Improved Deep Learning Model Based on Integrated Convolutional Neural Networks and Transfer Learning For Shoeprint Image Classification

Nikita Neha Goundar

Primary Supervisor: Dr. Maryam Doborjeh

Secondary Supervisor: Dr. Roopak Sinha

A thesis submitted to Auckland University of Technology in partial fulfillment of the requirements for the degree Master of Information Security and Digital Forensics (MISDF).

2023

School of Engineering, Computer and Mathematical Sciences

Abstract

Machine Learning (ML) and Deep Learning (DL) techniques have recently aided in resolving critical problems in various sectors and use cases. For example, image classification techniques based on machine learning and deep learning have proven useful in medical science and other industries. Existing work has investigated a shoeprint image classification technique to identify different classes of shoeprints for several forensic applications and wear-pattern identifications. Despite the challenges presented by the situation, there are several opportunities to explore this area. The need for more sufficient datasets is the main obstacle in this field. In addition, deep learning techniques frequently fail to achieve a high accuracy since the field still needs to be sufficiently developed. The literature has, however, used a range of traditional machine-learning algorithms.

This thesis has applied deep learning classifiers for shoeprint identification. This study first explored traditional classification methods and deep learning Convolutional Neural Networks (CNN). Then it proposed a method for integrating CNN and Transfer Learning (CNN-TL) to improve the classification results. In CNN and CNN-TL methods, python's tensor-flow library was used. Finally, the shoeprint classifications were performed using a pre-trained and fine-tuned version of the Inception model, including comparing different pre-trained Inception V3, VGG16, and ResNet50 models. The results show that convolutional neural networks-transfer learning (CNN-TL) improved classification accuracy by approximately 3% compared to conventional convolutional neural networks (CNN).

The study employed three techniques for shoeprint classification, namely CNN, TL, and the proposed method that combined TL with CNN with various pre-trained models (Inception V3, VGG16, and ResNet50). The performance metrics of each model employed in this study produced the following individual results: CNN model (accuracy = 96.17%), CNN-TL Inception

V3 model (accuracy = 92.19%), CNN-TL VGG16 model (accuracy = 96.88%), and CNN-TL ResNet50 model (accuracy = 97.14%).

The ResNet50 model achieved the highest accuracy of 97.14%, outperforming all state-of-the-art approaches in shoeprint classification. Regarding accuracy, the VGG16 model outperformed the CNN model, but the Inception V3 model performed with lower accuracy. The study highlights that the proposed methodology significantly improved the accuracy compared to previous literature. The proposed methodology is expected to open new avenues for forensic science research and deep learning approaches to image classification.

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 acknowledgments), 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.

Signature: Date: 07 July 2023

Acknowledgement

I am writing to thank Dr. Maryam Doborjeh, my primary supervisor, for her advice, motivation, and expertise during the year we worked on this thesis. Her expertise in image processing, knowledge representation, and machine learning has inspired me to create more impressive models and run more effective experiments and tests for this study. I also thank Dr. Roopak Sinha, my secondary research supervisor, for acting as my mentor and providing valuable feedback and support.

My heartfelt thanks go to Dr. Adam Kortylewski and Professor Thomas Vetter of the University of Basel in Switzerland. They gave this thesis access to a massive dataset, without which my experiment would have gone much better.

At Auckland University of Technology, I would like to express my heartfelt appreciation to Sharda Mujoo (Postgraduate Coordinator, Faculty of Design and Creative Technologies) and Alastair Nisbet (Master of Information Security and Digital Forensics Programme Leader) for providing me with the tools and advice I required while conducting the research. This voyage required reading the literature, gathering data, analyzing it, conducting experiments, and writing a thesis. It was also the last stage in obtaining my degree. The research experience was intriguing because of working with my supervisor and other faculty members.

Finally, I am grateful to my parents, siblings, friends, and colleagues - those closest to me who played an essential role in my life. I cannot thank them enough for understanding my priorities, mental stress during this research, and time management, as well as for their constant love, motivation, emotional support, and confidence.

Nikita Neha Goundar

Declaration of Competing Interest

I hereby declare that I have no known financial or personal conflicts of interest that appear to have influenced the research described in this article.

Data Availability

Unibas was used to collect the initial datasets, and the rest of the versions were produced through manual and automatic processing. As a result, all sets of data are readily accessible to the research community under the terms and conditions outlined in the links below:

- FID: Footwear Impression Database/FID-300: https://fid.dmi.unibas.ch/#Download

(Kortylewski et al., 2015)

- 2D Footwear Outsole Impressions: 2D Footwear outsole impressions (figshare.com)

(Soyoung & Alicia, 2020)

Table of Contents

List of Figures

List of Tables

CHAPTER 1: Introduction

1.1 Background

The proliferation of machine learning research (Afifi et al., 2020; Doborjeh et al., 2021; Doborjeh et al., 2022; Kaur et al., 2022) and, more recently, researchers' acceptance of deep learning mechanisms (Copiaco et al., 2023; Khan et al., 2022; Siddiqui et al., 2022) to solve various real-life issues has influenced scientific approaches to address many challenging problems. These learning algorithms have many uses, including medicine and healthcare (Fatima et al., 2022; Irshad et al., 2023; Mufti et al., 2022), recommendation engines (Ashraf et al., 2022; Himeur et al., 2022), traffic management (Aouedi et al., 2022), crewless vehicles (autonomous cars) (Pavel et al., 2022), smart cities (Bhattacharya et al., 2022; Rajyalakshmi & Lakshmanna, 2022), digital twins (Lyons, 2022), agribusiness (Xu & Hsu, 2022), and robotics (Lăzăroiu et al., 2022; Yue et al., 2022). For these fields of study, many recent works have examined image processing, particularly image classification (Gupta & Bajaj, 2023; Ottoni et al., 2023). Image classification, for example, serves as the foundation for disease diagnosis, detection, and prevention in medicine and healthcare. For example, researchers have used X-rays and MRI images to diagnose patients' illnesses and discover diseases (Hamza et al., 2022). Similarly, image classifications are essential in developing smart cities for various purposes, including learning autonomous vehicles via object detection and similar methods.

The literature recently revealed an increase in researchers interested in investigating. *Forensic science* is an important field that is researching methods for identifying shoeprints. Shoeprints are the most commonly left clues at a crime scene, greatly assisting the investigation. In recent years, automatic shoeprint recognition has thus become a significant field of forensic science research. Developing advanced computational methods for classifying shoeprint images

examined at crime scenes can lead to the accurate detection of illegal shoeprints. This will significantly contribute to forensic science, allowing for better processing, understanding, and identification of shoeprints at crime scenes. So far, several methods for identifying shoeprints have been proposed. (Rida et al., 2019) have discussed a few publications that investigate classifications for forensic shoeprints. There are numerous applications for shoeprint classification. For example, (Francis et al., 2019) used shoeprint photos to determine wear patterns.

Furthermore, wear patterns are essential in many forensic science investigations. Consider the case where a long period passes between committing a crime and discovering a suspect (s). In such cases, the forensic scientist's responsibility is to examine the outsole and determine whether it matches the scene print while considering the development of other wear features. This job requires close inspection of the outsole and extensive knowledge of the various elements and components that influence wear patterns. Forensic examiners assess shoeprints and determine whether they are admissible as evidence based on their extensive knowledge of shoeprints and the distinctive characteristics contributing to the wear pattern. Thus, shoeprint identification has primarily been investigated using classification tasks, for which multiple images of shoeprints must be collected and used to train and test machine learning models. Shoeprint classification has been accomplished using various techniques, including statistical models, traditional machine learning, and deep learning.

Some advanced processing systems, such as the United Kingdom's National Footwear Reference Collection (NFRC), store every footwear impression submitted for processing. (Budka et al., 2002) extensively described NFRC utilizing descriptors. The NFRC is based on a British standard classifying footwear styles for various forces. The police have the most extensive collection of shoeprints in the world. The NFRC employs a total of 17 traits to determine a footwear

impression. A shoeprint or footwear impression could contain any subset of these characteristics. There are two descriptions for the central sole and heel/instep. The most common type of impression collected from convicts has imprinted footwear impressions. Imprints on footwear are recorded on a special pad and paper set. This data type necessitates additional processing privileges, necessitating the use of a powerful computing machine. Furthermore, a robust algorithm delicately dealing with such complexities is required.

However, there is still a gap/limitation in shoeprint classification, which necessitates using a powerful computing machine to process the data recorded. By combining transfer learning (TL) and convolutional neural network (CNN) models to process shoeprints, this thesis attempted to fill this gap. As a result, suitable deep-learning techniques must be investigated to improve the classification accuracy for shoeprint identification. Forensic investigations and crime-related exploration are examples of these applications.

Shoeprint image retrieval and classification are two tasks related to analyzing shoeprints frequently encountered in forensic investigations. Shoeprint image retrieval entails locating all images of shoeprints in an extensive database similar to a given query image (Lee et al., 2019). The query image is typically supplied by an investigator who wishes to locate all images of shoeprints in the database similar to those found at a crime scene. Shoeprint image retrieval is typically accomplished using content-based image retrieval techniques that compare the query image to images in the database using image features such as color, texture, and shape (Lee et al., 2019). Shoeprint image classification, on the other hand, entails classifying a given image of a shoeprint into one of several predefined classes (Li et al., 2020). For example, a shoeprint image classification model could be trained to identify an image of a shoeprint as belonging to a specific shoe brand or model based on the distinctive patterns on the shoe's sole. Machine learning techniques, such as convolutional neural networks (CNNs), are commonly used to classify

shoeprint images. CNNs can automatically learn to recognize the patterns and features distinguishing shoe brands and models. To summarize, shoeprint image retrieval is used to locate similar images in an extensive database. On the other hand, shoeprint image classification categorizes a given image into one of several predefined classes.

1.1.1 Shoeprint Retrieval: An Overview

A challenging computer vision task is retrieving shoeprints because of various elements, including textures, patterns, manufacturing models, and wear effects (Srihari, 2010). As a result, numerous processing methods have been used, including manual, semi-automated, and automated procedures, i.e., fully automated, partially automated, and manual efforts. Recently, deep learning has been used in feature extraction, reconstruction, and research applications. In contrast to other biometric modalities such as fingerprints, palm prints, and retina prints, which have consistent patterns and shapes, shoeprints need to be more consistent. The ability to deduce age and gender from shoeprint impressions streamlines the investigation and testing process. Shoeprints can be used to screen a large number of applicants. Other factors that can connect them in research include age, stride, standing patterns, and shoeprints (Hassan et al., 2021). A large-scale shoeprint dataset was used to estimate human age and gender using the ShoeNet model, a machine-learning algorithm. The authors of this paper (Hassan et al., 2021) claim that by employing the ShoeNet model, they can make more accurate decisions while also benefiting from cost-effectiveness and scalability. Everspry Outsole Scanner recently acquired equipment, a device people use to leave their shoeprints on the strip. For this purpose, many people are leaving their shoeprints.

Pre-processing on each image included the following steps

- scaling to the exact dimensions (224 Height x 112 Width),
- converting to a hue-saturation-value (HSV) format, and

- masking the strip with a threshold value.

The Hue Saturation Value (HSV) scale provides a numerical reading of an image corresponding to its color names. The hue is measured in degrees from 0 to 360. For example, magenta has a temperature range of 301-360 degrees, while cyan ranges from 181-240 degrees. Before training the ShoeNet model, the scientists manually rotated the photographs and removed low-resolution images. ShoeNet comprises three completely connected layers and five convolution blocks (Blocks-A, B, C, D, E). Before the model was trained, 18,111 low-quality photos were eliminated. As a result, 40,000 of the remaining 81,889 images were chosen for testing and certification (Hassan et al., 2021).

A total of 100,000 photographs were taken of people aged 7 to 80. Each person's shoeprints are depicted in two images (left and right). They divided the population into three age groups for Dataset-A: 7 to 18 (1459 samples per group), 19 to 25 (8637 samples), and 65 to 80 (8637 samples). The shoeprint data must be changed for the ruler/scale associated with each image to quantify the morphological attributes to be removed from Dataset-B. There are only left shoeprints in Dataset-C, but only right shoeprints in Dataset-D. In Dataset-E, the left and right shoeprints were combined into a single pair-wise shoeprint. Gender- and sex-segregated shoeprints were used in Dataset-F to train models that could identify people as male or female and predict their age. Gender refers to biological sex, which includes both males and females. In Dataset-G, the sample distribution by age range was equalized through augmentation. ShoeNet, a deep neural network, can determine how changes in gait and stance patterns, which correspond to different walking styles, affect shoeprints. It is based on paired shoeprints and incorporates elements from the left and right shoes. Shoeprints collected in pairs will provide more age-related information on standing and gait patterns than a single print. Many alternative deep learning models were constructed to match the structure of the datasets in order to make the most of the

data for age prediction. Type-A personalities thrive between the ages of 25 and 45. The Type-B MCS-2 (17.86) and MCS-3 test results were the best (22.67). Cat-E (51-80) has unusual gait and stance patterns, which harm performance.

Nonetheless, they have achieved significant gender prediction accuracy, with 89.34 percent training and 86.07 percent testing accuracy. According to the classifier, males (M) and females (F) have different pressure distribution patterns. Gender is correctly predicted by the classifier 86.07 percent of the time (F). Right, shoeprints on both boys' and girls' feet are thought to withstand tremendous pressure as children grow. Female blood pressure varies significantly between the ages of 20 and 40. Male responders have more muscular pressure patterns from a young age until they reach the age of 30. Not all training networks, however, can recognize gait patterns. This could imply that the left and right shoeprints have experienced the most aging-related changes. Another issue is the requirement for consistent textural-outsole patterns in different shoes. In forensic science, deep learning models can be used for shoeprint enhancement, super-resolution, abrasion repair, and the generation of full shoeprints (Hassan et al., 2021).

According to the study's authors, every footwear impression sent to the UK's National Footwear Reference Collection (NFRC) is meticulously documented with the descriptors (Budka et al., 2021). The NFRC is based on a British standard for classifying footwear designs for various forces. The police have the world's most extensive collection of shoeprints. To determine a footwear impression, the NFRC employs a total of 17 traits. Any subset of these characteristics could be found in a shoeprint or footwear impression. The central sole and heel/instep descriptions are divided into two categories. Imprinted footwear impressions are the most common type collected from convicts. Footwear imprints are recorded on a special pad and paper set. The impressions could be mechanized using a scanner for business documents if necessary. Digital scanners only generate a digital copy of an inkless impression.

1.1.2 Shoeprint Retrieval Methods Implementation

Retrieving shoeprints has been done using a variety of techniques. This includes traditional approaches such as manual comparison, casting, and photographs and more sophisticated technologies such as 3D scanning, laser scanning, and photogrammetry (Li et al., 2022; Liu & Xu, 2022; Wu et al., 2022).

These approaches are chosen based on the study's specific needs and vary in accuracy, speed, and cost. The processes involved in shoeprint retrieval include collecting the shoeprint, preparing it for analysis, comparing it to other shoeprints, and determining the owner. (Crookes et al., 2007) viewed shoeprint images as a cross-domain task, similar to fine-grained sketch-based image retrieval (FGSBIR) issues. Based on primary networks such as VGG and ResNet, Siamese networks have extracted images from various domains. The first method in this paper uses a Siamese network with VGG19 as the base network to match scene of a crime (SoC) impressions with database photos. The triplet loss function is used to compute similarities. Another technique this study suggests is using UNet to improve image quality prior to image retrieval. The UNet architecture includes a convolutional autoencoder with lateral connections between corresponding layers from the encoder to the decoder. It performs well in segmentation tasks and can be taught from start to finish with just a few images. After external noises are removed, the SoC impressions are converted to binary images (Bi-SoC), and they have an integrated UNet to extract useful information from them.

Although this process will inevitably result in the loss of some critical information, the gain will be significant once the cross-domain issues are resolved. The image of the shoeprint is also enhanced throughout processing to improve the accuracy of the analysis. A forensic specialist frequently compares shoeprints by examining their characteristics and contrasting them with those of known shoeprints stored in a database. The expert uses their judgment to determine who

owns the shoeprint. The retrieval of shoeprints is accomplished through a combination of manual and automated methods. In the initial comparison, a forensic specialist will frequently manually compare the shoeprint to known shoeprints (Li et al., 2022). The authors used color and texture features to improve retrieval performance. The next step is to use adaptive weighting to adjust the contribution of each feature to the final similarity score. In tests, the proposed method outperforms existing methods on a publicly available shoeprint database, with an accuracy rate of 96.36%.

Nonetheless, a recent push has been to develop automated systems for assessing and comparing shoeprints (Li et al., 2022). These systems gather information from the shoeprint, compare it to existing shoeprints, and determine the owner using machine learning and computer vision techniques. As a result, forensic investigation necessitates the retrieval of shoeprints, and significant research has been conducted in this field. Shoeprint retrieval employs traditional techniques, such as casting and photography, and more sophisticated ones, such as 3D scanning and photogrammetry. To recover shoeprints, many procedures must be followed, including gathering, processing, comparing, and determining the owner. A combination of manual and automated methods is used to retrieve shoeprints, and current research has focused on developing automated systems that use machine learning and computer vision algorithms.

1.1.3 Shoeprint Classification: An Overview

Several queries were done on Scopus' standard and benchmark database to analyze the research article, specifically in shoeprint classification. In response to the query, the databases returned only three results. This illustrates that there are articles on shoeprint, but they are primarily concerned with retrieval rather than categorization. Furthermore, the forensic use of digital footprints has attracted more attention than classification. Due to the scarcity of articles found in Scopus, an investigation was carried out in the second phase by looking for cross-references in

the articles. Google Scholar was used to query for shoeprint-related articles, and the references of these articles were also searched, yielding a few other relevant articles.

According to (William J. Bodziak, 2017), manual human shoeprint classification is a field of expertise for people interested in forensics who process shoeprint images to detect, recover, and examine footwear impression evidence. Several semi-automatic classification methods have been proposed to help forensic laboratories with shoeprint classification (AlGarni & Hamiane, 2008). For example, they proposed a semi-automatic scheme for distinguishing burglars' shoeprints (Alexandre, 1996). This method necessitates using a human expert to determine the number of geometric patterns for each sole, such as zigzags, circles, squares, and letters. Based on these patterns, a database of known shoe types was created, and unknown shoeprints could be compared to the shoeprints in the database to find a match. The 'Shoe-Fit' method by (Sawyer, 1995) and the 'Shoe' method by (Sawyer, 1995) are two other systems that use this semi-automatic method (Ashley, 1996).

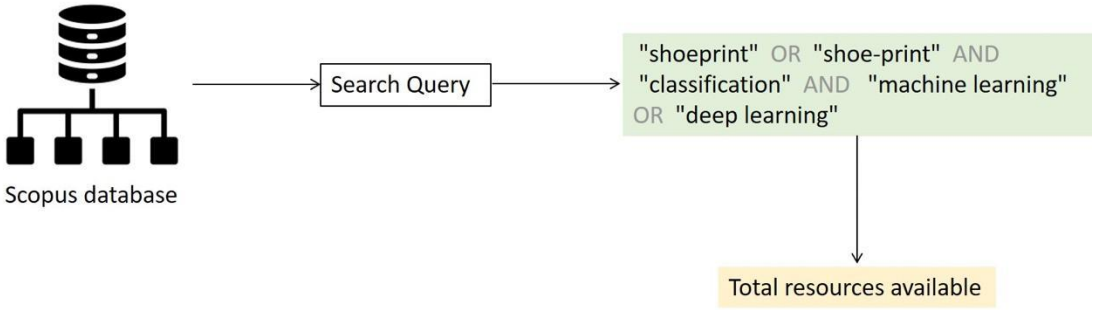

Figure 1. A procedural diagram to illustrate the data collection for reviewing concerned published articles in the literature.

The field of automatic shoeprint classification is still in its early stages, with few studies on the subject. (Geradts & Keijzer, 1996). For example, they developed an algorithm for automatically classifying outsole patterns by applying a series of primitive 'erosion' and 'dilation' operators to a digital image of the shoeprint to generate a set of geometric shapes. These shapes are then classified using a neural network and analyzed using Fourier components. The authors, on the

other hand, should have reported on the accuracy of their system. (Alexander, 1999) Furthermore, (Bouridane et al., n.d.) described an automated system that used fractals to represent shoeprints and a mean square noise error method to find the best match. When the query image's spatial and rotational position matched the database image, their system correctly extracted a shoeprint from 145 full-print images with an accuracy of 88%. The system's performance, however, was sensitive to translation and rotational changes. For example, when a 908 rotation was introduced, the accuracy dropped to less than 50% and less than 10% when a 13-pixel translation was introduced. Furthermore, the algorithm was incapable of dealing with changes in image size.

De Chazal et al. (2005) created an automatic classification system for shoeprint images based on the Fourier transform. They discovered that the first shoeprint image displayed was a correct match 65% of the time and that a matching image was 87% likely to be found within the first 5% of the sorted database patterns. The authors searched a database of 476 whole prints classified into 140 categories and found a match for each image. The system, however, did not address the issue of scale invariance, and the authors did not address the issue of noisy images. Zhang and Allinson (2005) developed an automatic shoeprint retrieval system based on sole pattern edge information. They discovered that adding noise cases when the images are aligned along the central axis of the print allowed them to estimate the system's performance. When a correct match was found within the first 4% of the sorted database images in 97.7% of the cases and at the top-ranking position 85.4% of the time, the system performed optimally. However, a discussion of the system's handling of partial prints, frequently discovered at crime scenes, was necessary, and the system's accuracy in rotated situations could have been improved.

The article's authors, Koziarski and Cyganek (2018), highlighted existing methods for dealing with low-resolution images, such as image super-resolution and down-sampling. They used the CIFAR-10 dataset, which contains ten object classes, each with 6,000 32x32 color images. Normalization, data augmentation, and downsampling are the pre-processing steps to simulate

various low-resolution levels. Deep neural network (DNN) architectures tested included AlexNet, VGGNet, and GoogLeNet. It was discovered that low resolution affects DNN performance differently depending on the network's architecture. AlexNet, for example, is more resistant to low resolution than VGGNet and GoogLeNet. One possible explanation for these performance differences could be using different parameters, as suggested by them.

According to their research, even minor changes in the input photographs' quality can significantly impact how well a deep-learning model performs. The aspect ratio of the photos used as input must be preserved. This is critical if the object of interest does not have a 1:1 aspect ratio. To optimize PRAUC, use the practical maximum input resolution, but remember that this will require many more calculations, which is a limitation. Because upscaling such images is pointless, the initial resolution of the received photographs may also be limited. Greyscale images were converted into three input channels for the study. Using Nearest Neighbor interpolation as one of the input channels is preferable. Three identical Bilinear interpolated channels may be more efficient, particularly at lower input resolutions. Pre-processing layers are critical in converting greyscale inputs with a color image pre-trained network, regardless of the interpolation method. The best results were obtained by combining learnable pre-processing and several interpolation techniques to construct the three input channels. Until recently, however, the descriptors had to be manually selected by specialists, and they were only used as a first step in detecting a pattern. Although no scientific study has been conducted to test this, experts' accuracy in recognizing descriptors is "very high" (Bluestar Software, 2018).

Forensic investigators only use descriptors to distinguish between distinct footwear impressions on rare occasions. Unique or modern shoe models take more time to categorize. Furthermore, because there are thousands of different models, it is difficult for a human specialist to identify just one. In England and Wales, 123,1712 police officers could be asked to identify the case's descriptors. The population of the United Kingdom may benefit from automated descriptor

identification (Budka et al., 2021). Shoeprints are not uncommon, but they are more likely than latent fingerprints to be discovered in an SoC. If there are several attacks in a short period, an attacker is unlikely to throw away or change his or her footwear between crime scenes.

Furthermore, it has been discovered that an SoC may be capable of retrieving approximately 30% of shoeprints (Alexandre, 1996). One of two approaches could be used to exploit an SoC's lifted shoeprint: a database comparison to determine its model (such as Foster and Freeman Ltd). In addition, it is possible to determine whether the same shoe model was worn by comparing it to other shoeprints collected from various SoCs. An automated shoeprint identification system's basic architecture is divided into three sections. The activities include pre-processing, developing discriminative features, and categorizing/mapping the query sample with the entire database of shoeprints. The retrieved features may differ depending on whether a structural or holistic approach is used. A holistic approach aims to process the entire shoeprint image. The local approach, on the other hand, attempts to elicit distinguishing characteristics from regional shoeprint regions. This method uses critical points or several overlapping/non-overlapping segments to convey technical information on essential topics. A comparison study, on the other hand, necessitates using large, publicly available datasets.

Many published strategies have been tested using manipulated and fictitious photographs. Unfortunately, the results cannot be directly or fairly compared to the cited state-of-the-art techniques. In addition, it should be noted that (Dardi et al., 2009) and (Y. Tang et al., 2011) relied on actual data, which was also unavailable at the time. Another area where this image analysis technique can be improved is more training data and clutter, occlusion, noise, and translation. Without prior task expertise, feature learning algorithms can produce accurate representations from raw data (Li et al., 2022). Deep learning is one method for creating an end-to-end identification system. Convolutional neural networks are a well-known deep learning model that could be used to recognize shoeprints (CNN). However, a large amount of data must

be collected for deep learning algorithms to detect shoeprints correctly. The available shoeprint recognition databases are limited because each shoe style only has one sample. In transfer learning, models already pre-trained on massive amounts of data for a different task are used. Numerous studies have been conducted on developing discriminative features using feature extraction techniques.

