Kiwifruit Detection and Tracking from A Deep Learning Perspective Using Digital Videos

Yi Xia

A thesis submitted to the Auckland University of Technology

in partial fulfillment of the requirements for the degree of

Master of Computer and Information Sciences (MCIS)

2023

School of Engineering, Computer & Mathematical Sciences

Abstract

With the growing popularity of ChatGPT, deep learning is rapidly advancing, leading to the development of new techniques and applications in various domains, including agriculture. Fruit detection, tracking, and counting play vital roles in crop management and yield prediction in the agricultural automation industry. However, conventional machine learning methods rely on manual inspection and are labor-intensive and prone to errors. In contrast, deep learning-based visual object detection and tracking algorithms have gained attention for their potential to improve the accuracy and speed of fruit detection and counting.

In this thesis, we propose a novel approach for kiwifruit counting in videos that integrates state-of-the-art models with Kalman filter algorithm. Our method leverages the visual object detection capability of the improved YOLOv8 model to identify and locate individual kiwifruits in images, while the Kalman filter tracks their position and trajectory over time, even when partially occluded or obscured by other objects. Duplicate counting is reduced using the Hungarian algorithm for matching.

We evaluate the effectiveness of our approach on a dataset of kiwifruit images and videos for training and performance assessment. Our results show that the proposed approach outperforms to the existing methods in terms of accuracy and robustness in detecting, tracking, and counting kiwifruits. Our kiwifruit detection module achieved a mean average precision at intersection over union of 95.6%, after combined with the kiwifruit tracking and counting module, resulted in an average counting accuracy of 0.782. Our research contributions include labeling a practical kiwifruit dataset, implementing attention mechanisms and modifying the IoU in the detection model to improve fruit detection accuracy, and enhancing the yield prediction model through the integration of a Kalman filter tracking model for kiwifruit counting.

Keywords: Object detection, YOLO, multiple object tracking, CNN, Kalman filtering, Hungarian algorithm, agricultural automation

Table of Contents

List of Figures

List of Tables

Attestation of Authorship

I hereby declare that this submission is my own work and that, to the best of my knowledge and belief, it contains no material previously published or written by another person (except where explicitly defined in the 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.

Date: 21 April 2023

Acknowledgment

I would like to take this opportunity to express my sincere gratitude to the individuals and organizations who have contributed to the completion of this thesis.

Firstly, I would like to extend my heartfelt thanks to my primary supervisor, Wei Qi Yan, for his unwavering support, encouragement, and invaluable guidance throughout the entire research process. His knowledge, expertise, and constructive feedback have been instrumental in shaping the direction and focus of this thesis. I am also deeply grateful to my second supervisor Minh Nguyen, for his insightful comments, helpful suggestions, and invaluable support in various stages of my research. His expertise and guidance have been of immense help in this thesis.

Furthermore, I would like to express my gratitude to my family and my wife for their unwavering support, encouragement, and understanding throughout my academic journey. I would also like to acknowledge and thank my colleagues in the laboratory who have provided valuable feedback, assistance, and support throughout my research. Their insights and expertise have been an important contribution to the success of this thesis.

Lastly, I would like to thank the Auckland University of Technology (AUT) for providing me with the necessary academic resources and facilities to pursue my academic goals. The academic and research opportunities provided by the university have been invaluable to my growth and development as a scholar.

Once again, I would like to express my gratitude to everyone who has contributed to the completion of this thesis. Your support and encouragement have been instrumental in my academic achievements, and I am truly grateful for your kindness and generosity.

Yi Xia

Auckland, New Zealand

April 2023

Chapter 1
Introduction

This chapter is composed of five parts. The first part shows an introduction to the background and motivation of yield prediction using deep learning methods. The subsequent parts cover the research questions, followed by the contributions, objectives, and structure of this thesis.

1.1 Background and Motivation

In recent years, deep learning has made remarkable progress, and various deep learning methodologies have been proposed for diverse applications (Carion et al., 2020). Meanwhile, visual object detection and tracking algorithms have emerged as a promising area of research, owing to their potential to enhance the efficiency and accuracy of crop management and yield prediction in the agricultural domain. Conventional techniques for crop management primarily rely on manual inspection, which is labor-intensive and time-consuming (Gu et al., 2017).

Deep learning-based visual object detection and tracking algorithms have been shown to provide efficient and accurate solutions for detecting, tracking, and counting fruits in agricultural settings. Kiwifruit industry in New Zealand is one of the country's significant industries and one of the world's largest kiwifruit exporters. The kiwifruit plantation area and production are considerable, mainly distributed in the coastal areas of the North and South Islands. New Zealand's kiwifruit is well known globally for its excellent quality, unique taste, and rich nutritional value and is widely exported to Europe, Asia, North America, and other regions. To maintain its competitive position, kiwifruit farmers in New Zealand have been striving to improve production efficiency, optimize management, and develop new technologies (Ferguson, 2004). In recent years, with the development of deep learning technology, more and more people have begun to explore its application in kiwifruit cultivation and management to improve production efficiency and quality (Yan, 2021).

Accurate prediction of kiwifruit yield is crucial for optimizing crop management and achieving maximum yield. However, traditional methods based on manual counting for kiwifruit yield prediction are prone to errors and time-consuming. Therefore, there is an urgent need for a reliable and efficient kiwifruit yield prediction method (Li et al., 2022). In this regard, deep learning-based algorithms have demonstrated substantial potential in increasing the accuracy and efficiency of kiwifruit yield prediction (He et al., 2022). The kiwifruit counting process may be automated, and the accuracy of yield forecast can be greatly improved, by using deep learning-based object identification and tracking algorithms (Chen et al., 2017).

The motivation for this study is to propose a kiwifruit yield prediction approach that is based on state-of-the-art deep learning methodologies. The proposed methodology aims to integrate advanced object detection and tracking algorithms to achieve kiwifruit counting and yield prediction. The motivation for this thesis originates from the need to overcome the limitations of traditional kiwifruit yield prediction methods, which are time-consuming and prone to errors (Dorj, Lee & Yun, 2017). By exploiting the superior performance of deep learning algorithms, we intend to provide a reliable and efficient solution for kiwifruit yield prediction, which can significantly enhance the efficiency and accuracy of crop management.

The proposed approach is expected to have significant implications for the agricultural industry, as it can provide farmers and growers with information on kiwifruit yield, enabling them to optimize crop management strategies (Fountas et al., 2020). Furthermore, the proposed approach can be extended to other fruit crops after training, thereby contributing to the advancement of precision agriculture (Koirala et al., 2017). Therefore, the motivation of this thesis is not only to propose a solution for kiwifruit yield prediction but also to make a significant contribution to the broader field of precision agriculture (Pan & Yan, 2020).

1.2 Research Questions

As previously mentioned, the objective of this thesis is to utilize deep learning algorithms for the purpose of kiwifruit counting in orchards, while also enhancing the accuracy and efficiency of kiwifruit recognition, tracking, and counting through model modifications. Consequently, the research inquiries for this thesis are as follows:

(1) *What is the effectiveness of different object detection models in detecting highly overlapping kiwifruits in orchards?*

The existing visual object detection models possess distinctive strengths. Our aim is to determine, through comparative experiments, which specific model is better suited for detecting small and highly overlapping kiwifruits.

(2) *How can existing object detection models be enhanced to improve their performance in detecting kiwifruits?*

In order to enhance the performance of the object detection model, a multitude of methods will be employed in this study, and their efficacy will be validated through ablation experiments.

(3) *Can the fusion of the improved object detection model with a Multiple Object Tracking (MOT) model enable kiwifruit counting in digital videos?*

A high-performance object detection model serves as the foundation for kiwifruit counting. However, it is only by feeding the output of the improved visual object detection model into a MOT model that kiwifruit counting can be achieved.

The central aim of this thesis is to accomplish the detection, tracking, and counting of kiwifruits in a real orchard. Therefore, it is imperative to enhance and evaluate the performance of various techniques applied to kiwifruit detection. Subsequently, we will input the prediction results into a suitable MOT model to achieve tracking. Based on comparative experiments and ablation studies regarding algorithm performance, we will select an appropriate methodology for our specific scenario..

1.3 Contributions

In this thesis, we address the challenge of kiwifruit detection, tracking and counting, which is a critical task in modern agriculture. In recent years, the use of deep learning algorithms has shown great potential for improving fruit detection and counting accuracy, as well as reducing the labor and cost required for manual harvesting (Gao, Yang & Fu, 2021; Gongal, Karkee & Amatya, 2018). However, there are still significant challenges to overcome, such as developing efficient and accurate models for detecting and tracking fruits in complex outdoor environments, where lighting, weather, and occlusions can vary widely. To address these challenges, we propose a novel approach that integrates several advanced techniques in computer vision. Our contributions are summarized as follows:

(1) We create a new kiwifruit dataset by collecting, preprocessing, and annotating images to facilitate the training and validation of kiwifruit detection and segmentation algorithms with higher specificity. The dataset contains a diverse range of kiwifruit images, including those with varying degrees of occlusion and lighting conditions, which is essential for the development of robust and generalizable algorithms. The dataset is publicly available, and we hope it will facilitate further research in the field of fruit detection and yield prediction.

(2) We propose improvements to the YOLOv8 model by introducing the attentional mechanisms strategy and adopting a modified loss function, which lead to significant performance gains in kiwifruit detection. Through extensive experiments and comparisons with state-of-the-art detection methods, we demonstrated the effectiveness and robustness of our proposed algorithm in detecting kiwifruits accurately and efficiently, even in challenging scenarios.

(3) We develop a kiwifruit tracking algorithm based on the Kalman filter and Hungarian algorithm, which can effectively track multiple kiwifruits simultaneously in a video sequence with high accuracy and efficiency. By integrating the proposed tracking algorithm with the improved YOLOv8 model, we achieved better kiwifruit counting performance, especially in crowded and occluded scenes. Our approach can be easily extended to other fruit crops and can contribute to precision agriculture and crop management.

Overall, we provide a reliable and efficient solution for kiwifruit detection and tracking, which can significantly improve the accuracy and efficiency of crop management in agriculture.

1.4 Objectives of This Thesis

The goal of this thesis is to develop a deep learning-based strategy for predicting kiwifruit production. This research project intends to provide a full pipeline for kiwifruit recognition, tracking, and counting utilizing cutting-edge deep learning techniques. The proposed method will be tested on our kiwifruit dataset to determine its accuracy and efficiency in real-world agricultural situations. The following thesis will be carried out to reach this goal:

Firstly, we conduct a comprehensive literature review of existing fruit detection, tracking, and counting methods in agriculture, with a focus on deep learning-based approaches.

Secondly, we will collect and pre-process a kiwifruit dataset for training and evaluating the proposed methods.

Thirdly, we will optimize an object detection model based on YOLOv8 to achieve high accuracy and efficiency in kiwifruit detection.

Fourthly, we will combine algorithms such as Kalman filtering and Hungarian algorithm with the high-precision output of the detection model to achieve kiwifruit tracking and counting.

Finally, we will compare the proposed methods with state-of-the-art kiwifruit yield prediction methods, including both target detection algorithms and multi-object tracking algorithms.

The ultimate goal of this thesis is to provide a reliable and efficient solution for kiwifruit yield prediction while overcoming the challenge of significant fruit overlap in kiwifruit detection. This solution will significantly improve the accuracy and efficiency of crop management in agriculture.

1.5 Structure of This Thesis

We provide a brief introduction to the background and significance of kiwifruit detection and tracking in agriculture in Chapter 1. We outline the challenges faced in this area, and how our proposed research can contribute to address these challenges. Moreover, we provide a clear statement of our research questions, objectives, and hypotheses to guide the development of our methodology.

In Chapter 2, we discuss the related work on fruit detection and tracking in agriculture, highlighting the strengths and limitations of existing approaches. This discussion provides a foundation for our proposed research and helps to identify the gaps in the existing literature that our research aims to address.

In Chapter 3, we describe the suggested research methodologies in depth, including the exact deep learning models, optimisation techniques, and assessment measures that will be employed in our trials. We also go through the experimental design, which includes the data collection procedure, pre-processing stages, and experimental setup. This chapter clarifies our technique and guarantees that our experiments are carried out in a systematic and thorough manner.

In Chapter 4, we report the outcomes of our trials, including a quantitative evaluation of our suggested technique. We also present outcomes performance to assist in the analysis and comprehension of our findings.

In Chapter 5, we summarize the experimental results, and discuss the implications of our findings for kiwifruit detection and tracking in agriculture. We also compare our results to the existing literature and highlight the contributions of our proposed method.

Finally, in Chapter 6, we conclude our work by summarizing the key findings and contributions of our research, as well as discussing the limitations and future directions of our proposed approach. We also reflect on the broader impact of our research on the field of agriculture and deep learning and provide recommendations for future research.

Chapter 2
Literature Review

The objective of this thesis is deep learning-based visual object detection and multiobject tracking. The goal of this chapter is to give a thorough assessment of the literature on important classical and deep learning approaches. The literature review will include previous theoretical and empirical research, with a focus on significant ideas, hypotheses, and findings pertinent to our enquiry.

2.1 Introduction

In recent years, deep learning has turned up as a powerful technology that has achieved significant breakthroughs across various domains. Numerous industries, including agriculture, have experienced considerable improvements and advancements due to the widespread application of deep learning. The chapter of this thesis primarily focuses on the applications of deep learning in the agricultural sector, particularly in the detection, tracking, and counting of fruit crops. These processes play a crucial role in crop management and yield prediction.

Traditional fruit detection and counting techniques predominantly rely on labour-intensive and error-prone manual inspection. However, deep learning-based object detection and tracking algorithms have garnered considerable attention due to their potential to accelerate these processes while enhancing the accuracy of fruit detection and counting. By leveraging the exceptional performance of deep learning algorithms, these techniques provide efficient and precise solutions for fruit detection, tracking, and counting in agricultural settings (Zhao & Yan, 2021).

In this chapter, we begin by introducing the developmental background of deep learning and its applications in the agricultural sector. Following that, we will delve into the exploration of deep learning applications in agriculture (Olaniyi, Oyedotun & Adnan, 2016; Nguyen et al., 2016). Subsequently, we will introduce the techniques for fruit detection and counting. Next, we will discuss the applications and achievements of deep learning-based object detection models in the domain of fruit detection. Furthermore, we will examine the applications of object tracking techniques in fruit tracking. Afterward, we will explore the integration of these technologies to achieve fruit counting in videos. Finally, this chapter will summarize existing research on deep learning in fruit detection, tracking, and counting, and propose future research directions and potential challenges. The goal of this chapter is to offer readers with a full overview of deep learning applications in fruit recognition, tracking, and counting, allowing them to have a deeper appreciation of its usefulness and promise in agricultural automation.

2.2 Background of Deep Learning

2.2.1 Artificial Neural Networks

Artificial neural networks are modelled after the way of human brain functions, which is made up of linked neurones arranged in layers (McCulloch & Pitts, 1943). By sending signals between linked neurones, these computational models imitate the information processing capabilities of organic neural networks. A neural network is composed of an input layer, one or multiple hidden layers, and an output layer. Neurons within each layer are connected to neurons in the subsequent layer through weighted connections, and each neuron processes incoming information using an activation function. The primary objectives of neural networks are pattern learning, prediction, and data classification based on input-output relationships. Deep neural networks, comprising multiple hidden layers, have emerged as powerful tools for various tasks, including image recognition, natural language processing, and speech recognition (LeCun, Bengio, & Hinton, 2015). These networks' deep architectures enable the extraction of higher-level features and representations, resulting in superior performance compared to shallow networks.

In digital image processing tasks, deep neural networks have set new benchmarks, surpassing human-level performance in some instances (Krizhevsky, Sutskever, & Hinton, 2017). They have been successfully applied to object detection, facial recognition, and image segmentation (Lv et al., 2019), significantly impacting computer vision research and applications. Furthermore, deep neural networks have completely changed how robots comprehend and produce human language in the field of natural language processing. Modern breakthroughs have been made in tasks including machine translation, sentiment analysis, and question answering by models like BERT (Kamath, Graham, & Emara, 2022) and GPT (Radford et al., 2018). Deep neural network has made tremendous advancements in speech recognition as well. These networks have achieved more accurate and robust speech-to-text conversion, even in noisy environments or with varying accents. The success of deep neural networks can be attributed to several factors, including advancements in hardware such as GPUs (Graphic Processing Unit), which provide powerful computational capabilities for effectively training large-scale models (Raina et al., 2009), and the availability of vast amounts of labeled data, which facilitate supervised learning (Sun et al., 2017). Additionally, the development of novel optimization algorithms and activation functions has promoted more efficient training processes (Kingma & Ba, 2014).

Artificial neural networks, especially deep neural networks with multiple hidden layers, have proven to perform very well in a variety of tasks, including speech recognition, digital image recognition, and natural language processing. Ongoing advancements in neural network architectures, optimization techniques, and hardware capabilities are expected to further enhance their performance and expand their applicability across diverse domains.

2.2.2 Convolutional Neural Networks

Convolutional Neural Networks (CNNs) have been designed explicitly to handle grid-like data, prominently images (LeCun et al., 1998). Convolutional Neural Networks (CNNs) possess a structure that comprises convolutional, pooling, and fully connected layers that collaborate to accurately identify local attributes and develop spatial hierarchies from the input data. CNNs have demonstrated their outstanding potential in a range of computer vision applications, including but not limited to image classification, object detection, and segmentation.

Convolutional Neural Networks (CNNs), as proposed by LeCun et al. (1998), have revolutionized the field of computer vision by incorporating a distinct method for image processing. This technique involves the application of convolutional operations, allowing CNNs to effectively identify patterns and features across a range of scales and complexities. These networks' convolutional layers are made to scan the input data by using a variety of filters or kernels. In order to create feature maps that emphasise the most important information, these filters glide across the input data while conducting element-wise multiplication and aggregation (Goodfellow, Bengio, & Courville, 2016). Contrarily, pooling layers are essential for lowering the spatial dimensions of the feature maps, which enhances computing efficiency and makes the feature maps more resilient to changes in the input data (Goodfellow, Bengio, & Courville, 2016). By employing such unique processing techniques, CNNs have successfully demonstrated their ability to recognize and extract meaningful information from complex visual data, making them an indispensable tool in various computer vision tasks.

