLU (1st)
All Combined Conv2D + ReLU (2nd)


Organ - First Sample
Stft
Stft Conv2D + ReLU (1st)
Stft Conv2D + ReLU (2nd)
Log_mel
Log_mel Conv2D + ReLU (1st)
Log_mel Conv2D + ReLU (2nd)
Mfcc
Mfcc Conv2D + ReLU (1st)
Mfcc Conv2D + ReLU (2nd)
Chroma
Chroma Conv2D + ReLU (1st)
Chroma Conv2D + ReLU (2nd)
Spectral_contrast
Spectral_contrast Conv2D + ReLU (1st)
Spectral_contrast Conv2D + ReLU (2nd)
Tonnetz
Tonnetz Conv2D + ReLU (1st)
Tonnetz Conv2D + ReLU (2nd)
Raw Audio
All Combined Spectrogram
All Combined Conv2D + ReLU (1st)
All Combined Conv2D + ReLU (2nd)


Reed - First Sample
Stft
Stft Conv2D + ReLU (1st)
Stft Conv2D + ReLU (2nd)
Log_mel
Log_mel Conv2D + ReLU (1st)
Log_mel Conv2D + ReLU (2nd)
Mfcc
Mfcc Conv2D + ReLU (1st)
Mfcc Conv2D + ReLU (2nd)
Chroma
Chroma Conv2D + ReLU (1st)
Chroma Conv2D + ReLU (2nd)
Spectral_contrast
Spectral_contrast Conv2D + ReLU (1st)
Spectral_contrast Conv2D + ReLU (2nd)
Tonnetz
Tonnetz Conv2D + ReLU (1st)
Tonnetz Conv2D + ReLU (2nd)
Raw Audio
All Combined Spectrogram
All Combined Conv2D + ReLU (1st)
All Combined Conv2D + ReLU (2nd)


String - First Sample
Stft
Stft Conv2D + ReLU (1st)
Stft Conv2D + ReLU (2nd)
Log_mel
Log_mel Conv2D + ReLU (1st)
Log_mel Conv2D + ReLU (2nd)
Mfcc
Mfcc Conv2D + ReLU (1st)
Mfcc Conv2D + ReLU (2nd)
Chroma
Chroma Conv2D + ReLU (1st)
Chroma Conv2D + ReLU (2nd)
Spectral_contrast
Spectral_contrast Conv2D + ReLU (1st)
Spectral_contrast Conv2D + ReLU (2nd)
Tonnetz
Tonnetz Conv2D + ReLU (1st)
Tonnetz Conv2D + ReLU (2nd)
Raw Audio
All Combined Spectrogram
All Combined Conv2D + ReLU (1st)
All Combined Conv2D + ReLU (2nd)


Vocal - First Sample
Stft
Stft Conv2D + ReLU (1st)
Stft Conv2D + ReLU (2nd)
Log_mel
Log_mel Conv2D + ReLU (1st)
Log_mel Conv2D + ReLU (2nd)
Mfcc
Mfcc Conv2D + ReLU (1st)
Mfcc Conv2D + ReLU (2nd)
Chroma
Chroma Conv2D + ReLU (1st)
Chroma Conv2D + ReLU (2nd)
Spectral_contrast
Spectral_contrast Conv2D + ReLU (1st)
Spectral_contrast Conv2D + ReLU (2nd)
Tonnetz
Tonnetz Conv2D + ReLU (1st)
Tonnetz Conv2D + ReLU (2nd)
Raw Audio
All Combined Spectrogram
All Combined Conv2D + ReLU (1st)
All Combined Conv2D + ReLU (2nd)

Appendix 4: Attention Mechanism – Channel Attention

This mechanism is implemented using two dense layers that generate an attention map by computing weights for each channel. The input feature is first averaged and max-pooled, then passed through shared dense layers to produce the attention map, which is then multiplied with the original input feature to enhance the informative channels

```
def channel_attention(input_feature, ratio=8):
    channel = input_feature.shape[-1]

    shared_layer_one = Dense(channel // ratio,
                             activation='relu',
                             kernel_initializer='he_normal',
                             use_bias=True,
                             bias_initializer='zeros')
    shared_layer_two = Dense(channel,
                             kernel_initializer='he_normal',
                             use_bias=True,
                             bias_initializer='zeros')

    avg_pool = GlobalAveragePooling2D()(input_feature)
    avg_pool = Reshape((1, 1, channel))(avg_pool)
    avg_pool = shared_layer_one(avg_pool)
    avg_pool = shared_layer_two(avg_pool)

    max_pool = GlobalMaxPooling2D()(input_feature)
    max_pool = Reshape((1, 1, channel))(max_pool)
    max_pool = shared_layer_one(max_pool)
    max_pool = shared_layer_two(max_pool)

    cbam_feature = Add()([avg_pool, max_pool])
    cbam_feature = Activation('sigmoid')(cbam_feature)

    attention_map = Multiply()([input_feature, cbam_feature])
    return attention_map, cbam_feature
```