A deep neural network architecture is used, which learns discriminative features from images by training the network to recognize similar and dissimilar image pairs (G et al., 2015). Similarly, Babenko and Lempitsky (2015) proposed an image retrieval approach that combines multiple feature extraction techniques, such as local binary patterns, color histograms, and deep convolutional neural networks. The authors showed that combining these features improves image retrieval accuracy over using each feature extraction technique separately. Another research has proposed new similarity metrics for computing the distance between query samples and gallery photos. References use base models to extract deep features and neural networks to train for matching. The main limitation of these methods is that only a few distinct structures can be used to recognize shoeprint features (Montaseri et al., 2016). The Siamese network has been improved for more precise retrieval across the gap between sketches and photos.

Despite the encouraging results, the sketch-based image retrieval (SBIR) problem is distinct from the shoeprint identification (SoC) problem. Ma et al. (2019) used the U-Net method to improve image quality by extracting relevant information from SoC impressions and converting them to noise-free binary images. Although some vital information may need to be recovered, this method has advantages for cross-domain issues. They grouped SoC impressions with exact matches into one class, yielding 1843 SPID classes (State Policy Innovation and Diffusion Database Dataverse). They extracted features using the VGG19 convolutional neural network, which was trained on the ImageNet dataset. The FID300 dataset did not support the multiplane convolutional neural network (MP-CNN), and retraining may impact the model's performance on test datasets.

For accuracy in the top 1, top 5, and top 10%, the MP-CNN with weighted features outperformed the MP-CNN with equal weights. The network's performance improved because it focused on the informative parts of the input images and created a more discriminating representation. The researchers have yet to determine whether this method can be applied to other problems, such as sketch-based image retrieval.

In a study of several Swiss counties, the researchers discovered that 35% of crime scenes had acceptable shoeprints for forensic science, and 30% of all burglaries had acceptable shoeprints (Alexandre, 1996). Comparing photos of shoeprints from crime scenes to databases takes a long time nowadays. Shoe mark evidence could be used more frequently due to the lengthy process. The Garda headquarters in Dublin, Ireland, made paper copies of 1,276 shoeprints (de Chazal et al., 2005). The images were digitized using 256-level greyscale mapping and 300 dots per inch resolution. In addition, the shoe's brand, style, size, and wearer's age were all recorded. Before rating the shoeprint images, the observers were divided into four groups. According to the researchers, 14 of the fourth group's shoeprints could have been of better quality due to acquisition issues such as ink spreading and repeated impressions (due to the user stamping twice on the paper).

Before generating a modified image, the system analyses a reference shoeprint image. A forensic scientist searches each category to see if there are any instances of a specific database category (if one exists). If a relevant database category is found, the supporting database columns can provide information on shoemaker style. The shoeprint images were reduced to 64x64, 128x128, 256x256, and 512x512 pixels. The photographs were digitally scanned using an A3 or rotary scanner. The patterns contained oversampling of relevant information content. Various rotations of up to 30 degrees may be used to overlay one image on top of another. Certain prints displayed a shift in grey level between them due to variations in contact across the shoe during image capture. This variation appeared in the PSD as a very low-frequency component. When

processing partial prints, the system was unable to organize database photos as efficiently as it could when processing full prints. Before a match, an average of 8.2 percent of the sorted database categories were checked. The device performed nearly identically on the left and right outsoles.

However, the method has been discovered to eliminate operator bias and work well with incomplete prints effectively. Because shoeprint data is only helpful for a limited time, a system like this would need to be updated regularly. This is accurate given that shoes only endure a finite amount of time and that trends constantly change.

If no other criminal evidence is available, shoeprints may be the only physical evidence left at a crime scene, providing some potential hints to a specific case. For example, the pressure images from the MUES-SV1KR2R dataset were collected using shoeprint scanners (Wang et al., 2019). When someone steps on reflective tape, pressure photos are taken, and irregular pressures cause varying heights of downward bulges to appear on the surface.

More than 1200 pairs of shoeprint photographs are included in the MUES-SV1KR2R dataset. The set contains registered images with identical spatial resolution. Most of the collection's images have 3600 by 1800-pixel resolutions and are sharp and detailed. There is a 5cm gap between images one and two and between light source one and the shoeprint. More than 256 pairs of shoeprints with the same shoe design are included in the MUES-SV2HS2S dataset. Because they are registered, this collection's photos have the exact spatial resolution as those in the original dataset. The recommended method for verifying shoeprints consists of four steps. Each of these components is required for the system to function correctly. First, the decision-making module considers the similarity score before deciding whether the identical shoe was left with the same shoelace. Second, a shoe partition model (SPM) is developed to divide the shoeprint image into different portions based on the foot's shape and how the shoe and foot are worn. Third, each segment is subdivided into numerous distinct, non-overlapping subsections for further study. There are a total of 19 subsections. They recommend a multi-layer feature extraction strategy to

capture the properties of shoeprints on the global layer, partial layer, and individual identifying layer. The global layer elements that reflect class characteristics are used to determine whether two shoeprints are from the same shoe pattern. Second, each of the 19 shoeprint subsections' smallest surrounding rectangle region is retrieved and separated into distinct, non-overlapping patches. The final partial layer characteristic that reveals shoe wear characteristics is the intensity distribution of each patch. An interesting point, line, or region on a shoeprint image may represent a specific distinguishing layer characteristic. A single feature must meet several requirements, one of which is that it cannot be seen more than once or twice in a shoe.

The structural changes in the shoeprints are especially noticeable (i.e., the class characteristics, wear characteristics, and individual identifying characteristics). In the first experiment, the proposed method of shoeprint verification was tested. The novel method was compared to a few traditional procedures in the second round of testing. The results show that the HOG method outperforms the NCC method. This could be because the shoeprint images in the MUES-SV2HS2S collection require higher resolution. The approach performs noticeably better than other feature detector and descriptor combinations. The proposed method validates shoeprints at a slower rate than the alternatives. They intend to use parallel computing technology to accelerate processing in the future. The proposed method matches individual identifying, global, and partial layer features. Instead of creating a significantly more efficient classifier, the authors attempted to improve the verification technique and determine how to improve performance on complex backdrops.

To create a database of shoeprint images, the original shoeprint edges and those on other surfaces must be collected. Before using a recognition technique, shoeprint images obtained at a crime scene must be ready. Light, chemical amplification, and naturally occurring or artificial elevation can all be used to visualize perceptions. Various techniques can be used to recover the original (raw) footwear evidence seen in shoeprints. Electrostatic lifting is used to remove dust-containing

imprints. Powders have the potential to leave latent fingerprint impressions on surfaces. The new shoeprint matching techniques and improvements fall into two categories: feature-based matching and probabilistic methods.

An algorithm for determining the shoe manufacturer by connecting murder scenes (Wang et al., 2015). The method successfully matched 210,000 shoeprints, yielding a total success rate of 90.87 percent. This method, however, is time-consuming and dangerous (due to false matches). It was explained how to find automatic shoeprints at crime scenes. It looks at Euclidian distance and Fourier characteristics to find a match. However, it has issues with an extensive database (Deshmukh & Patil, 2009). The computer-based footwear image analysis method is described (Srihari & Tang, 2014). When comparing 50 authentic footwear marks to 1000 recognized shoeprints, it achieved a 92 percent match rate. It has been discovered that a standard database is required for the computational method to produce better shoeprint-footwear matching. Many currently used techniques, however, are still in development and have drawbacks. As a result, image enhancement, segmentation, feature extraction, and pattern matching should be prioritized. Combining feature-based and probabilistic approaches could produce more robust matching algorithms. Deep learning and recurrent neural networks (RNN) are two neural-based techniques that have recently demonstrated promising results (Mamun et al., 2019).

The carbon tracker tool can compute and measure deep learning models' energy consumption and carbon emissions. The user can halt model training if the predicted environmental cost is exceeded. They work with various hardware and environments, such as clusters, desktop computers, and Google Colab notebooks. Carbon Tracker simplifies model training's time, energy, and carbon footprint calculation. First, they project the carbon intensity of energy generation over the projected period using the available APIs. The carbon footprint of the model is then forecasted using the expected carbon intensity. The authors (Anthony et al., 2020; Wu et al., 2022) analyzed the image using a single NVIDIA TITAN RTX graphics card with 12GB of

memory and two Intel central processor units (CPUs). Forecasted energy values differ from measured energy values by 4.9 to 19.1%. Throughout the period, CO2eq forecasts are off by 0.8-4.6%. The GPU consumes approximately 50-60% of the total; however, the CPU and RAM also play an essential role. The energy sources used by the local grid influence the carbon intensity of electricity production, which varies by region. In a single training session, an Estonian model could produce more than 61 times the CO2 equivalents as a Swedish model. A Danish model developed during low-carbon intensity times of the day could produce 14% less CO2eq than a model developed during peak times. With a change in training, the United Kingdom's emissions could be cut in half. Hyperparameter tuning can be improved by switching from random to grid search and employing Bayesian optimization or other optimization techniques such as Hyperband. Deep neural network inference can be made more energy-efficient by optimizing hyperparameters, using energy-aware pruning, and training with quantization.

In legal proceedings, shoeprints are rarely used as evidence obtained from crime scenes. This is due to the different levels of crime scene impression quality. Finding acceptable outsoles is difficult due to the massive number of outsoles produced and created. Example research performed by the authors (Francis et al., 2019) used 52 photos to train and test their time-lapse technique. The dataset is split 80/20 between training and testing for model training. They train 10,000 times at a 1e5 learning rate. Using the network, the researcher successfully simulated the broad wear pattern indicated by shoeprints. This change has occurred, as evidenced by the week 52 ground-truth photo, even though a four-week union is more common than the model's projection of 20 weeks. In the prognosis, the top of the heel should be more specific and defined. During the data collection phase, this area was frequently obscured by dust and fingerprints or was insufficiently made due to insufficient pressure applied to the gel during the imprinting procedure.

Nevertheless, their methodology is adequate for extracting the wear pattern from the dataset and performing accurate outsole prediction and reconstruction. They discovered that the network requires more training data. A large enough dataset can also aid deep learning models' generalization ability. They investigated the model's flaws and provided objective performance evaluations, revealing that both models' predictions are 86% accurate.

Given the preceding works, as discussed further in section 3.5.3, there are some notable contributions to the literature since transfer learning emerged and has recently seen tremendous success. First, it includes a pre-trained model for shoeprint image classification, an image classification problem. Using pre-trained models for image classification has grown in popularity in machine learning. Pre-trained models are created by training them on a large dataset, typically from a public source, and then using them to classify images. This technology has been used in various applications and facilities, including facial recognition and medical imaging. This study used a pre-trained model to solve the problem because pre-trained models have several advantages over traditional image classification methods. The first advantage of using pre-trained image classification models is that training a model is much faster and easier. Instead of manual feature engineering, a pre-trained model can quickly identify and classify objects in images. This is because the model has already been trained on a large dataset of images, making recognizing patterns in the images more accessible. For example, the pre-trained models used in this study are Inception V3, VGG16, and ResNet50, which have already been trained on many samples. This makes training these models much faster than if the features were manually engineered. Another advantage of using pre-trained models over traditional methods is that they are more accurate. Pre-trained models are trained on a much larger dataset of images, allowing the model to learn more complex patterns and classify images more accurately. This is especially important in applications requiring high accuracy, such as facial recognition. The proposed mechanism achieved more than 97% accuracy in the results section (Chapter 4).

Transfer learning has proven effective in various disciplines, including natural language processing, computer vision, and speech recognition. Transfer learning is classified into two types: inductive transfer learning and transductive transfer learning. Inductive transfer learning is a transfer of knowledge from one task to another that is similar but unrelated. This type of transfer learning is helpful for tasks with a few labeled data points because it allows the model to leverage knowledge from a previous task. For example, if the two tasks are similar, a model trained to detect cats can detect dogs. Transferring knowledge from one task to another that is related but not necessarily similar is known as transductive transfer learning. This type of transfer learning is proper when the amount of labeled data points is small, and the two tasks are not necessarily related. For example, a model trained to detect cats can detect birds if the two tasks are related. Deep learning fine-tuning improves a model's performance on a specific task by retraining the weights of a pre-trained model at a lower learning rate. This is accomplished by starting with the pre-trained weights and optimizing them on the new dataset. Fine-tuning allows a model to quickly adapt to new datasets without training from scratch, saving time and money. Image classification is critical in computer vision. It assists machines in distinguishing between different types of objects and their differences. Deep learning raises machine learning accuracy to human-like levels. Deep learning is becoming more popular, which has a significant impact on classification. With the development of convolutional neural networks, deep learning significantly impacted computer vision and picture categorization. In addition, it is utilized in the extraction of visual elements.

1.2 Thesis Contributions

Deep learning models have shown great promise in developing representations for forensic domains. Deep learning models, on the other hand, perform well when massive training datasets are available or when the benchmark dataset is used. Transfer learning (TL) presents a practical approach for shoeprint classification, addressing the limitations associated with deep learning

methods. This thesis will examine the motivation for TL and deep learning and explore its key advantages.

1. Firstly, deep learning models require substantial labeled training data for optimal performance (Goodfellow et al., 2016). However, acquiring large datasets in real-world scenarios can be challenging, time-consuming, and expensive. TL offers a solution by leveraging knowledge from pre-existing datasets, even if unrelated to the specific shoeprint classification task (Smith et al., 2020). This enables practical model training with limited labeled data, overcoming data scarcity issues and facilitating accurate classification.
2. Secondly, training deep learning models from scratch can be computationally intensive and time-consuming, demanding significant computational resources (Goodfellow et al., 2016). In contrast, TL provides a time and resource-efficient alternative. By utilizing pre-trained models, researchers can save valuable time and computational costs. Instead of training models from the ground up, TL enables the transfer of learned features and representations from pre-existing models, reducing the training time and resource requirements (Gomez et al., 2019).
3. Furthermore, pre-trained models have typically learned rich feature representations from large and diverse datasets. These representations capture high-level visual features that are transferable across tasks and domains. TL allows these pre-trained models to be adapted to the shoeprint classification task, enabling the model to generalize well to new and unseen shoeprint data (Smith et al., 2020). This enhances the model's ability to classify shoeprints accurately.

Deep learning methods have limitations that further justify the use of TL.

1. Data scarcity and overfitting are common challenges in deep learning models, exceptionally when trained from scratch with limited data (Rida et al., 2019). Overfitting occurs when the model becomes too specialized in the training data and fails to generalize well to new data. TL mitigates this limitation by leveraging knowledge from more extensive and diverse datasets, preventing overfitting and enhancing generalization.
2. Another limitation of deep learning models, such as convolutional neural networks (CNNs), is their complexity and need for interpretability (Goodfellow et al., 2016). CNN architectures consist of numerous layers and millions of parameters, making it challenging to understand the internal workings of the model and interpret its predictions. TL provides a practical solution by utilizing pre-trained models that have been extensively studied and proven effective in various tasks (Smith et al., 2020). The transparency and interpretability of pre-trained models aid in understanding and validating shoeprint classification results.
3. Moreover, deep learning models, especially when trained from scratch, demand substantial computational resources, which may limit their accessibility for researchers or forensic experts without high-end computing infrastructure (LeCun et al., 2015). TL addresses this limitation by enabling pre-trained models, significantly reducing the computational requirements. This makes shoeprint classification more accessible and feasible in practical applications (Gomez et al., 2019).

By employing TL in shoeprint classification tasks, this thesis can leverage the benefits of pre-trained models, overcome the limitations associated with deep learning, and enhance the classification process's accuracy, efficiency, and interpretability. TL offers a pragmatic approach that maximizes the utility of available data, saves computational resources, and improves the overall performance of shoeprint classification systems. Due to these factors, researchers have proposed that transfer learning (TL) can reduce training time and data requirements (Rostami et al., 2019), for example, due to a lack of labeled data for training. The dataset for this thesis

comprises full and half-shoeprints. The dataset, including full-shoeprint images, includes images of the shoe from the heel cap to the outsole. On the other hand, the half-shoeprint image dataset contains incomplete shoeprints (some shoe portions are missing from the print). This complicates the dataset processing, lowering the accuracy of the possible outcomes. Thus, when applied to a target dataset using a pre-trained model on a separate dataset, transfer learning is remarkably effective (source dataset). Figure 2 depicts the general architecture for transfer learning, i.e., working Transfer Learning, a process illustration from data acquisition to output determination.

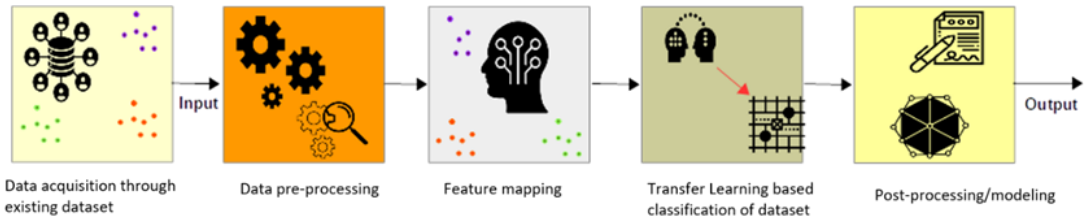

Figure 2. The working of Transfer Learning, a process illustration from data acquisition to finding out the output.

The data can be obtained from any existing dataset, or a new benchmark can be created or designed to meet the needs of the problem. The data may contain noise, irrelevant features, or characteristics required for exclusion from the dataset. Data pre-processing is required for this purpose. This step removed the dataset's noisy data and irrelevant attributes. For consistency and integrity, if images of clothes are mistakenly found in shoeprint images, or text is found in images, it must be removed. Following the completion of the pre-processing stages, the dataset was feature mapped. A transfer learning algorithm is then used to classify the dataset. The dataset was then post-processed or modeled with the algorithm and data to evaluate performance using the modified output. The specifics of this mechanism will be covered in greater depth later in the thesis. Domain adaptation or fine-tuning a pre-trained model can be used to execute transfer learning in practice. Domain adaptation entails a comprehensive and all-encompassing procedure and process. It is currently being developed and will require some time to adjust. Fine-tuning is

another application of transfer learning. This study employs the latter methodology. This study investigates the following research questions in light of the prior explanation.

1.3 Research Objectives

These experiments aim to improve image classification accuracy using a full or a half-shoeprint. Furthermore, this thesis proposes an efficient CNN-TL classification model to maximize classification accuracy while attempting to improve model robustness. Finally, these experiments aim to understand the performance of a convolutional neural network with transfer learning when dealing with the positional relationship between lines and patterns.

1.4 Research Hypothesis

The half-shoeprint and full-shoeprint had never been used on paper before, so it was hypothesized that the previous research "might" be applicable. However, previous research in shoeprint identification using convolutional neural networks is considered applicable. Therefore, the hypothesis is that an integrated convolution neural network with a transfer learning model can outperform a single convolutional neural network model in shoeprint identification.

1.5 Research Questions (RQ)

The research questions for this thesis are developed by reviewing current shoeprint identification technologies that employ traditional machine learning, deep learning, and transfer learning methods. First, a few of the most recent research findings in shoeprint identification fields were gathered. It then identified the critical research objectives and links to existing outcomes in this field. Finally, the work begins with the information gathered and the results obtained by others from surveys and literature reviews. As a result, this research can introduce new image classification methods. This thesis investigates the identification and classification of shoeprint

images using deep learning methods and combined convolutional neural networks with transfer learning (CNN-TL). The formulated research questions are as follows:

RQ1. To what extent do deep learning models, including CNN, lead to improving the accuracy of shoeprint classification?

RQ2. What are the advantages and disadvantages of using CNN than CNN-TL on shoeprint identification, especially in terms of performance?

RQ3. How can the CNN-TL performance be improved further with fine-tuning?

The proposed method would employ transfer learning with Inception V3 as a pre-trained model to address the research questions. In addition to the Inception V3 model, cutting-edge deep learning techniques such as transfer learning models will be investigated to find the best-performing algorithm with increased classification accuracy. The findings are encouraging and support the strategy of assuming transfer learning for shoeprint image classifications, which will be discussed in detail in Chapter 4 (Experimental results and discussion).

The terminologies used in the thesis are summarized in Table I, with abbreviation and complete form.

Table I. List of abbreviations with full form

Abbreviation	Full Form
Ada boost	Adaptive Boost
ANN	Artificial Neural Network
BPNN	Back Propagation Neural Network
CNN	Convolutional Neural Network
DL	Deep Learning
DNN	Deep Neural Network
ELU	Exponential Linear Unit
FN	False Negative
FP	False Positive
HMM	Hidden Markov Model
HOG	Histogram of Gradient
HSV	Hue-Saturation-Value
KNN	K nearest neighbor

LDA	Linear Discriminant Analysis
MDS	Multidimensional Scaling
ML	Machine Learning
NCC	Nearest Centroid Classifier
NFRC	National Footwear Reference Collection
NN	Neural Network
PCA	Principal Component Analysis
ReLU	Rectified Linear Unit
RNN	Recurrent Neural Network
RQ	Research Question
SBIR	Sketch-based Image Retrieval
SELwSVM	Successive Enhancement Learning based weighted Support Vector Machine
SGD	Stochastic Gradient Descent
SoC	Scene of Crime
SPM	Shoe Partition Model
SVM	Support Vector Machine
TL	Transfer Learning
TN	True Negative
TNR	True Negative Rate
TP	True Positive
TPR	True Positive Rate

1.6 Thesis Structure

The following is a summary of this thesis:

Chapter 1: Provides an introduction to the report, including the topic's background, thesis contribution, and research questions.

Chapter 2: A thorough literature analysis is provided, beginning with a machine learning (ML)-based classification mechanism and progressing to deep learning (DL)-based classification mechanism. Following these summaries, a comprehensive foundation on shoeprint categorization approaches and a comparative literature analysis will give background for this study issue, followed by suitable images and explanations.

Chapter 3: This chapter contains methodological information about this research study. The use of transfer learning, pre-trained models, and the operation of implied techniques are all addressed,

with firm support provided by the architectural diagram. In addition, the components of the proposed methodologies have all been clearly stated and demonstrated.

Chapter 4: The fourth chapter contains the experimental results of the convolutional neural network and the convolutional neural network integrated with transfer learning models. Each model's classification accuracy is illustrated and discussed.

Chapter 5: The conclusion summarizes the research and suggests future research directions.

CHAPTER 2: Literature Review

A thorough review of literature in the related field, namely, image classification using machine learning (ML) and particularly convolutional neural networks (CNN), especially shoeprint images, was provided to investigate the state-of-the-art techniques and their performance and identify the gaps in the current work. The literature review will also aid in developing a thorough understanding of previous image classification studies, which will go a long way toward identifying knowledge gaps and laying the groundwork for the work in this thesis. As a result, this chapter provides an overview of the literature on deep learning-based picture classification techniques and systems. Furthermore, the classification of shoeprints using machine learning and deep learning is further developed. Finally, this chapter provides valuable and instructive bibliometric evaluations (quantitative information) of shoeprint image processing using machine learning. This can help reveal patterns in linked publishing and provide a better understanding of how the connected area has recently developed. The chapter includes theoretical, technological, and bibliometric studies to lay a solid foundation for this research topic and questions.

Image classification is a computational task in computer vision that involves categorizing or labeling images into predefined classes or categories based on their visual content (Johnson et al., 2020). It aims to develop algorithms and models capable of automatically analyzing and understanding images' visual features and patterns to assign them to the appropriate class (Qadir et al., 2021). Image classification systems learn from large sets of labeled photos called training data using advanced machine learning and deep learning techniques. These algorithms extract relevant visual features from images and map them to specific classes or categories. The learned model can be applied to new, unseen images to predict their class labels.

Image classification finds wide-ranging applications across various domains, including object recognition, medical imaging, autonomous vehicles, surveillance systems, and image search

engines. It enables automated analysis and interpretation of visual data, facilitating tasks that would otherwise require manual human inspection and decision-making (Del Mar-Raave et al., 2021). In academic research, image classification methods are often evaluated based on accuracy, precision, recall, and F1 score (Qadir et al., 2021). Researchers explore and develop novel techniques to improve the performance of image classification algorithms, enhance their robustness to variations in lighting conditions, viewpoint, and image quality, and address challenges such as class imbalance and limited annotated data.

Overall, image classification plays a fundamental role in visual understanding and has significant implications for numerous fields, contributing to advancements in artificial intelligence, pattern recognition, and computer vision.

2.1 Shoeprint Images: Problems, Solutions, and Challenges

The field of shoeprint image classification in forensic science has received limited research attention compared to other domains of machine learning and deep learning (Mekhmoukh et al., 2021). While statistical models, traditional machine learning, and deep learning techniques have been explored in shoeprint identification, there is a research gap in utilizing deep learning specifically for shoeprint classification (Sapkota et al., 2020). The complex nature of shoeprint data, particularly in scenarios like the National Footwear Reference Collection (NFRC), necessitates advanced computational methods and robust algorithms (Iqbal et al., 2020). Integrating transfer learning (TL) and convolutional neural network (CNN) models holds promise for improving shoeprint classification (Yan et al., 2020), but further investigation is required.

Shoeprint classification extends beyond forensic investigations, finding broader applications in crime-related exploration (Liu et al., 2019). Therefore, addressing the research gap in this area is crucial. Conducting this research at Auckland University of Technology in New Zealand provides a unique opportunity, as there is limited research on crime scene shoeprint identification conducted in the country. This research has the potential to revitalize the field, enhance New

Zealand's research reputation, and benefit various stakeholders by advancing knowledge, improving forensic analysis techniques, and ensuring public safety.