The remarkable performance of Convolutional Neural Networks (CNNs) can be significantly attributed to the integration of various optimization techniques and advanced activation functions. Adaptive optimization algorithms, such as Adam, have been instrumental in offering efficient approaches for updating network weights during the training process. These algorithms enable faster convergence rates and result in improved overall performance, thereby making them essential for the success of CNN architectures. Moreover, the incorporation of advanced activation functions has played a vital role in enhancing the capabilities of CNNs. Rectified Linear Units (ReLU) (Nair & Hinton, 2010) and Leaky ReLU (Maas, Hannun, & Ng, 2013) are prime examples of such functions. These activation functions aid in addressing the vanishing gradient issue, which has historically plagued the training of deep neural networks. By mitigating the vanishing gradient problem, these activation functions allow for the effective training of deeper and more complex network architectures. Consequently, this has enabled CNNs to achieve superior performance in various computer vision tasks, demonstrating the importance of these advancements in optimization techniques and activation functions.

Data augmentation techniques have been instrumental in significantly improving the performance of Convolutional Neural Networks (CNNs) across various tasks. These techniques involve applying diverse transformations to training images, such as rotations, translations, and flips, which enhances the diversity of the training set. Consequently, this leads to the development of more robust and generalizable models, allowing CNNs to effectively handle a wider range of input variations and perform well in real-world scenarios.

Additionally, transfer learning has emerged as a powerful approach in the application of CNNs, particularly in situations where access to large amounts of labeled data is limited (Yosinski, Clune, Bengio, & Lipson, 2014). By using this technique, it is possible to fine-tune previously trained models for new tasks with comparatively lesser quantities of accessible data after they have been originally trained on large-scale datasets like ImageNet (Deng et al., 2009). By using the information gained from previously learned features and patterns, transfer learning enables quicker convergence rates and greater performance on target tasks. As a consequence, this strategy dramatically increased the applicability of CNNs across a variety of domains and problems, highlighting the need for data augmentation and transfer learning approaches for CNN architectures' ongoing success (Li et al., 2021).

The advancements in CNNs have significantly impacted the field of computer vision and have been applied to numerous applications beyond image recognition. For example, CNNs have been employed in digital video analysis tasks, such as action recognition, as well as in medical imaging for tasks like tumor segmentation (Milletari, Navab, & Ahmadi, 2016) and disease diagnosis (Gulshan et al., 2016). Additionally, CNNs have been integrated with other deep learning models, such as Recurrent Neural Networks (RNNs), to tackle problems that involve both spatial and temporal information, like image captioning (Vinyals et al., 2015) and visual question answering (Antol et al., 2015).

In recent years, neural networks have shown remarkable performance across various domains, such as speech recognition, image identification, and natural language processing. Deep neural networks with multiple hidden layers, in particular, have been found to be especially effective. Convolutional Neural Networks (CNNs), a specialized type of deep learning model, have made significant advancements in the field of computer vision by addressing key problems such as image classification, object recognition, and segmentation. The success of CNNs can be attributed to the integration of various optimization techniques, activation functions, data augmentation strategies, and transfer learning approaches. Additionally, CNNs have been employed in areas such as medical image processing, video analysis, and have been integrated with other deep learning models such as Recurrent Neural Networks (RNNs) to address complex problems involving both spatial and temporal information, such as image captioning and visual question answering (Bazame et al., 2021). It is anticipated that with the ongoing advancements in neural network architectures, optimization techniques, and hardware capabilities, the performance of deep learning models will continue to improve, finding applications across a broader range of domains.

2.3 Deep Learning in Agriculture

2.3.1 Crop Disease Detection

Deep learning has been increasingly employed in the realm of crop disease detection, with Convolutional Neural Networks (CNNs) demonstrating their efficacy in identifying and classifying plant diseases based on leaf images. One of the pioneering projects in this domain was conducted by Mohanty et al. (2016), who developed a CNN-based model to detect 14 crops and 26 diseases, highlighting the potential of deep learning in automated diagnosis of agricultural diseases.

Subsequent research has furthered the application of deep learning in plant disease monitoring, exploring various network architectures, optimization techniques, and data augmentation strategies to enhance detection performance. For instance, Ramcharan et al. (2017) developed a CNN-based approach for detecting diseases in cassava plants, achieving over 90% accuracy by leveraging transfer learning and fine-tuning pre-trained models on their dataset. Similarly, Ferentinos (2018) employed deep learning techniques for early detection of plant diseases using thermal and hyperspectral imagery, demonstrating the versatility of deep learning in handling diverse data sources.

Moreover, the incorporation of ensemble learning techniques has been explored to improve the robustness and generalization capabilities of deep learning models in plant disease detection. For example, Too et al. (2019) proposed a model ensemble approach that combined multiple CNNs to increase the reliability and accuracy of disease classification. By aggregating the predictions of individual CNNs, their ensemble model achieved better performance compared to single CNN-based approaches.

In addition to CNNs, other deep learning models, such as Generative Adversarial Networks (GANs), have been applied to synthesize realistic leaf images with various disease symptoms, effectively augmenting training datasets and enhancing model performance (Barbedo, 2019). This demonstrates the potential of GANs in addressing data scarcity issues, which often pose challenges in training deep learning models for crop disease detection.

These research contributions underscore the growing importance of deep learning in crop disease monitoring and the myriad of possibilities for further advancements in the field. Future research directions may include the development of more efficient and lightweight network architectures suitable for deployment on mobile devices or drones, the integration of multi-modal data sources such as satellite imagery, and the investigation of unsupervised and semi-supervised learning approaches to address the scarcity of labeled data in agricultural applications.

2.3.2 Weed Identification

Weed identification is another essential application of deep learning in agriculture, with considerable implications for precision farming and resource management. A deep learning-based method (Hung & Sukkarieh, 2014) was proposed for detecting and classifying weeds using Unmanned Aerial Vehicle (UAV) imagery. The approach achieved high accuracy, demonstrating the potential of deep learning in this domain.

Following Hung and Sukkarieh (2014) pioneering work, various deep learning techniques and approaches were developed to improve weed identification and classification. For example, Milioto et al. (2018) proposed a semantic segmentation method based on CNNs to identify and distinguish between crops and weeds. Their approach utilized RGB images captured by a camera mounted on a mobile robot, emphasizing the integration of deep learning with robotics for autonomous weed management. Moreover, researchers have investigated the fusion of different data modalities, such as multispectral and hyperspectral imagery, to enhance weed detection and classification performance. Skovsen et al. (2017) employed a CNN-based approach to classify weeds using multispectral images, illustrating the benefits of incorporating spectral information beyond the visible range.

Similarly, Li et al. (2021) explored the use of hyperspectral imaging and deep learning for early weed detection, highlighting the potential for more timely and targeted interventions in weed management. In addition to the use of CNNs, other deep learning models have been applied to weed identification tasks. For instance, Urmashev et al. (2021) employed a combination of deep learning and reinforcement learning to develop a precision agriculture robotic system capable of detecting and removing weeds. Their approach demonstrated the potential for integrating advanced learning techniques with agricultural robotics to automate weed management processes. As with crop disease detection, the application of deep learning in weed identification has been extended to address data scarcity and augmentation issues. Researchers have explored the use of GANs for generating synthetic images of weeds to augment existing datasets, thereby improving model performance and generalization capabilities (Fawakherji et al., 2020).

Deep learning has been effectively applied to weed identification in agriculture, contributing to precision farming and resource management. Researchers have developed various techniques, including CNN-based semantic segmentation, fusion of multispectral and hyperspectral imagery, and integration with reinforcement learning and robotics for autonomous weed management. Additionally, they have addressed data scarcity and augmentation challenges by employing GANs to generate synthetic weed images, enhancing model performance and generalization. These advancements demonstrate the potential of deep learning for improving weed detection and classification in agriculture.

2.3.3 Crop Yield Prediction

Deep learning has increasingly gained prominence in the field of crop yield prediction, with various models demonstrating promising results for assisting farmers in making informed decisions concerning crop management. Khalil and Abdullaev (2021) explored the use of Convolutional Neural Networks (CNNs) and Recurrent Neural Networks (RNNs) for predicting crop yields by leveraging satellite imagery and historical yield data. Their findings revealed that deep learning models could yield accurate predictions, highlighting their potential for practical applications in agriculture.

Building on the pioneering work, alternative deep learning architectures and data sources have been investigated to further enhance crop yield prediction capabilities. For instance, some studies have incorporated Long Short-Term Memory (LSTM) networks, a type of RNN, to model temporal dependencies in crop yield data (Sun, Di, & Fang, 2018). By effectively capturing temporal patterns, these models have shown improved performance in predicting crop yields over traditional machine learning techniques.

In addition to satellite imagery, researchers have begun to explore the integration of other data sources, such as weather data, soil information, and agricultural management practices, to enrich the input features for deep learning models. This multimodal approach aims to capture the complex relationships between various factors affecting crop yields, ultimately leading to more accurate and reliable predictions (Chlingaryan, Sukkarieh, & Whelan, 2018). Moreover, recent studies have focused on the development of region-specific and crop-specific deep learning models to address the unique challenges associated with different agricultural contexts. For example, Li et al. (2020) proposed a deep learning framework for maize yield prediction in Northeast China, incorporating region-specific factors to enhance prediction accuracy.

In summary, deep learning has exhibited significance in the domain of crop yield prediction. With ongoing research focusing on the development of more sophisticated architectures, the integration of diverse data sources, and the consideration of region-specific factors, it is anticipated that deep learning models will continue to advance and provide valuable support to farmers in their crop management decisions.

2.4 Fruit Detection and Counting Techniques

2.4.1 Image Segmentation Methods

Image segmentation techniques have gained considerable attention in the context of fruit detection and counting tasks, given the ability to separate and identify different regions within an image. Ninomiya (2022) proposed a method for apple detection that relied on the excess green index (ExG) and the scale-invariant feature transform (SIFT) algorithm. Despite the efficacy of these approaches, they may encounter difficulties when handling occlusions, varying lighting conditions, and diverse fruit shapes, prompting researchers to explore alternative techniques to address these challenges (Liu et al., 2019).

In response to the limitations of traditional image segmentation methods, several studies have investigated the application of deep learning techniques, particularly Convolutional Neural Networks (CNNs), for fruit detection and counting tasks. CNN-based approaches have demonstrated a higher degree of robustness and adaptability in handling complex scenes, occlusions, and lighting variations, compared to traditional image processing techniques (Rahnemoonfar & Sheppard, 2017). For example, Bargoti and Underwood (2017) employed a deep learning-based approach to detect and count fruits in orchard environments. Their model utilized a CNN to process high-resolution images, successfully identifying and counting fruits while accounting for occlusions and varying illumination conditions. This research highlights the potential of deep learning models to overcome the challenges associated with traditional image segmentation techniques. Another avenue of research in fruit detection and counting has focused on fusing different data modalities, such as depth and spectral information, to improve detection performance.

Gene-Mola et al. (2019) proposed a multi-modal deep learning framework that combined color and depth information to detect fruits in complex scenes. Their approach demonstrated improved performance in handling occlusions and diverse fruit shapes compared to single-modal methods. Furthermore, researchers have explored the use of advanced deep learning architectures, such as Mask R-CNN and YOLO (You Only Look Once), to enhance fruit detection and counting capabilities (Machefer et al., 2020). These models have shown promising results in terms of accuracy and performance, indicating their potential for practical applications in agriculture (Song, Chen & Liu, 2019).

In summary, while traditional image segmentation techniques have been utilized for fruit detection and counting tasks, deep learning approaches have emerged as a promising alternative. By offering enhanced robustness and adaptability, deep learning models have demonstrated their potential to overcome the challenges associated with occlusions, varying lighting conditions, and diverse fruit shapes, opening up new possibilities for advanced fruit detection and counting techniques in agricultural contexts.

2.4.2 Edge-Based Approaches

Edge-based fruit detection techniques focus on identifying the boundaries between objects and their surroundings, which can facilitate the recognition of fruits in agricultural settings. By detecting edges within an image, these approaches can isolate fruit regions and subsequently classify and count them.

Nanaa et al. (2014) developed an edge-based approach for mango detection, employing the Canny edge detection and Hough transform algorithms. The Canny edge detection algorithm works by identifying areas of rapid intensity change in an image, while the Hough transform is utilized to find shapes, such as circles or lines, within the detected edges. By combining these two algorithms, Nanaa et al. (2014) were able to accurately identify and count mangoes in their study. Despite the potential of edge-based approaches, they may have some limitations. For example, these methods can be sensitive to noise, which can adversely affect the detection of edges and, consequently, the identification and counting of fruits. Additionally, edge-based techniques may struggle with occlusions, varying lighting conditions, and diverse fruit shapes, which could lead to inaccurate detection and counting results. To overcome these challenges, researchers have proposed various enhancements to edge-based methods. For instance, adaptive thresholding techniques have been employed to improve the performance of Canny edge detection in the presence of noise (Lakshmi & Sankaranarayanan, 2010). Moreover, the integration of edge-based approaches with other computer vision techniques, such as region-based or texture-based methods, can help to achieve more robust fruit detection and counting performance in complex agricultural environments (Hassoon, 2021).

In a nutshell, while edge-based approaches have demonstrated their effectiveness in fruit detection and counting tasks, there is still room for improvement and further research to address their limitations and enhance their performance in real-world agricultural applications.

2.4.3 Region-based Methods

Region-based fruit detection techniques have become increasingly popular in agricultural applications due to their ability to partition images into distinct regions and subsequently classify them as fruit or non-fruit areas. Guadagna et al. (2023) effectively employed the region growing algorithm for grapevine detection and segmentation, yielding encouraging results. However, these methods can encounter challenges when dealing with occlusions, uneven illumination, and varying fruit sizes.

To overcome these obstacles, researchers have investigated various strategies and refinements aimed at enhancing the performance of region-based fruit detection techniques. For instance, some studies have combined region-based approaches with color-based techniques, leveraging the distinctive color attributes of different fruits to improve the discrimination between fruit and non-fruit regions in images (Gongal et al., 2015).

Another in-line research work involves integrating region-based methods with traditional machine learning techniques, such as support vector machines (SVM) or decision trees. By incorporating features extracted from region-based techniques as input to classifiers, researchers have been able to improve fruit detection performance (Kanimozhi & Latha, 2015). Furthermore, advanced image processing algorithms and techniques, including morphological operations and texture analysis, have been explored to enhance fruit detection and segmentation in complex agricultural settings (Awate et al., 2015).

Despite the progress made in region-based fruit detection techniques, further research and development are necessary to address the inherent challenges associated with occlusions, uneven illumination, and varying fruit sizes. As the field advances, it is anticipated that more sophisticated and robust approaches will emerge, enabling accurate and efficient fruit detection and counting across diverse agricultural contexts.

2.4.4 Deep Learning-based Approaches

Deep learning-based fruit detection methods have emerged as a promising approach for addressing complex environments and occlusions, offering superior performance compared to traditional computer vision techniques. Cecotti et al. (2020) successfully employed a CNN-based approach for grape detection, achieving high accuracy and underscoring the advantages of deep learning for fruit detection tasks. In addition to fruit detection, deep learning techniques have been applied to fruit counting tasks, demonstrating remarkable progress in recent years. For instance, Bargoti and Underwood (2017) utilized a CNN-based method for apple counting in orchards, achieving high precision and recall rates.

Similarly, Kamilaris and Prenafeta-Boldú (2018) proposed a deep learning framework for counting citrus fruits using RGB and depth data, demonstrating the potential of fusing different data modalities to improve counting performance. Researchers have also explored various deep learning architectures and strategies to further enhance fruit detection and counting capabilities. For example, fully convolutional networks (FCNs) have been adopted for semantic segmentation, enabling the simultaneous detection and localization of fruits in digital images (Toda & Okura, 2019). Moreover, approaches such as Faster R-CNN and YOLOv3 have been adapted for fruit detection and counting tasks, offering processing capabilities and higher accuracy rates (Buzzy et al., 2020). Transfer learning, a technique that leverages pre-trained deep learning models, has also been employed in fruit detection and counting tasks. By fine-tuning pre-trained models on domain-specific data, researchers have been able to achieve superior performance with relatively small datasets (De Luna et al., 2020).

In short, deep learning-based approaches have shown significant promise for fruit detection and counting tasks, overcoming challenges related to occlusions, complex environments, and varying fruit sizes. With continued advancements in deep learning techniques and the growing availability of annotated datasets, it is anticipated that these approaches will become increasingly robust and widely adopted in the agricultural domain.

2.5 Object Detection Models for Fruit Detection

2.5.1 R-CNN, Fast R-CNN, and Faster R-CNN

Region-based Convolutional Neural Networks (R-CNNs) (Girshick et al., 2014), Fast R-CNNs (Girshick, 2015), and Faster R-CNNs (Ren et al., 2015) are deep learning-based object detection models that have been instrumental in advancing the field. These models have distinct architectures and methodologies for extracting region proposals from images and classifying them using CNNs.

R-CNN (Girshick et al., 2014), the earliest of these models, consists of three main steps: (1) generating region proposals using an external method like selective search, (2) extracting features from each region proposal using a CNN, and (3) classifying the features using a support vector machine (SVM) classifier. While effective, this process is computationally expensive due to the separate steps of region proposal generation and feature extraction.

By combining the feature extraction and classification phases, Fast R-CNN enhances the R-CNN model (Girshick, 2015). Fast R-CNN uses the CNN to the entire image once and creates a feature map rather than extracting features for each area proposed individually. Then, it extracts features from the feature map that match the area suggestions using a region of interest (ROI) pooling layer. In order to categorise the areas, it uses a fully connected layer and a softmax classifier.