Appendix 5: Attention Mechanism – Coordinates Attention

This mechanism divides the input feature into two parts, capturing spatial information along the height and width dimensions separately. This helps the model focus on essential spatial regions and long-range dependencies.

```
def coordinate_attention(inputs, reduction_ratio=8):
    def h_swish(x):
        return x * tf.nn.relu6(x + 3) / 6

    _, h, w, c = inputs.shape

    h_avg_pool = tf.reduce_mean(inputs, axis=2, keepdims=True)
    w_avg_pool = tf.reduce_mean(inputs, axis=1, keepdims=True)

    h_avg_pool = tf.transpose(h_avg_pool, [0, 2, 1, 3])
    h_avg_pool = Conv2D(filters=c // reduction_ratio, kernel_size=1, activation=h_swish)(h_avg_pool)
    h_avg_pool = Conv2D(filters=c, kernel_size=1, activation='sigmoid')(h_avg_pool)
    h_avg_pool = tf.transpose(h_avg_pool, [0, 2, 1, 3])

    w_avg_pool = Conv2D(filters=c // reduction_ratio, kernel_size=1, activation=h_swish)(w_avg_pool)
    w_avg_pool = Conv2D(filters=c, kernel_size=1, activation='sigmoid')(w_avg_pool)

    attention_map = inputs * h_avg_pool * w_avg_pool
    return attention_map, h_avg_pool, w_avg_pool
```

Appendix 6: Model Structure

This appendix details the architecture of a hierarchical residual network designed for multi-label classification. The model integrates several layers of residual blocks, each followed by coordinate attention mechanisms applied at different stages—early, mid, and late—to enhance feature extraction. Channel attention is applied before the final classification layers to further refine the learned representations. The network culminates in a dense layer with a sigmoid activation function, which is well-suited for multi-label classification tasks. The model is optimized using the Adam optimizer and employs binary cross-entropy as the loss function, aiming to achieve accurate classification performance across multiple classes.

```
def create_training_model(input_shape, num_classes):
    inputs = Input(shape=input_shape)

    x = residual_block(inputs, 32)
    x = MaxPooling2D(pool_size=(2, 2))(x)
     # print(f"Shape after MaxPooling2D: {x.shape}")

    x, _, _ = coordinate_attention(x) # early attention

    x = residual_block(x, 64)
    x = MaxPooling2D(pool_size=(2, 2))(x)
    # print(f"Shape after MaxPooling2D: {x.shape}")

    x, _, _ = coordinate_attention(x) # mid attention

    x = residual_block(x, 128)
    x = MaxPooling2D(pool_size=(2, 2))(x)
     # print(f"Shape after MaxPooling2D: {x.shape}")

    x = residual_block(x, 256)
    x = MaxPooling2D(pool_size=(2, 2))(x)
     # print(f"Shape after MaxPooling2D: {x.shape}")

    x, _ = channel_attention(x)
     # print(f"Shape after channel_attention: {x.shape}")
    x, _, _ = coordinate_attention(x)
     # print(f"Shape after coordinate_attention: {x.shape}")

    x = Flatten()(x)
     # print(f"Shape after Flatten: {x.shape}")
    x = Dense(512, activation='relu', kernel_regularizer=l2(0.0001))(x)
     # print(f"Shape after Dense(512): {x.shape}")
    x = Dropout(0.40)(x)
    outputs = Dense(num_classes, activation='sigmoid')(x)  # Use sigmoid for multi-label classification
     # print(f"Shape after Dense(output): {outputs.shape}")

    model = Model(inputs, outputs)

    # fixed weighted
    # model.compile(optimizer='adam', loss='custom_weighted_loss', metrics=['accuracy'], weighted_metrics=[])
    # fixed weighted

    model.compile(optimizer='adam', loss='binary_crossentropy', metrics=['accuracy'], weighted_metrics=[])
    return model
```

Appendix 7: Best Results on the Open-MIC Dataset

This appendix presents the best performance metrics achieved by our model on the Open-MIC dataset. The results include the exact match ratio (EMR) and mean average precision (mAP), showcasing the model's effectiveness in multi-label classification tasks. The metrics highlight the model's ability to accurately predict multiple labels simultaneously, demonstrating its robustness and precision in handling complex musical instrument classification challenges.

	precision	recall	f1-score	support
accordion	1.00	0.00	0.01	233
banjo	1.00	0.07	0.14	244
bass	1.00	0.02	0.04	226
cello	0.91	0.03	0.06	302
clarinet	0.58	0.02	0.04	321
cymbals	0.69	0.48	0.56	317
drums	0.91	0.17	0.29	311
flute	0.78	0.17	0.28	328
guitar	0.57	0.36	0.44	315
mallet_percussion	0.80	0.20	0.32	280
mandolin	0.84	0.19	0.30	329
organ	0.90	0.04	0.08	221
piano	0.97	0.27	0.43	304
saxophone	0.76	0.26	0.39	389
synthesizer	0.75	0.31	0.44	305
trombone	0.88	0.06	0.11	393
trumpet	0.82	0.14	0.24	453
ukulele	0.80	0.24	0.36	314
violin	0.83	0.22	0.35	458
voice	0.76	0.46	0.58	236
micro avg	0.77	0.19	0.30	6279
macro avg	0.83	0.19	0.27	6279
weighted avg	0.82	0.19	0.28	6279
samples avg	0.21	0.18	0.18	6279

```
Exact Match Ratio (EMR): 0.27472959685349063
Mean Average Precision (mAP): 0.8272766709680545
```