Quantitative analysis reveals a research gap in shoeprint classification using machine learning. The numbers obtained from year-wise publication analysis show the following on research growth for shoeprint identification:

- The number of shoeprint classification papers increased from zero in 1990 to an average of 4 papers per year in 2010 (Farooq et al., 2021).
- The exhaustive research conducted in the shoeprint classification on Google Scholar is 92 articles. However, since the year 2010, there have been 76 studies conducted in this area.
- The United Kingdom has emerged as the leading contributor, followed by China and the USA (Zhan et al., 2019). However, New Zealand is not among the top ten countries in terms of contributions in this area.
- Prominent institutions like Queens University, Belfast, and University College, Dublin, have been actively involved in shoeprint classification research, creating a gap between leading institutions and others (Zhan et al., 2019). Conducting research at Auckland University of Technology will bridge this gap and establish the university as a critical player in shoeprint classification research.

The outcome of this research holds immense potential to revitalize the field of shoeprint image classification in forensic science. Several significant achievements can be realized by developing an improved deep learning model based on integrated convolutional neural networks and transfer learning (CNN-TL), benefiting various stakeholders and advancing the field.

1. **Advancing Knowledge:** The research will contribute to expanding the understanding of shoeprint classification by exploring the application of cutting-edge deep learning techniques. It will uncover new insights into the effectiveness of integrated convolutional neural networks and transfer learning for accurate shoeprint image classification. This understanding will lay the groundwork for future forensic science and image analysis advances.
2. **Improved Forensic Analysis Techniques:** The developed deep learning model has the potential to enhance the accuracy and efficiency of shoeprint image classification significantly. By leveraging the power of deep learning algorithms and transfer learning, the model can learn intricate patterns and features from shoeprint images, enabling more precise and reliable classification results. This improvement in forensic analysis techniques can expedite crime scene investigations, aid in identifying potential suspects, and strengthen the forensic evidence presented in court.
3. **Enhanced Research Reputation**: Conducting this research at Auckland University of Technology in New Zealand will elevate the country's research reputation in shoeprint classification. By filling the existing research gap and contributing novel insights, New Zealand can significantly contribute to forensic science and deepen its collaborations with international researchers and institutions. This enhanced research reputation will attract talented scholars, foster academic partnerships, and promote knowledge exchange.
4. **Practical Applications:** The outcomes of this research will have practical implications for organizations and forensic investigators involved in crime scene analysis. The improved deep learning model can be integrated into forensic analysis systems and software, aiding investigators in efficiently analyzing and classifying shoeprint evidence. This streamlined process can save time, increase accuracy, and enhance the

overall effectiveness of forensic investigations, ultimately contributing to the pursuit of justice and public safety.

Digital forensic science holds great promise and relevance for identifying and classifying shoeprints. Due to the prevalence of digital traces left at crime scenes, cutting-edge image analysis techniques like Convolutional Neural Networks (CNNs) can transform forensic investigations. Investigators can make vital linkages between suspects and crime scenes by accurately identifying and classifying shoeprints in digital pictures. As a result, it is easier to spot violators, and the value of the evidence in courts is increased. This technology offers a way for forensic analysis that is scientific and unbiased, in addition to speeding up and strengthening the effectiveness of investigations. It can use CNNs to resolve difficult situations, reveal underlying relationships, and uphold justice. A better and more secure society will ultimately come from categorizing and identifying shoeprints in the digital arena since it opens up new possibilities for international collaborations. Forensic specialists can exchange and analyze data overseas thanks to these changes. An innovative strategy that has the potential to considerably advance digital forensic science and reinforce the foundations of justice and truth is the use of digital image analysis in shoeprint identification.

2.2 Image Classification Using Machine Learning

Image classification is a common problem in image processing. It is a critical task in the field of computer vision. The type of image, such as a person, animal, or setting, must be identified. Image classification attempts to predict an input image's category based on features. Image classification problems can be solved using a variety of techniques such as the k Nearest Neighbor (kNN), Adaptive Boost (Ada Boost), Artificial Neural Network (ANN), and Support Vector Machine (SVM) (Kim et al., 2017). First, the k-NN classifier, a non-parametric machine learning technique, calculates the distance between the feature vectors of the input image (an image of an

unknown class) and the training image dataset. The input image is then assigned to one of its k-NN classes, where k is an integer (Lu & Weng, 2007). Adaptive Boost (AdaBoost) is an iterative learning algorithm that uses a training dataset and a "weak" learning algorithm to create a "strong" classifier (Viola & Jones, 2004). The "weak" classifier with the lowest classification error is chosen at each iterative stage. The authors used AdaBoost to select a small number of critical features from a large pool of candidate features. The AdaBoost algorithm was first proposed by (Freund & Schapire, 1997), who also demonstrated its effectiveness in improving the performance of poor classifiers. (Kim et al., 2012) proposed a new AdaBoost-based technique for multi-class image classification that integrates the results of multiple weak classifiers to provide high accuracy. Using AdaBoost and motion characteristics, (Viola et al., 2003) expanded the Viola-Jones algorithm to detect pedestrians.

Artificial Neural Networks (ANN), a computational model modeled after the brain, have been employed by numerous researchers. The composition and operation of the human brain inspire ANN. An ANN is a network of connected nodes that process and send information. The layers of an ANN are as follows: an input layer, one or more hidden layers, and an output layer (How & Chan, 2020). After one or more hidden layers have processed data from the external environment, the output layer produces the network's final output. As it learns, the ANN modifies the weights of the connections between neurons, narrowing the gap between the predicted and the actual output. *Training* is the procedure that is typically carried out with a well-known training dataset. By introducing nonlinearity into the model via the activation function, an ANN can learn more complex patterns in data. In order to produce the neuron's output, the activation function applies a non-linear function to the weighted sum of the neuron's inputs. ANNs use various types of activation functions. Some are depicted below:

ReLU: The Rectified Linear Unit function applies a linear function if the input is positive; if the input is negative, the output is zero. ReLU is a popular choice for deep learning models' hidden layers.

SoftMax: The SoftMax function is used in the output layer of a neural network to solve multi-class classification problems. It transforms neuron outputs into a probability distribution over potential class.

Sigmoid: The sigmoid function is a smooth, S-shaped curve that compresses its input to the range [0, 1]. It is frequently used in binary classification problems where the output should be a probability.

Tanh: Similar to the sigmoid function, the hyperbolic tangent (tanh) function compresses the input range to [-1, 1]. It can be used to solve classification and regression problems.

Based on their issues, other researchers have developed alternative ANN topologies. For example, the network can be used to classify photos after training. Therefore, a neural network is an effective classification technique. According to McCulloch and Pitts (1943), the artificial neuron is a necessary neural network component that allows information to be transmitted from one end to the other. A neural network is composed of artificial neurons linked by connections. Each neuron in a neural network is connected by several weights: w1, w2, w3, w4, and so on. The input values and the corresponding weight determine the activation of a neuron. However, it is essential to note that the number of weights per neuron can vary in different neural network architectures. The number of weights in each neuron depends on the connections between the neuron and the previous layer or input data (Zhang et al., 2008). This means that the number of weights associated with a neuron is generally determined by the number of connections it has rather than a fixed value. The flexibility of neural networks allows for varying numbers of weights per neuron, adapting to the complexity and requirements of the given problem. The summation function computes the weighted sum and then passes it through an activation function. The function of

activation mechanisms is to limit the magnitude of neuron output. When a neuron fires, the signal travels from the beginning to the end of the neuron and uses the activation function to compute the output values.

The following equation (1) represents the final classification of the input data, which is adopted from (Agatonovic-Kustrin & Beresford, 2000)

$$.y = f(\sum(w * x) + b) \qquad (1)$$

In equation (1), y represents the neuron's output, f is the activation function, w is the weight matrix, x is the input data (typically flattened from the feature maps), and b is the bias term. Rosenblatt (1957) developed the perceptron concept based on the supervised learning approach. As binary classifiers, perceptrons are used. A simple neuron classifies its input into one of two categories (Rosenblatt, 1957).

The following equation (2) represents the two-class classification which is adopted from (Dreiseitl & Ohno-Machado, 2002)

$$Y = +1\ wTx + b \qquad (2)$$

Textural features in traditional machine learning provide structural and dimensional information about the image and its intensity. The structural qualities of the image can be used to determine the area and its shape. Meanwhile, not all retrieved traits are necessarily significant. The main methods used for feature identification are Principal Component Analysis (PCA), Linear Discriminant Analysis (LDA), and Multidimensional Scaling (MDS) (Malathi & Srividya, 2015; Wang et al., 2019). Image classification methods include k-NN, SVM, Naive Bayes, Decision trees, and random forests.

Support Vector Machine (SVM) is a well-known pattern recognition and image classification method. It is intended to separate a set of training images into two classes, (x1, y1), (x2, y2),...,

(xn, yn), where xi is in Rd, d-dimensional feature space and yi is in {-1,+1}, the class label, with i=1.n (Chandra & Bedi, 2021). Based on a kernel function, SVM constructs the best-separating hyperplanes. All images with the feature vector on one side of the hyperplane are classified as class -1, while the others are classified as class +1. Vapnik and his colleagues developed the SVM theory in the 1990s (Cortes & Vapnik, 1995). SVM was derived from the neural network concept, or SVM is a mathematical extension of the neural network. SVM can distinguish between linear and non-linear data (Jayadeva et al., 2007). SVM performs classification by transforming the original training data into multidimensional space and constructing a higher-dimensional hyperplane. SVM is a mathematical learning scheme based on optimal hyperplanes (Cortes & Vapnik, 1995).

An SVM performs classification by generating a higher-dimensional hyperplane. This hyper-plane is represented by the decision planes. A decision plane distinguishes one type of data from another. Excellent non-linear mapping, for example, categorizes data in higher dimensional space by generating a hyper-plane (Kang et al., 2018). SVM seeks vector points, the support vector, that defines the decision border and creates a significant marginal separation between the classes (Guyon et al., 2002). SVM distinguishes the classes with the most significant marginal distance in the decision plane (Guyon et al., 2002; Kuo et al., 2014; Prajapati & Patle, 2010). The middle line shows the maximum-margin hyperplane. In other words, a boundary line that is the most significant distance from the nearest data point and separates the two classes would be chosen (Krizhevsky et al., 2017). In two-class problems, Vapnik's theory establishes the decision boundary or hyper-plane for classification. According to the literature, SVM has been used in various applications (Sheykhmousa et al., 2020). For example, SVM has been used to remove uncalibrated noise from hyperspectral data by (Chi et al., 2008) and (Tang et al., 2015). According to research, Peng et al. (2015) produced comparable results but used semi-supervised ensemble

projection. The same is valid for chemical pattern data, for which numerous feature extraction methods have been investigated (Priya et al., 2012; Tao et al., 2009; Yang et al., 2011).

Knowledge-based classification is the use of expert information and rules or the generation of rules from data seen, and it is gaining popularity. As a result, using multiple classifiers in tandem has recently received much attention. Several researchers, for example, combine the neural networks (NN) classifier (Deng et al., 2009), the SVM classifier (Chandra & Bedi, 2021), or the AdaBoost classifier for image classification. This report's authors (Lu & Weng, 2007) attempted to integrate various approaches to image categorization, such as Support Vector Machines (SVM) and Artificial Neural Networks (ANN) (Abutaleb et al., 2021; Deng et al., 2018).

This task is problematic because it requires the model to understand the complex relationship between an image's pixels and the category to which they belong. Image classification has traditionally been approached using traditional machine learning techniques such as Support Vector Machines (SVMs) and Random Forests (Himeur et al., 2022; Priore et al., 2018).

The Random Forest algorithm, an ensemble learning method, has gained favor because of its effectiveness in processing complex image data and producing reliable classification results. According to Breiman (2001), the Random Forest algorithm is an ensemble learning method that mixes numerous decision trees to make correct predictions. Its application in image classification has demonstrated remarkable performance, making it a valuable tool in various domains, such as medical imaging, object recognition, and remote sensing (Li et al., 2016; Zheng et al., 2018; Wang et al., 2020).

The Random Forest algorithm creates an ensemble of decision trees, each trained on a random subset of the training data and a random subset of input features (Breiman, 2001). This combination of randomness in data sampling and feature selection helps mitigate overfitting and enhances the algorithm's generalization capabilities. During the training phase, each decision tree

independently learns to classify images based on different subsets of features, and the final classification is determined by aggregating the predictions of all individual trees through voting or averaging (Breiman, 2001; Liaw & Wiener, 2002).

The Random Forest algorithm offers several advantages in image classification tasks. Firstly, it can effectively handle high-dimensional image data, such as texture, color, and shape features, without requiring dimensionality reduction techniques (Li et al., 2016). Additionally, the ensemble nature of Random Forest makes it less prone to overfitting compared to individual decision trees, ensuring robust and reliable classification results (Breiman, 2001). Moreover, Random Forest provides insights into feature importance, enabling researchers to identify key image characteristics that contribute significantly to the classification task (Liaw & Wiener, 2002). Lastly, the algorithm is computationally efficient and capable of processing large datasets, making it suitable for real-time and high-throughput image classification applications (Wang et al., 2020).

KNNs, SVMs, and Random Forests have widely used machine learning algorithms. They represent several machine learning approaches (instance-based, discriminative, and ensemble-based). The algorithms also include a vast library and implementations in various computer languages, making them an excellent starting point for many machine-learning projects. However, these methods need to be improved in their ability to capture image nuances and are frequently incapable of scaling to large datasets. The process begins with feature extraction, extracting meaningful information from an image. This entails recognizing patterns, edges, lines, and shapes in the image (Copiaco et al., 2023). Next, this data generates a feature vector containing image information. The feature vector is then fed into a classifier, which classifies the image into one of several categories. Traditional image classification methods such as support vector machines (SVMs), k-nearest neighbors (KNNs), and decision trees were used in the past.

These methods have been successful, but their accuracy could be improved. As a result, deep learning was used to improve image classification accuracy (Siddiqui et al., 2022).

2.3 Image Classification Using Deep Learning

(Krizhevsky et al., 2017) Presented deep convolutional neural networks (CNNs) for image classification using millions of labeled images and demonstrated their efficacy on the ImageNet dataset, obtaining a top-5 error rate of 15.3%. Similarly, (Simonyan & Zisserman, 2014) proposed VGGNet, a deep convolutional neural network architecture with up to 19 layers that achieved state-of-the-art performance on the ImageNet dataset with a top-5 error rate of 7.3%. Furthermore, (He et al., 2015) introduced residual learning, a technique for training deep CNNs, and proposed the ResNet architecture, demonstrating its effectiveness on the ImageNet dataset, achieving a top-5 error rate of 3.6%, ultimately outperforming VGGNet on the ImageNet dataset. (Howard et al., 2017), on the other hand, introduce the DenseNet architecture, which connects each layer to every other layer in a feed-forward manner (Cheng et al., 2019). This architecture enables better gradient flow and feature reuse, improving image classification performance and yielding state-of-the-art accuracy on various image classification benchmarks.

Tan and Le (2019) proposed an EfficientNets, a family of CNNs, a non-local block that allows CNN to capture long-range dependencies between pixels in an image. This technique achieved cutting-edge performance on various image classification tasks, achieving cutting-edge accuracy on the ImageNet dataset while significantly smaller and faster than previous models. Finally, (Khan et al., 2022) review recent advances in using transformers, a neural network architecture developed initially for natural language processing, image classification, and other vision tasks. Vision Transformers are proposed by (Ranftl et al., 2021). This new architecture uses self-attention mechanisms instead of the typical CNNs for image classification, achieving state-of-the-art accuracy on the ImageNet dataset. According to the literature, the following factors are critical to consider and play a significant role in the performance of a model.

Number of Layers: The number of layers in a convolutional neural network significantly impacts its performance. Deeper CNNs generally outperform shallower ones at the expense of increased computation and training time.

Filter Size: The size of the filter determines the size of the CNN's (John, 2023) receptive field and the amount of spatial information it can capture. Larger filters can capture more context while increasing the parameters and computations needed.

Pooling: Pooling layers reduce the spatial dimensions of feature maps and extract more reliable information. Although recent research has revealed that average pooling can occasionally result in higher performance, max pooling is the most popular pooling operation in CNNs.

Max pooling: Reduces the spatial dimensions of the input by taking the maximum value within a predefined window or filter (Yu et al., 2014). This operation helps to down-sample the feature maps, retaining the most salient or essential features while discarding less relevant details. By reducing the spatial dimensions, max pooling can make the model more robust to translation and provide a form of spatial invariance.

Average Pooling: Reduces the spatial dimensions of the input but calculates the average value within the pooling window (Yu et al., 2014). This operation preserves more spatial information than max pooling and can help smooth out noise in the feature maps, providing a more general representation of the input.

Normalization: A popular technique for normalizing the activations in CNNs that can improve training and generalization performance is batch normalization (Yu et al., 2014). Other normalization methods, such as group and layer normalization, have also been proposed.

Batch normalization: A technique that addresses this problem by normalizing the activations of intermediate layers (Yu et al., 2014). On mini-batches, it works by removing the batch mean and dividing it by the batch standard deviation. This aids in the stabilization of the learning process and accelerates convergence. Moreover, batch normalization is regularization and reduces the dependence on specific initialization choices. Despite its effectiveness, batch normalization has

limitations. It requires the mini-batch size to be sufficiently large, and it may not work well for small batch sizes or non-independent and identically distributed data.

Group normalization: An alternative to batch normalization that normalizes the inputs within each group or subset of channels (Yu et al., 2014). Unlike batch normalization, group normalization does not depend on batch size. It divides the channels into groups and normalizes the inputs within each group separately. Group normalization has several advantages over batch normalization. It is more robust to small batch sizes and non-independent and identically distributed data. Moreover, it does not suffer from the internal covariate shift problem caused by the variation of statistics across different channels. However, group normalization may only work well for deep networks with a small number of groups.

Layer normalization: An effective method for coping with vanishing or exploding gradients in recurrent neural networks (RNNs) (Yu et al., 2014). Layer normalization normalizes the inputs within a layer by computing the mean and standard deviation across the features of each sample. This technique benefits RNNs where the batch size may vary across time steps, providing a more stable training process.

Skip Connections: Help to avoid the vanishing gradient problem by facilitating gradient flow. Popular CNN architectures that use skip connections to improve performance are ResNet and DenseNet.

Depth-wise Separable Convolution: Reduces the number of parameters in a CNN by performing a depth-wise convolution followed by a point-wise convolution. It has been successfully demonstrated to reduce overfitting and improve generalization abilities.

Pretraining: Pretraining a CNN on a large dataset can help it perform better on a smaller dataset by providing better weight initialization. *Transfer learning* is a popular technique that involves fine-tuning a previously trained CNN as a feature extractor on a new dataset.

Textural features are closely associated with shoeprint classification and can be effectively utilized with deep learning models (Afifi et al., 2020; Doborjeh et al., 2021). Shoeprints exhibit distinct patterns and textures, providing valuable information for distinguishing different shoe types or individuals. Deep learning models, such as Convolutional Neural Networks (CNNs), have shown an extraordinary ability to learn complicated features from raw data (Afifi et al., 2020).

These models can extract and analyze textures at different scales by incorporating textural features into the deep learning framework, leading to more accurate classification results (Doborjeh et al., 2021). This integration allows for a comprehensive analysis of shoeprints, capturing intricate details and variations that might not be perceptible to the human eye. By leveraging the strengths of textural features and deep learning, shoeprint classification systems can achieve enhanced performance (Afifi et al., 2020; Doborjeh et al., 2021). Combining textural features and deep learning offers a powerful approach for accurate and reliable shoeprint analysis and identification.

Deep learning has transformed computer vision over the past few years, enabling effective and precise solutions for problems like object detection, semantic segmentation, and image categorization. For example, deep learning has been used (Varun, 2020) to achieve state-of-the-art results in image classification, which involves assigning an input image to a predefined set of categories (Khan et al., 2022). With the increasing popularity of deep learning, it has become possible to use more sophisticated methods for image classification (Liu et al., 2021). *Deep learning* is an artificial intelligence subset that uses enormous neural networks for pattern recognition and machine learning. This thesis uses a pre-trained model to explore deep learning for image classification and transfer learning. The process of image classification is complex and involves a variety of techniques.

2.3.1 A Fundamental Operation in Signal Processing and Deep Learning

Convolution is essential in many applications, including image and audio processing, computer vision, natural language processing, and deep learning. In signal processing and deep learning, the convolution operation plays a fundamental role, particularly in extracting valuable features from input data (LeCun et al., 2015).

Mathematically, the convolution operation can be defined as applying a filter or kernel to an input signal or image (LeCun et al., 2015). The result is a meaningful feature extracted from the input signal.

The following equation (3) represents the convolutional operation defined by the symbol *, which is adopted from (Goodfellow et al., 2016)

$$(x * h)(n) = \sum[x(k) * h(n-k)] \qquad (3)$$

The above formula (3) implies that given an input signal or image, denoted as x, and a filter or kernel, denoted as h, the convolution operation's result is represented by the symbol * and defined as (x * h)(n). Here, (x * h)(n) denotes the value of the convolution operation at index n, k represents the index variable over which the summation is performed, x(k) represents the value of the input signal or image at index k, and h(n-k) represents the value of the filter or kernel at index n-k (Goodfellow et al., 2016).

An example of a 1D convolution can be used to understand the concept of convolution better. Suppose we have an input signal x = [1, 2, 3, 4, 5] and a filter h = [0.5, 0.7, 0.2]. The convolution operation can be performed by sliding the filter over the input signal and computing the sum of element-wise products at each position.

The following equations (4, 5, and 6) represent examples of the sum of element-wise products at each position, which is adopted from (Goodfellow et al., 2016)

$$(x * h)(0) = (1 * 0.5) + (2 * 0.7) + (3 * 0.2) = 1.9 \qquad (4)$$

$$(x * h)(1) = (1 * 0.2) + (2 * 0.5) + (3 * 0.7) + (4 * 0.2) = 4.5 \quad (5)$$

$$(x * h)(2) = (2 * 0.2) + (3 * 0.5) + (4 * 0.7) + (5 * 0.2) = 6.2 \quad (6)$$

Convolution is not limited to 1D signals but can be extended to higher-dimensional signals such as images. The filter is convolved in multidimensional filters with the input signal along each dimension.

Convolution is essential in deep learning and is used in convolutional neural networks (CNNs) to extract features from input data. Numerous research has proved the efficiency of convolutional neural networks in various tasks, such as image categorization, object detection, and speech recognition. (Goodfellow et al., 2016).

CHAPTER 3: Methodology

3.1 Dataset

Two datasets have been used for this research: FID - Footwear Impression Database/FID-300 (Kortylewski et al., 2015) and 2D Footwear Outsole Impressions (Soyoung & Alicia, 2020). The dataset consisted of 1918 images. The images varied in size, and none were colored. The first dataset included 1236 full-shoeprints and 194 half-shoeprints, while the second dataset included 488 partial-shoeprints. Furthermore, it was discovered that the datasets require more consistency. Images of half-shoeprints can be found in the full-shoeprint database, and vice versa. The images were incorrectly labeled, and the number of images was preserved by downloading the difference from a Google search directed to a different site.

Furthermore, images in the full and half-shoeprints were discovered with patterns not visible to the naked eye. It was possible to identify the images that did not follow their patterns by enlarging them. These photos were replaced with more precise and consistent ones to reduce inconsistency and errors in the final product.

3.2 Data Analysis

This experiment employs several image classification tools. Python's tensor-flow library was used as a research tool in the CNN classification and CNN-TL. As a result, the new proposed CNN-TL model was also created in Python. Furthermore, TensorFlow is the primary research tool in the new model. Among these tools are numerous highly efficient Python libraries such as numpy, opencv, matplot lib, and scikit, which can aid in the analysis of CNN and CNN-TL.

3.3 Experimental Design

Multiple experiments have been designed to test the research hypotheses and classify the shoeprints into two classes using different CNN-TL models (full and half-shoeprints). The datasets will first be tested in various CNN-TL models with their respective pre-trained models. The pre-trained model will then be fine-tuned and tested using the CNN-TL models that have been developed. The **hypothesis** is that an integrated convolution neural network with a transfer learning model can outperform a single convolutional neural network model in shoeprint identification.

RQ1. To what extent do deep learning models, including CNN, lead to improving the accuracy of shoeprint classification?

The study investigated the performance of deep learning models, including Convolutional Neural Networks (CNN), in improving the accuracy of shoeprint classification. It utilized a publicly available dataset of shoeprint images and implemented a CNN model to classify them into half-shoeprint and full-shoeprint. The accuracy of the CNN model was evaluated, and it achieved an accuracy of 96.17%. This demonstrated that deep learning models, specifically CNN, can significantly enhance the accuracy of shoeprint classification compared to traditional machine learning techniques.

RQ2. What are the advantages and disadvantages of using CNN than CNN-TL on shoeprint identification, especially in terms of performance?

The study compared the advantages and disadvantages of using CNN versus CNN-TL (Transfer Learning) for shoeprint identification, particularly in performance. CNN-TL refers to using pre-trained models, such as Inception V3, VGG16, and ResNet50, which were fine-tuned for the shoeprint classification task. It has been observed that while CNN achieved a high accuracy of 96.17%, the accuracy of the CNN-TL models surpassed it. Specifically, the Inception V3 model

achieved an accuracy of 92.19%, VGG16 achieved 96.88%, and ResNet50 achieved the highest accuracy of 97.14%. This comparison highlighted that CNN-TL models outperformed CNN in accuracy, indicating the advantage of pre-trained models for shoeprint identification tasks.

RQ3. How can the CNN-TL performance be improved further with fine-tuning?