To improve object identification, Faster R-CNN introduces a Region Proposal Network (RPN) that generates region proposals directly within the neural network. This eliminates the need for external region proposal generating methods used in previous models like R-CNN and Fast R-CNN. The RPN is a fully convolutional network that shares the convolutional layers of the detection network, making the process more efficient. The RPN-generated region proposals are combined with CNN-generated feature maps, and then passed through ROI pooling, fully connected layers, and a softmax classifier for final classification (Ren et al., 2017).

In the context of fruit detection, R-CNN and Fast R-CNN have been successfully applied to various tasks. For example, Gao et al. (2016) implemented an R-CNN-based approach to detect apples in orchard images, achieving promising results. Likewise, Quan et al. (2019) proposed a method for maize seedling detection in complex backgrounds using Faster R-CNN and demonstrated its effectiveness in dealing with complex backgrounds and occlusions. Despite their success, R-CNN and Fast R-CNN suffer from computational inefficiencies, mainly due to the separate steps of region proposal generation and classification. This characteristic hinders their suitability for applications, particularly in scenarios where rapid processing is essential, such as in autonomous harvesting robots or on-the-fly yield estimation.

Ren et al. (2017) proposed Faster R-CNN as an improved object detection model that reduces computational limitations experienced in R-CNN and Fast R-CNN. The key innovation of Faster R-CNN is the introduction of a Region Proposal Network (RPN) that generates region proposals directly within the neural network, resulting in a more streamlined detection process. Several studies have applied Faster R-CNN in fruit detection tasks and have reported successful outcomes. For example, Yu et al. (2019) utilized Faster R-CNN for strawberry detection, achieving high accuracy and demonstrating its potential for various applications. Similarly, Wan and Goudos (2020) utilized Faster R-CNN for mango detection and achieved remarkable performance in terms of both precision and recall.

While Faster R-CNN overcomes some of the computational challenges of R-CNN and Fast R-CNN, it still requires considerable computational resources, limiting its applicability in resource-constrained environments or on low-power devices. The advancements in object detection models, such as R-CNN, Fast R-CNN, and Faster R-CNN, have significantly contributed to the progress of fruit detection tasks. However, researchers continue to explore more efficient and accurate models to meet the demands of applications and resource-limited scenarios.

2.5.2 Single Shot MultiBox Detector (SSD)

Single Shot MultiBox Detector (SSD) (Liu et al., 2016) is a single-stage object detection model that streamlines the object detection process by eliminating the need for separate region proposal and classification steps. This simplification results in faster detection performance, making SSD particularly suitable for applications. The SSD architecture consists of a base network, typically a pre-trained CNN, followed by a series of convolutional layers with varying sizes. These layers are designed to detect objects at different scales and aspect ratios. SSD employs default bounding boxes, or anchor boxes, for each feature map cell. During training, the model predicts both the class scores and the box offsets relative to the anchor boxes. The final predictions are obtained by applying non-maximum suppression (NMS) to the combined set of predicted boxes.

SSD has been successfully applied to a variety of applications in the field of fruit detection. Wang et al. (2022), for example, offer a lightweight SSD object detection approach for detecting Lingwu long jujubes in a natural setting. The improved SSD technique achieves excellent detection accuracy without the need of pre-trained weights and reduces complexity to allow for mobile platform adoption. The addition of a coordinate attention module and a global attention method improves object detection accuracy. Furthermore, the SSD model has been customised and optimised for certain fruit detecting tasks. To increase the model's performance and resilience, researchers investigated the use of data augmentation techniques such as random cropping, flipping, and colour distortion (Li et al., 2018).

In summary, the SSD model offers a fast and efficient single-stage object detection approach well-suited for fruit detection tasks. Its streamlined architecture and adaptability make it a valuable tool in the agricultural domain, particularly when combined with other deep learning techniques or data modalities.

2.5.3 YOLO (You Only Look Once) Family

The YOLO (You Only Look Once) family of models (Redmon et al., 2016) encompasses a collection of single-stage object detection architectures that treat object detection as a regression problem. These models have gained popularity due to their simplicity, speed, and capabilities, rendering them highly suitable for a variety of fruit detection tasks in agricultural applications.

In essence, the YOLO architecture divides the input image into a grid, with each grid cell responsible for predicting bounding boxes and class probabilities. The model undergoes end-to-end training to minimize the combined localization and classification loss. The final predictions are obtained by applying non-maximum suppression (NMS) to the predicted boxes. The YOLO model has been effectively employed in numerous fruit detection tasks, such as tomato detection (Liu et al., 2022), highlighting its potential in agricultural settings. Our previous study proposed an improved kiwifruit detection model based on YOLOv7 (Xia, Nguyen & Yan, 2023). The model was trained using a manually labeled and data-augmented kiwifruit image dataset. To improve the identification of visual features, an attention module was included to YOLOv7. The findings demonstrated that the proposed strategy outperformed the original YOLOv7 model in terms of detection accuracy.

The YOLO family, consisting of YOLOv2 (Redmon and Farhadi, 2017), YOLOv3, YOLOv4, YOLOv5, YOLOv6, YOLOv7 (Xia, Nguyen & Yan, 2023), and the current state-of-the-art model YOLOv8 (Lou et al., 2023), has undergone iterative development resulting in notable improvements in accuracy, speed, and overall performance. These advancements are realized through architectural refinements, such as the adoption of anchor boxes, enhanced feature extraction via skip connections, and the incorporation of various loss functions.

In conclusion, the YOLO family of models offers a fast and efficient single-stage object detection approach for fruit detection tasks in agriculture. Their simplicity, adaptability, and capabilities render them invaluable tools for various fruit detection applications, particularly when integrated with other deep learning techniques or data modalities.

2.6 Object Tracking Techniques for Fruit Tracking

2.6.1 Optical Flow

Optical flow is a popular technique in object tracking for predicting object motion across consecutive frames in a video clip. Optical flow can be used to follow the movement and growth of fruits in diverse agricultural situations in the context of fruit tracking. Optical flow is a technique that uses intensity changes between consecutive frames to estimate the apparent velocity of objects in a video clip. The fundamental assumption is that the intensity of a moving item remains constant during a short period of time. The optical flow field, represented by a vector field, indicates the motion and direction of objects in the image. Several algorithms for computing optical flow have been developed, including the Lucas-Kanade method (Lucas and Kanade, 1981) and the Horn-Schunck method (Horn and Schunck, 1981).

Optical flow has been applied in various fruit tracking tasks to estimate the motion and growth of fruits over time. For instance, Yang et al. (2020) used the Lucas-Kanade optical flow method for tracking apple movements in an orchard, achieving accurate tracking results despite the presence of occlusions and varying lighting conditions. In another study, a deep learning object detector based on Faster R-CNN architecture and optical flow for object tracking was used to count fruit on apple trees in RGB video sequences (Fu et al., 2020). The methodology minimized counting errors due to occluded and clustered fruit (Jarvinen et al., 2018).

Despite the successes of optical flow in fruit tracking, challenges remain. Noise, illumination changes, and occlusions can negatively impact the accuracy of optical flow estimations. Additionally, the computational complexity of optical flow algorithms can hinder their applicability in scenarios. To overcome these challenges, researchers have proposed combining optical flow with other tracking techniques, such as deep learning-based methods, to improve tracking performance (Hur & Roth, 2020). Furthermore, the development of more efficient algorithms for optical flow computation can enhance its applicability in fruit tracking applications.

In short, optical flow has demonstrated its potential for fruit tracking in various agricultural settings. By addressing the challenges and exploring the integration of optical flow with other tracking techniques, optical flow-based fruit tracking can continue to advance, ultimately benefiting the agricultural sector.

2.6.2 Mean Shift

Mean Shift is a non-parametric clustering algorithm that has been applied to object tracking tasks, including fruit tracking, due to its robustness to noise, shape changes, and illumination variations (Comaniciu & Meer, 1999). The algorithm is based on the assumption that different clusters of data conform to different probability density distributions. It seeks to determine the direction of maximum density increase for any given data point (the Mean Shift) and assumes that data points convergent on the same local maximum are from the same cluster. Mean Shift has been widely employed in image segmentation, clustering, and video tracking applications. The mean shift technique can be used to track the location of fruits based on colour, texture, or form attributes in the context of object tracking.

Mean shift has been employed in various fruit tracking tasks, demonstrating its effectiveness in handling challenging conditions such as occlusions and illumination changes. For instance, Yang et al. (2020) proposed a tree-structured image segmentation method that combines adaptive mean shift and image abstraction to improve the segmentation of tree images with complex backgrounds in natural environments. This method focuses on the background and tree-specific features of the image and has achieved good results.

While the mean shift algorithm has shown promise in fruit tracking tasks, it is not without limitations. One notable challenge is selecting an appropriate bandwidth parameter, which can significantly impact the performance of the algorithm. Additionally, the mean shift may struggle with tracking objects that undergo substantial changes in appearance over time or in situations where similar objects with overlapping features are present. To address these challenges, researchers have proposed various methods to adapt and extend the mean shift algorithm, such as adaptive mean shift and scale-adaptive mean shift, aiming to improve the performance of the algorithm by adjusting the bandwidth parameter or incorporating scale information.

In short, the mean shift algorithm has exhibited considerable potential in tracking fruits in diverse agricultural settings. However, it is crucial to consider its limitations in the context of object tracking. The algorithm is sensitive to scale and initialization, which can lead to inaccuracies when tracking fruits that experience significant changes in size or appearance. Furthermore, the mean shift algorithm may encounter difficulties when tracking multiple targets or handling occlusions, where one fruit may be partially or fully obscured by another. To overcome these limitations, researchers have been exploring the integration of mean shift with other state-of-the-art tracking techniques, such as deep learning-based methods, particle filters, or Kalman filters. Combining the mean shift algorithm with complementary techniques can result in more accurate and robust fruit tracking in challenging agricultural environments.

2.6.3 Kalman Filter

Kalman Filter, originally proposed by Kalman in 1960, is a recursive estimation algorithm that has been extensively applied in diverse object tracking tasks, including fruit tracking, owing to its capability to estimate the state of a system under conditions of noise and uncertainty (Kalman, 1960). The algorithm functions by continuously updating the system state using observed measurements and a dynamic model that describes the temporal changes of the system. In the context of fruit tracking, the Kalman Filter can be utilized to forecast the position and velocity of fruits based on their preceding states, enabling the tracking of fruits in agricultural settings.

In the literature, several studies have employed the Kalman Filter for fruit tracking. For instance, Gao et al. (2022) applied the Kalman Filter to track apple positions in orchards using video streams. By incorporating a dynamic model of apple motion and employing the filter to estimate apple position and velocity, they were able to obtain accurate tracking results despite noise and uncertainty in the measurements. In another study, Itakura et al. (2021) proposed study utilized YOLOv2 to automatically detect pears and apples in videos captured during walking, followed by identifying the same fruits in successive frames using a Kalman filter. The proposed method achieved automatic fruit counting in videos, even under unstable lighting conditions and with green-colored fruits.

Despite its advantages, the Kalman Filter has certain limitations when applied to fruit tracking. The algorithm assumes that system dynamics and measurement noise follow Gaussian distributions, which may not always be the case in real-world agricultural settings. Additionally, the Kalman Filter is a linear estimation algorithm, and its performance may degrade when dealing with nonlinear motion models or non-Gaussian noise distributions. To address these limitations, researchers have developed the Extended Kalman Filter (Julier and Uhlmann, 1997), which can more effectively handle nonlinear systems and non-Gaussian noise distributions.

Ultimately, Kalman Filter has demonstrated its potential for tracking fruits in various agricultural applications. By addressing its limitations and exploring the integration of the Kalman Filter with other tracking techniques, such as deep learning-based methods, particle filters, or mean shift algorithms, fruit tracking performance can be further improved (Apolo-Apolo et al., 2020).

2.7 Multi-object Tracking Models

2.7.1 Data Association Techniques

Data association techniques play a pivotal role in multi-object tracking (MOT), as they facilitate the establishment and maintenance of correspondences between object detections and their respective trajectories across consecutive frames. In this section, we provide a brief overview of several data association techniques employed in MOT, and describe the fundamental principles underlying these algorithms.

Hungarian algorithm, proposed by Kuhn (1955), was designed to solve assignment problems. Within the context of MOT, it is utilized to find the optimal matching between object detections and existing tracks, minimizing the total cost of the assignment. Costs are typically defined as distance measures, such as the Euclidean distance between predicted and observed object positions. The algorithm involves constructing a cost matrix, reducing the matrix, and iteratively updating the matrix until the optimal assignment is found.

Global Nearest Neighbor (GNN) algorithm is a straightforward data association technique that assigns each detection to the closest track based on a distance metric, such as the Euclidean distance (Radosavljevic, 2006). GNN operates on a frame-by-frame basis, rendering it computationally efficient, but prone to errors when objects are closely spaced or have similar appearances.

Joint Probabilistic Data Association (JPDA) algorithm, introduced by Fortmann, Bar-Shalom and Scheffe (1983), takes into account the uncertainty in both measurements and associations. It calculates the probabilities of all potential associations between detections and tracks and updates the tracks based on these probabilities. This approach enables JPDA to handle closely spaced objects and multiple detections more effectively than GNN.

Multiple Hypothesis Tracking (MHT) algorithm, proposed by Blackman (2004), maintains multiple tracking hypotheses for each object, considering various possible connections between detections and tracks. It evaluates each hypothesis based on a likelihood function and selects the most probable hypothesis as the final trajectory. Although MHT is more robust than other data association techniques, its computational complexity can be prohibitive for applications.

These pioneering data association methods have significantly advanced the field of multi-object tracking, paving the way for a variety of applications. However, each technique has its own limitations. The Hungarian algorithm, while effective in solving assignment problems, may struggle with occlusions or rapidly changing object dynamics (Kuhn, 1955). GNN's simplicity makes it computationally efficient, but it is susceptible to errors in cases of closely spaced or visually similar objects. JPDA addresses the uncertainty inherent in measurements and associations, but its probabilistic nature might lead to computational challenges as the number of objects increases. MHT, despite its robustness, suffers from high computational complexity, which restricts its suitability for applications. To overcome these limitations, researchers have continued to develop novel data association techniques and hybrid approaches, aiming to achieve a balance between accuracy, computational efficiency, and robustness in multi-object tracking.

2.7.2 Probabilistic Graphical Models

Probabilistic graphical models have emerged as a powerful framework for multi-object tracking (MOT) due to their ability to model complex dependencies between variables and to efficiently perform inference on these dependencies. In this section, we provide an overview of several probabilistic graphical models used in MOT and discuss the fundamental principles behind these algorithms.

Bayesian networks, alternatively referred to as directed graphical models, describe the conditional relationships between random variables via a directed acyclic graph (DAG) (Pearl, 1988). In the domain of MOT, Bayesian networks offer a means to model the connections between object detections, tracks, and their associated uncertainties. By applying Bayesian inference, it becomes possible to estimate the posterior distribution of the object tracks, accounting for both prior knowledge and observed data. Nonetheless, the primary constraint of Bayesian networks lies in their need for numerous parameters to model intricate dependencies, which makes them computationally demanding.

In the field of multiple object tracking (MOT), Markov random fields (MRFs) are utilized as undirected graphical models to depict the joint probability distribution of random variables using an undirected graph (Kindermann & Snell, 1980). By doing so, MRFs can reflect the spatial and temporal dependencies between object detections and tracks. The optimal data association can be determined by minimizing an energy function, which is established based on the graph's structure and the inter-relationships among its nodes. However, determining the optimal solution can be a computationally challenging task, as it frequently entails solving an NP-hard problem.

Factor graphs provide a more general framework for representing probabilistic graphical models, as they can capture both directed and undirected relationships between variables (Kschischang et al., 2001). In MOT, factor graphs can be used to model the dependencies between object detections, tracks, and other relevant variables, such as object appearances or motion patterns. Inference on factor graphs can be performed efficiently using techniques such as the sum-product algorithm or the max-product algorithm. Nevertheless, the complexity of the inference process can still be an issue for large-scale or MOT applications.

Probabilistic graphical models in MOT present several advantages, such as their ability to model complex dependencies between variables, incorporate prior knowledge, and perform efficient inference (Pearl, 1988; Kindermann & Snell, 1980). These models have been shown to provide more accurate and robust tracking performance compared to some traditional data association techniques, especially in challenging scenarios with closely spaced objects, occlusions, or varying appearances. However, these methods also have some limitations. One of the main challenges associated with probabilistic graphical models is the computational complexity involved in performing inference, which may hinder their applicability in real-time tracking applications. This is particularly true for large-scale problems or scenarios with a high number of objects and potential associations.

Moreover, selecting an appropriate model structure and defining meaningful probability distributions for the variables can be a difficult task that requires expert knowledge in the specific tracking domain. Additionally, while these models can handle uncertainty and ambiguity to some extent, they may still be susceptible to errors when faced with severe occlusions, drastic appearance changes, or highly cluttered environments. Further research is needed to address these limitations and develop more efficient and robust probabilistic graphical models for MOT, potentially by combining them with other complementary techniques, such as deep learning (LeCun et al., 2015) or optimization-based approaches (Papadimitriou & Steiglitz, 1982).

2.7.3 Deep Learning-based Approaches

Over the past decade, deep learning techniques have profoundly impacted computer vision research, leading to remarkable advancements in multi-object tracking (MOT) (LeCun et al., 2015). These methods have demonstrated superior performance in handling intricate scenarios, including occlusions, varying object appearances, and highly cluttered environments (Bewley et al., 2016).

One popular deep learning-based approach for MOT is the use of convolutional neural networks (CNNs) to extract robust and discriminative features from object detections (Wojke et al., 2017). These features can then be used for data association by comparing the similarities between detected objects and existing tracks. CNN-based feature extraction has been shown to improve tracking performance by providing more accurate and robust appearance information compared to traditional hand-crafted features

(Farkhodov, Lee & Kwon, 2020).

Another approach is to employ recurrent neural networks (RNNs), specifically long short-term memory (LSTM) networks, for modeling the temporal dependencies between consecutive frames. LSTM-based trackers can learn to predict the future position and appearance of tracked objects, which can improve the robustness of data association in challenging scenarios with occlusions and varying object appearances.

Additionally, Graph convolutional networks (GCNs) have been employed to model complex object relationships in the context of MOT. GCNs can capture both spatial and temporal relationships among objects, improving track continuity and reducing the impact of occlusions or detection failures. Integrating GCN-based methods with other data association techniques can provide a more comprehensive understanding of the scene, resulting in more accurate and robust tracking performance.

Attention mechanisms, inspired by human visual perception, have been incorporated into deep learning-based MOT approaches to selectively focus on important regions in the scene. Attention-based methods can adaptively weigh different regions based on their relevance to the tracking task, enhancing tracking accuracy and robustness, particularly in cases of partial occlusions or similar object appearances (Vaswani et al., 2017).

In conclusion, deep learning-based approaches in MOT offer several advantages over traditional data association techniques and probabilistic graphical models. They excel in handling complex scenarios involving occlusions, varying object appearances, and highly cluttered environments due to their ability to automatically learn powerful feature representations from data. Moreover, with the integration of Siamese networks, graph convolutional networks, and attention mechanisms, deep learning-based methods can capture complex object relationships and selectively focus on relevant regions in the scene, resulting in enhanced tracking accuracy and robustness.

However, it is important to note that deep learning approaches typically require large amounts of annotated training data and may struggle to generalize to unseen scenarios.

Additionally, the computational demands of these models can be substantial, which can pose challenges for real-time tracking applications (Ngo, 2019). As research in the field of MOT continues to progress, it is likely that deep learning techniques will be further refined and combined with other complementary methods to address these limitations and achieve even more accurate and efficient tracking performance (An & Yan, 2021).

2.8 Summary

In this chapter, a comprehensive review was presented with various techniques and models utilized for fruit detection and tracking in agricultural applications. Initially, the concept of deep learning and its related subfields, namely, neural networks and convolutional neural networks, were introduced. The applications of deep learning in agriculture were then discussed, including crop disease detection, weed identification, and crop yield prediction. Subsequently, different techniques for fruit detection and counting were examined, including image segmentation, edge-based approaches, region-based methods, and deep learning-based approaches.

Additionally, visual object detection models such as R-CNN, Fast R-CNN, Faster R-CNN, Single Shot MultiBox Detector (SSD), and the YOLO (You Only Look Once) family were studied for fruit detection. Moreover, several object tracking techniques for fruit tracking were reviewed, including optical flow, mean shift, and Kalman filter. Lastly, multi-object tracking models were presented, which included data association techniques such as the Hungarian algorithm, global nearest neighbor algorithm (GNN), joint probabilistic data association (JPDA), and multiple hypothesis tracking (MHT); probabilistic graphical models; and deep learning-based approaches.

In summary, fruit yield prediction, which involves fruit detection and tracking, is faced with several challenges in current research. One major challenge is the complexity of occlusion and appearance variations in fruits, which can make it difficult for traditional methods to accurately detect and track fruits in real-world scenarios. Another challenge is the lack of high-quality training data, particularly for targeted or seasonal fruits, which may limit the performance of deep learning models.

Furthermore, deploying deep learning-based methods on resource-constrained edge devices may be challenging due to their high computational requirements. To address these challenges, further research is needed to develop more efficient and accurate deep learning-based models for fruit detection and tracking in agriculture. One potential direction is to explore new network architectures and training strategies that can better handle occlusion and appearance variations in fruits, such as attention mechanisms or domain adaptation. Additionally, efforts should be made to collect more diverse and representative training datasets, which can lead to the development of robust deep learning models for various fruits and environments.

Finally, developing lightweight and efficient deep learning models that can be deployed on edge devices can facilitate the practical implementation of fruit detection and tracking systems in real-world agricultural environments. Solving these challenges and developing more powerful and efficient deep learning-based models can greatly promote the development of fruit detection and tracking systems, ultimately leading to more accurate and efficient fruit yield prediction methods such as the proposed kiwifruit yield prediction method.

Chapter 3
Methodology

In this chapter, we describe the technical details of our proposed method for kiwifruit yield prediction based on deep learning. In this chapter, we present the process of our research, including the collection and preparation of the kiwifruit dataset, the design and implementation of the deep learning model, and the evaluation of the model's performance. We will also describe the algorithms used in our method, including the object detection algorithm, the tracking algorithm, and the filtering algorithm. Moreover, we will discuss the details of the hardware and software systems used to implement our method. The goal of this chapter is to provide a comprehensive understanding of our approach, enabling readers to reproduce our method and evaluate its effectiveness.

3.1 Introduction

The purpose of this chapter is to propose a computer vision solution for estimating kiwifruit yield by detecting, tracking, and counting fruits in videos. Traditional kiwifruit industry relies heavily on human resources and physical effort to achieve yield prediction, which is subject to various factors and often results in high errors, making it unreliable as a data source for agricultural automation. Therefore, the development of an efficient and high-precision kiwifruit yield prediction model can lay the technical foundation for information agriculture.

In this chapter, we introduce the methodology for developing a deep learning-based kiwifruit yield prediction system. The proposed system consists of two main components, kiwifruit detection and tracking, which also include other necessary steps such as data collection, pre-processing, and counting. Each of these components plays a critical role in the overall system performance. Therefore, we will discuss each of these components in detail, including the methods used and their associated benefits and limitations.

Firstly, we will introduce the kiwifruit dataset that we used in our study and the pre-processing techniques that we adopted to ensure the dataset's quality and accuracy. To ensure dataset diversity and model robustness, we employed various data augmentation techniques such as rotation, flipping, and cropping. Secondly, we will present the YOLOv8 object detection algorithm that we used in our study and its optimization process. YOLOv8 is a high-performing object detection algorithm with excellent accuracy. We enhanced the YOLOv8 model by incorporating an attention mechanism and modifying the IoU method. We will discuss the implementation details of our improved YOLOv8 algorithm, including its advantages and limitations.

Lastly, we will describe the tracking and counting methods that we employed to estimate kiwifruit yield. We used the Kalman filter and the Hungarian algorithm to track kiwifruits detected in consecutive frames and estimate their number. Furthermore, we addressed issues such as ID duplication resulting from prediction results' occlusion. We will provide a detailed account of our tracking and counting algorithms' implementation and their associated advantages and limitations.

In this chapter, we provide a detailed introduction to the methodology for developing a kiwifruit yield prediction system. The techniques and methods introduced are crucial for developing an accurate and effective kiwifruit yield prediction system and can be applied to similar applications. Our research results demonstrate the potential of deep learning in kiwifruit yield prediction and lay the foundation for future research in this field.

3.2 Research Design

3.2.1 Overview of Kiwifruit Yield Prediction Model

In this thesis, we propose a method for kiwifruit yield prediction based on an improved YOLOv8 network and Kalman filter algorithm. The method can automatically detect the position of kiwifruit in a video and track its motion trajectory, which enables the display of the kiwifruit count and prediction of kiwifruit yield. The practical application of this method can significantly enhance the production efficiency and quality of kiwifruit.

The implementation of the proposed model is illustrated in Figure 3.1. Firstly, the video is processed using an improved YOLOv8 network to detect kiwifruits with high accuracy. The YOLOv8 network is enhanced with an attention module, and the intersection over union (IoU) is improved to boost the detection performance of the model (Dong & Duoqian, 2023). For each frame of the video, the algorithm outputs a set of kiwifruit detection boxes' coordinates and corresponding feature vectors.

Next, we employ the Kalman filter algorithm to predict the target's position and status in the next frame of the video. The Kalman filter algorithm is a widely used motion target tracking algorithm that predicts the target's position and status based on the current state and historical information. The algorithm calculates each kiwifruit target's position and status in the next frame using its previous frame's status and prediction model. Then, the predicted results are matched using the Hungarian algorithm to obtain the target's trajectory in the video between the previous and current frames.

Hungarian algorithm completes large-scale target matching in a short time and has lower optimized complexity than the original algorithm. For each frame, the algorithm matches the kiwifruit targets of the current frame with those of the previous frame, updates the targets' positions and status based on the matching results, and outputs the prediction results. If a target fails to match for 30 consecutive frames, it is considered disappeared, and its trajectory is deleted. Finally, a counter is used to display the prediction results, showing the kiwifruit count and achieving the final kiwifruit yield prediction.

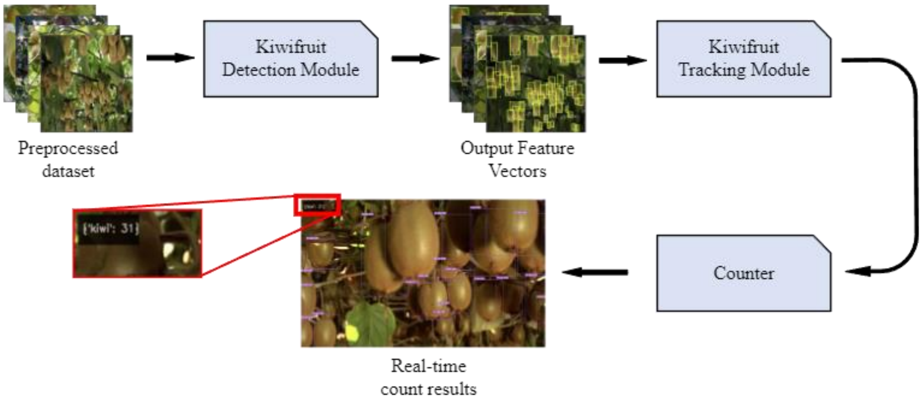


Figure 3.1: Overall architecture of the kiwifruit counting model.

3.2.2 Research Design of the Kiwifruit Detection Module

As the state-of-the-art (SOTA) object detection algorithm, YOLOv8 is the next major updated version of YOLOv5, which was open-sourced by Ultralytics on January 10th, 2023, represents a significant advancement over previous versions of YOLO, as well as other object detection algorithms. This algorithm is designed to improve the accuracy and processing speed of object detection from digital images and videos. YOLOv8 is built on a single-shot multibox detector (SSD) architecture, which allows the model to predict bounding boxes and class probabilities for objects in an image in a single forward pass. This algorithm also incorporates anchor boxes and a "mosaic data augmentation" technique to enhance the model's ability to detect objects of varying scales. A key innovation of YOLOv8 is the incorporation of a “scale-aware training” method, which improves the model's ability to handle visual objects of different sizes in an image. This is achieved by training the model on a diverse set of images, including images with objects of varying scales, using a “mosaic data augmentation” that combines multiple images into a single training image.

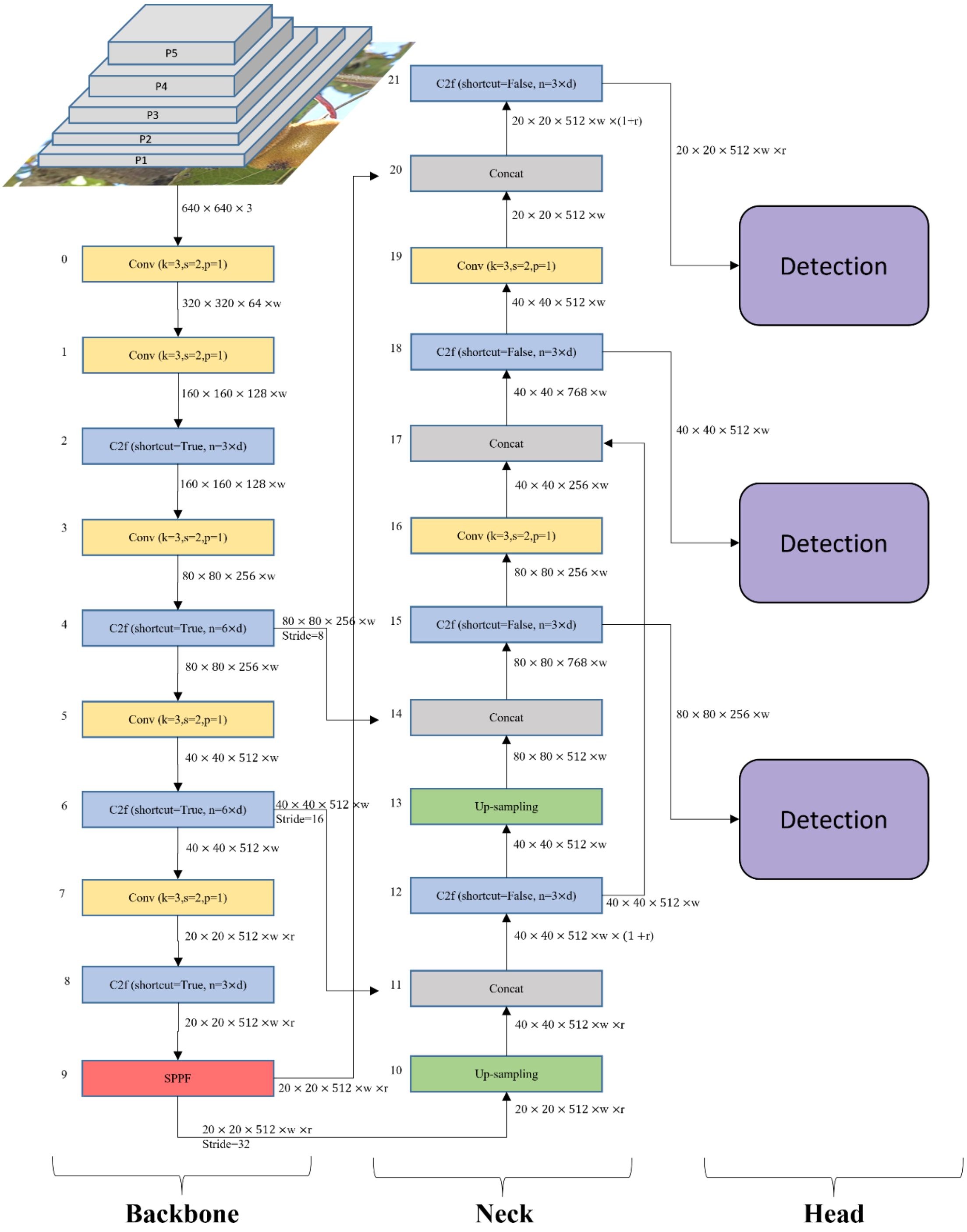
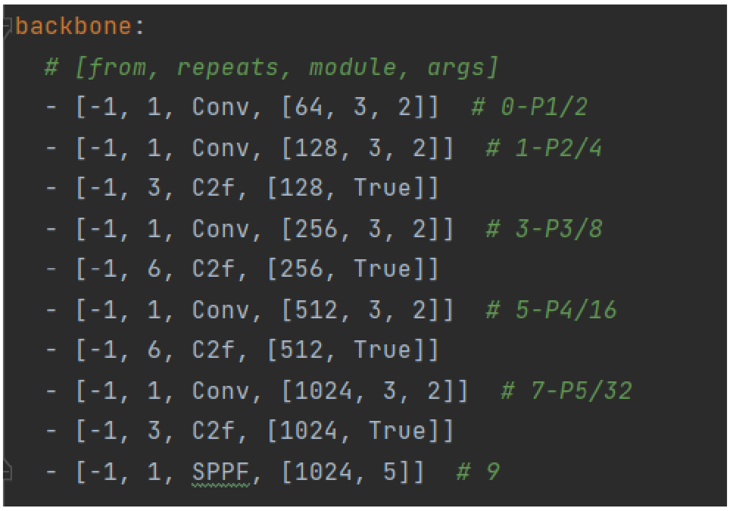


Figure 3.2: The network architecture diagram of YOLOv8

Additionally, YOLOv8 utilizes an efficient implementation of the architecture, allowing it to process images at a higher frame rate and making it well-suited for various applications. The YOLOv8 network architecture is illustrated in Figure 3.2, and takes advantage of a more complex network architecture than its predecessors. This enables the model to detect visual objects with greater accuracy and generalization. Overall, the YOLOv8 algorithm represents a significant advancement in the field of object detection and has the potential to significantly improve the accuracy and efficiency of object detection from digital images and videos.

YOLOv8 is the latest version of YOLO, which consists of three key modules including Backbone, Neck and Head. The elements of the Backbone module are shown in Figure 3.3. Each row of the list contains four elements representing [from, number, module, args]. The first column 'from' with a value of -n represents the input obtained from the previous n layers, where -1 represents the input from the previous layer. 'number' represents the number of layers, 'module' represents the name of the network module, and 'args' represents the initialization parameters of the class. The Backbone and neck modules adopt the concept of Cross-Stage Partial networks (CSP), where the C3 module in YOLOv5 is replaced by the C2f module.



```
backbone:
  # [from, repeats, module, args]
  - [-1, 1, Conv, [64, 3, 2]]  # 0-P1/2
  - [-1, 1, Conv, [128, 3, 2]]  # 1-P2/4
  - [-1, 3, C2f, [128, True]]
  - [-1, 1, Conv, [256, 3, 2]]  # 3-P3/8
  - [-1, 6, C2f, [256, True]]
  - [-1, 1, Conv, [512, 3, 2]]  # 5-P4/16
  - [-1, 6, C2f, [512, True]]
  - [-1, 1, Conv, [1024, 3, 2]]  # 7-P5/32
  - [-1, 3, C2f, [1024, True]]
  - [-1, 1, SPPF, [1024, 5]]  # 9
```

Figure 3.3: The architecture of backbone module in the YOLOv8s model

The traditional C3 module utilized the idea of CSPNet to extract the branching, combined with the concept of residual structure, and designed the C3 Block. The CSP main branch gradient module is the BottleNeck module, which is a residual module. As shown in Figure 3.4, YOLOv8 uses the c2f structure, which is a lightweight version of c3. It replaces the convolution layer with split to layer-wise features, resulting in further lightweighting. YOLOv8 also uses the Spatial Feature Fusion (SPPF) module, which is a feature extraction method for image classification and object detection in deep learning. The main idea is to divide the feature map of the convolutional neural network (CNN) into grids of different sizes, perform pooling operations on the features in each grid, and then concatenate the pooling results of different sizes to form a fixed-size feature vector. This method can handle inputs of different scales and sizes and avoids traditional image scaling operations, thus exhibiting good performance in tasks such as object detection.