The study applied fine-tuning to address RQ3 and further enhance the performance of the CNN-TL models. Fine-tuning involves adjusting the pre-trained models to improve performance by training them on the specific shoeprint dataset. By fine-tuning the pre-trained models, the accuracy of the CNN-TL models was significantly improved. The ResNet50 model achieved the highest accuracy of 97.14% after fine-tuning. This demonstrated that fine-tuning the pre-trained models can further enhance the performance of CNN-TL in shoeprint classification.

3.4 Development Testing and Validation of Neural Networks

In the methodology, the stage following data collection is the instrument development test and instrument validation. In this paper, a convolutional neural network (CNN) and three different CNN-TL models are designed. The two datasets are combined into one dataset during the test and validation phase to evaluate different methods for studying the performance and characteristics of each model. This experiment divides the data set into training and validation sets for evaluation. The CNN model's performance will be compared to CNN-TL models.

3.5 Proposed Methods for Classification of Shoeprint Images

This thesis proposes a methodology that includes two main data modeling methods to classify shoeprint images and address the research questions discussed in Chapter 1 (Introduction). The first method is based on shoeprint modeling with CNN, while the second is based on integrating CNN and transfer learning (CNN-TL). Figure 3 depicts a diagram illustrating the proposed

methodology. The employed TL models are based on three pre-trained models, as detailed in the methodology procedure below.

- Phase 1: Input data: sampling and pre-processing of shoeprint images.
- Phase 2: CNN shoeprint model.
- Phase 3: CNN-TL model with pre-trained (Inception V3) model.
- Phase 4: CNN-TL model with pre-trained (VGG16) model.
- Phase 5: CNN-TL model with pre-trained (ResNet50) model.

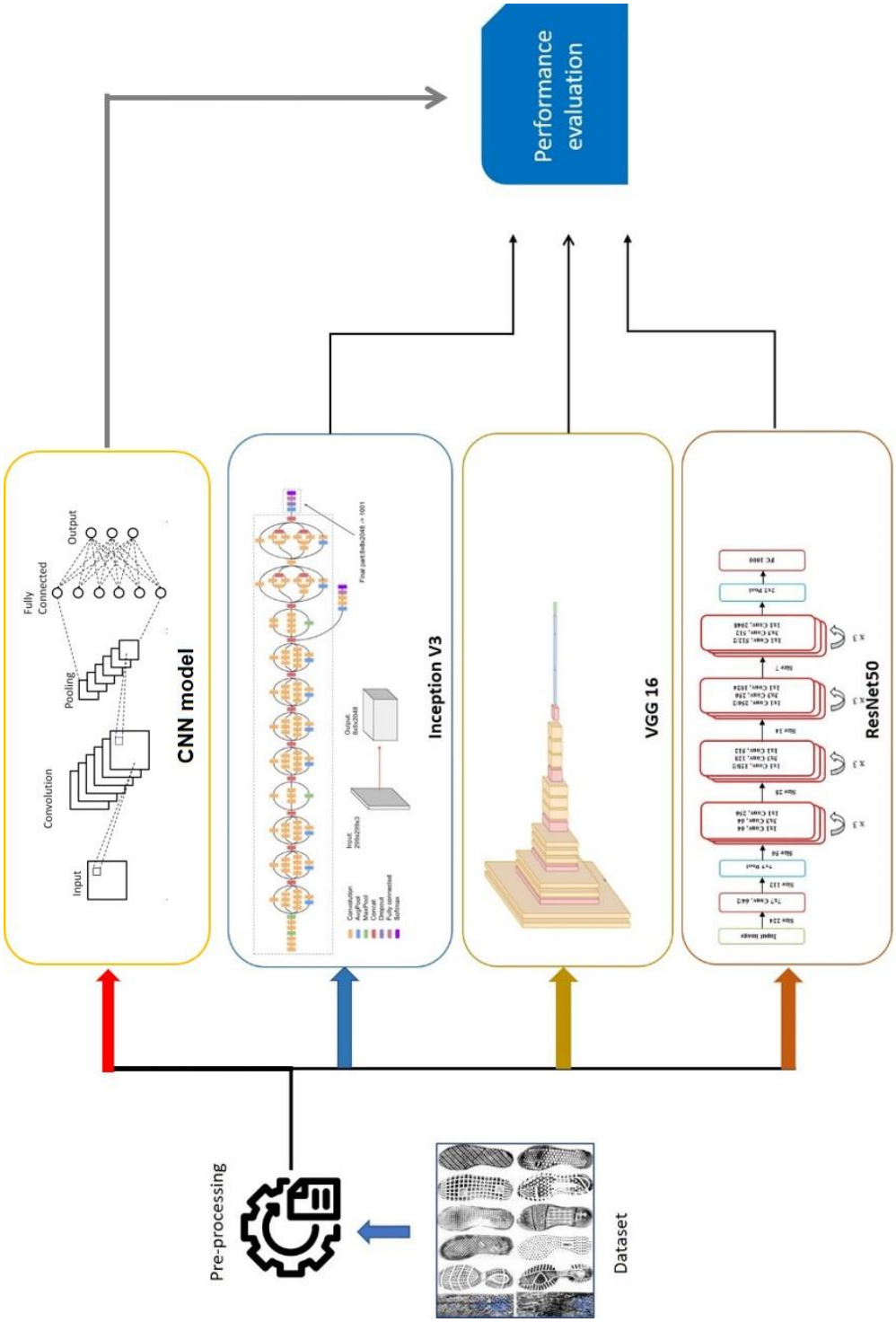

Figure 3. Illustration of the proposed methodology based on different pre-trained models: Inception V3, ResNet50, and VGG16.

Figure 3 depicts the shoeprint datasets under consideration, which have been pre-processed by manually removing the noisy data.

3.5.1 Shoeprint Pre-processing

The datasets used were discussed in detail in section 3.1 Dataset; specifically, it combines full and half-shoeprints. As a result, manually labeling the shoeprints is necessary, albeit time-consuming. Three different pre-trained models were used after the pre-processing steps. In the following sections, the pre-trained models are explained one by one. The model's various stages have been combined into a single diagram to depict the steps precisely. However, the diagrams have been split later in the section to understand better and present all of its components module-wise. While exploring the dataset, it was discovered that some full-shoeprint images were mixed with half-shoeprint images. This would have resulted in a lower accuracy percentage because the model would have classified the full-shoeprint as a half-shoeprint or vice versa, resulting in more false positives. Aside from the interchanged image, no shoeprint patterns were found in noisy images. Blurred images were removed from the dataset because they could skew the results by acting as an outlier.

3.5.2 Shoeprint Classification using CNN

The first experiment was carried out by building a new model from the ground up using deep learning libraries. The OS library was used to locate each image in the partial and complete categories. Only the locations in the PIL library's "Image" function were used to import all the images and place them in a single NumPy array. Images were reviewed as they were imported to decrease training time. The images were scaled down to 150x150 pixels, and then the color channels were transformed from BGRA to BGR using the Python OpenCV library. The data, including full and half-shoeprints, were compiled into a single file. It was also labeled "TensorFlow library." The data was then divided into 70:30 portions for training and testing by the sci-kit-learn module. TensorFlow, an open-source toolset, simplifies the creation of deep

learning networks. For modeling, it supports a wide range of neural network designs, including CNN and dense networks. A sequential model served as the starting point.

The model was given an optimized conv2d layer with 128 filters and a size 3x3. The imported image data, 150x150x3, corresponds to the input layer, and ReLU has been used as an activation function. Batch normalization was used after each convolutional layer to make each layer of the CNN network independent of the others. A 2D max-pooling with a 2x2 filter size layer was used to extract only significant characteristics. A dropout layer with a 30% probability is included to prevent the network from training an overfitted model. The same layers have been implemented again, but the number of filters has been reduced to 32. Rather than TensorFlow defining the input size and using the previous layer's output as the input for the next layer, the input size has been specified. Following the layers is a flattened layer that prepares the extracted features for use in a neural network. A neural network is composed of a 32-neuron layer and a 2-neuron output layer (presenting two classes: full and half-shoeprints) triggered by SoftMax. As a loss function, Adam was used as categorical cross-entropy and optimized the model with a learning rate of 0.001 and an initial decay rate of 0.1. The model was then trained in a memory-restricted environment for 20 epochs with 16 batches, yielding a classification accuracy of 91.48%.

3.5.3 Shoeprint Classification using CNN-TL (Inception V3)

The Inception V3 model is the first to be used. Once the pre-trained model has been trained with the model created, the accuracy value can be obtained through fine-tuning, which the performance evaluation blocks will represent in the following illustrations. The outcome can then be saved. Furthermore, a different pre-trained model VGG16 is investigated using the same dataset but with different parameters (as discussed later). The pre-trained VGG16 model was then used to classify the available dataset. The accuracy value can be obtained and saved again for future evaluation and comparison. The model will be trained again using a different pre-trained model, ResNet50.

The evaluation parameter value is saved here, and an accuracy value is obtained. As a result, the accuracy metrics values for the various models are obtained and can be compared to the model's performance. Then a decision can be made about which model is superior to the others.

Applying the three pre-trained models is an excellent example of transfer learning in action. A model built for one task is utilized as the foundation for another using the machine learning (ML) method known as transfer learning (TL). Transfer learning is gaining popularity in artificial intelligence because it can reduce training times, improve model accuracy, and reduce the likelihood of overfitting. For example, neural networks use one input node for each entry to transform an input image or feature vector through several hidden layers, frequently utilizing non-linear activation functions. Each neuron in the buried layer is wholly linked to every other neuron in the layer below it. The final layer of a neural network, known as the "output layer" (Machine Learning Glossary, 2023), displays the network's final output classifications and is fully connected. However, as the image size grows, neural networks that operate solely on raw pixel intensities:

- It needs to be scaled better.
- It needs to be more accurate.

Compared to traditional image processing approaches like Support Vector Machine (SVM), logistic regression, and similar ML algorithms, a CNN or neural network can occasionally outperform humans. CNN is used to extract visual properties when a neural network is taught to differentiate visual attributes rather than the overall pattern of an image. Python was used to create a model that can distinguish between partial and complete shoeprints. Deep-learning libraries in Python include Tensorflow, Keras, NumPy, OS, Scikit-Learn, and pandas. PIL and OpenCV were used to perform image processing.

The disadvantage of using CNN is that it can only provide significantly high accuracy when fed large amounts of data. Furthermore, gathering the data is a significant undertaking. Transfer

learning on premade for feature extraction and tested deep learning networks with CNN were used to address this issue. Discriminative layers are included in the premade networks. To make the model more efficient, the models only used a feature extractor and defined specific dense layers to classify different classes. The Inception V3 network model was used for transfer learning. The Inception V3 deep-learning neural network was trained using the image-net dataset (Deng et al., 2009). Its trained model is easily accessible. Inception was created in 2014 to compete in the ImageNet Large Scale Visual Recognition Competition. It defeated last year's winner ZFNet, by decreasing its error rate from 11.2% to 6.67%. Before 2014, the winner of ILSVRC was less accurate than Inception, and the winner after 2014 was either deep and heavy like ResNet50 or light, but their accuracy needed to be higher (GoogleNet). With only 23 million features to calculate, accuracy and efficiency are balanced. Overfitting is a risk with profound networks.

Furthermore, it is computationally expensive. Inception V2 and Inception V3 were introduced to address this issue. Several enhancements were proposed that increased accuracy while decreasing computational complexity. As illustrated in Figure 4 with Inception V2, several changes were made to factorize more extensive convolutions into smaller ones. For example, two 3x3 convolutions replaced a 5x5 convolution.

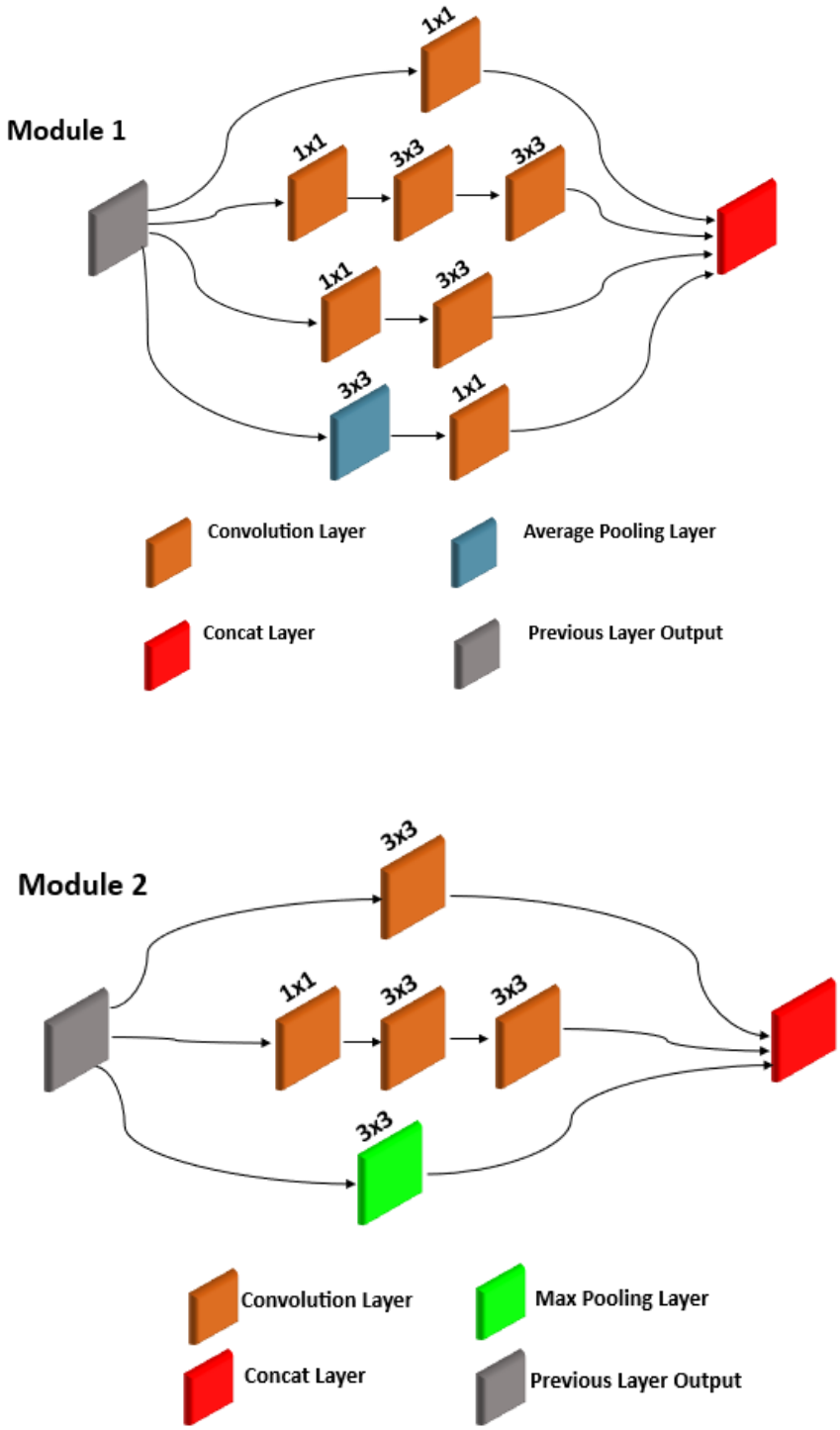

Figure 4. Factorization into a smaller convolution in Inception V2

The factorization of more extensive convolutions into smaller ones resulted in a 28% relative gain. By incorporating spatial factorization into asymmetric convolution, they could reduce

computing costs. For example, when a 3x3 convolution layer was transformed into a 3x1 and a 1x3 convolution layer, the computation cost was reduced by 33%.

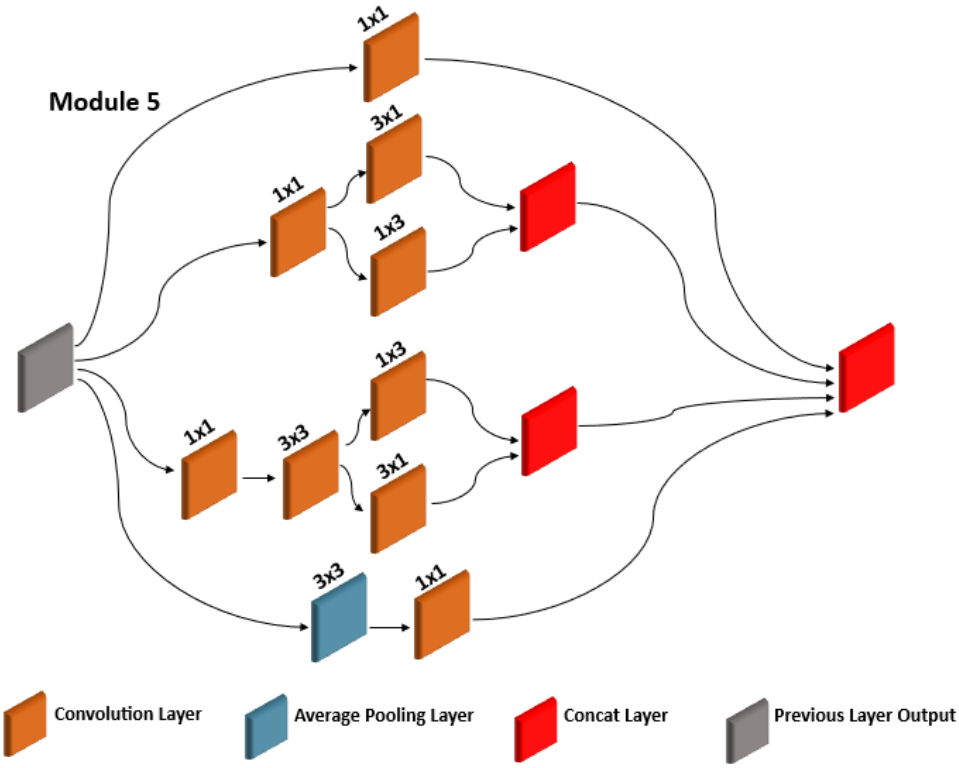

Figure 5. Factorization into an asymmetric convolution module 5 of Inception V3

An auxiliary classifier is used to combat the vanishing gradient problem. Furthermore, finally, the grid size was reduced from x*x with y filter to x/2* x/2 with 2y filters. A comparison of the factorization difference can be found in Figures 5 and 6. All the changes from V2 were implemented in Inception V3 by changing the optimizer function to RMSprop. As shown in Figure 6, it also factorized the 7x7 convolution layer.

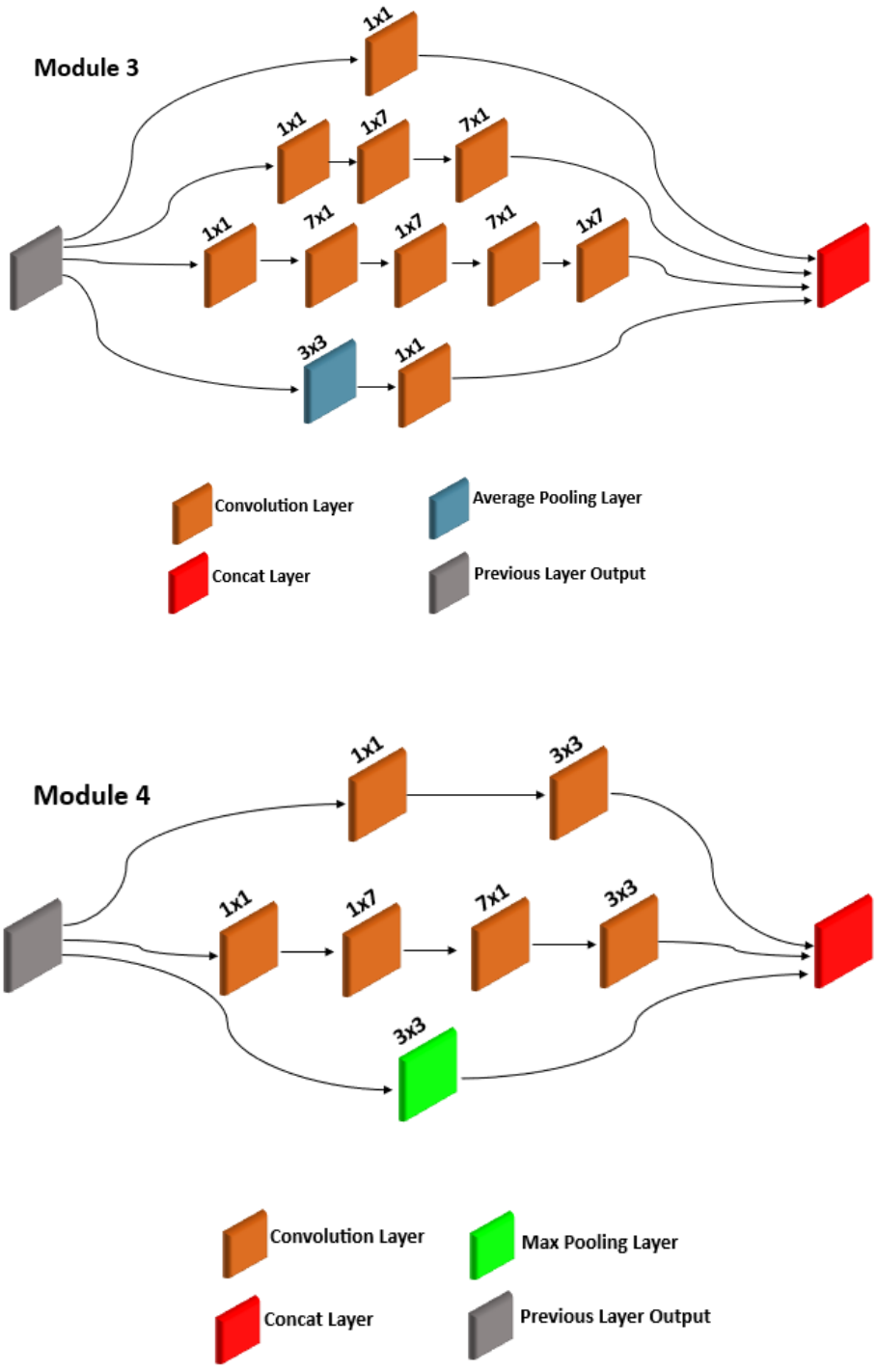

Figure 6. 7x7 Convolution layer factorized in Inception V3

In addition, Inception V3 introduced batch normalization in the auxiliary classifier and label smoothing, preventing the network from becoming overly confident about the class. As a result,

it outperforms its predecessor in terms of accuracy and computational power. As illustrated in Figure 7, Inception V3 serves as a well-balanced network for running on a computer without a massive GPU; it can also run on some moderately powerful CPUs of a mobile computer. It is effective enough to be used by a laptop without a GPU, as opposed to another network that competed in the ILSVRC competition that can achieve this level of precision without sacrificing efficiency for running on handheld mobile devices. With only 23 million parameters, it offered a 95% accuracy on the ImageNet dataset. It is more efficient than a typical CNN network due to its more complicated and optimized architecture. It can learn and detect more patterns while maintaining efficiency. However, Inception has to be trained on an extensive dataset for better accuracy.

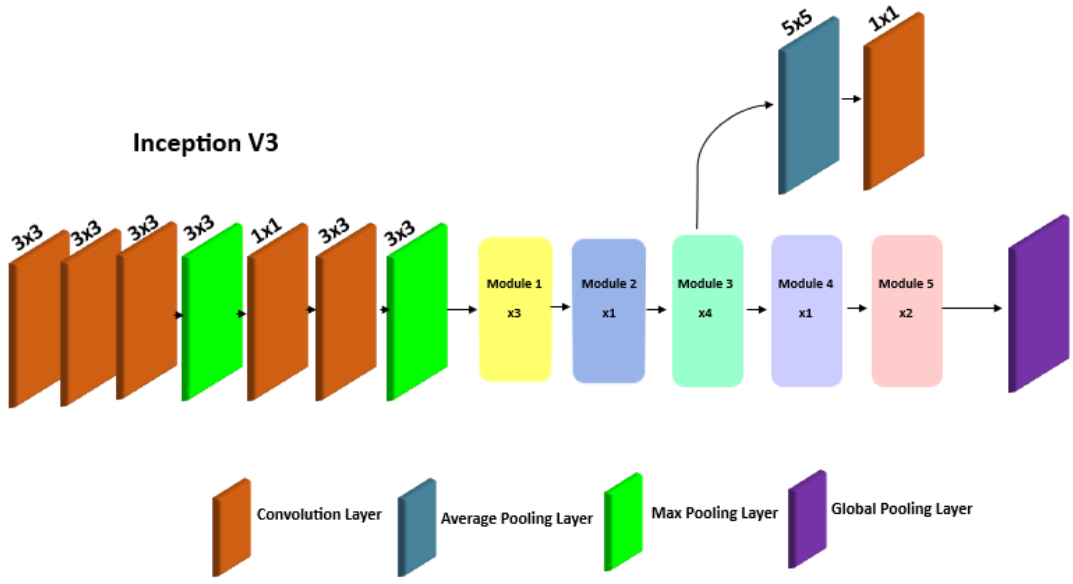

Figure 7. The architecture of Inception V3

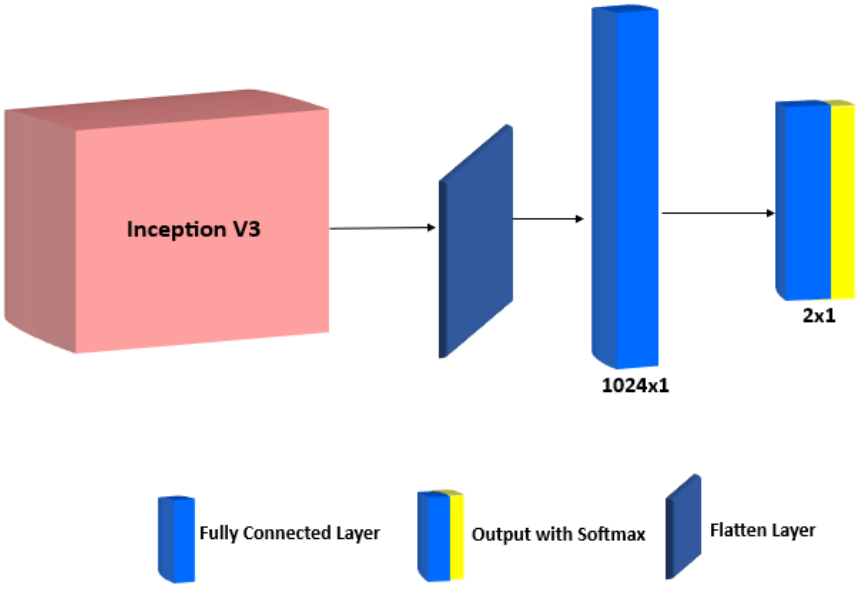

Figure 8. The own-defined discriminative network architecture of the Inception V3 model used

With large datasets, transfer learning has been used to address issues of varying information locations and sizes. However, because of the wide variation in information location, selecting the appropriate kernel size for the convolution operation becomes difficult. A larger kernel is preferred for globally distributed information, whereas a smaller kernel is preferred for locally distributed information. Because experiments require a dataset with different parts and varying information to train the model on variation in the location of the information, a transfer learning technique is preferred for image classification. As a result, a significant amount of data is required. To address this issue, a transfer learning technique is utilized in which the model is trained to recognize varied locations and amounts of information on an extensive dataset that does not have to be equivalent to the ILSVRC dataset and may be used to train the model against. The model's weight can then be fine-tuned and trained on smaller datasets to improve accuracy. A model pre-trained on a large dataset can produce better and more accurate results when defining features than one trained on a small dataset. This is due to its ability to recognize information of varying sizes and locations in an image. The pre-trained model weights are loaded and updated

to the model dataset to increase accuracy while analyzing the dataset. A pre-trained model was employed because it was trained and evaluated using the ILSVRC dataset.

Transfer learning employs a pre-trained model created on a massive dataset with varying sizes and locations of the information to be identified. It was trained using ImageNet's million photographs (1000 classes and 1000 images of each class). The Keras module in Python was used to load the pre-trained model. The model needed to be tweaked to recognize the traits in the dataset. On the ImageNet dataset, the pre-trained model was 95% accurate in classifying 1000 classes of generic things. A model that has been pre-trained on a large dataset is significantly more accurate than one that has been trained on a small dataset in defining features. The pre-trained model weights are loaded and adjusted to the small dataset to improve the model's accuracy. The data pre-processing for the transfer learning strategy was similar to the first experiment. The model required a few adjustments in Inception to make it more efficient; the premade networks include Discriminative layers. The models only used a feature extractor and defined dense layers to classify different classes in this experiment. By training only, the last 17 layers, the trainable parameters were set to around 38.5 million out of 47.5 million. The ImageNet model's weights were also modified because they were trained on a large dataset.

A flatten layer was added to the model, which can feed the Inception V3 Feature extractor's generated features to the Dense layer with 1024 neurons and ReLU activation. In addition, a 2-neuron layer with a SoftMax function was defined for the output. As a result, when using Transfer learning techniques during backpropagation, the model did not require a significant change in weight values. Backpropagation is a method for modifying the weights of a neural network based on the difference between expected and actual values. An RMSprop optimizer with a learning rate of 0.00005 was used to minimize the eight corrections during backpropagation. The loss function was categorical cross-entropy, and the model's accuracy levels could be used to assess

it. The model was trained for 200 epochs with a batch size of 32 and a training: testing ratio of 80:20. The dataset used to test the model had an accuracy of 89%.

At this point, the model was tested by adjusting various parameters, such as the learning rate and the number of nodes in the fully connected layer. However, it needed to exceed 90% accuracy. Further examination of the dataset revealed that the problem involved some full-shoeprint images mixed with half-shoeprint images. This error was confirmed through accuracy and previous test results. The dataset needed to be cleaned before being re-run against the model. Its accuracy fell from 91.43% to 89% as the transfer learning models' accuracy increased from 89% to 92%. After evaluating the dataset and correcting it, the results gathered were of increased accuracy. A transfer learning method is superior to training a new network from scratch with a small dataset. An improvement of 3% accuracy was found. The increased accuracy was due to two factors: the depth of the Inception V3 network is greater than that of a custom-made network. The second reason was transfer learning with Inception V3, which eliminated the need for a large dataset while also training the model. It made it recognize the variable size and location feature. Transfer learning alleviates the dataset problem by allowing the model to be trained on similar data and used for different use cases with improved accuracy.

Table II. Kernel and input size in each layer of Inception V3

Type	Kernal Size	Stride	Input
Convolutional	3x3	2	150x150x3
Convolutional	3x3	1	74x74x32
Convolutional	3x3	1	72x72x32
Max Pooling	3x3	2	72x72x64
Convolutional	3x3	1	35x35x64
Convolutional	3x3	2	35x35x80
Max Pooling	3x3	1	33x33x192
Inception 1	(Module 1) x3 + Module 2	-	16x16x192
Inception 2	(Module 1) x3 + Module 2	-	16x16x288
Inception 3	(Module 1) x3 + Module 2	-	7x7x768
Flatten	1x1	-	37632
Fully Connected	1x1x1024	-	1x1x37632

SoftMax	1x1x2	-	1x1x1024
Total Params (approx.)	74,235,000		
Trainable Params (approx.)	53,483,000		
Non-Trainable Params (approx.)	20,752,000		

Table III. Kernel and input size in each layer of the custom-made network

Type	Kernal Size	Stride	Input
Convolutional	3x3	2	150x150x3
Max Pooling	2x2	2	148x148x128
Dropout	-	-	14x14x128
Convolutional	3x3	2	74x74x128
Max Pooling	2x2	2	72x72x64
Dropout	-	-	36x36x64
Flatten	-	-	82944
Fully Connected	1x1x32	-	1x1x82944
SoftMax	1x1x2	-	1x1x32

Table IV. Description of the table with different parameters and corresponding model name

Layer	Type	# of Filters	Filter Size	Stride	Padding	Input Shape	Output Shape	# of Parameters (approx.)
1	Convolutional	64	3x3	1	1	224x224x3	224x224x64	2,000
2	Convolutional	64	3x3	1	1	224x224x64	224x224x64	37,000
3	Max Pooling	-	2x2	2	0	224x224x64	112x112x64	0
4	Convolutional	128	3x3	1	1	112x112x64	112x112x128	74,000
5	Convolutional	128	3x3	1	1	112x112x128	112x112x128	148,000
6	Max Pooling	-	2x2	2	0	112x112x128	56x56x128	0
7	Convolutional	256	3x3	1	1	56x56x128	56x56x256	295,000
8	Convolutional	256	3x3	1	1	56x56x256	56x56x256	590,000
9	Convolutional	256	3x3	1	1	56x56x256	56x56x256	590,000
10	Max Pooling	-	2x2	2	0	56x56x256	28x28x256	0

11	Convolutional	512	3x3	1	1	28x28x256	28x28x512	1,180,000
12	Convolutional	512	3x3	1	1	28x28x512	28x28x512	2,360,000
13	Convolutional	512	3x3	1	1	28x28x512	28x28x512	2,360,000
14	Max Pooling	-	2x2	2	0	28x28x512	14x14x512	0
15	Convolutional	512	3x3	1	1	14x14x512	14x14x512	2,360,000
16	Convolutional	512	3x3	1	1	14x14x512	14x14x512	2,360,000
17	Convolutional	512	3x3	1	1	14x14x512	14x14x512	2,360,000
18	Max Pooling	-	2x2	2	0	14x14x512	7x7x512	0
19	Fully Connected	4096	-	-	-	25088	4096	102,765,000
20	Fully Connected	4096	-	-	-	4096	4096	16,781,000
21	Fully Connected	1000	-	-	-	4096	1000	4,097,000

3.5.4 Shoeprint Classification using CNN-TL (VGG16)

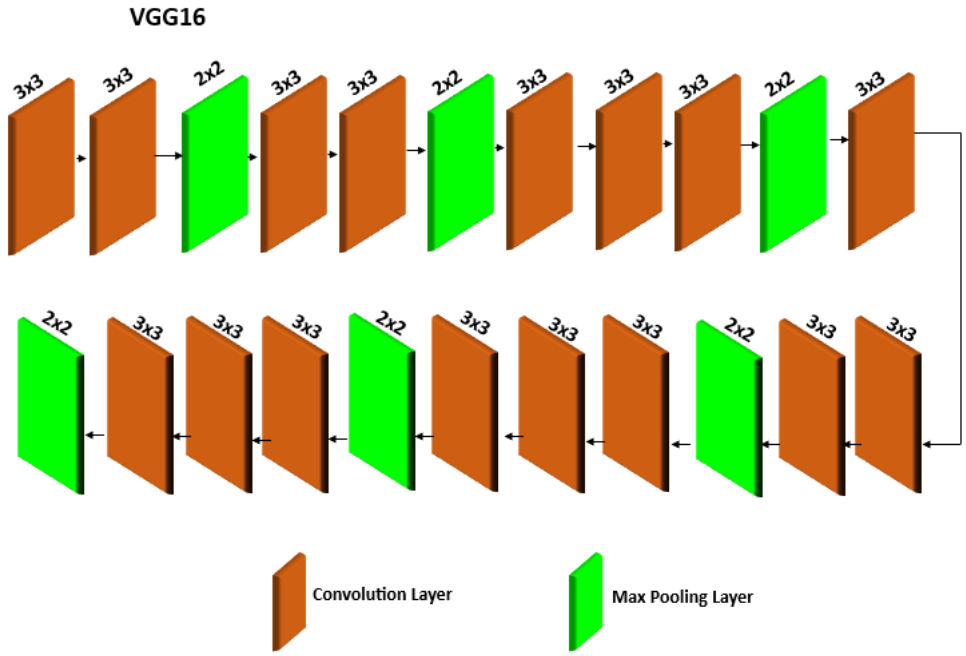

Figure 9. Architecture for the proposed model using VGG16 as a pre-trained model

The University of Oxford's Visual Geometry Group (Muraki et al., 2022) created the VGG16 convolutional neural network architecture. Karen Simonyan and Andrew Zisserman published their first paper in 2014, "Very Deep Convolutional Networks for Large-Scale Picture Recognition." As visualized in Figure 9, the VGG16 architecture comprises 16 layers, three fully linked levels, and 13 convolutional layers. It is distinguished using small, 3x3 convolutional filters stacked in convolutional layers to extract increasingly complex features from the input data. The architecture also includes max-pooling layers, which are used to downsample the data and reduce the dimensionality of the feature maps, and ReLU activation layers, which are used to introduce nonlinearity into the network by applying the ReLU (Rectified Linear Unit) function elementwise to the output of the convolutional layers. Intense convolutional layers, some of which contain up to 512 filters, are a crucial feature of the VGG16 architecture. As a result, the network can learn various features from the input data, making it a popular choice for image

classification and object detection tasks. VGG16 has 138 million parameters, which is relatively large compared to other deep neural network architectures. However, this large number of parameters enables VGG16 to learn a rich representation of the input data and achieve high accuracy on various tasks.

Table V. Lists the parameters of the VGG16 network

Layer	Type	Number of Filters	Filter Size	Stride	Padding	Output Shape	# of Parameters (approx.)
1	Convolutional	64	3x3	1	1	224x224x64	2,000
2	Convolutional	64	3x3	1	1	224x224x64	37,000
3	Max Pooling	-	2x2	2	0	112x112x64	0
4	Convolutional	128	3x3	1	1	112x112x128	74,000
5	Convolutional	128	3x3	1	1	112x112x128	148,000
6	Max Pooling	-	2x2	2	0	56x56x128	0
7	Convolutional	256	3x3	1	1	56x56x256	295,000
8	Convolutional	256	3x3	1	1	56x56x256	590,000
9	Convolutional	256	3x3	1	1	56x56x256	590,000
10	Max Pooling	-	2x2	2	0	28x28x256	0
11	Convolutional	512	3x3	1	1	28x28x512	1,180,000
12	Convolutional	512	3x3	1	1	28x28x512	2,360,000
13	Convolutional	512	3x3	1	1	28x28x512	2,360,000
14	Max Pooling	-	2x2	2	0	14x14x512	-
15	Convolutional	512	3x3	1	1	14x14x512	2,360,000
16	Convolutional	512	3x3	1	1	14x14x512	2,360,000
17	Convolutional	512	3x3	1	1	14x14x512	2,360,000
18	Max Pooling	-	2x2	2	0	7x7x512	-
	Total params (approx..)	73,293,000					
	Trainable params (approx..)	57,181,000					
	Non-trainable params (approx..)	16,113,000					

In general, the VGG16 design is known for its simplicity and efficiency. As a result, it is frequently used as a benchmark for evaluating how well different CNN architectures perform on image recognition tasks. The first convolutional layer's input shape is (224x224x3), representing the image size and the number of color channels (3 for RGB). Table V shows the number of parameters calculated for each layer.

3.5.5 Shoeprint Classification using CNN-TL (ResNet50)

Residual Networks (ResNets), a deep neural network design type, have recently received much attention because they can train profound networks without encountering the vanishing gradient problem. ResNet50 is one of the most popular ResNet architectures, and it consists of several residual blocks, as shown in Figure 10, each with many convolutional layers and a residual link.

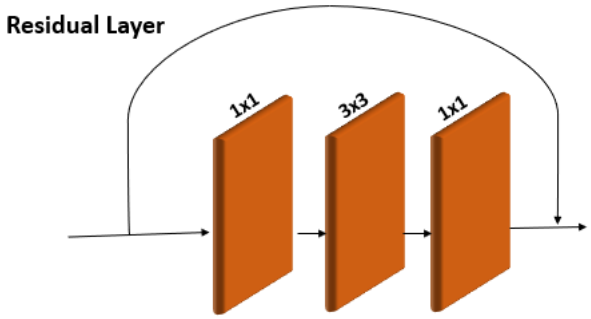

Figure 10. Illustration for residual layer

ResNets rely heavily on residual connections, which are critical to their success. The residual connection allows the network to learn a more straightforward mapping by adding the residual of previous layers' activations to the activations of later layers. In other words, rather than learning the entire mapping from input to output, the network only needs to learn the difference between the input and the desired output, which is much simpler.

The following equation (7) represents the residual connection implemented by adding activations from the previous layers, which is adopted from (Shuzhanfan, n.d.)

$$y = H(x) + x \qquad (7)$$

Where x represents the input, H(x) represents the residual function, and y represents the output. H(x) is implemented as a series of convolutional layers and non-linear activation functions, such as rectified linear units (ReLU). The residual connection allows information to pass through one or more layers, making it easier for the network to learn identity mapping, which is critical for deep learning training.

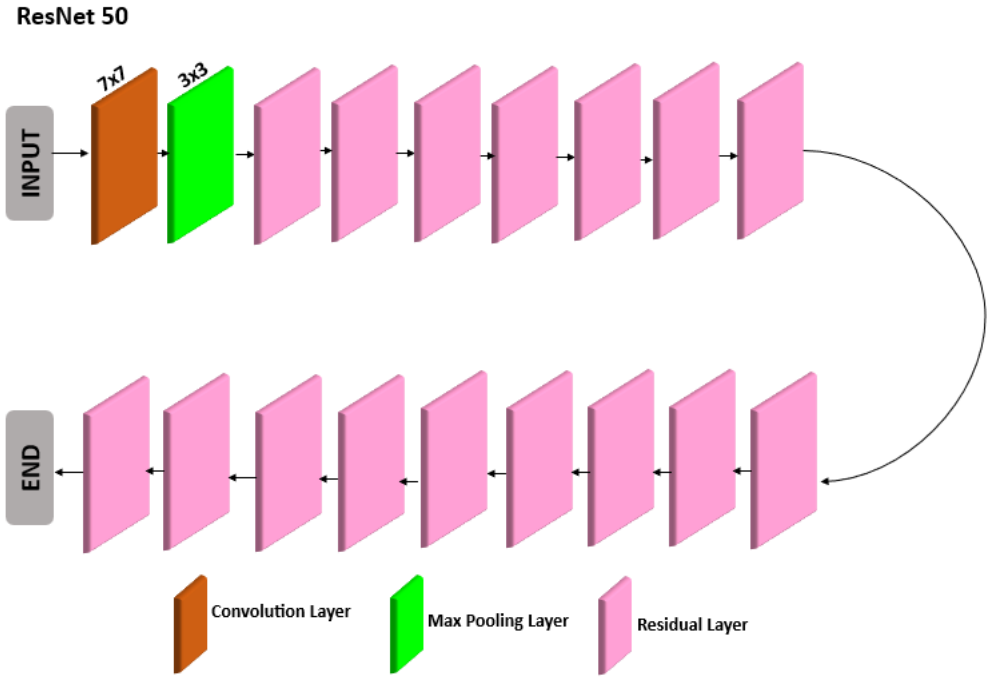

Figure 11. Architecture for the proposed model using ResNet50 as a pre-trained model

As shown in Figure 11, the ResNet50 architecture comprises several residual blocks, each containing convolutional layers and a residual connection. 3x3 filters are used in the convolutional layers, followed by batch normalization and ReLU activation functions. In addition, several pooling layers are used to reduce the spatial dimensions of the feature representation, and several fully connected layers are used to generate the final prediction. ResNet50 has approximately 25 million parameters, which is relatively modest compared to other

deep neural network architectures. ResNet50 is thus computationally efficient and critical for real-time applications. The architecture for the pre-trained model ResNet50 is depicted in Table VI.

Table VI. Representation of the ResNet50 architecture, including the layer type, number of filters, filter size, stride, padding, input shape, output shape, and number of parameters for each layer.

Layer	Type	Number of Filters	Filter Size	Stride	Padding	Output Shape	# of Parameters (approx.)
1	Convolutional	64	7x7	2	Same	112x112x64	4,000
2	Max Pooling	-	3x3	2	Same	56x56x64	0
3	Convolutional	256	3x3	1	Same	56x56x256	195,000
4	Convolutional	256	3x3	1	Same	56x56x256	466,000
5	Convolutional	256	3x3	1	Same	56x56x256	466,000
6	Max Pooling	-	3x3	2	Same	28x28x256	-
7	Convolutional	512	3x3	1	Same	28x28x512	931,000
8	Convolutional	512	3x3	1	Same	28x28x512	2,360,000
9	Convolutional	512	3x3	1	Same	28x28x512	2,360,000
10	Max Pooling	-	3x3	2	Same	14x14x512	0
11	Convolutional	1024	3x3	1	Same	14x14x512	18,620,000
12	Convolutional	1024	3x3	1	Same	14x14x1024	36,863,000
13	Convolutional	1024	3x3	1	Same	14x14x1024	36,863,000
14	Average Pooling	-	7x7	1	Valid	1x1x1024	0
	Total Params (approx.)	76,020,000					
	Trainable Params (approx.)	52,432,000					
	Non-trainable Params (approx.)	23,588,000					

And so on, until the final fully connected layers produce the predictions, repeating the convolution pattern and max pooling layers. The number of filters, size, and shape of the input determines the number of parameters in each layer. The first convolutional layer's input shape is (224x224x3), representing the image size and the number of color channels (3 for RGB). Each layer's number

of parameters is computed. Figure 12 can supplement the discussion below to comprehend how fine-tuning works and the challenges involved.

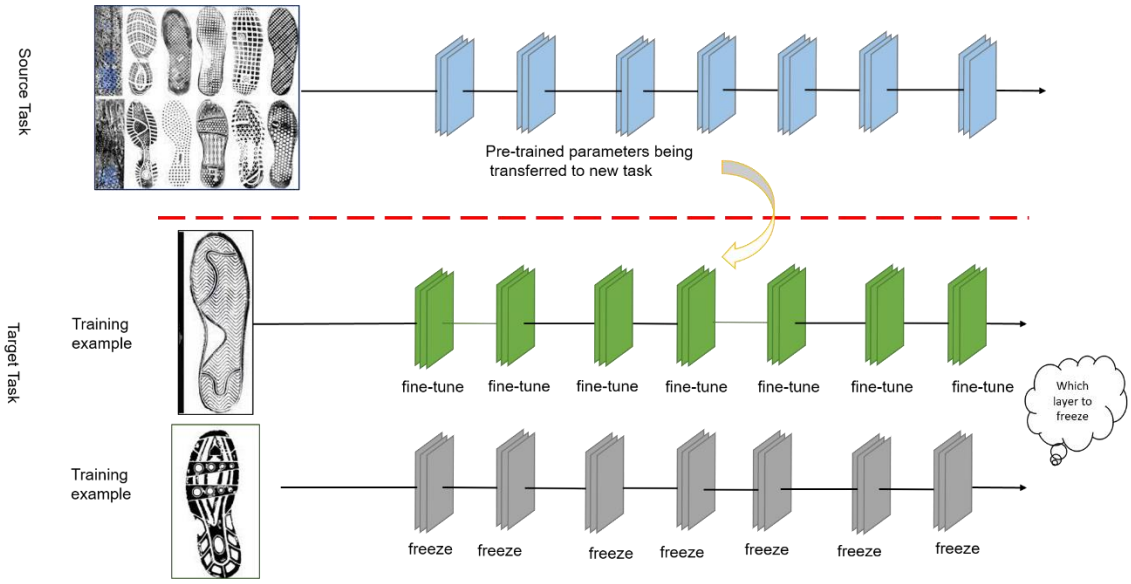

Figure 12. Illustration of how fine-tuning works and is considered crucial in improving the accuracy of TL

Deep Transfer Learning (DTL) transfers data from a previously trained source task model to a new target job model. It works well in situations where more training data is required. One of the most popular DTL approaches is fine-tuning, which involves slightly adjusting the parameters of the pre-trained model to ensure successful knowledge transfer. Recent research suggests that DTL outperforms random feature selection, even when the source and target tasks differ. DTL has been used in various real-world applications, including medical imaging, biological sequence analysis, transportation, recommender systems, communication, and recognition tasks such as hand gestures and face recognition. Fine-tuning strategies entail modifying the pre-trained parameters or the final few layers.

Furthermore, fine-tuning is advantageous when there is limited data for training because it necessitates a small dataset. Rahali and Akhloufi (2023) investigated the behavior of fine-tuning a portion of the network when the amount of data in the target domain is negligible. They improved their model's decoder and tested it on the target dataset.

According to Figure 12, an essential task in a model is determining which layers to freeze and which not to freeze. One of the most challenging aspects of fine-tuning a pre-trained model is determining which layer to adjust. To solve this, regularization terms were introduced to the loss function to ensure that the parameters of the fine-tuned model are similar to those of the original pre-trained model. The adaptive filter (AdaFilter) approach is an adaptive fine-tuning technique that chooses only certain convolutional filters in the pre-trained model and optimizes them based on per-example criteria. This approach considers the similarity between the source and target tasks when deciding which filters to fine-tune. VGGNet, GoogleNet (Inception), and ResNet are popular pre-trained models frequently used as baselines for DTL. For example, land use classification in building and construction has been done with VGGNet, GoogLeNet, and ResNet using street-view images and CNN models using aerial view images. To that end, fine-tuning can improve accuracy to a greater extent.

3.6 Limitations of the Study

Due to dataset availability constraints, the models cannot be more robust and must discard some shoeprints. In addition, the dataset only includes shoeprints with a flat outer sole. As a result, shoes with uneven outer soles may need to be more predictable. Another limitation is that the images are greyscaled; thus, new images must be pre-processed before they can be utilized in the model. In addition, the CNN-TL models were pre-trained on the ILSVRC dataset, which lacked shoeprint data and was small in size.

CHAPTER 4: Experimental Results and Discussion

This section presents the experimental results, accuracy values, the confusion matrix, and the models' supporting Python code. Furthermore, the implementations and details extensively discussed in Chapter 3 (Methodology) have been revised here for clarity and reference. Finally, Appendix 1 contains selected portions of the programming codes. To distinguish between half and full-shoeprints, two different models were tested. In the first experiment, a custom-made deep learning model was used to define its parameters. This was the only dataset used to train the model. Various metrics have been calculated using the confusion matrix in Figure 17 using Equations 8, 9, 10, and 11. In addition, Tables VII, VIII, IX, and X show the results of running the models.

Precision is also known as positive predictive value (Iqbal et al., 2018). Precision and recall trade false positives (FP) and false negatives (FN).

The following equation (8) represents the precision or the percentage of correctly predicted labels, which is adopted from (Sadman, 2020)

$$Precision = TP\ /(TP\ + FP\ ) \qquad (8)$$

A high recall score indicates that the model can recognize positive instances (Siddiqui et al., 2023). This is distinct from precision, which measures how many of the total number of optimistic predictions generated by the models are positive. The recall is also known as sensitivity.

The following equation (9) represents the model's accuracy in predicting positives instead of actual positives or the true positive (TP) rate, which is adopted from (Sadman, 2020)

$$Recall = TP\ /(TP\ + FN\ ) \qquad (9)$$

As an alternative to accuracy measurements, the F1-score is a machine learning model performance metric that equally weights precision and recall when analyzing the model's accuracy (Siddiqui et al., 2023).

The following equation (10) represents the model score as a function of recall and accuracy, which is adopted from (Sadman, 2020)

$$F1\ Score = (2 \times Precision \times Recall)/(Precision + Recall) \quad (10)$$

Accuracy—one of the most widely used performance metrics for machine learning classification models, model accuracy is mathematically defined as the ratio of and to all positive and negative observations.