The architecture of this module is depicted in Figure 3.5, which is composed of three layers: the feature extraction layer, the pyramid pooling layer, and the fully connected layer. The feature extraction layer is responsible for extracting features, while the pyramid pooling layer generates grids of various sizes, pooling the features in each grid. Finally, the fully connected layer produces the ultimate prediction result.

The Conv structure in the model refers to the combination of convolutional layer (Convolution), batch normalization (Batch Normalization), and SiLU activation function. The structure of Conv is shown in Figure 3.6, where the Conv structure first performs convolutional operation on the input using the convolutional layer, then performs batch normalization on the convolutional result, and finally nonlinearly maps the result through the SiLU activation function. Batch normalization and SiLU activation function improve network stability and accuracy, and also have a certain regularization effect, which helps prevent overfitting.

The Bottleneck structure diagram is shown in Figure 3.7, which uses a 1x1 convolution to reduce dimensionality, followed by a 3x3 convolution operation, and finally a 1x1 convolution to expand dimensionality, forming the basic convolution block. This design aims to reduce network computation and parameter quantity while maintaining good performance. Using a 1x1 convolution layer for dimensionality reduction and expansion reduces the number of parameters without affecting network performance. Additionally, using a 3×3 convolution layer for convolutional operation increases the receptive field of the network, improving its feature extraction ability.

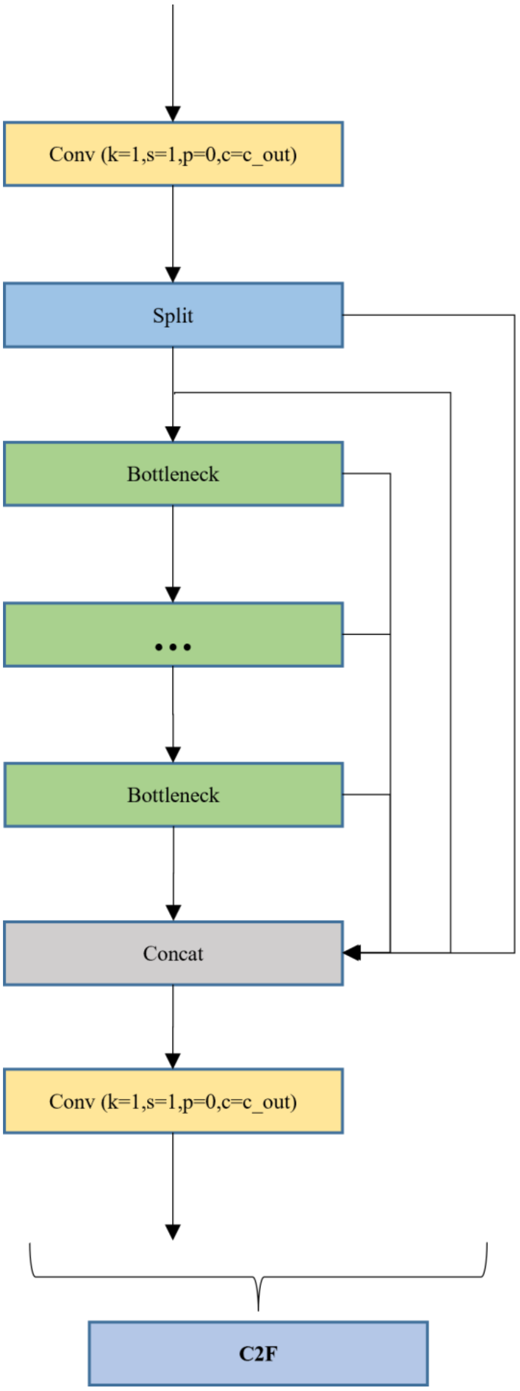


Figure 3.4: The network architecture diagram of C2f module in YOLOv8

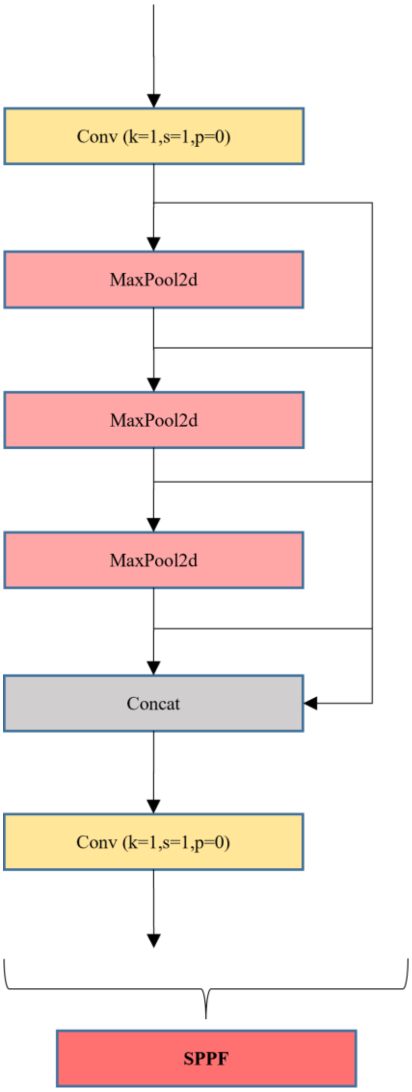


Figure 3.5: The network architecture diagram of SPP-fast module in YOLOv8

The elements of the Head module are shown in Figure 3.8, which converts the traditional coupled head to a decoupled head. As shown in Figure 3.9. YOLOv8's Head module no longer has the objectness branch as in previous versions, only the decoupled classification and regression branches, and the regression branch uses the integral form of the Distribution Focal Loss.

YOLOv8 loss function takes use of a multitask loss that was designed to optimize the performance of the object detection model. YOLOv8 adopts Vari Focal (VFL) loss as the classification loss function, Dostronition Focal Loss (DFL) and CIOU loss as the regression loss function (Zhang et al., 2022; Zheng et al., 2020).

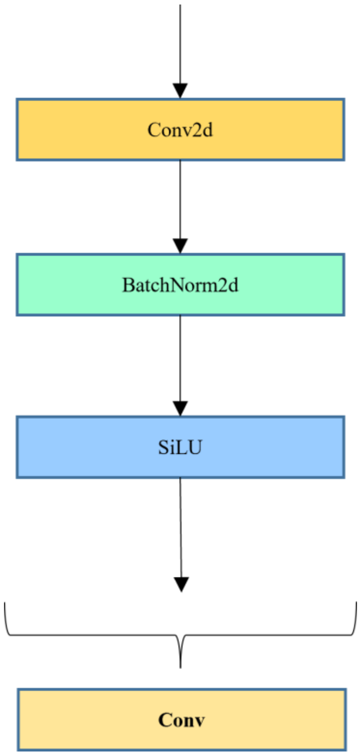


Figure 3.6: The network architecture diagram of convolution module in YOLOv8

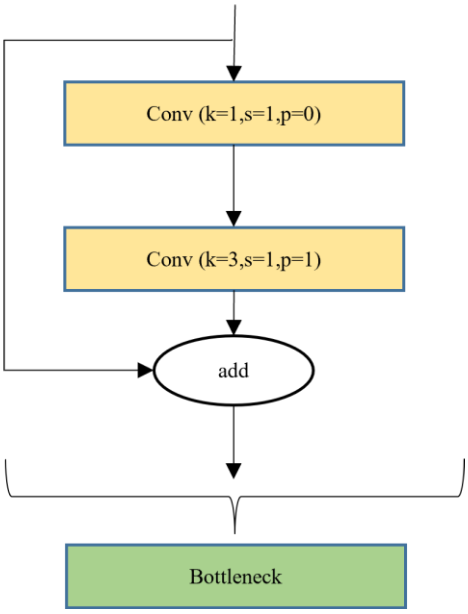
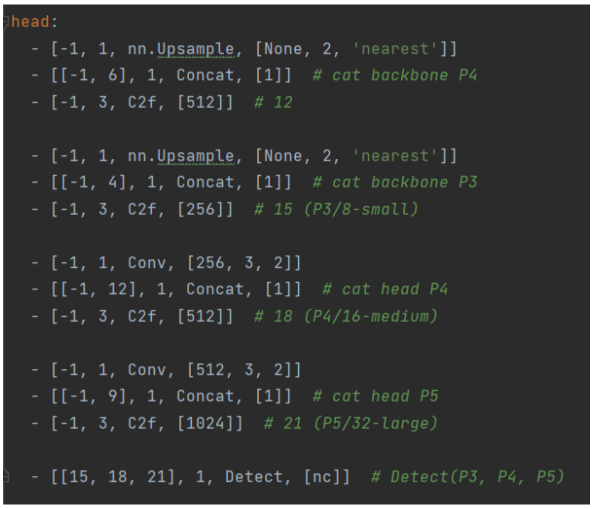


Figure 3.7: The network architecture diagram of bottleneck module in YOLOv8

Classification Loss Function. Enhancing classifier performance is a crucial aspect of detector optimization. Focal loss is a modified version of the conventional cross-entropy loss function that addresses class imbalance between negative and positive samples or between easy and hard samples. Quality Focal Loss (QFL) extends Focal Loss by jointly considering classification scores and localization quality to address inconsistent use of quality estimation and classification between training and inference. As expressed in Eq.

(3.1), where p represents the predicted value, “l” represents the label, and α is the hyperparameter. VariFocal Loss (VFL), which is derived from Focal Loss, addresses positive and negative samples asymmetrically by considering the varying importance levels of each sample type. This approach balances the learning signals from both types of samples. Consequently, the YOLOv8 model utilizes VFL as the classification loss function.

$$VFL(p,l) = \begin{cases} -l(l\,log(p) + (1-l)\,log(1-p)) & l > 0 \\ -\alpha p^{\gamma}\,log(1-p) & l = 0 \end{cases} \tag{3.1}$$



```
head:
  - [-1, 1, nn.Upsample, [None, 2, 'nearest']]
  - [[-1, 6], 1, Concat, [1]]  # cat backbone P4
  - [-1, 3, C2f, [512]]  # 12

  - [-1, 1, nn.Upsample, [None, 2, 'nearest']]
  - [[-1, 4], 1, Concat, [1]]  # cat backbone P3
  - [-1, 3, C2f, [256]]  # 15 (P3/8-small)

  - [-1, 1, Conv, [256, 3, 2]]
  - [[-1, 12], 1, Concat, [1]]  # cat head P4
  - [-1, 3, C2f, [512]]  # 18 (P4/16-medium)

  - [-1, 1, Conv, [512, 3, 2]]
  - [[-1, 9], 1, Concat, [1]]  # cat head P5
  - [-1, 3, C2f, [1024]]  # 21 (P5/32-large)

  - [[15, 18, 21], 1, Detect, [nc]]  # Detect(P3, P4, P5)
```

Figure 3.8: The architecture of the neck and the head module in the YOLOv8s model

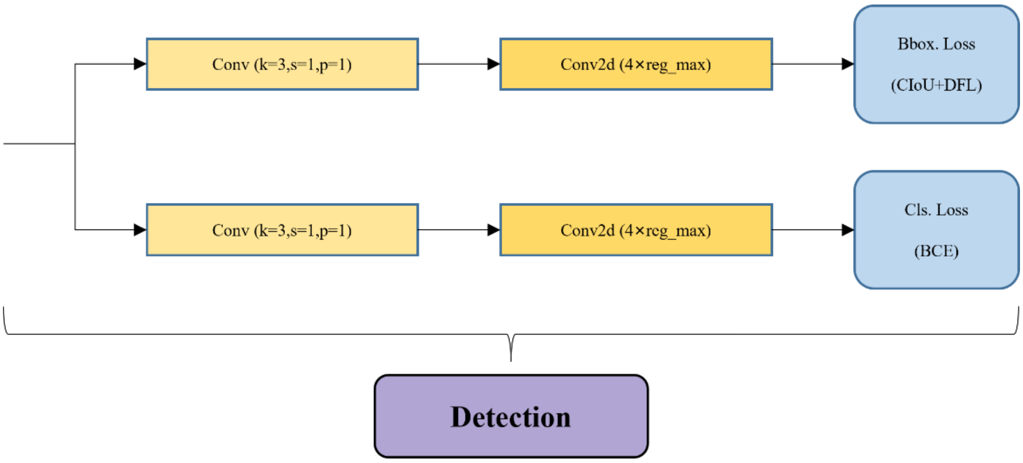


Figure 3.9: The network architecture diagram of detection module in the head section of YOLOv8

Regression Loss Function. DFL is a method that discretizes the underlying continuous distribution of box positions to consider ambiguity and uncertainty in data without introducing additional prior information, leading to improved box localization accuracy, especially in cases where the boundaries of ground truth boxes are unclear. In the YOLOv8 model, DFL models the box location as a general distribution, allowing the network to quickly focus on the distribution of locations close to the target and increase the likelihood of object detection. The CIoU loss function involves three geometric parameters, including overlapping area, center point distance, and aspect ratio, with α and v representing the aspect ratio and a positive trade-off parameter, respectively. The equation is shown in Eq. (3.2) and Eq. (3.3), with v measuring the consistency of aspect ratio.

$$\mathcal{L}_{CIoU} = 1 - IoU + \frac{\rho^2(b, b^{gt})}{c^2} + \alpha v \tag{3.2}$$

$$\alpha = \frac{v}{(1 - IoU) + v'} \tag{3.3}$$

In the training of a convolutional neural network (CNN), the selection and design of the loss function play a critical role in measuring the discrepancy between the predicted output and the true label of a sample. By minimizing the value of the loss function, the CNN model aims to maximize the similarity or closeness between the output distribution of the model and the sample label distribution. The loss function is capable of adjusting the weight parameters in the model, which guides the learning process in the CNN. Therefore, selecting an appropriate loss function is crucial in revealing the characteristics of the current model, and can significantly impact the model's overall performance.

In the previously introduced CIoU loss function, the penalty term used includes the distance and relative proportion of the bounding boxes. Zhang et al. proposed the Focal-EIoU loss function to solve the problem of severe oscillation of loss values caused by low-quality samples. This loss function differs from the CIoU loss function in YOLOv8, as the Focal-EIoU loss function directly adopts the side length as a penalty term.

EIoU Loss. The EIoU loss function is an improvement of the CIoU loss function, which addresses the issue of the penalty term becoming ineffective in certain situations where the predicted bounding box width and height satisfy specific conditions in the CIoU loss function. The definition of the EIoU loss function is given by Eq. (3.4) , where c_w and c_h are defined as the width and height of two rectangular anchor boxes. $\boldsymbol{\mathcal{L}_{IoU}}$, $\boldsymbol{\mathcal{L}_{dis}}$, and $\boldsymbol{\mathcal{L}_{asp}}$ are the IoU loss, distance loss, and aspect ratio loss, respectively.

FocalL1 Loss. The FocalL1 loss is a modified version of the focal loss that addresses the imbalanced problem in regression problems. In object detection, most of the predicted boxes based on anchor boxes have low Intersection over Union (IoU) values with the ground truth, leading to high fluctuations in loss values when training on such low-quality samples. The purpose of FocalL1 is to resolve the imbalance between high- and low-quality samples. By assigning a smaller gradient to low-quality samples, FocalL1 loss suppresses the impact of these samples. As shown in Eq. (3.5), FocalL1 loss calculates the regression loss by summing up the deviations of x, y, w, and h.

Focal-EIoU Loss. The Focal-EIoU loss function is a combination of the EIoU loss function and the FocalL1 loss function mentioned above. As shown in Eq. (3.6), the hyperparameter y is employed to control the curvature of the curve.

$$\mathcal{L}_{EIoU} = \mathcal{L}_{IoU} + \mathcal{L}_{dis} + \mathcal{L}_{asp} = 1 - IoU + \frac{\rho^2(\boldsymbol{b}, \boldsymbol{b^{gt}})}{c^2} + \frac{\rho^2(w, w^{gt})}{c_w^2} + \frac{\rho^2(h, h^{gt})}{c_h^2} \quad (3.4)$$

$$\mathcal{L}_{FocalL1} = \sum_{i \in \{x,y,w,h\}} \mathcal{L}_{Focal}\left(\left|\boldsymbol{B}_i - \boldsymbol{B}_i^{gt}\right|\right) \quad (3.5)$$

$$\mathcal{L}_{Focal-EIoU} = IoU^{\gamma}\mathcal{L}_{EIoU} \quad (3.6)$$

Furthermore, attention mechanisms have become increasingly popular in the area of computer vision, enabling models to concentrate on specific regions of an image during prediction. This is particularly important when dealing with images that contain multiple objects or areas of interest that are relevant to the task at hand. In computer vision, attention mechanisms are similar to those found in natural language processing and are commonly implemented as a neural network layer that receives a set of feature maps as input.

The incorporation of attention mechanisms in computer vision has become prevalent in recent years. Attention mechanisms in computer vision are a valuable tool that allows models to concentrate on specific regions of an image during predictions, particularly in situations where the image encompasses numerous objects or regions of interest that are crucial for the task at hand. These mechanisms are implemented as a neural network layer that takes in a set of input feature maps and generates a new set of output feature maps. The output feature maps are a combination of the input feature maps, where the weights are computed by examining the similarity between the query and key vectors. The highest weights are assigned to the most similar pairs, enabling the model to selectively concentrate on the most significant image regions. Attention mechanisms have been proven to enhance the performance of computer vision models across various tasks, including visual object detection, image segmentation, and image captioning. Additionally, attention mechanisms have been found to facilitate the interpretation of neural network models' decisions, as the attention weights offer insight into the image regions utilized by the model for its predictions.