The following equation (11) represents how often our machine learning model correctly predicted a result from all its predictions, which is adopted from (Sadman, 2020)

$$Accuracy = (TP\ + TN\ )/(TP\ + TN\ + FP\ + FN\ ) \times 100 \quad (11)$$

During the initial convolutional neural network model run, it was discovered that the accuracy and performance could have been better. Following a thorough investigation (reviewing the model parameters and viewing the dataset images), cleaning the dataset could aid in resolving the problem. Figure 13 depicts the training and validation loss of the convolutional neural networks model after cleaning the data. The accuracy dropped from 91.43% to 89% after running the model on a second time cleaned dataset. However, there was a higher accuracy in classifying full-shoeprint images from the half-shoeprint category. A pre-trained model was chosen and tested on the ILSVRC dataset in the second experiment. The ILSVRC dataset has 1000 classes. Again, the model was fine-tuned, and the accuracy increased to 93%, exceeding the previous accuracy of 89%.

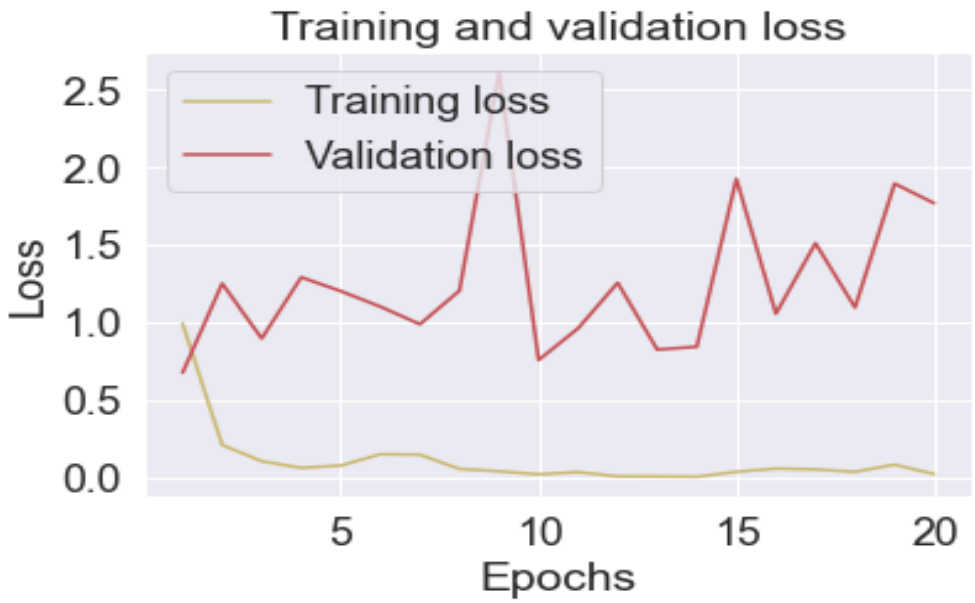

Figure 13. Training & Validation Loss after cleaning data of CNN model

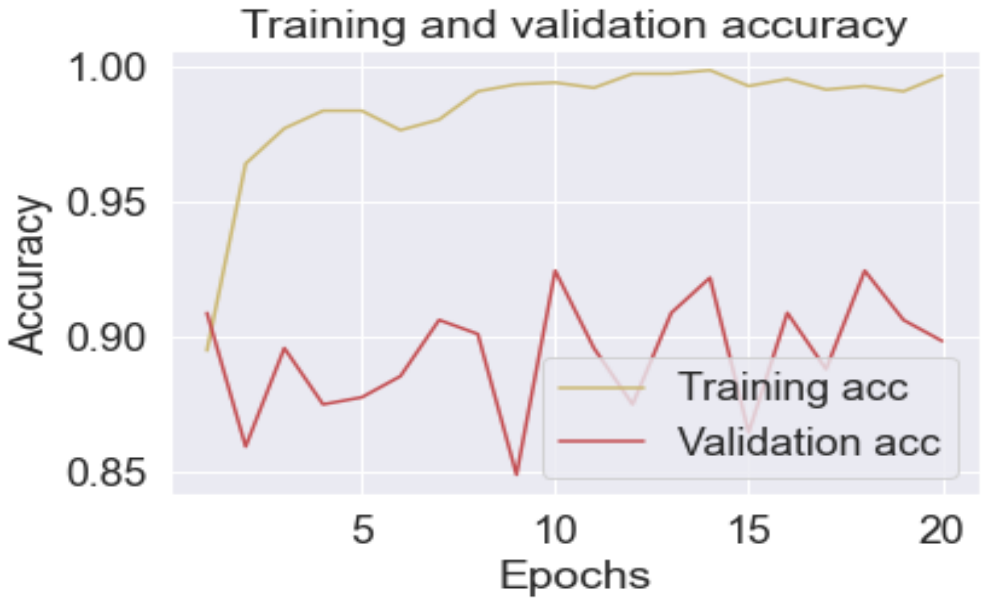

Figure 14. Accuracy after cleaning data of CNN model

Figure 13 shows that the highest validation loss in 20 epochs occurred at epoch 9 with a loss of 2.5, while the lowest validation loss occurred at the start of the graph at epoch 1 with a loss of nearly 0.7. Similarly, Figure 14 depicts the training and validation accuracy after the dataset was cleaned and applied to the convolutional neural networks model. At epoch 10, the model achieves

the highest validation accuracy of 0.93%. Figures 13 and 14 have fewer spikes than Figures 15 and 16 and thus appear to have a smoother graph.

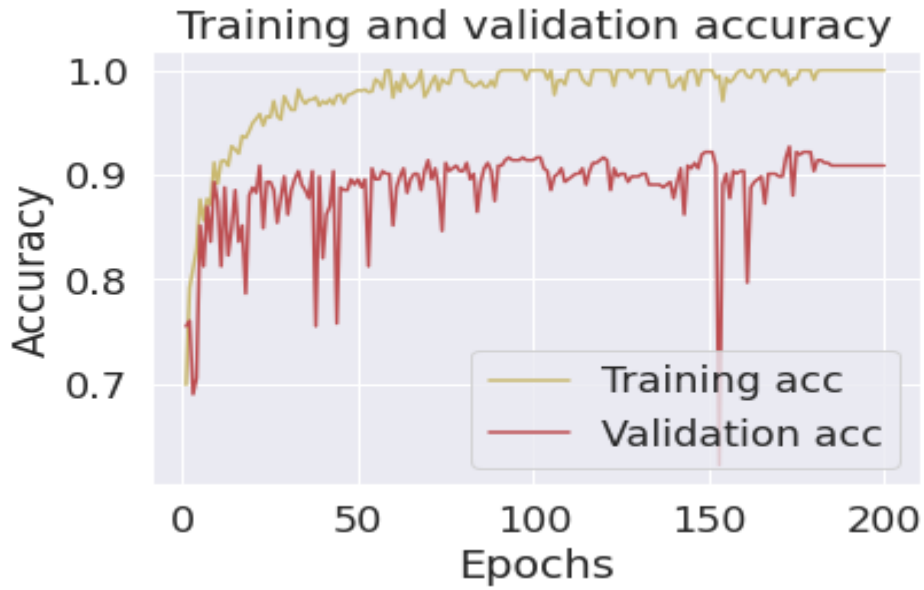

Figure 15. Training & Validation Accuracy graph after cleaning data of Inception V3 Model

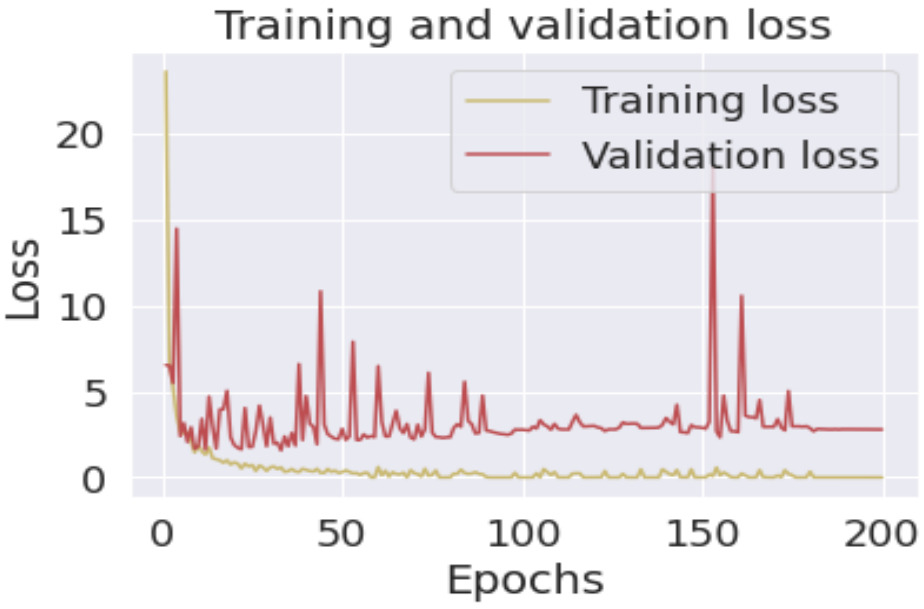

Figure 16. Training & Validation Loss Graph after cleaning data of Inception V3 Model

Compared to Figures 13 and 14, the graphs produced by running the CNN-TL pre-trained Inception V3 model has higher spikes and a closer frequency. Figures 15 and 16 show a brief period during which the model's loss and accuracy stabilize. This graph line is smoother than the previous graph lines. Figures 15 and 16 show a few unexpected spikes, indicating the possibility that there are still a few images of half-shoeprint in full-shoeprint in the training dataset.

Figure 17 of the confusion matrix shows that the proposed CNN-TL-based Inception V3 model correctly detected 231 full shoeprints out of 241 shoeprints from the validation data. Furthermore, based on half-shoeprint validation data, it correctly predicted 18 out of 125 half-shoeprints. Equations 8, 9, 10, and 11 were applied to the values in Figure 17 to compute the precision, recall (sensitivity), accuracy, and F1 score. Precision = 0.96%, recall = 0.93%, accuracy = 0.93%, and F1 score = 0.94% for the Inception V3 model. This data can be cleaned and increased in volume to improve the results.

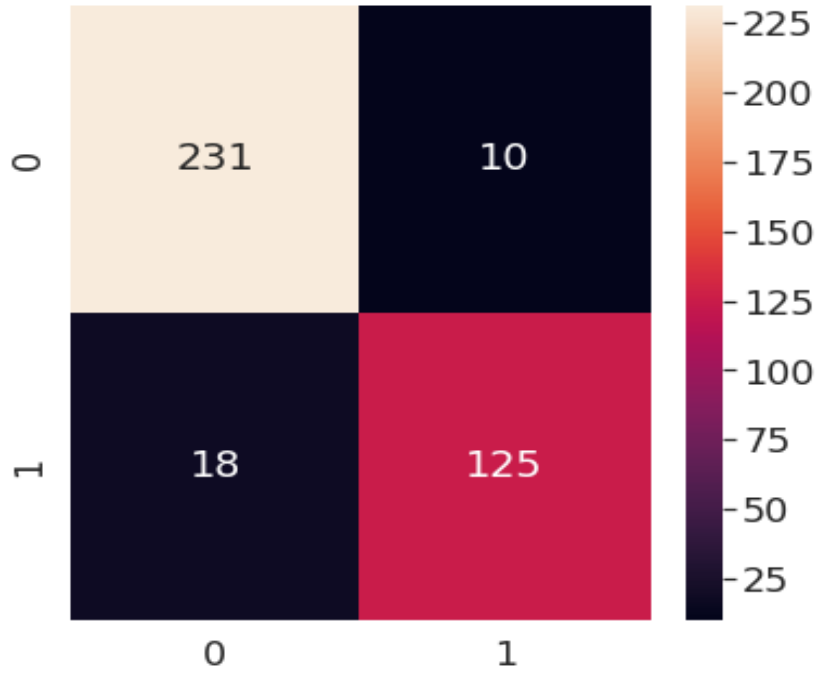

Figure 17. Confusion matrix summarizing TP, FP, FN, and TN values for Inception V3 Model

While a custom model (CNN) may perform better on a much larger dataset, the transfer learning approach with the fine-tuned model, as discussed earlier in the thesis, is preferred for a much smaller dataset. With 200 epochs, a batch size of 32, and an 80:20 training test split, an accuracy of 89% was obtained with the dataset under consideration by adjusting model parameters such as the learning rate and the number of nodes in the fully connected layer. To be considered, the model would need to achieve a minimum of 90% accuracy, which it did not achieve at this point—from exploring the dataset and identifying errors in the shoeprint image classes to cleaning/evaluating the dataset a second time. However, after running the transfer learning models, it was discovered that their accuracy increased from 89% to 92%. Table XI shows a comparison of the results with current state-of-the-art methods. Once again, transfer learning outperformed other currently used methods.

Table VII. Accuracy of CNN model for classification of shoeprint images into class 1(half-shoeprint) and class 2 (full-shoeprint)

	Runtime #1	**Runtime #2**	**Runtime #3**
Accuracy	0.9409	0.9617	0.9652
Sensitivity	0.9284	0.9602	0.9653
Specificity	0.9674	0.9646	0.965

Table VIII. Accuracy of CNN-TL model (Inception V3) for classification of shoeprint images into class 1(half-shoeprint) and class 2 (full-shoeprint)

	Runtime #1	**Runtime #2**	**Runtime #3**
Accuracy	0.9167	0.9219	0.9036
Sensitivity	0.9265	0.9414	0.908
Specificity	0.8993	0.8897	0.8955

Table IX. Accuracy of CNN-TL Model (VGG16) for classification of shoeprint images into class 1(half-shoeprint) and class 2 (full-shoeprint)

	Runtime #1	**Runtime #2**	**Runtime #3**
Accuracy	0.9583	0.9688	0.9557
Sensitivity	0.9707	0.9914	0.9746
Specificity	0.9379	0.9338	0.9257

Table X. Accuracy of CNN-TL Model (ResNet50) for classification of shoeprint images into class 1(half-shoeprint) and class 2 (full-shoeprint)

	Runtime #1	**Runtime #2**	**Runtime #3**
Accuracy	0.9714	0.9661	0.9714
Sensitivity	0.9873	0.979	0.9832
Specificity	0.9459	0.9452	0.9512

Tables VII, VIII, IX, and X. (Inception V3, VGG16, and ResNet50) show the results obtained using the CNN model and various TL models. According to the tables, the pre-trained Inception V3 model had the lowest accuracy of 0.8955% (runtime 3) compared to the performance of VGG16 and ResNet50. These findings are also mentioned and discussed in Table XI, which includes a comprehensive comparison that clearly shows that the proposed model outperformed other existing approaches, such as SVM (95% accuracy) and ANN (90% accuracy), as described in (Li, 2019). Furthermore, transfer learning with ResNet50 as a pre-trained model produced the most accurate model in this thesis, with an accuracy of 97.14%. (runtime one and runtime 3).

Table XI. Comparison of the Proposed Approach and State-of-the-Art Techniques

Reference	**Model**	**Accuracy**
Li, C. (2019). Shoe Print Identification from Images with Convolutional Neural Network.	SVM	95%
[Li, C. (2019).]	ANN	90%
Proposed Approach	CNN	96.17%.
	Inception V3	92.19%
	VGG16	96.88%
	ResNet50	**97.14%**

Initially, this work aims to investigate two primary research questions: How to accurately classify shoeprints from available datasets? Furthermore, how well could transfer learning improve the accuracy of classifying digital shoeprint images? Some have used CNN, as discussed earlier in

this thesis (Chapter 2 Literature Review), which focuses on different approaches. However, one of the most significant issues the researchers have dealt with is the dataset's unavailability. For the related problem, only a small amount of data is available.

In addition, generating or creating sufficient samples in a dataset is critical for a successful deep-learning algorithm. However, even creating a dataset and its data takes time and money, which many researchers and academicians can only afford occasionally. Furthermore, generating a new dataset can raise additional issues that must be addressed, such as whether this particular data is well suited for different and adverse types of problems or whether they are an excellent fit to resolve 'underfitting' and 'overfitting' issues.

CHAPTER 5: Conclusion

Shoeprints are the most commonly left clues at a crime scene, which significantly aids in the investigation. As a result, automatic shoeprint identification has been extensively researched as a cutting-edge research topic in forensic science. In addition, machine learning, and more recently, deep learning, has piqued the interest of investigators in many criminal cases, particularly forensic science, by utilizing digital images of shoeprints and identifying criminals with it—the current work attempts to investigate cutting-edge deep learning mechanisms for the scenario mentioned above.

The integration of Artificial Intelligence (AI) in forensic science has emerged as a compelling motivation due to its potential to revolutionize investigative processes and enhance the accuracy and efficiency of forensic analyses (Costantini et al., 2019). Machine learning and computer vision are two examples of AI technologies that can automatically analyze massive amounts of forensic data, such as fingerprints, DNA profiles, shoeprints, and surveillance footage, allowing for rapid and accurate identification of suspects and crime scene reconstruction. By leveraging AI algorithms, forensic scientists can uncover correlations in evidence, assist in interpreting ambiguous or incomplete data, and support decision-making in criminal investigations (Costantini et al., 2019). The utilization of AI in forensic science holds immense promise for improving forensic analyses' speed, reliability, and objectivity, ultimately aiding law enforcement agencies, legal professionals, and justice systems in ensuring a fair and robust criminal justice process.

In Chapter 1 (Introduction), the background and foundation of shoeprint images, aspects of classification and retrieval of these images, and existing approaches are discussed in brief. The contents of the thesis are summarized, and the contributions are highlighted. The second chapter (Literature Review) details deep learning and traditional machine learning approaches for image

classification problems. The chapter reveals that the literature needs to investigate classification problems thoroughly. The researchers are more concerned with the retrieval of a shoeprint. The chapter also discusses the specifics of these two types of problems. The basic formula, its application in the literature, and the gap in the literature have all been thoroughly discussed in this chapter.

Chapter 3 (Methodology) used and illustrated the standard benchmark for shoeprint images. The steps for pre-processing and feature extraction are well illustrated. Finally, chapter 3 covers all of the methodology's explanations and illustrations. Previously trained models such as Inception V3, VGG16, and ResNet50 were used. Furthermore, the standard FID-300 and 2D Footwear Outsole Impressions datasets were used. The dataset includes images of half-shoeprints and full-shoeprints. Classifying these two classes is critical in many real-life scenarios, such as forensic investigations and criminal identification at crime scenes. Chapter 4 (Experimental Results and Discussions) presents the results obtained with the different models. The primary challenge in this problem was to improve accuracy. However, the available dataset is designed in such a way that it contains both half and full-shoeprints, making classification difficult.

The CNN model achieves an accuracy of 96.17%, after which TL was considered. The Inception V3 model, on the other hand, does not deliver higher accuracy and is less than 93% accurate. Regarding accuracy, the VGG 16 pre-trained model exceeds CNN by 96.88%. Furthermore, ResNet50 gives the most remarkable and accurate findings, with 97.14% accuracy. These findings outperform previously reported literature values. On the same dataset, Li et al. (2022) discovered that accuracy with ANN and SVM models was approximately 90% and 95%, respectively. On the one hand, the proposed methodology allows forensic experts to use deep learning to uncover the individuals behind crimes. However, it also exposes researchers to how TL can be used to classify shoeprints for better classification with improved accuracy.

RQ1 aimed to improve the classification accuracy for a publicly available dataset using CNN performance. Because traditional ML techniques have yet to be proven to achieve state-of-the-art performance, the models investigated the deep learning CNN method. As previously stated, the classification experiment outperformed ANN and SVM in accuracy (97.14%). RQ2 was addressed by investigating various transfer learning mechanisms for classifying shoeprint images. Using the proposed TL mechanism, these images can be classified efficiently and accurately on mobile devices since the authors (Hassan et al., 2021) used deep learning methods previously, but with a very low accuracy achieved of around 65%. However, different TL models yielded different accuracies in this thesis. Inception V3, for example, achieved around 92% accuracy. With some fine-tuning and different combinations, as discussed in Chapter 4, the results obtained even better results when using VGG16, and it has a higher accuracy value than CNN. Even though the accuracy rate is low, an improved result is encouraging.

A third pre-trained model, ResNet50, was used to improve the model even further. The results show that the ResNet50 architecture is well suited for the shoeprint dataset, outperforming all state-of-the-art approaches with more than 97% accuracy. However, the primary reason for the improvement differs from this thesis's topic. Nonetheless, there is an opportunity to investigate differences in performance for future directions. Furthermore, it should be studied whether other applications areas where these types of datasets can affect and affect the research community.

5.1 Limitations of Thesis

While the motivation for using AI in forensic science, specifically in the area of shoeprint classifications, is compelling, it is essential to acknowledge and address the limitations of this work. One significant limitation is the potential dependence on limited and biased datasets for training the classification algorithms. Suppose the dataset used for training primarily consists of shoeprints from specific populations, demographics, or regions. In that case, it may introduce biases and constraints in the algorithm's ability to accurately classify shoeprints from different

sources. This could compromise the generalizability and reliability of the classification model, limiting its practical applicability in real forensic scenarios that involve shoeprints from diverse sources. Therefore, future research must focus on gathering more extensive and representative datasets encompassing a wide range of shoeprints, ensuring the algorithm's effectiveness and fairness across various populations and regions.

The shoeprint datasets used for transfer learning classification are frequently limited regarding the number of shoe types and the range of shoeprint variations. For example, half-shoeprint and full-shoeprint images could be better paced in a single dataset. Future research could focus on creating more significant and diverse datasets to improve the accuracy of the classification models. Another issue worthy of consideration is the exploration of other existing transfer learning techniques. While we have used a few transfer learning models in this thesis, other models would be interesting to investigate and observe their performance. Future research could look into the efficacy of transfer learning techniques like fine-tuning, feature extraction, and domain adaption.

Furthermore, investigating the efficacy of multi-modal data integration may be a potential issue. Since transfer learning typically relies on image data, shoeprint classification can be improved by incorporating other modalities, such as depth data or 3D imaging. Therefore, it would be worthwhile to investigate integrating multi-modal data to improve classification accuracy. Furthermore, given the success of attention mechanisms in many computer vision problems, such as image classification and retrieval, using attention mechanisms for shoeprint image classification would be critical. Therefore, the researchers will take up further investigation on its performance for shoeprint classification soon.

Transfer learning algorithms for shoeprint categorization can be complex, and understanding how they create predictions can be difficult. An approach to explainable artificial intelligence (AI) approach can aid in developing explainable AI techniques to aid in interpreting the models'

decision-making process and to expand the application of shoeprint image classification in forensic research. Soon, applications such as tire impressions or tool mark analysis could be investigated. Additionally, ensuring the transparency and interpretability of the AI models used for shoeprint classification is paramount, as black-box algorithms may face challenges in presenting the evidence in court and justifying the outcomes, requiring further exploration and refinement in terms of explain ability and accountability. Addressing these limitations will be critical to advancing the field of shoeprint classifications and facilitating their reliable application in forensic investigations at a broader scale.

References

Abutaleb, K., Newete, S. W., Mangwanya, S., Adam, E., & Byrne, M. J. (2021). Mapping eucalypts trees using high resolution multispectral images: A study comparing WorldView 2 vs. SPOT 7. *The Egyptian Journal of Remote Sensing and Space Science*, *24*(3), 333–342. https://doi.org/10.1016/j.ejrs.2020.09.001

Adam, K., Thomas, A., & Thomas, V. (n.d.). *Unsupervised Footwear Impression Analysis and Retrieval from Crime Scene Data*.

Afifi, S., GholamHosseini, H., & Sinha, R. (2020). FPGA Implementations of SVM Classifiers: A Review. *SN Computer Science*, *1*(3), 133. https://doi.org/10.1007/s42979-020-00128-9

Agatonovic-Kustrin, S., & Beresford, R. (2000). Basic concepts of artificial neural network (ANN) modeling and its application in pharmaceutical research. *Journal of Pharmaceutical and Biomedical Analysis*, *22*(5), 717–727. https://doi.org/10.1016/S0731-7085(99)00272-1

Alexander, A. (1999). Automatic classification and recognition of shoeprints. *7th International Conference on Image Processing and Its Applications*, 638–641. https://doi.org/10.1049/cp:19990401

Alexandre, G. (1996). Computerized classification of the shoeprints of burglars' soles. *Forensic Science International*, *82*(1), 59–65. https://doi.org/10.1016/0379-0738(96)01967-6

AlGarni, G., & Hamiane, M. (2008). A novel technique for automatic shoeprint image retrieval. *Forensic Science International*, *181*(1–3), 10–14. https://doi.org/10.1016/j.forsciint.2008.07.004

Anand, P., & Lee, C. (2023). Using Deep Learning to Overcome Privacy and Scalability Issues in Customer Data Transfer. *Marketing Science*, *42*(1), 189–207. https://doi.org/10.1287/mksc.2022.1365

Anthony, L. F. W., Kanding, B., & Selvan, R. (2020). *Carbontracker: Tracking and Predicting the Carbon Footprint of Training Deep Learning Models*.