An attention mechanism called Convolution Block Attention Module (CBAM) has been proposed for computer vision tasks to selectively focus on important regions of an image (Woo et al., 2018). CBAM employs both channel-wise and spatial attention mechanisms, as shown in Figure 3.10. The channel attention mechanism in CBAM is used to emphasize the most important channels in the feature maps, as depicted in Figure 3.11. This is achieved by generating weights for each channel using a fully connected layer, which takes the average and maximum values of each channel. These weights are then used to weight the channels in the feature maps (Shan & Yan, 2021). Similarly, the spatial attention mechanism in CBAM is employed to emphasize important regions of the image, as illustrated in Figure 3.12. A 2D attention map is generated by passing the feature maps through a convolutional layer, which is then used to weight the feature maps. By integrating both channel-wise and spatial attention mechanisms, CBAM enables the model to selectively focus on the most important channels and regions of the image, leading to improved performance on a range of computer vision tasks such as visual object detection, semantic segmentation, and image classification.

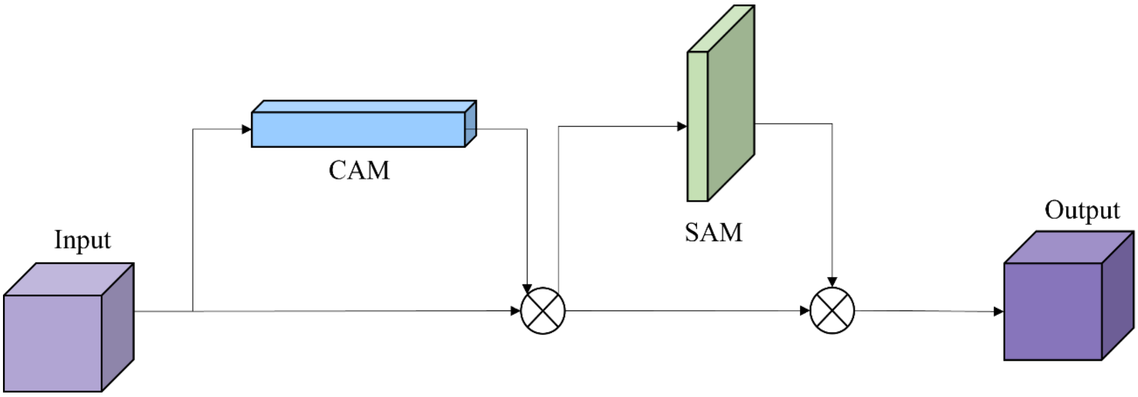


Figure 3.10: The flowchart of convolutional block attention module

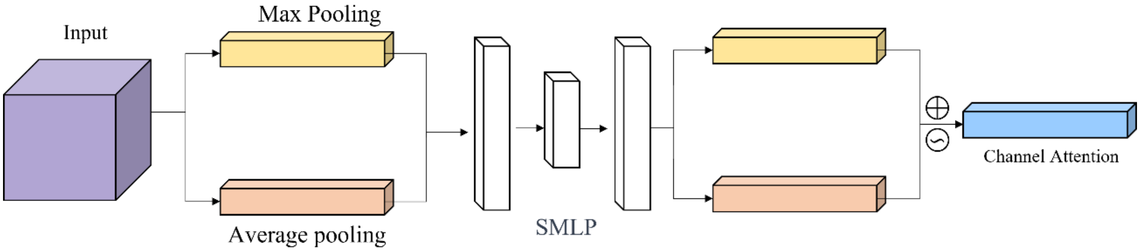


Figure 3.11: The flowchart of channel attention module

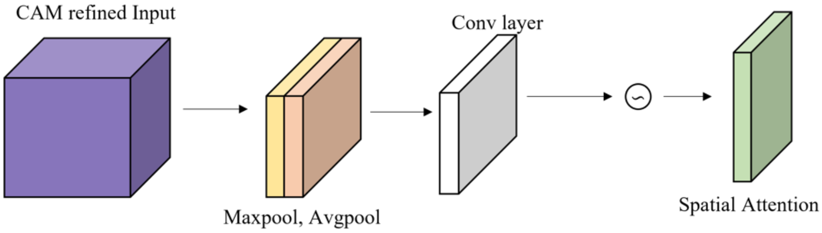


Figure 3.12: The flowchart of spatial attention module

In this thesis, we conducted our experiments based on detecting kiwifruit using YOLOv8 as a baseline, with the aim of predicting yield by inputting the output results into a kiwifruit tracking and counting model. We proposed a method for kiwifruit detection that involves inserting a CBAM module into the main structure of YOLOv8 and replacing the CIoU loss function with the Focal-EIoU loss function, which differs from the original YOLOv8 model. The Convolution Block Attention Module applies weights to channel and spatial features in the feature map, allowing the model to focus on the target object while suppressing attention to non-targets. The CBAM module comprises two parts: the channel attention module and the spatial attention module. As depicted in

Figure 3.13, the CBAM module is inserted prior to the convolution layer in the main network, and the model takes a 640x640 size image as input into the main network and outputs the prediction result to achieve object detection.

In YOLOv8 model, the CIoU loss function considers the center distance, overlap area, and aspect ratio of the bounding box, which are discussed in detail in the relevant literature. However, the aspect ratio difference measured by the function does not reflect the true difference between the anchor box's length and width and its confidence, which sometimes hinders effective similarity optimization. Additionally, low-quality samples can cause sharp fluctuations in the loss value, which can be problematic. To address these issues, we replaced the CIoU loss function with the Focal-EIoU loss function in this experiment. The Focal-EIoU loss function combines the EIoU loss function and the FocalL1 loss function, as illustrated in Figure 3.14, and has been shown to perform better than the CIoU loss function.

The purpose of this module is to improve the performance of kiwifruit detection by enhancing the accuracy and efficiency of the YOLOv8 model. The proposed method can be applied to the kiwifruit tracking and counting model to achieve accurate yield prediction. In addition, we also expanded our dataset by manually collecting and preprocessing kiwifruit images to better train the model.

3.2.3 Research Design of the Kiwifruit Counting Module

Kalman filter is a mathematical algorithm extensively employed in signal processing and control systems to estimate unknown variables based on a sequence of measurements. It operates by anticipating the state of a system at a specific time and then refining the prediction using fresh measurements to enhance the state estimate. The process commences with an initial state estimate and a covariance matrix that reflects the error in the estimate. The algorithm then proceeds through two primary steps: prediction and update.

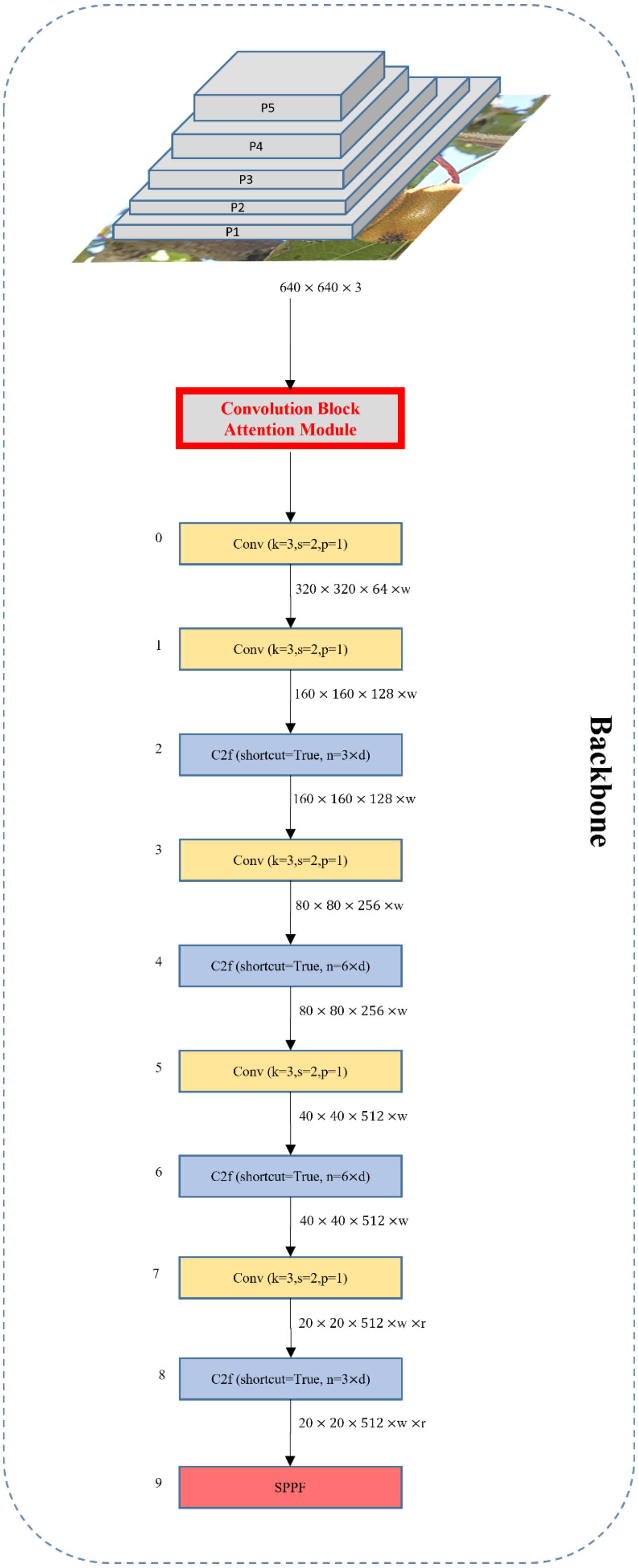


Figure 3.13: The network architecture diagram of backbone network with CBAM modules inserted.

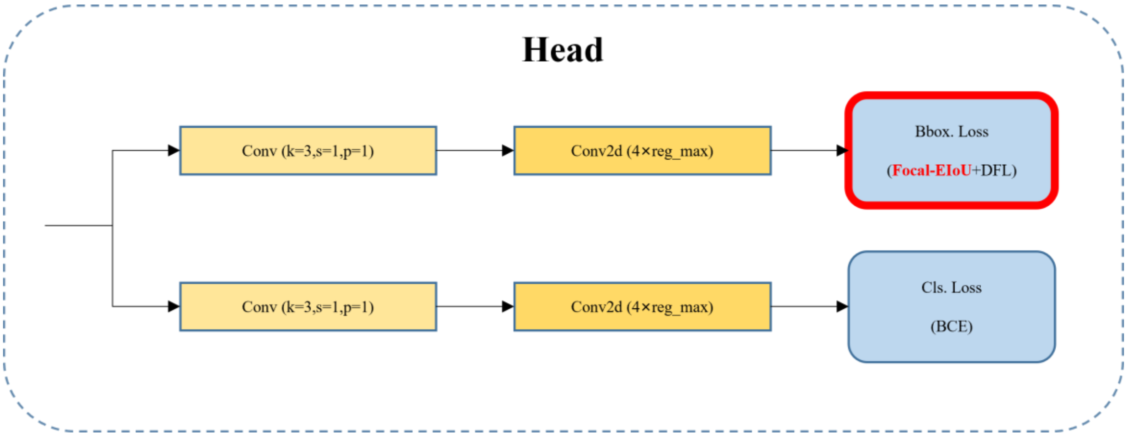


Figure 3.14: The network architecture diagram of head network with improved IoU loss function

During the prediction phase, the filter predicts the system's state based on the previous state estimate and the dynamic model of the system. In the update phase, the filter merges the anticipated state with the new measurement to obtain a more precise state estimate. This is achieved using the Kalman gain, which adjusts the weight given to the predicted state and the measurement based on their respective error covariances. The process then repeats, with the updated state estimate and error covariance used as the new initial conditions for the next iteration.

As shown in Figure 3.15, the input value of the algorithm is a measurable quantity, which can be any quantity that can be measured, and the accuracy of the measurement is known. With this measurement value, we can estimate the true output of the system based on the measurement value and provide an estimate of the accuracy of the new estimated value within a certain range. This is the work done by the Kalman filter, but this work is constantly ongoing, measuring the system continuously and estimating continuously, so that after a period, a very accurate output value of the system can be estimated. It should be noted that the measurement value may be very inaccurate, and the estimate value may also be very inaccurate, which is in line with many work situations in engineering. However, based solely on these two inaccurate values, a relatively accurate system output value can be estimated, which is the role of the Kalman filter.

Measured value of the system state variable
Kalman Filter
Optimal estimate of the system state variable

Figure 3.15: The working principle of the Kalman filter

Eq. (3.7) is the equation of state for the Kalman filter. The state equation predicts the current state based on the previous state and control variables. The Gaussian-distributed noise W_k, which represents the prediction error, is added to the state equation, and it corresponds to the noise in each component of the state vector X_k. The noise has an expected value of 0 and a covariance matrix of Q and is characterized as Gaussian white noise with a distribution of W_k-N (0, Q). Q is referred to as the process noise or process covariance matrix, which represents the uncertainty in the system model and the unmodeled dynamics.

Eq. (3.8) is the observation equation for the Kalman filter. Here, Z_k denotes the observation value of the system, X_k represents the estimated state value, H_k is the observation matrix, and v_k is the observation noise. The observation matrix H_k maps the state vector to the observation vector, The observation equation describes the relationship between the estimated state and the observation values. It provides a means of combining the estimated state and the observed data to obtain an optimal estimate of the state. In the context of the Kalman filter, the observation equation is used to update the state estimate, which improves the accuracy and precision of the state estimation process. Table 3.1 describes the parameters in the state and prediction equations of the Kalman filter.

$$X_k = A_k X_{k-1} + B_k u_k + W_k \tag{3.7}$$

$$Z_k = H_k X_k + v_k \tag{3.8}$$

In the prediction step, we make use of a priori information to predict the state of the system and consists of two parts, state prediction and covariance prediction. The goal of state prediction is to use the prior information of the system to make predictions about the state at the next moment. Assuming that the current moment is moment k, the system state is $\hat{X}_{\bar{k}}$, the input is u_k and the state transfer matrix is A_k, the state prediction equation is shown in Eq. (3.9), where $\hat{X}_{k-1}$ is the predicted value of the state and B_k is the input matrix. If the system has no inputs, then B_k is 0. The goal of covariance prediction is to use the system's prior information to predict the uncertainty of the next moment's state prediction. Assuming that the current moment is moment k, the state prediction is P_{k-1}, the state transfer matrix is A_k^T, the process noise is Q_k, and the covariance matrix is $P_{\bar{k}}$, the covariance prediction equation is shown in Eq. (3.10), where Q_k is the covariance matrix of the process noise.

Table 3.1: The descriptions of the parameters in Eq. (3.7) and Eq. (3.8)

Parameters	Descriptions
X_k	State Vector at Time k
A_k	State Transition Matrix
B_k	Control Input Matrix
u_k	Control Input Vector
W_k	Covariance Matrix of Process Noise
Z_k	Observation Vector at Time k
H_k	Observation Matrix
v_k	Covariance Matrix of Observation Noise

$$\hat{X}_{\bar{k}} = A_k \hat{X}_{k-1} + B_k u_k \tag{3.9}$$

$$P_{\bar{k}} = A_k P_{k-1} A_k^T + Q_k \tag{3.10}$$

During the update stage, the Kalman filter corrects the state of the associated track based on the detection received at time T, resulting in a more accurate estimation. The Kalman filter computes the Kalman gain matrix, which describes the relationship between sensor measurements and predicted values. The higher the Kalman gain matrix, the greater the influence of the sensor measurement on the state estimate. Then, the Kalman filter performs a weighted average of the sensor measurement and the state prediction to generate the final state estimate.

$$y_k = z_k - H_k \hat{X}_{\bar{k}} \tag{3.11}$$

$$S_k = H_k P_{\bar{k}} H_k^T + R \tag{3.12}$$

$$K_k = \frac{P_{\bar{k}} H_k^T}{S_k} \tag{3.13}$$

$$\hat{X}_k = \hat{X}_{\bar{k}} + K_k (z_k - H_k \hat{X}_{\bar{k}}) \tag{3.14}$$

$$P_k = (I - K_k H_k)P_{\bar{k}} \tag{3.15}$$

In Eq. (3.11), the mean vector of the detection excluding the velocity components is denoted as z_k, which is equal to [cx, cy, r, h]. The measurement matrix H_k maps the estimated mean vector $\hat{X}_{\bar{k}}$ of the track to the detection space, and the equation calculates the mean error between the detection and track. In Eq. (3.12), the noise matrix of the detector is represented as R, which is a diagonal matrix of size 4x4, and its values correspond to the noise in the x and y coordinates of the center point, as well as the width and height of the detection. It is commonly initialized with arbitrary values, and the noise in the width and height is usually set higher than that in the center point. The equation first maps the covariance matrix $P_{\bar{k}}$ to the detection space and then adds the noise matrix R. Eq. (3.13) computes the Kalman gain K_k, which is employed to weigh the importance of the estimation error. Eq. (3.14) and (3.15) provide the updated mean vector $\hat{X}_k$ and covariance matrix P_k.

The objective of this thesis is to enhance the YOLOv8 model for object detection and state prediction of targets through the utilization of the Kalman filter. To evaluate the similarity between the detection results of the improved YOLOv8 model and the prediction results of the Kalman filter model, we use the Euclidean distance as shown in Eq. (3.16), where x_d, and y_d denote the horizontal and vertical coordinates of the anchor boxes for object detection, and x_t and y_t represent the corresponding coordinates of the targets being tracked.

$$d = \sqrt{(x_d - x_t)^2 + (y_d - y_t)^2} \tag{3.16}$$

However, object detection and prediction are usually based on a certain time interval, and due to errors, occlusion, and other reasons, a target detected at different times may be considered as different targets. Therefore, we combine Kalman filtering and the Hungarian algorithm, where Kalman filtering is used to predict the target's state and position, and the Hungarian algorithm is used to match newly detected targets with existing ones.

The core idea of the Hungarian algorithm is to establish a bipartite graph by using the distance between predicted targets and detected targets as weights, and to use the maximum matching algorithm in graph theory to find the optimal matching scheme. The algorithm considers predicted targets and detected targets as the left and right vertex sets of the bipartite graph, respectively, and establishes a set of edges between them, with each edge weight being the distance between the predicted target and the detected target, which can be calculated based on the target's position, speed, and other states. Then, the Hungarian algorithm takes this bipartite graph as input and uses the augmenting path algorithm to find the maximum weight matching. Maximum weight matching refers to a set of edges selected in the graph, whose weight sum is maximum while ensuring that each node is matched with only one adjacent node.

Finally, the Hungarian algorithm assigns the maximum weight matching as the target ID allocation scheme. Specifically, for each predicted target, the algorithm finds the corresponding detected target according to the matching scheme, and then combines their states together to update the predicted target's state. At the same time, for the detected targets that were not matched, the algorithm assumes that they are new targets and assigns them a new ID. This combination method can improve the accuracy and stability of multi-target tracking.

Nevertheless, during the process of multi-object tracking using improved YOLOv8 combined with Kalman filtering and the Hungarian algorithm, target matching failure may occur due to detection errors or prolonged occlusion. To address this issue, we perform Intersection over Union (IoU) matching between the detected targets and tracked targets and determine the maximum threshold to remove low correlation matches between detection and tracking anchor boxes.

To avoid the immediate deletion of temporarily occluded or lost targets that were not correctly matched, failed targets are temporarily retained and prediction continues until a target fails to match for 30 consecutive frames, at which point the model considers it a lost target and deletes its tracking trajectory.

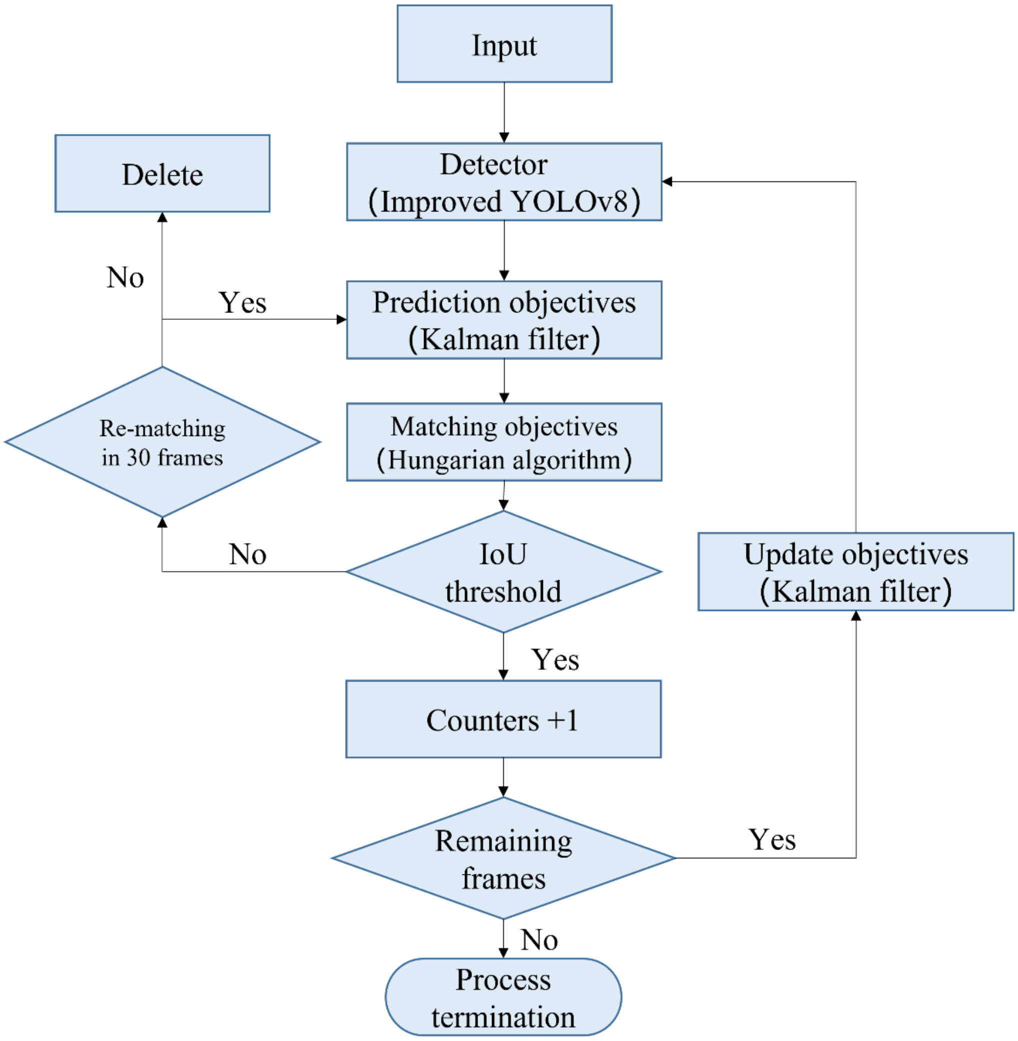


Figure 3.16: Detailed workflow of kiwifruit yield forecast model

Finally, IDs are assigned to the detected targets in the order they appear in the video frames, and a counter is used to display the total number of detected targets in the upper left corner of the video, ultimately achieving the counting of the number of targets. As previously described, the flowchart in Figure 3.16 provides a detailed description of the proposed workflow of our kiwifruit yield prediction model.

3.3 Evaluation Methods

Assessing the performance of a computer vision system is a vital aspect that determines the accuracy and dependability of predictions. The advancement of deep learning has led to remarkable progress in a range of computer vision tasks, including segmentation, recognition, and object detection. The current thesis concentrates on evaluating the performance of the proposed models in kiwifruit detection, tracking and counting.

After completing the model training, it is essential to use appropriate evaluation metrics to assess its performance. In this study, we divided the model into two main parts: kiwifruit detection and kiwifruit counting. When evaluating the performance of the kiwifruit object detection module, a series of metrics is needed to measure the model's accuracy, efficiency, and stability. These metrics can provide insights into the model's performance in different aspects and help compare performance among different models. To assess the performance of our object detection model, we will employ several standard evaluation metrics, including precision and recall, as well as mAP at different thresholds, such as mAP@0.5 and mAP@0.5:0.95. By utilizing these metrics, we will be able to evaluate the model's performance across different datasets and scenarios, as well as pinpoint specific areas for potential enhancements and optimizations.

In the kiwifruit counting module, our primary evaluation metrics will be mean absolute error (MAE), mean squared error (MSE), and root mean squared error (RMSE), which will provide insights into the model's ability to accurately predict fruit yield.

3.3.1 Evaluation of the Kiwifruit Detection Module

An accurate object detection model serves as the foundation for implementing the kiwifruit counting model. In order to evaluate the improved performance of the kiwifruit detection model proposed in this thesis, we were use of industry-standard evaluation metrics commonly used in the field.

The Confusion Matrix is a commonly used tool for evaluating the performance of classification models, particularly in binary classification problems. As shown in Figure 3.17, it presents the relationship between the model's predicted results and the true results in the form of a table. Typically, the confusion matrix includes four important metrics: True Positive (TP), False Positive (FP), True Negative (TN), and False Negative (FN), which respectively represent the classification of positive and negative instances by the classifier.

As shown in Table 3.2, four fundamental metrics are commonly used to evaluate the performance of object detection algorithms during the evaluation process. True Positive (TP) refers to the number of instances of existing objects correctly detected by the model. In other words, if a kiwifruit instance is correctly detected, then this detection is a true positive. False Positive (FP) refers to the number of instances of non-existing objects incorrectly detected by the model. In other words, if a non-existent kiwifruit is mistakenly detected, then this detection is a false positive. True Negative (TN) refers to the number of instances of non-existing objects correctly identified as such by the model. In other words, if a non-existent kiwifruit is correctly ignored, then this ignore is a true negative. False Negative (FN) refers to the number of instances of existing objects incorrectly identified as non-existing by the model. In other words, if an existing kiwifruit is mistakenly ignored, then this ignore is a false negative.

These four metrics play a critical role in the evaluation of object detection and can help us assess performance indicators such as accuracy and recall rate. Additionally, they can also help us further optimize the performance of the algorithm to improve the accuracy and reliability of object detection.

Table 3.2: Explanation of confusion matrix metrics

Metric	**Definition**
True Positive (TP)	The number of actual positive examples that the model correctly identified as positive
False Positive (FP)	The number of negative examples that the model incorrectly identified as positive
True Negative (TN)	The number of actual negative examples that the model correctly identified as negative
False Negative (FN)	The number of positive examples that the model incorrectly identified as negative

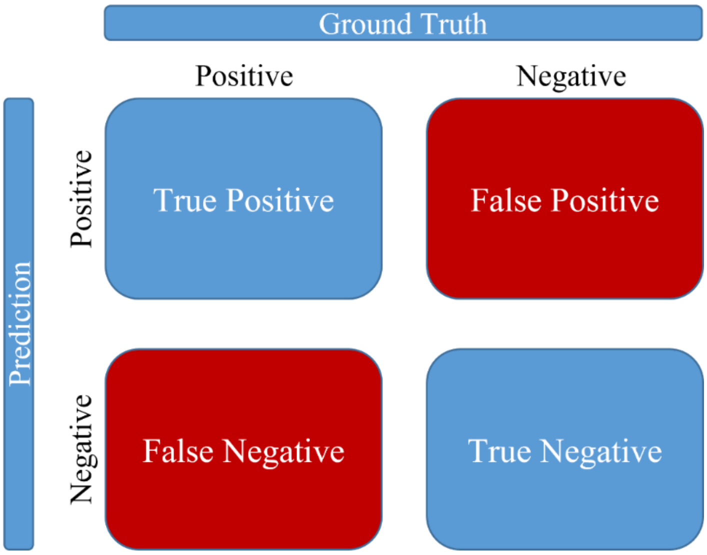


Figure 3.17: An illustration of confusion matrix comprising predicted and ground truth.

Precision and Recall are commonly used metrics to evaluate the performance of object recognition models, particularly in binary classification tasks. Precision calculates the proportion of true positive predictions among all positive predictions generated by the classifier. Eq. (3.17) provides the mathematical definition of Precision, where TP and FP represent the numbers of true positive and false positive predictions, respectively. A high Precision score indicates that the trained model is effective at identifying positive instances and has few false positive errors. Recall, also known as sensitivity or true positive rate, measures the proportion of true positive predictions among all actual positive instances in the dataset. Eq. (3.18) provides the mathematical definition of Recall, where TP and FN represent the numbers of true positive and false negative predictions, respectively. A high Recall score indicates that the trained model is good at detecting positive instances and has few false negative errors. These metrics are essential in evaluating the performance of object recognition models and can help improve the accuracy and reliability of the algorithm.

$$Precision = \frac{TP}{TP + FP} = \frac{TP}{Total\ Positive\ Results} \tag{3.17}$$

$$Recall = \frac{TP}{TP + FN} = \frac{TP}{Total\ Ground\ Truths} \tag{3.18}$$

IoU is a widely-used evaluation metric in the field of object detection that measures the performance of detectors by quantifying the degree of overlap between the predicted and ground-truth instances. Its application can be traced back to early computer vision tasks like object tracking and image segmentation.

With the advent of deep learning techniques, IoU has become increasingly popular in assessing the effectiveness of models in tasks such as object detection and semantic segmentation. The IoU is determined as the ratio of the intersection area between the predicted bounding box and the ground-truth bounding box to the union area of the two boxes, as shown in Eq. (3.19). The Overlap Area represents the intersection area between the predicted bounding box and the ground-truth bounding box, while the Union Area denotes the union area of the two boxes. Figure 3.18 depicts the four potential scenarios of IoU, where the blue box represents the ground-truth box and the red box represents the predicted box. The IoU value gets closer to 1 as the overlap between the two boxes increases, indicating better prediction results from the model.

$$IoU = \frac{Overlap\ Area}{Union\ Area} \tag{3.19}$$

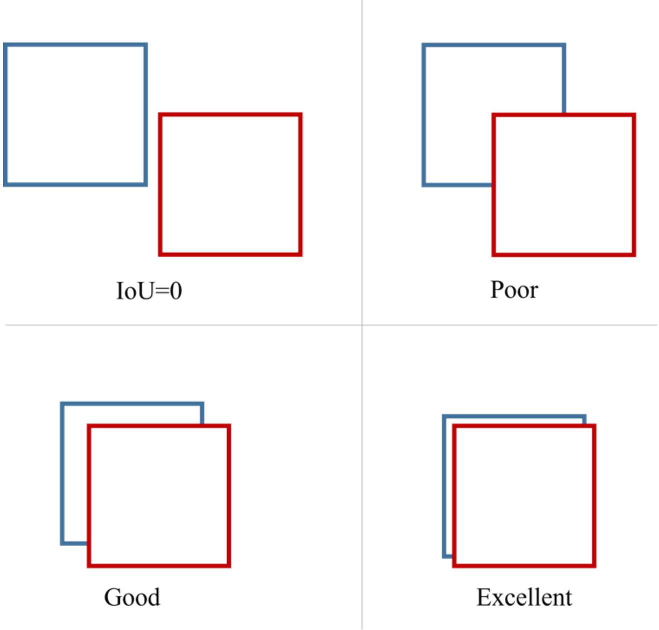


Figure 3.18: An illustration of intersection over unions for various bounding boxes

The Precision-Recall (PR) curve is another metric used to evaluate the performance of object detection models, especially in multi-object detection scenarios. The PR curve is calculated based on the model's prediction results at different thresholds, showing the model's performance at different levels of precision and recall. In object detection tasks, the detection model predicts the confidence score and class probability of each bounding box. For a specific class, precision and recall can be calculated at different confidence thresholds. Assuming that the model outputs n bounding boxes, they can be sorted based on their confidence scores and divided into positive class (containing the object) and negative class (not containing the object). The mathematical equation for confidence score is shown in Eq. (3.20), where $Pr(Object)$ represents the object prior probability, which is 1 if the box contains an object and 0 otherwise. IoU_{pred}^{truth} represents the intersection-over-union between the predicted box and the ground truth box, with values ranging from 0 to 1.

Therefore, the confidence score ranges from 0 to 1 and reflects two pieces of information: the confidence that the object is contained in the predicted box and the accuracy of the predicted box. In the context of object detection tasks, we adjust the IoU threshold value based on the output results during the model training, testing, and validation phases to obtain better predicted bounding boxes. As shown in Figure 3.19, we set the IoU threshold value to 0.6, and if the predicted IoU value of the model in Figure 3.19 (a) is approximately 0.7, which exceeds the threshold, the prediction result is considered a true positive (TP). On the other hand, if the predicted IoU value in Figure 3.19 (b) is approximately 0.3, which is lower than the threshold, we classify it as a false positive (FP). That also means that for a prediction, we may get different binary TRUE or FALSE positives, by changing the IoU threshold.

$$Confidence = Pr(Object) \times IoU_{pred}^{truth} \tag{3.20}$$

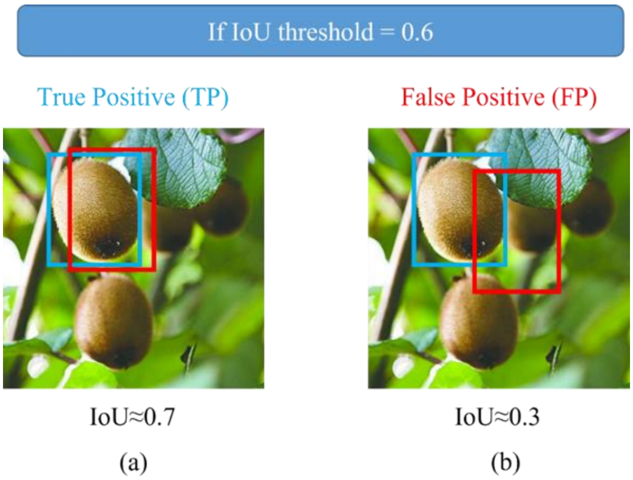


Figure 3.19: An IoU threshold of 0.6, predicted bounding boxes with different IoU values are shown in the image, where the blue bounding boxes represent ground truth, and the red bounding boxes represent predicted bounding boxes from the model.

To measure the accuracy of object detection algorithms, Mean Average Precision (mAP) is used, which calculates the average precision for each object class and takes the mean of all the classes. Higher mAP values indicate better performance. To compute mAP, it is necessary to first calculate the Precision-Recall curves for each category. This involves performing a threshold scan on the model's prediction results. By applying different confidence thresholds to the model's predictions, Precision and Recall values can be calculated for each threshold. The Precision-Recall curve for each category can be plotted, and the area under the curve (AUC) can be calculated. Finally, the average AUC value for each category can be obtained to yield the overall average precision (AP) of the dataset. The mathematical expressions for AP and mAP are defined by Eq. (3.21) and Eq. (3.22), respectively. As the dataset used in this thesis contains only one object class, the AP value of the object detection model is equal to its mAP value.

$$AP_i = \int_0^1 p(r)dr \tag{3.21}$$

$$mAP = \frac{1}{N}\sum_{i=1}^{N} AP_i \tag{3.22}$$

In addition, to evaluate the performance of the proposed object detection module, it is necessary to specify the IoU threshold when calculating the mean average precision (mAP). Two methods were used in this study to reflect the evaluation results of the model. The first method involved using a single IoU threshold of 0.5 (mAP@0.5), where mAP@0.5 represents the average precision of the model when the overlap threshold between predicted and ground-truth bounding boxes is set to 0.5. The second method used a range of values from 0.5 to 0.95, with an increment of 0.05 (mAP@0.5:0.95). The mAP@0.5:0.95 represents the average precision of the model when the overlap threshold between predicted and ground-truth bounding boxes ranges from 0.5 to 0.95 with a step size of 0.05. The mAP value will decrease as the IoU threshold increases because of more restrictive requirements. The mAP for each range value is calculated, and the average is taken.