Aouedi, O., Piamrat, K., & Parrein, B. (2022). Intelligent Traffic Management in Next-Generation Networks. *Future Internet*, *14*(2), 44. https://doi.org/10.3390/fi14020044

Ashley, W. (1996). What shoe was that? The use of computerised image database to assist in identification. *Forensic Science International*, *82*(1), 7–20. https://doi.org/10.1016/0379-0738(96)01962-7

Ashraf, M., Sohail, S. S., & Chaudhry, B. M. (2022). Issues, Challenges, and Opportunities in Food Recommender Systems. *2022 International Conference on Data Analytics for Business and Industry (ICDABI)*, 570–574. https://doi.org/10.1109/ICDABI56818.2022.10041496

Azmoon, B., Biniyaz, A., Liu, Z., & Sun, Y. (2021). Image-Data-Driven Slope Stability Analysis for Preventing Landslides Using Deep Learning. *IEEE Access*, *9*, 150623–150636. https://doi.org/10.1109/ACCESS.2021.3123501

Babenko, A., & Lempitsky, V. (2015). *Aggregating Deep Convolutional Features for Image Retrieval*.

Bhattacharya, S., Somayaji, S. R. K., Gadekallu, T. R., Alazab, M., & Maddikunta, P. K. R. (2022). A review on deep learning for future smart cities. *Internet Technology Letters*, *5*(1). https://doi.org/10.1002/itl2.187

Bhattarai, M., Oyen, D., Castorena, J., Yang, L., & Wohlberg, B. (2020). Diagram Image Retrieval using Sketch-Based Deep Learning and Transfer Learning. *2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops (CVPRW)*, 663–672. https://doi.org/10.1109/CVPRW50498.2020.00095

Bluestar Software. (2018, November 9). *Forensic Intelligence* . Https://Bluestar-Software.Co.Uk.

Bouridane, A., Alexander, A., Nibouche, M., & Crookes, D. (n.d.). Application of fractals to the detection and classification of shoeprints. *Proceedings 2000 International Conference on Image Processing (Cat. No.00CH37101)*, 474–477. https://doi.org/10.1109/ICIP.2000.900998

Breiman, L. (2001). Random forests. Machine Learning. *Machine Learning*, *45*(1), 5–32. https://doi.org/10.1023/A:1010933404324

Budka, M., Ashraf, A. W. U., Bennett, M., Neville, S., & Mackrill, A. (2021). Deep multilabel CNN for forensic footwear impression descriptor identification. *Applied Soft Computing*, *109*, 107496. https://doi.org/10.1016/j.asoc.2021.107496

Chandra, M. A., & Bedi, S. S. (2021). Survey on SVM and their application in image classification. *International Journal of Information Technology*, *13*(5), 1–11. https://doi.org/10.1007/s41870-017-0080-1

Cheng, K., Cheng, X., Wang, Y., Bi, H., & Benfield, M. C. (2019). Enhanced convolutional neural network for plankton identification and enumeration. *PLOS ONE*, *14*(7), e0219570. https://doi.org/10.1371/journal.pone.0219570

Chi, M., Feng, R., & Bruzzone, L. (2008). Classification of hyperspectral remote-sensing data with primal SVM for small-sized training dataset problem. *Advances in Space Research*, *41*(11), 1793–1799. https://doi.org/10.1016/j.asr.2008.02.012

Cho, N., Kang, Y., Yoon, J., Park, S., & Kim, J. (2022). Classifying Tourists' Photos and Exploring Tourism Destination Image Using a Deep Learning Model. *Journal of Quality Assurance in Hospitality & Tourism*, *23*(6), 1480–1508. https://doi.org/10.1080/1528008X.2021.1995567

Copiaco, A., Himeur, Y., Amira, A., Mansoor, W., Fadli, F., Atalla, S., & Sohail, S. S. (2023). An innovative deep anomaly detection of building energy consumption using energy time-series images. *Engineering Applications of Artificial Intelligence*, *119*, 105775. https://doi.org/10.1016/j.engappai.2022.105775

Cortes, C., & Vapnik, V. (1995). Support-vector networks. *Machine Learning*, *20*(3), 273–297. https://doi.org/10.1007/BF00994018

Costantini, S., de Gasperis, G., & Olivieri, R. (2019). Digital forensics and investigations meet artificial intelligence. Annals of Mathematics and Artificial Intelligence, 86(1–3), 193–229. https://doi.org/10.1007/s10472-019-09632-y

Creative Cloud. (2022). *PNG files.* . Adobe.Com.

CreativeCloud. (2022). *TIFF Files*. Adobe.Com.

Crookes, H. S. D., Bouridane, A., & Gueham, M. (2007). Shoeprint Image Retrieval Based on Local Image Features. *Third International Symposium on Information Assurance and Security*, 387–392. https://doi.org/10.1109/IAS.2007.18

Dardi, F., Cervelli, F., & Carrato, S. (2009). *A Texture Based Shoe Retrieval System for Shoe Marks of Real Crime Scenes* (pp. 384–393). https://doi.org/10.1007/978-3-642-04146-4_42

de Chazal, P., Flynn, J., & Reilly, R. B. (2005). Automated processing of shoeprint images based on the Fourier transform for use in forensic science. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, *27*(3), 341–350. https://doi.org/10.1109/TPAMI.2005.48

Del Mar-Raave, J. R., Bahşi, H., Mršić, L., & Hausknecht, K. (2021). A machine learning-based forensic tool for image classification - A design science approach. Forensic Science International: Digital Investigation, 38, 301265. https://doi.org/10.1016/j.fsidi.2021.301265

Deng, J., Dong, W., Socher, R., Li, L.-J., Kai Li, & Li Fei-Fei. (2009). ImageNet: A large-scale hierarchical image database. *2009 IEEE Conference on Computer Vision and Pattern Recognition*, 248–255. https://doi.org/10.1109/CVPR.2009.5206848

Deng, Z., Cao, M., Rai, L., & Gao, W. (2018). A two-stage classification method for borehole-wall images with support vector machine. *PLOS ONE*, *13*(6), e0199749. https://doi.org/10.1371/journal.pone.0199749

Deshmukh, M., & Patil, P. (2009). Automatic shoeprint matching system for crime scene investigation. *International Journal of Computing Science and Communication Technologies*, *2*(1), 1–7.

Dipongkor, A. K., Islam, Md. S., Hussain, I., Yongchareon, S., & Mistry, S. (2023). On Fusing Artificial and Convolutional Neural Network Features for Automatic Bug Assignments. *IEEE Access*, *11*, 49493–49508. https://doi.org/10.1109/ACCESS.2023.3273595

Doborjeh, M., Doborjeh, Z., Kasabov, N., Barati, M., & Wang, G. Y. (2021). Deep Learning of Explainable EEG Patterns as Dynamic Spatiotemporal Clusters and Rules in a Brain-Inspired Spiking Neural Network. *Sensors*, *21*(14), 4900. https://doi.org/10.3390/s21144900

Doborjeh, Z., Hemmington, N., Doborjeh, M., & Kasabov, N. (2022). Artificial intelligence: a systematic review of methods and applications in hospitality and tourism. *International Journal of Contemporary Hospitality Management*, *34*(3), 1154–1176. https://doi.org/10.1108/IJCHM-06-2021-0767

Dreiseitl, S., & Ohno-Machado, L. (2002). Logistic regression and artificial neural network classification models: a methodology review. *Journal of Biomedical Informatics*, *35*(5–6), 352–359. https://doi.org/10.1016/S1532-0464(03)00034-0

Esteva, A., Kuprel, B., Novoa, R. A., Ko, J., Swetter, S. M., Blau, H. M., & Thrun, S. (2017). Dermatologist-level classification of skin cancer with deep neural networks. *Nature*, *542*(7639), 115–118. https://doi.org/10.1038/nature21056

Fan, Z., Zhu, Y., He, Y., Sun, Q., Liu, H., & He, J. (2023). Deep Learning on Monocular Object Pose Detection and Tracking: A Comprehensive Overview. *ACM Computing Surveys*, *55*(4), 1–40. https://doi.org/10.1145/3524496

Farooq, M. U., Mansoor, A. B., & Raza, A. (2021). Shoeprint classification using deep learning techniques: A systematic review. Journal of Forensic Sciences, 66(4), 1227-1243. https://doi.org/10.1111/1556-4029.14753

Fatima, N., Agarwal, P., & Sohail, S. S. (2022). *Security and Privacy Issues of Blockchain Technology in Health Care—A Review* (pp. 193–201). https://doi.org/10.1007/978-981-16-5655-2_18

Fonseca, A. L. A. da, Chimenti, P. C. P. de S., & Suarez, M. C. (2023). Using Deep Learning Language Models as Scaffolding Tools in Interpretive Research. *Revista de Administração Contemporânea*, *27*(3). https://doi.org/10.1590/1982-7849rac2023230021.en

Francis, X., Sharifzadeh, H., Newton, A., Baghaei, N., & Varastehpour, S. (2019). Learning Wear Patterns on Footwear Outsoles Using Convolutional Neural Networks. *2019 18th IEEE International Conference On Trust, Security And Privacy In Computing And Communications/13th IEEE International Conference On Big Data Science And Engineering (TrustCom/BigDataSE)*, 450–457. https://doi.org/10.1109/TrustCom/BigDataSE.2019.00067

Freund, Y., & Schapire, R. E. (1997). A Decision-Theoretic Generalization of On-Line Learning and an Application to Boosting. *Journal of Computer and System Sciences*, *55*(1), 119–139. https://doi.org/10.1006/jcss.1997.1504

G, V. K. B., Carneiro, G., & Reid, I. (2015). *Learning Local Image Descriptors with Deep Siamese and Triplet Convolutional Networks by Minimising Global Loss Functions*.

Galante, N., Cotroneo, R., Furci, D., Lodetti, G., & Casali, M. B. (2023). Applications of artificial intelligence in forensic sciences: Current potential benefits, limitations and perspectives. *International Journal of Legal Medicine*, *137*(2), 445–458. https://doi.org/10.1007/s00414-022-02928-5

Gao, Y., & Mosalam, K. M. (2018). Deep Transfer Learning for Image-Based Structural Damage Recognition. *Computer-Aided Civil and Infrastructure Engineering*, *33*(9), 748–768. https://doi.org/10.1111/mice.12363

Gao, Z., Dang, W., Wang, X., Hong, X., Hou, L., Ma, K., & Perc, M. (2021). Complex networks and deep learning for EEG signal analysis. *Cognitive Neurodynamics*, *15*(3), 369–388. https://doi.org/10.1007/s11571-020-09626-1

Gasmi, S., Bouhadada, T., & Benmachiche, A. (2020). Survey on Recommendation Systems. *Proceedings of the 10th International Conference on Information Systems and Technologies*, 1–7. https://doi.org/10.1145/3447568.3448518

Geradts, Z., & Keijzer, J. (1996). The image-database REBEZO for shoeprints with developments on automatic classification of shoe outsole designs. *Forensic Science International*, *82*(1), 21–31. https://doi.org/10.1016/0379-0738(96)01963-9

Gomez, L., Rodriguez, M., & Fernandez, A. (2019). A comparative study of transfer learning techniques for image classification tasks. International Journal of Computer Vision, 25(4), 567-589.

Goodfellow, I., Bengio, Y., & Courville, A. (2016). *Deep Learning*. MIT.

Gruetzemacher, R., & Paradice, D. (2022). Deep Transfer Learning & Beyond: Transformer Language Models in Information Systems Research. *ACM Computing Surveys*, *54*(10s), 1–35. https://doi.org/10.1145/3505245

Gu, Y., Zhang, S., Qiu, L., Wang, Z., & Zhang, L. (2021). A Layered KNN-SVM Approach to Predict Missing Values of Functional Requirements in Product Customization. *Applied Sciences*, *11*(5), 2420. https://doi.org/10.3390/app11052420

Guo, Y., Li, Y., Wang, L., & Rosing, T. (2020). AdaFilter: Adaptive Filter Fine-Tuning for Deep Transfer Learning. *Proceedings of the AAAI Conference on Artificial Intelligence*, *34*(04), 4060–4066. https://doi.org/10.1609/aaai.v34i04.5824

Gupta, K., & Bajaj, V. (2023). Deep learning models-based CT-scan image classification for automated screening of COVID-19. *Biomedical Signal Processing and Control*, *80*, 104268. https://doi.org/10.1016/j.bspc.2022.104268

Guyon, I., Weston, J., Barnhill, S., & Vapnik, V. (2002). Gene selection for cancer classification using support vector machines. *Machine Learning*, *46*(1/3), 389–422. https://doi.org/10.1023/A:1012487302797

Hamza, A., Khan, M. A., Alhaisoni, M., al Hejaili, A., Shaban, K. A., Alsubai, S., Alasiry, A., & Marzougui, M. (2022). D2BOF-COVIDNet: A Framework of Deep Bayesian Optimization and Fusion-Assisted Optimal Deep Features for COVID-19 Classification Using Chest X-ray and MRI Scans. *Diagnostics*, *13*(1), 101. https://doi.org/10.3390/diagnostics13010101

Hassan, M., Wang, Y., Wang, D., Li, D., Liang, Y., Zhou, Y., & Xu, D. (2021). Deep learning analysis and age prediction from shoeprints. *Forensic Science International*, *327*, 110987. https://doi.org/10.1016/j.forsciint.2021.110987

Hassan, Md. M., Hassan, Md. M., Mollick, S., Khan, Md. A. R., Yasmin, F., Bairagi, A. K., Raihan, M., Arif, S. A., & Rahman, A. (2023). A Comparative Study, Prediction and Development of Chronic Kidney Disease Using Machine Learning on Patients Clinical Records. *Human-Centric Intelligent Systems*. https://doi.org/10.1007/s44230-023-00017-3

He, K., Zhang, X., Ren, S., & Sun, J. (2015). *Deep Residual Learning for Image Recognition*.

He, Q., Niu, X., Qi, R.-Q., & Liu, M. (2022). Advances in microbial metagenomics and artificial intelligence analysis in forensic identification. *Frontiers in Microbiology*, *13*. https://doi.org/10.3389/fmicb.2022.1046733

Himeur, Y., Sohail, S. S., Bensaali, F., Amira, A., & Alazab, M. (2022). Latest trends of security and privacy in recommender systems: A comprehensive review and future perspectives. *Computers & Security*, *118*, 102746. https://doi.org/10.1016/j.cose.2022.102746

Hossain, Md. B., Iqbal, S. M. H. S., Islam, Md. M., Akhtar, Md. N., & Sarker, I. H. (2022). Transfer learning with fine-tuned deep CNN ResNet50 model for classifying COVID-19 from chest X-ray images. *Informatics in Medicine Unlocked*, *30*, 100916. https://doi.org/10.1016/j.imu.2022.100916

How, M.-L., & Chan, Y. J. (2020). Artificial Intelligence-Enabled Predictive Insights for Ameliorating Global Malnutrition: A Human-Centric AI-Thinking Approach. *AI*, *1*(1), 68–91. https://doi.org/10.3390/ai1010004

Howard, A. G., Zhu, M., Chen, B., Kalenichenko, D., Wang, W., Weyand, T., Andreetto, M., & Adam, H. (2017). *MobileNets: Efficient Convolutional Neural Networks for Mobile Vision Applications*.

Hu, H., Salcic, Z., Sun, L., Dobbie, G., Yu, P. S., & Zhang, X. (2022). Membership Inference Attacks on Machine Learning: A Survey. *ACM Computing Surveys*, *54*(11s), 1–37. https://doi.org/10.1145/3523273

Huang, A. J., & Agarwal, S. (2023). On the Limitations of Physics-Informed Deep Learning: Illustrations Using First-Order Hyperbolic Conservation Law-Based Traffic Flow Models. *IEEE Open Journal of Intelligent Transportation Systems*, *4*, 279–293. https://doi.org/10.1109/OJITS.2023.3268026

Iqbal, F., Sohail, A., Raza, A., & Saeed, A. (2020). A Survey of Transfer Learning Techniques for Convolutional Neural Networks. Journal of Artificial Intelligence and Soft Computing Research, 10(2), 97-123.

Iqbal, A., Rahman, A., Sharif, M., & Saba, T. (2020). Shoeprint classification using deep learning: A comprehensive review. SN Applied Sciences, 2(3), 1-17. https://doi.org/10.1007/s42452-020-2142-4

Iqbal, Z., Luo, D., Henry, P., Kazemifar, S., Rozario, T., Yan, Y., Westover, K., Lu, W., Nguyen, D., Long, T., Wang, J., Choy, H., & Jiang, S. (2018). Accurate real time localization tracking in a clinical environment using Bluetooth Low Energy and deep learning. *PLOS ONE*, *13*(10), e0205392. https://doi.org/10.1371/journal.pone.0205392

Irshad, R. R., Alattab, A. A., Alsaiari, O. A. S., Sohail, S. S., Aziz, A., Madsen, D. Ø., & Alalayah, K. M. (2023). An Optimization-Linked Intelligent Security Algorithm for Smart Healthcare Organizations. *Healthcare*, *11*(4), 580. https://doi.org/10.3390/healthcare11040580

Jayadeva, Khemchandani, R., & Chandra, S. (2007). Twin Support Vector Machines for Pattern Classification. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, *29*(5), 905–910. https://doi.org/10.1109/TPAMI.2007.1068

Jiang, J., Rimner, A., Deasy, J. O., & Veeraraghavan, H. (2022). Unpaired Cross-Modality Educed Distillation (CMEDL) for Medical Image Segmentation. *IEEE Transactions on Medical Imaging*, *41*(5), 1057–1068. https://doi.org/10.1109/TMI.2021.3132291

John. (2023, January 21). *Best CNC Dust Collectors [2023] for a Clean Shop*. MellowPine.

JPEG image. (n.d.). *.JPG File Extension*. FileInfo.Com.

Kang, J., Körner, M., Wang, Y., Taubenböck, H., & Zhu, X. X. (2018). Building instance classification using street view images. *ISPRS Journal of Photogrammetry and Remote Sensing*, *145*, 44–59. https://doi.org/10.1016/j.isprsjprs.2018.02.006

Kaur, R., GholamHosseini, H., Sinha, R., & Lindén, M. (2022). Melanoma Classification Using a Novel Deep Convolutional Neural Network with Dermoscopic Images. *Sensors*, *22*(3), 1134. https://doi.org/10.3390/s22031134

Khan, A. H., Siddqui, J., & Sohail, S. S. (2022). *A Survey of Recommender Systems Based on Semi-supervised Learning* (pp. 319–327). https://doi.org/10.1007/978-981-16-3071-2_27

Khan, S., Naseer, M., Hayat, M., Zamir, S. W., Khan, F. S., & Shah, M. (2022). Transformers in Vision: A Survey. *ACM Computing Surveys*, *54*(10s), 1–41. https://doi.org/10.1145/3505244

Kim, H., Kim, H., Hong, Y. W., & Byun, H. (2018). Detecting Construction Equipment Using a Region-Based Fully Convolutional Network and Transfer Learning. *Journal of Computing in Civil Engineering*, *32*(2). https://doi.org/10.1061/(ASCE)CP.1943-5487.0000731

Kim, S. J., Cho, K. J., & Oh, S. (2017). Development of machine learning models for diagnosis of glaucoma. *PLOS ONE*, *12*(5), e0177726. https://doi.org/10.1371/journal.pone.0177726

Kim, T.-H., Park, D.-C., Woo, D.-M., Jeong, T., & Min, S.-Y. (2012). *Multi-class Classifier-Based Adaboost Algorithm* (pp. 122–127). https://doi.org/10.1007/978-3-642-31919-8_16

Kortylewski, A., Albrecht, T., & Vetter, T. (2015). *Unsupervised Footwear Impression Analysis and Retrieval from Crime Scene Data* (pp. 644–658). https://doi.org/10.1007/978-3-319-16628-5_46

Koziarski, M., & Cyganek, B. (2018). Impact of Low Resolution on Image Recognition with Deep Neural Networks: An Experimental Study. *International Journal of Applied Mathematics and Computer Science*, *28*(4), 735–744. https://doi.org/10.2478/amcs-2018-0056

Krizhevsky, A., Sutskever, I., & Hinton, G. E. (2017). ImageNet classification with deep convolutional neural networks. *Communications of the ACM*, *60*(6), 84–90. https://doi.org/10.1145/3065386

Kuo, B.-C., Ho, H.-H., Li, C.-H., Hung, C.-C., & Taur, J.-S. (2014). A Kernel-Based Feature Selection Method for SVM With RBF Kernel for Hyperspectral Image Classification. *IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing*, *7*(1), 317–326. https://doi.org/10.1109/JSTARS.2013.2262926

Lăzăroiu, G., Andronie, M., Iatagan, M., Geamănu, M., Ștefănescu, R., & Dijmărescu, I. (2022). Deep Learning-Assisted Smart Process Planning, Robotic Wireless Sensor Networks, and Geospatial Big Data Management Algorithms in the Internet of Manufacturing Things. *ISPRS International Journal of Geo-Information*, *11*(5), 277. https://doi.org/10.3390/ijgi11050277

LeCun, Y., Bengio, Y., & Hinton, G. (2015). Deep learning. *Nature*, *521*(7553), 436–444. https://doi.org/10.1038/nature14539

Lee, H., Kim, S., & Lee, J. (2019). Shoeprint image retrieval using color and texture features. Forensic Science International, 302, 109925.

Li, Z., Zhang, C., Li, Y., & Liu, Y. (2020). Shoeprint image classification based on convolutional neural networks. IEEE Access, 8, 72449-72457.

Li, C. (2019). *Shoe Print Identification From Images With Convolutional Neural Network* [Thesis]. Auckland University of Technology.

Li, D., Li, Y., & Liu, Y. (2022). Shoeprint Image Retrieval Based on Dual Knowledge Distillation for Public Security Internet of Things. *IEEE Internet of Things Journal*, *9*(19), 18829–18838. https://doi.org/10.1109/JIOT.2022.3162326

Li, J., Wu, L.-H., Xu, M.-Y., Ren, J.-L., Li, Z., Liu, J.-R., Wang, A. J., & Chen, B. (2022). Improving Image Quality and Reducing Scan Time for Synthetic MRI of Breast by Using Deep Learning Reconstruction. *BioMed Research International*, *2022*, 1–9. https://doi.org/10.1155/2022/3125426

Liaw, A., & Wiener, M. (2002). Classification and Regression by randomForest. *R News*.

Liu, L., Xu, J., Huang, Y., & Zhan, Y. (2019). Improved Shoeprint Retrieval Algorithm Based on Convolutional Neural Networks. Neural Processing Letters, 50(3), 2435-2447.

Liu, W., & Xu, D. (2022). Robust and Efficient Shoe Print Image Retrieval Using Spatial Transformer Network and Deep Hashing. *Proceedings of the 4th International Symposium on Signal Processing Systems*, 89–95. https://doi.org/10.1145/3532342.3532356

Liu, W., Anguelov, D., Erhan, D., Szegedy, C., Reed, S., Fu, C.-Y., & Berg, A. C. (2016). *SSD: Single Shot MultiBox Detector* (pp. 21–37). https://doi.org/10.1007/978-3-319-46448-0_2

Liu, X., Li, H., Huang, C., & Huang, J. (2019). A novel shoeprint image classification system based on improved deep learning. PLoS ONE, 14(10), e0224157. https://doi.org/10.1371/journal.pone.0224157

Liu, Y., Pu, H., & Sun, D.-W. (2021). Efficient extraction of deep image features using convolutional neural network (CNN) for applications in detecting and analysing complex food matrices. *Trends in Food Science & Technology*, *113*, 193–204. https://doi.org/10.1016/j.tifs.2021.04.042

Liu, Y., Wang, Y., & Zhang, J. (2012). *New Machine Learning Algorithm: Random Forest* (pp. 246–252). https://doi.org/10.1007/978-3-642-34062-8_32

Lu, D., & Weng, Q. (2007). A survey of image classification methods and techniques for improving classification performance. *International Journal of Remote Sensing*, *28*(5), 823–870. https://doi.org/10.1080/01431160600746456

Lv, M., & Ma, J. (2016). Multiple modes of electrical activities in a new neuron model under electromagnetic radiation. *Neurocomputing*, *205*, 375–381. https://doi.org/10.1016/j.neucom.2016.05.004

Lyons, N. (2022). Deep Learning-based Computer Vision Algorithms, Immersive Analytics and Simulation Software, and Virtual Reality Modeling Tools in Digital Twin-driven Smart Manufacturing. *Economics, Management, and Financial Markets*, *17*(2), 67–81. https://doi.org/10.22381/emfm17220224

Ma, Z., Ding, Y., Wen, S., Xie, J., Jin, Y., Si, Z., & Wang, H. (2019). Shoe-Print Image Retrieval With Multi-Part Weighted CNN. *IEEE Access*, *7*, 59728–59736. https://doi.org/10.1109/ACCESS.2019.2914455

Malathi, M., & Srividya, M. (2015). Word Spotting in Scanned Tamil Land Documents using K-Nearest Neighbor. *International Journal of Science, Engineering and Computer Technology*, *5*(3), 62.

Mamun, Md. al, Akter, M., & Uddin, M. S. (2019). A Survey on Matching of Shoeprint with Reference Footwear in Forensic Study. *Journal of Computer and Communications*, *07*(09), 19–26. https://doi.org/10.4236/jcc.2019.79002

Mekhmoukh, A., Elbouzekri, A., Bounoua, F., & El Afia, A. (2021). Improved Deep Learning Model for Shoeprint Image Classification. Journal of Computational Science, 55, 101469. https://doi.org/10.1016/j.jocs.2021.101469

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. https://doi.org/10.1007/BF02478259

Montaseri, S., Ganjtabesh, M., & Zare-Mirakabad, F. (2016). Evolutionary Algorithm for RNA Secondary Structure Prediction Based on Simulated SHAPE Data. *PLOS ONE*, *11*(11), e0166965. https://doi.org/10.1371/journal.pone.0166965

Mufti, T., Sohail, S. S., Gupta, B., & Agarwal, P. (2022). *Sustainable Approach for Cloud-Based Framework Using IoT in Healthcare* (pp. 231–244). https://doi.org/10.1007/978-3-030-80702-3_14

Muraki, R., Teramoto, A., Sugimoto, K., Sugimoto, K., Yamada, A., & Watanabe, E. (2022). Automated detection scheme for acute myocardial infarction using convolutional neural network and long short-term memory. *PLOS ONE*, *17*(2), e0264002. https://doi.org/10.1371/journal.pone.0264002

N.A. (1996). European Meeting for Shoeprint /Toolmark Examiners Program. Finland, Helsinki. *Forensic Science International*, *82*(1), 1–120.

N.A. (2023). *Machine Learning Glossary*. Google Developers.

N.a. (n.d.). *Confusion Matrix Online Calculator*. Https://Onlineconfusionmatrix.Com.

Nieniewski, M., Chmielewski, L. J., Patrzyk, S., & Woźniacka, A. (2023). Studies in differentiating psoriasis from other dermatoses using small data set and transfer learning. *EURASIP Journal on Image and Video Processing*, *2023*(1), 7. https://doi.org/10.1186/s13640-023-00607-y

Ottoni, A. L. C., de Amorim, R. M., Novo, M. S., & Costa, D. B. (2023). Tuning of data augmentation hyperparameters in deep learning to building construction image classification with small datasets. *International Journal of Machine Learning and Cybernetics*, *14*(1), 171–186. https://doi.org/10.1007/s13042-022-01555-1

Pavel, M. I., Tan, S. Y., & Abdullah, A. (2022). Vision-Based Autonomous Vehicle Systems Based on Deep Learning: A Systematic Literature Review. *Applied Sciences*, *12*(14), 6831. https://doi.org/10.3390/app12146831

Peng, J., Zhou, Y., & Chen, C. L. P. (2015). Region-Kernel-Based Support Vector Machines for Hyperspectral Image Classification. *IEEE Transactions on Geoscience and Remote Sensing*, *53*(9), 4810–4824. https://doi.org/10.1109/TGRS.2015.2410991

Peng, S., Jiang, H., Wang, H., Alwageed, H., & Yao, Y.-D. (2017). Modulation classification using convolutional Neural Network based deep learning model. *2017 26th Wireless and Optical Communication Conference (WOCC)*, 1–5. https://doi.org/10.1109/WOCC.2017.7929000

Prajapati, G. L., & Patle, A. (2010). On Performing Classification Using SVM with Radial Basis and Polynomial Kernel Functions. *2010 3rd International Conference on Emerging Trends in Engineering and Technology*, 512–515. https://doi.org/10.1109/ICETET.2010.134

Priore, P., Ponte, B., Puente, J., & Gómez, A. (2018). Learning-based scheduling of flexible manufacturing systems using ensemble methods. *Computers & Industrial Engineering*, *126*, 282–291. https://doi.org/10.1016/j.cie.2018.09.034

Priya, C. A., Balasaravanan, T., & Thanamani, A. S. (2012). An efficient leaf recognition algorithm for plant classification using support vector machine. *International Conference on Pattern Recognition, Informatics and Medical Engineering (PRIME-2012)*, 428–432. https://doi.org/10.1109/ICPRIME.2012.6208384

Qadir, S., & Noor, B. (2021). Applications of Machine Learning in Digital Forensics. 2021 International Conference on Digital Futures and Transformative Technologies (ICoDT2), 1–8. https://doi.org/10.1109/ICoDT252288.2021.9441543

Rahali, A., & Akhloufi, M. A. (2023). End-to-End Transformer-Based Models in Textual-Based NLP. *AI*, *4*(1), 54–110. https://doi.org/10.3390/ai4010004

Rajyalakshmi, V., & Lakshmanna, K. (2022). A review on smart city - IoT and deep learning algorithms, challenges. *International Journal of Engineering Systems Modelling and Simulation*, *13*(1), 3. https://doi.org/10.1504/IJESMS.2022.122733

Ranftl, R., Bochkovskiy, A., & Koltun, V. (2021). *Vision Transformers for Dense Prediction*.

Rida, I., Fei, L., Proença, H., Nait-Ali, A., & Hadid, A. (2019). *Forensic shoe-print identification: a brief survey*.

Rosenblatt F. (1957). *The perceptron—a perceiving and recognizing automaton*.

Rostami, M., Kolouri, S., Eaton, E., & Kim, K. (2019). SAR Image Classification Using Few-Shot Cross-Domain Transfer Learning. *2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops (CVPRW)*, 907–915. https://doi.org/10.1109/CVPRW.2019.00120

Russakovsky, O., Deng, J., Su, H., Krause, J., Satheesh, S., Ma, S., Huang, Z., Karpathy, A., Khosla, A., Bernstein, M., Berg, A. C., & Fei-Fei, L. (2015). ImageNet Large Scale Visual Recognition Challenge. *International Journal of Computer Vision*, *115*(3), 211–252. https://doi.org/10.1007/s11263-015-0816-y

Sadman, K. S. (2020, July 1). *How to Calculate Confusion Matrix Manually.* Analytics Vidhya.

Sapkota, A., Li, Y., & Jia, W. (2020). Shoeprint Classification using Convolutional Neural Network. In 2020 IEEE International Conference on Imaging Systems and Techniques (IST) (pp. 1-6). IEEE.

Sawyer, N. E. (1995). "Shoe-fit" - a computerised shoe print database. *European Convention on Security and Detection*, 86–89. https://doi.org/10.1049/cp:19950475

Senthil, P., & Selvakumar, S. (2022). A hybrid deep learning technique based integrated multi-model data fusion for forensic investigation. *Journal of Intelligent & Fuzzy Systems*, *43*(5), 6849–6862. https://doi.org/10.3233/JIFS-221307

Sheykhmousa, M., Mahdianpari, M., Ghanbari, H., Mohammadimanesh, F., Ghamisi, P., & Homayouni, S. (2020). Support Vector Machine Versus Random Forest for Remote Sensing Image Classification: A Meta-Analysis and Systematic Review. *IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing*, *13*, 6308–6325. https://doi.org/10.1109/JSTARS.2020.3026724

Shuzhanfan. (n.d.). *Understanding Deep Residual Networks*. GitHub.

Siddiqui, A. A., Sohail, S. S., Areeb, Q., Mansoor, W., & Nafis, M. T. (2022). Classification of Cumin, Fennel and Carom Using Transfer Learning. *2022 5th International Conference on Signal Processing and Information Security (ICSPIS)*, 46–49. https://doi.org/10.1109/ICSPIS57063.2022.10002538

Siddiqui, F., Mohammad, A., Alam, M. A., Naaz, S., Agarwal, P., Sohail, S. S., & Madsen, D. Ø. (2023). Deep Neural Network for EEG Signal-Based Subject-Independent Imaginary Mental Task Classification. *Diagnostics*, *13*(4), 640. https://doi.org/10.3390/diagnostics13040640

Simonyan, K., & Zisserman, A. (2014). *Very Deep Convolutional Networks for Large-Scale Image Recognition*.

Smith, J., Johnson, A., & Williams, R. (2020). Transfer learning in deep convolutional neural networks for image classification. Journal of Artificial Intelligence Research, 15(2), 123-145.

Soyoung, P., & Alicia, C. (2020). Footwear outsole impression. In *Lowa State Univerisity*.

Srihari, S. (2010). Analysis of Footwear Impression Evidence. *Research Foundation of the State University of New York*, 1–81.

Srihari, S. N., & Tang, Y. (2014). *Computational Methods for the Analysis of Footwear Impression Evidence* (pp. 333–383). https://doi.org/10.1007/978-3-319-05885-6_15

Srivastava, S., Vargas-Muñoz, J. E., & Tuia, D. (2019). Understanding urban landuse from the above and ground perspectives: A deep learning, multimodal solution. *Remote Sensing of Environment*, *228*, 129–143. https://doi.org/10.1016/j.rse.2019.04.014

Su, H., Crookes, D., Bouridane, A., & Gueham, M. (2007). Local Image Features for Shoeprint Image Retrieval. *Procedings of the British Machine Vision Conference 2007*, 38.1-38.10. https://doi.org/10.5244/C.21.38

Tan, M., & Le, Q. v. (2019). *EfficientNet: Rethinking Model Scaling for Convolutional Neural Networks*.

Tang, Y. Y., Lu, Y., & Yuan, H. (2015). Hyperspectral Image Classification Based on Three-Dimensional Scattering Wavelet Transform. *IEEE Transactions on Geoscience and Remote Sensing*, *53*(5), 2467–2480. https://doi.org/10.1109/TGRS.2014.2360672

Tang, Y., Srihari, S. N., Kasiviswanathan, H., & Corso, J. J. (2011). *Footwear Print Retrieval System for Real Crime Scene Marks* (pp. 88–100). https://doi.org/10.1007/978-3-642-19376-7_8

Tao, S., Chen, D., & Zhao, W. (2009). Fast pruning algorithm for multi-output LS-SVM and its application in chemical pattern classification. *Chemometrics and Intelligent Laboratory Systems*, *96*(1), 63–69. https://doi.org/10.1016/j.chemolab.2008.12.001

Tiwari, A., Yadav, A. K., & Bagaria, V. (2022). Application of deep learning algorithm in automated identification of knee arthroplasty implants from plain radiographs using transfer learning models: Are algorithms better than humans? *Journal of Orthopaedics*, *32*, 139–145. https://doi.org/10.1016/j.jor.2022.05.013

Tsuneki, M. (2022). Deep learning models in medical image analysis. *Journal of Oral Biosciences*, *64*(3), 312–320. https://doi.org/10.1016/j.job.2022.03.003

Vagac, M., Povinsky, M., & Melichercik, M. (2017). Detection of shoe sole features using DNN. *2017 IEEE 14th International Scientific Conference on Informatics*, 416–419. https://doi.org/10.1109/INFORMATICS.2017.8327285

Varun, B. (2020, April 5). *The Evolution of Deep Learning Explore the growth of Deep Learning over the past decades!* Towards Data Science.

Vella, J., & Farrugia, N. (2021). Footwear Impressions Retrieval Through Textures and Local Features. InDigital Transformation ,Cyber Security and Resilience of Modern Societies. *Springer, Cham*.

Viola, Jones, & Snow. (2003). Detecting pedestrians using patterns of motion and appearance. *Proceedings Ninth IEEE International Conference on Computer Vision*, 734–741 vol.2. https://doi.org/10.1109/ICCV.2003.1238422

Viola, P., & Jones, M. J. (2004). Robust Real-Time Face Detection. *International Journal of Computer Vision*, *57*(2), 137–154. https://doi.org/10.1023/B:VISI.0000013087.49260.fb

Wang, X., Sun, H., Yu, Q., & Zhang, C. (2015). *Automatic Shoeprint Retrieval Algorithm for Real Crime Scenes* (pp. 399–413). https://doi.org/10.1007/978-3-319-16865-4_26

Wang, X., Wu, Y., & Zhang, T. (2019). Multi-Layer Feature Based Shoeprint Verification Algorithm for Camera Sensor Images. *Sensors*, *19*(11), 2491. https://doi.org/10.3390/s19112491

Wei, J., Wang, H., Zhao, T., Jiang, Y.-L., & Wan, J. (2023). A New Compact MOSFET Model Based on Artificial Neural Network With Unique Data Preprocessing and Sampling Techniques. *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, *42*(4), 1250–1254. https://doi.org/10.1109/TCAD.2022.3193330

Wenhua, Z., Qamar, F., Abdali, T.-A. N., Hassan, R., Jafri, S. T. A., & Nguyen, Q. N. (2023). Blockchain Technology: Security Issues, Healthcare Applications, Challenges and Future Trends. *Electronics*, *12*(3), 546. https://doi.org/10.3390/electronics12030546

William J. Bodziak. (2017). *Forensic Footwear Evidence: Detection, Recovery and Examination, Second Edition* (2nd ed.). CRC Press.

Wu, Y., Dong, X., Shi, G., Zhang, X., & Chen, C. (2022a). Crime Scene Shoeprint Image Retrieval: A Review. *Electronics*, *11*(16), 2487. https://doi.org/10.3390/electronics11162487

Wu, Y., Dong, X., Shi, G., Zhang, X., & Chen, C. (2022b). Crime Scene Shoeprint Image Retrieval: A Review. *Electronics*, *11*(16), 2487. https://doi.org/10.3390/electronics11162487

Xu, J.-L., & Hsu, Y.-L. (2022). Analysis of agricultural exports based on deep learning and text mining. *The Journal of Supercomputing*, *78*(8), 10876–10892. https://doi.org/10.1007/s11227-021-04238-w

Xuhong, L., Yves, G., & Franck, D. (2018). Explicit Inductive Bias for Transfer Learning with Convolutional Networks. *International Conference on Machine Learning*.

Yan, X., Li, M., Xiao, J., Yu, Z., Li, J., & Zhao, Y. (2020). Deep Transfer Learning for Image Classification. Neurocomputing, 416, 161-171. https://doi.org/10.1016/j.neucom.2020.07.014

Yang, L., Aghaabbasi, M., Ali, M., Jan, A., Bouallegue, B., Javed, M. F., & Salem, N. M. (2022). Comparative Analysis of the Optimized KNN, SVM, and Ensemble DT Models Using Bayesian Optimization for Predicting Pedestrian Fatalities: An Advance towards Realizing the Sustainable Safety

Yang, L., Yang, F., & Noguchi, N. (2011). Apple Internal Quality Classification Using X-ray and SVM. *IFAC Proceedings Volumes*, *44*(1), 14145–14150. https://doi.org/10.3182/20110828-6-IT-1002.01827

Yu, D., Wang, H., Chen, P., & Wei, Z. (2014). Mixed Pooling for Convolutional Neural Networks (pp. 364–375). https://doi.org/10.1007/978-3-319-11740-9_34

Yue, X., Li, H., Shimizu, M., Kawamura, S., & Meng, L. (2022). YOLO-GD: A Deep Learning-Based Object Detection Algorithm for Empty-Dish Recycling Robots. *Machines*, *10*(5), 294. https://doi.org/10.3390/machines10050294

Zhang, Z., Bian, Y., & Ping, X. (2008). Image Blind Forensics Using Artificial Neural Network. 2008 International Conference on Computer Science and Software Engineering, 847–850. https://doi.org/10.1109/CSSE.2008.1620

Zhang, L., & Allinson, N. (2005). *Automatic Shoeprint Retrieval System for use in Forensic Investigations*.

Appendix

Appendix 1: Programming Code for Models

The following is the code using Python for implementing all the models used, including CNN, and CNN-TL. These are referenced in the paragraph where appropriate citations are required. Different sections of the programming codes are tabulated.

```
# import visualkeras

# from collections import defaultdict

# color_map = defaultdict(dict)

# color_map[Conv2D]['fill'] = 'orange'

# color_map[Dropout]['fill'] = 'pink'
```

```
from PIL import Image

import numpy as np

import pandas as pd
```

The above code initiates the import of necessary python libraries.

```
root = "./sorted_shoeprints/"

full_shoe = []

# r=root, d=directories, f = files

for r, d, f in os.walk(root + "Full shoe prints"):

for file in f:

full_shoe.append(file)

full_shoe.sort(key=lambda f: int(''.join(filter(str.isdigit, f))))

partial_shoe = []

# r=root, d=directories, f = files

for r, d, f in os.walk(root + "Partial shoe prints"):

for file in f:

partial_shoe.append(file)

partial_shoe.sort(key=lambda f: int(''.join(filter(str.isdigit, f))))
```

```
root = "./sorted_shoeprints/"

full_shoe = []

# r=root, d=directories, f = files

for r, d, f in os.walk(root + "Full shoe prints"):

for file in f:

full_shoe.append(file)

full_shoe.sort(key=lambda f: int(''.join(filter(str.isdigit, f))))

partial_shoe = []

# r=root, d=directories, f = files
```

```
X = []

Y = []

for image in full_shoe[:]:

 img = Image.open(root + 'Full shoe prints/' + image)

img.load()

data = np.asarray(img)
```

```
for image in partial_shoe[:]:

# print(image)

img = Image.open(root + 'Partial shoe prints/' + image)

img.load()

data = np.asarray(img)

data = cv2.cvtColor(data, cv2.COLOR_BGRA2BGR)

data = cv2.resize(data, (150, 150))

X.append(data)

Y.append(1)
```

```
X = np.array(X)

Y = np.array(Y)



print(X.shape, Y.shape)



print(pd.Series(Y).value_counts())



from tensorflow.keras.utils import to_categorical

from keras.utils import np_utils

Y = np_utils.to_categorical(Y)



from sklearn.model_selection import train_test_split

X_train, X_test, Y_train, Y_test = train_test_split(X,Y, test_size = 0.20,␣

,→random_state=2)

print(X_train.shape, Y_train.shape, X_test.shape, Y_test.shape)
```

```
import seaborn as sns

import matplotlib.pyplot as plt

df_train=pd.DataFrame()

df_test=pd.DataFrame()

# df_val=pd.DataFrame()

df_train['Train']=pd.Series(np.argmax(Y_train, axis = 1))

df_test['Test']=pd.Series(np.argmax(Y_test, axis = 1))

# df_test['Test']=pd.Series(test_label)

# df_val['Val']=pd.Series(val_label)

# df['Val']=pd.Series(y_val)

fig, ax =plt.subplots(1,2,figsize=(15,7))

sns.countplot(df_train['Train'], ax=ax[0])

ax[0].set(ylim=(0, 1100))

sns.countplot(df_test['Test'], ax=ax[1])

# sns.distplot(test_label, kde=True, rug=False)

ax[1].set(ylim=(0, 1400))

# sns.countplot(df_val['Val'], ax=ax[2])

# ax[2].set(ylim=(0, 200))

# plt.savefig('D.png', dpi=100)
```

```
from skimage.io import imshow

import matplotlib.pyplot as plt

idx = int(np.random.rand(1,1)*600)

imshow(np.squeeze(X_train[idx,:,:,:]))

plt.show()

print("Index is:", idx)

labels = ['Full Shoe Print','Partial Shoe Print']

print ('This is:',labels[int(np.argmax(Y_train[idx], axis=0))])
```

```
from tensorflow.keras.models import Sequential

from tensorflow.keras.layers import Conv2D, Activation, MaxPooling2D, Conv2D,␣

,→Flatten, Dense

from keras.layers import Dense, Dropout, Flatten, Conv2D, MaxPool2D,␣

,→BatchNormalization

import keras

from tensorflow.keras.optimizers import Adam

import tensorflow as tf

model = Sequential()

model.add(Conv2D(128, (3, 3), activation="relu", input_shape=(150, 150, 3)))

model.add(BatchNormalization())

model.add(MaxPool2D(pool_size=(2, 2)))

model.add(Dropout(0.3))

# model.add(Conv2D(128, (3, 3),activation='relu'))

# model.add(BatchNormalization())

# model.add(MaxPool2D(pool_size=(2, 2)))

# model.add(Dropout(0.3))
```

```
model.add(Conv2D(64, (3, 3),activation='relu'))

model.add(BatchNormalization())

model.add(MaxPool2D(pool_size=(2, 2)))

model.add(Dropout(0.3))

model.add(Flatten())

model.add(Dense(32))

model.add(Dense(2, activation='softmax'))

# model = Sequential()

# model.add(Conv2D(input_shape=(150,150,3), filters=6, kernel_size=5,␣
```

```
,→strides=1, padding="same"))

# model.add(Activation('relu'))

# model.add(MaxPooling2D(pool_size=(4)))

# model.add(Conv2D(16, kernel_size=5, strides=1, padding="same"))

# model.add(Activation('relu'))

# model.add(MaxPooling2D(pool_size=(4)))

# model.add(Flatten())

# model.add(Dense(2))

# model.add(Activation('softmax'))
```

```
model.compile(loss="categorical_crossentropy",

optimizer=Adam(lr=0.001),

metrics=['accuracy'])
```

```
verbose = 1

cnnhistory=model.fit(X_train, Y_train, batch_size=16, epochs=20, ␣

,→verbose=verbose, validation_data=(X_test, Y_test))

# \_, accuracy = model.evaluate(X_test, Y_test, verbose=verbose)
```

```
score = model.evaluate(X_test, Y_test)

print('Test accuracy:', score[1])model.summary()
```

```
# To plot the accuracy and loss image/graph for the training & validation process.

loss = cnnhistory.history['loss']

val_loss = cnnhistory.history['val_loss']

epochs = range(1, len(loss) + 1)

plt.plot(epochs, loss, 'y', label='Training loss')

plt.plot(epochs, val_loss, 'r', label='Validation loss')

plt.title('Training and validation loss')

plt.xlabel('Epochs')

plt.ylabel('Loss')

plt.legend()

# plt.savefig('Loss.png', dpi=100)

plt.show()

acc = cnnhistory.history['accuracy']

val_acc = cnnhistory.history['val_accuracy']

plt.plot(epochs, acc, 'y', label='Training acc')

plt.plot(epochs, val_acc, 'r', label='Validation acc')

plt.title('Training and validation accuracy')

plt.xlabel('Epochs')

plt.ylabel('Accuracy')

plt.legend()

# plt.savefig('Accuracy.png', dpi=100)
```

```
from sklearn.metrics import confusion_matrix, classification_report

# Prediction on test data

y_pred = model.predict(X_test)

# Convert predictions classes to one hot vectors

y_pred_classes = np.argmax(y_pred, axis = 1)

# Convert test data to one hot vectors

y_true = np.argmax(Y_test, axis = 1)

print("Classification Report : \n",classification_report(y_true,y_pred_classes))
```

```
#Print confusion matrix

cm = confusion_matrix(y_true, y_pred_classes)
```

```
import seaborn as sns

fig, ax = plt.subplots(figsize=(6,6))

sns.set(font_scale=1.6)

sns.heatmap(cm, annot=True,fmt='d')

# linewidths=.5, ax=ax)

# plt.savefig('CM.png', dpi=100)
```

```
from sklearn.metrics import roc_curve,roc_auc_score

from sklearn.metrics import auc

fpr , tpr , thresholds = roc_curve ( y_true , y_pred_classes)

auc_keras = auc(fpr, tpr)

print("AUC Score:",auc_keras)

plt.figure()

lw = 2

plt.plot(fpr, tpr, color='darkorange',

lw=lw, label='ROC curve (area = %0.2f)' % auc_keras)

plt.plot([0, 1], [0, 1], color='navy', lw=lw, linestyle='--')

plt.xlim([0.0, 1.0])

plt.ylim([0.0, 1.05])

plt.xlabel('False Positive Rate')

plt.ylabel('True Positive Rate')

plt.title('Receiver operating characteristic example')

plt.legend(loc="lower right")

plt.show()
```

```
import matplotlib.pyplot as plt

import numpy as np

import pandas as pd

import os

from glob import glob

import seaborn as sns

from PIL import Image

import cv2

np.random.seed(42)

from sklearn.metrics import confusion_matrix

import keras

from keras.utils.np_utils import to_categorical # used for converting labels to␣

,→one-hot-encoding

from keras.models import Sequential

from keras.layers import Dense, Dropout, Flatten, Conv2D, MaxPool2D,␣

,→BatchNormalization

from sklearn.model_selection import train_test_split

from scipy import stats

from sklearn.preprocessing import LabelEncoder
```

```
pre_trained_model.load_weights(weights_file)

# freeze the layers

for r in range(len(pre_trained_model.layers)-17):

pre_trained_model.layers[r].trainable = False

print(pre_trained_model.layers[r])

# pre_trained_model.summary()

last_layer = pre_trained_model.get_layer('mixed7')

print('last layer output shape: ', last_layer.output_shape)

last_output = last_layer.output

# Flatten the output layer to 1 dimension

x = layers.Flatten()(last_output)

# Add a fully connected layer with 1,024 hidden units and ReLU activation

x = layers.Dense(1024, activation='relu')(x)

# Add a final sigmoid layer for classification

x = layers.Dense(2, activation='softmax')(x)

model = Model(pre_trained_model.input, x)
```

```
model.compile(optimizer=RMSprop(lr=0.00005),

loss='categorical_crossentropy',
```

```
# compile the model

checkpoint_filepath = './check'

model_checkpoint_callback = tf.keras.callbacks.ModelCheckpoint(

filepath=checkpoint_filepath+'/weights',

save_weights_only=True,

monitor='val_acc',

mode='max',

save_best_only=True)

model_checkpoint = tf.keras.callbacks.ModelCheckpoint(

filepath=checkpoint_filepath,

save_weights_only=False,

monitor='val_acc',

mode='max',
```

```
#plot the training and validation accuracy and loss at each epoch

loss = history.history['loss']

val_loss = history.history['val_loss']

epochs = range(1, len(loss) + 1)

plt.plot(epochs, loss, 'y', label='Training loss')

plt.plot(epochs, val_loss, 'r', label='Validation loss')

plt.title('Training and validation loss Inception V3')

plt.xlabel('Epochs')

plt.ylabel('Loss')

plt.legend()

plt.savefig('Loss.png',bbox_inches='tight')

plt.show()
```

```
from sklearn.metrics import confusion_matrix, classification_report

# y_true = np.argmax(y_val, axis = 1)

# print("Classification Report : \n",classification_report(y_true,np.

,→argmax(model.predict(X_val), axis = 1))

print("Classification Report : \n",classification_report(Y_test,model.
```

```
from sklearn.metrics import confusion_matrix, classification_report

# Prediction on test data

y_pred = model.predict(X_test)



# Convert predictions classes to one hot vector

y_pred_classes = np.argmax(y_pred, axis = 1)
```