By leveraging these evaluation metrics, we can objectively quantify the performance of the enhanced kiwifruit detection model and compare it with other object detection models in the field. The use of these metrics ensures a rigorous and comprehensive evaluation of the proposed kiwifruit detection model, enabling us to accurately assess its performance and compare it with other state-of-the-art models. Furthermore, the evaluation results provide valuable insights into the strengths and weaknesses of the model, allowing for further refinement and improvement of the model's performance.

3.3.2 Evaluation of the Kiwifruit Counting Module

The model proposed in this thesis is used to predict the yield of kiwifruit orchards. Therefore, the output results of the object detection module were fed into the object tracking and counting model to generate the final counting results. To validate the reliability of the kiwifruit counting model, we compared its output results with manually counted results.

During the manual counting process, we ensured the thoroughness and accuracy of the counting. Initially, we recorded the number of kiwifruit in the first frame of each video and then tracked the number of new kiwifruit in subsequent frames. By summing up the new kiwifruit count in each frame, we obtained the total number of kiwifruit in the video as a baseline fact for the experiment. This approach enabled us to establish a reliable and accurate reference for comparing the counting results generated by the kiwifruit counting module. To assess the performance of the kiwifruit counting model in this study, we compared its counting results with the ground truth obtained from manual counting. M Average Counting Precision (ACP) were employed for this purpose.

ACP (Average Counting Precision) is a metric used to evaluate the performance of object counting models, which indicates the matching degree between the model's output counting results and the ground truth counting results. Its calculation is based on the results of object detection and tracking. In this thesis, we made use of visual object detection and tracking algorithms to obtain the position and quantity of each kiwifruit and compared them with the manually counted results. The calculation equation for ACP is shown in Eq. (3.23), where n represents the number of videos, y_i^{truth} represents the manual counting results for the nth sample, and y_i^{pred} represents the model's predicted counting results for the nth sample. ACP reflects the matching degree between the model's counting results and the actual results, and its value ranges from 0 to 1. The higher the ACP value, the higher the matching degree between the model's counting results and the actual number of targets.

$$ACP = \frac{1}{n}\sum_{i=1}^{n}\left(1 - \frac{\left|y_i^{pred} - y_i^{truth}\right|}{y_i^{truth}}\right) \tag{3.23}$$

Chapter 4
Results

In this chapter, we begin by preparing the necessary data for our experiments, providing a detailed description of the methods used to collect, clean, label, augment, and partition our newly proposed kiwifruit database. We then proceed to present the experimental results and model performance of both the kiwifruit detection module and tracking and counting module.

4.1 Data Preparation

The role of data in training deep learning models is fundamental, as it serves as the basis for both model training and evaluation. Specifically, training data is utilized to train the model parameters, whereas evaluation data is employed to assess the model's accuracy and generalization performance. During the data preparation process, data must undergo a series of operations such as pre-processing, cleaning, and labelling to ensure its quality and usability. Only high-quality data can provide accurate training signals, enabling the model to learn effective features and exhibit good generalization ability. If the training data is insufficient or not representative, the model will have difficulty generalizing to new and unseen data. Therefore, data preparation is a necessary step in training deep learning models, which is crucial for the final model's performance and accuracy.

In our thesis, we collected a diversity of kiwifruit image and video datasets from multiple sources to ensure the robustness of our model. To ensure the quality and accuracy of the dataset, we adopted preprocessing techniques such as image cropping, resizing, and augmentation to increase its diversity. The following sections will describe in detail the preparation of the dataset used in this thesis.

4.1.1 Data Collection

In the field of deep learning-based computer vision, large and high-quality datasets are essential for developing and evaluating machine learning models. Four commonly used object detection datasets are COCO, VOC, ImageNet, and Open Images. However, these datasets do not contain kiwifruit object instances, which is not suitable for our specific task of kiwifruit detection and tracking.

The COCO dataset contains over 330,000 images, with over 25,000 object instances from more than 80 object categories, including three types of fruits: bananas, apples, and oranges.

The PASCAL VOC dataset contains 5,717 images, with annotations for multiple object instances in each image, but there is no fruit-related image data.

The ImageNet dataset contains over 14 million images, covering over 20,000 categories, with one set of images for each category. While the dataset includes a fruit category, there is no separate kiwifruit category.

The Open Images dataset contains over 9 million images, with over 900 object categories, including a fruit category, but there is no separate kiwifruit category. Although these datasets cover a wide range of object categories, they do not include kiwifruit object instances.

Therefore, it is necessary to collect our own kiwifruit dataset, customized specifically for our task and with carefully planned annotations to ensure accuracy and consistency. By training our machine learning model on our own dataset, we can ensure that the model is well-suited for the specific conditions and characteristics of our proposed kiwifruit detection and tracking model.

We downloaded kiwifruit orchard videos from the internet and used the “video_to_frames” function by python code to split the collected videos into frames. In addition, we downloaded kiwifruit images from the internet to increase the robustness of our dataset. As shown in Figure 4.1, we collected a total of 3000 kiwifruit images. To comprehensively understand the growth and development of kiwifruit, we collected data from multiple sources. We collected video data from different kiwifruit orchards that were collected at different stages of kiwifruit growth to represent different environments and conditions. The images we collected came from different regions and were captured under different lighting and weather conditions to increase the diversity of our dataset.

Furthermore, we manually inspected each image and video frame to ensure the relevance and quality of the collected data. We also eliminated duplicate and visually similar images to avoid any biases in the dataset. Collecting high-quality and diverse data is crucial for training deep learning models. By ensuring that our dataset is diverse and representative of real-world environments, we can train models that can accurately perform and generalize well to new and unseen data. Therefore, we were very careful in collecting the dataset to ensure its comprehensiveness, high quality, and diversity. Furthermore, to achieve the goal of counting kiwifruits in videos, we collected a total of 20 videos from kiwifruit orchards or packaging lines. These videos were selected to serve as a validation dataset for evaluating the detection and counting performance of our model.

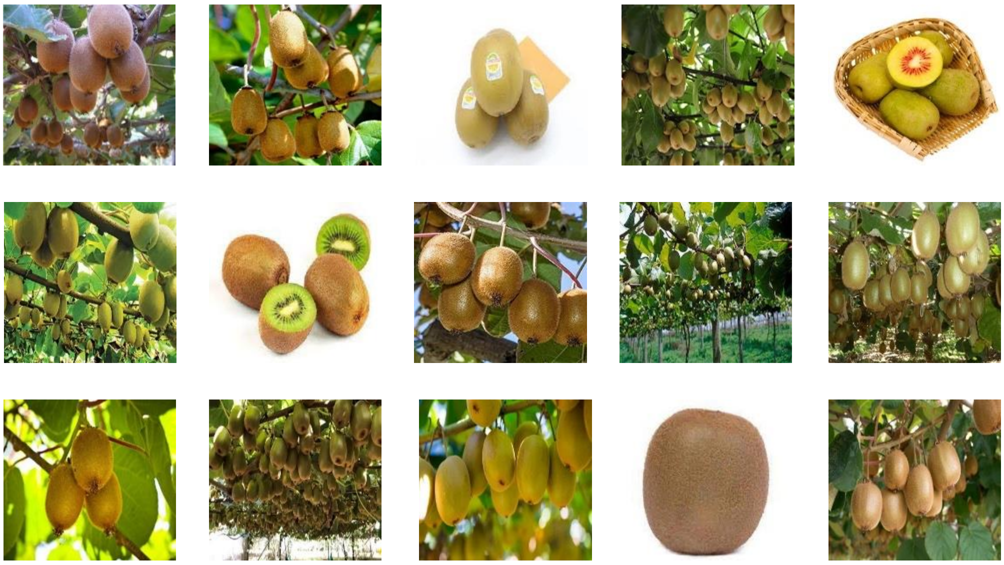


Figure 4.1: The selected samples of the collected image dataset

In summary, our data collection process involved downloading kiwifruit orchard videos from the internet, segmenting them into frames, and downloading kiwifruit images from the internet. We collected data from different orchards videos that were captured at different stages of kiwifruit growth and under different lighting and weather conditions. The quality of the collected data was ensured through manual inspection of each image and video frame, and elimination of duplicates and visually similar images. Our comprehensive and diverse dataset is crucial for training accurate and powerful deep learning models.

4.1.2 Data Cleaning

Data cleaning is a critical step in data preparation that involves identifying and addressing errors, inconsistencies, and inaccuracies in the collected data. This process is crucial for ensuring the quality and reliability of the dataset, which in turn affects the accuracy and effectiveness of the trained model on the data. The process of data cleaning typically involves several steps, including data transformation, data standardization, missing data imputation, and noise removal.

In this thesis, we performed various data cleaning operations to ensure the accuracy and reliability of the dataset used. Firstly, we used deduplication to eliminate duplicate frames from the dataset. This not only reduces the size of the dataset but also avoids bias introduced by duplicate data during model training.

Additionally, we performed deduplication on images using an image hashing algorithm to calculate the hash value of each image. We set the threshold value to 1 and deleted images with hash values lower than 1 that were highly similar to other images, to detect and remove duplicate images. This operation not only helps to reduce the size of the dataset but also avoids introducing noise and unnecessary computational burden caused by duplicate images during model training.

On the other hand, the image sizes in the dataset we collected were not uniform, with some images being too large or too small. The size distribution of the collected image dataset is presented in Figure 4.2, where the red dots represent images that are too large, and the green dots represent images that are too small. The horizontal axis represents the image width, the vertical axis represents the image height, and the points closer to the diagonal line indicate that the aspect ratio of the image is closer to 1. To facilitate the training, validation, and testing of the kiwifruit detection model, we resized all images to a resolution of 640x640. The choice of this resolution was based on a balance between model performance and computational efficiency. To ensure that resizing did not affect the quality of the images, we used the bilinear interpolation method. This method calculates pixel values by interpolating adjacent pixels. Specifically, bilinear interpolation estimates the new pixel value by taking a weighted average of the four nearest known pixels around the target pixel position. This method assumes that the intensity of each pixel varies linearly within the neighborhood of the pixel. Bilinear interpolation is a relatively simple and efficient method that can help adjust image size while preserving the quality and integrity of the image content. In the field of deep learning, bilinear interpolation can be used for preprocessing data, preparing input images for training or testing neural networks. Due to its effectiveness and computational efficiency, it has been widely applied in various computer vision applications such as object detection, image segmentation, and classification. Additionally, we visually inspected the resized image samples to confirm that no obvious distortion or artifacts appeared during the resizing process.

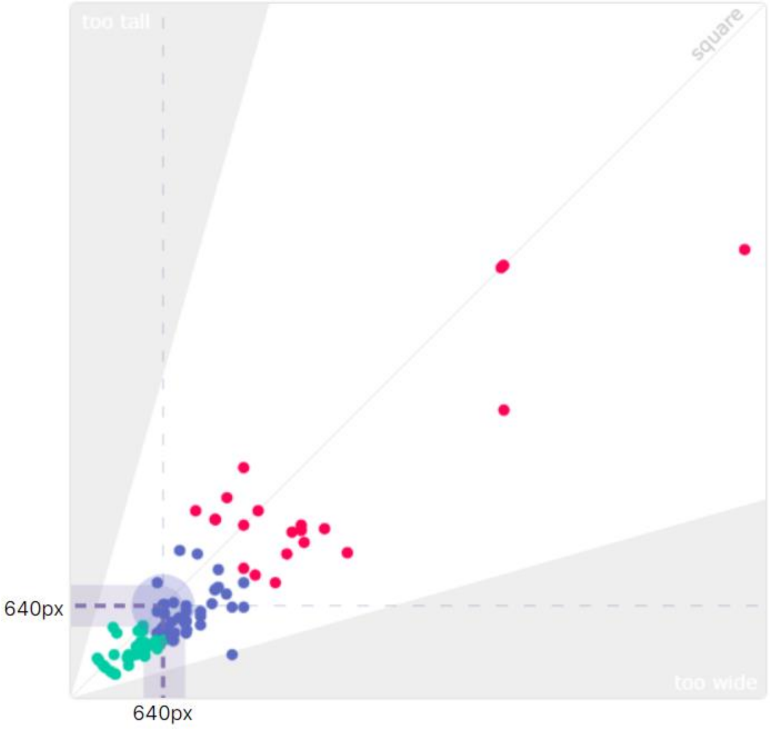


Figure 4.2: The example of labelling objects in Roboflow platform

Finally, we manually checked each video and image to ensure that the content in the dataset was relevant, correct, and of high quality. If any quality issues or irrelevant content were found, they were removed from the dataset.

After the data cleaning process, we ultimately created a kiwifruit dataset containing 1224 images. The data cleaning process is crucial for ensuring the accuracy and reliability of the dataset, which in turn affects the effectiveness of the trained model on the data. In my research, the data cleaning process involved several steps, including deduplication, noise removal, and the establishment of a diverse and representative dataset to ensure the best performance of the improved kiwifruit detection and tracking model.

4.1.3 Data Labelling and Splitting

Data labelling is a critical step in machine learning and computer vision, where labels required for object recognition and tracking are combined with images. The labelled dataset can be used to train and test machine learning models to help computers automatically identify and locate target objects. In this study, we used the Roboflow platform to label 1224 collected kiwifruit images. Figure 4.3 illustrates the process of labelling objects in the Roboflow platform. Roboflow is a comprehensive tool for image labelling and data management, which can automatically perform common data preprocessing tasks such as image scaling, cropping, rotation, and labelling. Using Roboflow can improve the efficiency and accuracy of data labelling, as well as reduce the number of human errors.

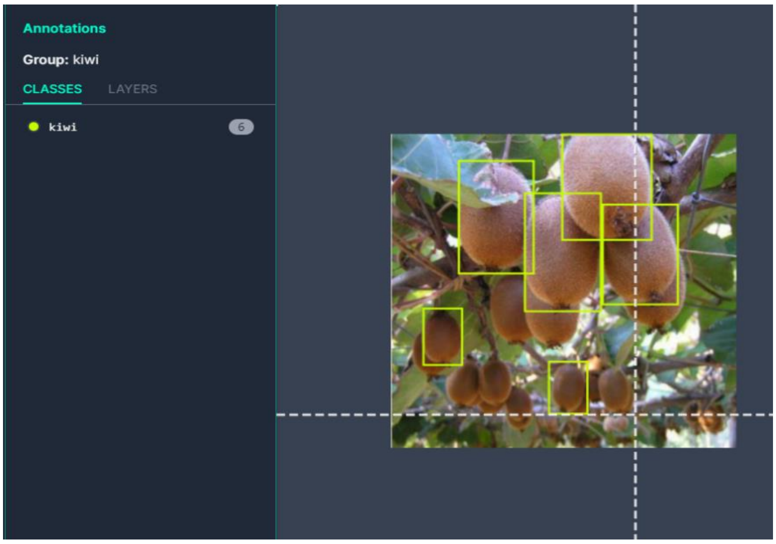


Figure 4.3: The example of labelling objects in the Roboflow platform

In our data labelling process, we first uploaded each kiwifruit image to the Roboflow platform. Then, we manually labelled each kiwifruit and assigned a specific class label for each. In this process, we used the labelling tools provided by Roboflow to label the bounding boxes and categories of each kiwifruit. Finally, we manually labelled a total of 12869 ground truth labels, meaning there are 12869 kiwifruits in our image dataset.

Upon completion of the labelling process, our team leveraged the data management tool provided by Roboflow to efficiently organize and manage our dataset. In particular, we partitioned the dataset into three distinct subsets, namely the training set, validation set, and test set, comprising 855, 246, and 123 original images, respectively. The use of a separate test set was critical to assessing the generalization performance of the trained model. Furthermore, the validation set facilitated hyperparameter tuning and allowed us to select the most optimal model during the training process. By dividing the dataset into subsets, we ensured that the trained model did not overfit to the training data and could generalize well to previously unseen data. Notably, we were able to effortlessly view and modify our labelled data on the Roboflow platform and export the dataset in various formats based on our specific needs. With the labelled dataset, we were able to train our models effectively and enhance the precision and dependability of our kiwifruit yield prediction model.

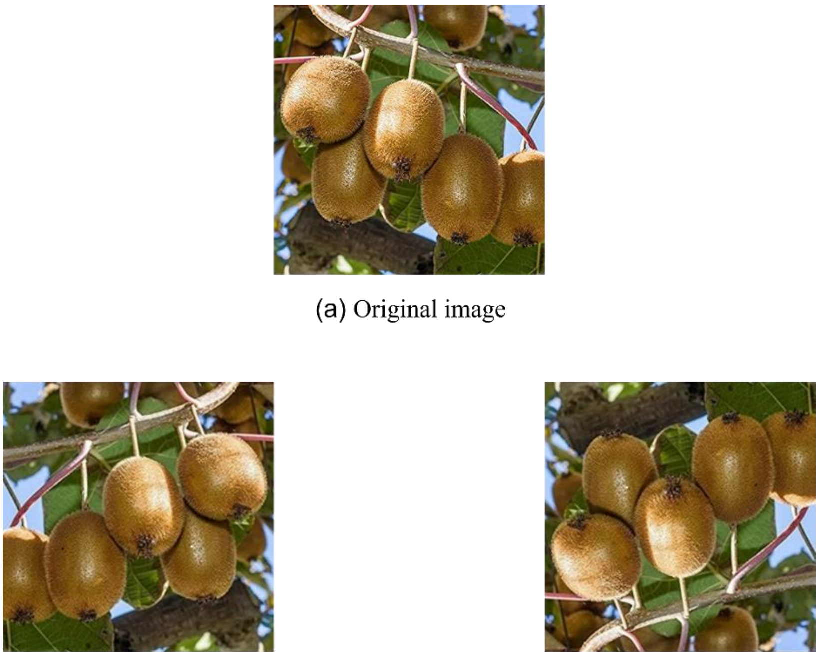
4.1.4 Data Augmentation

In the field of deep learning, having a large and diverse training dataset is crucial for developing and evaluating robust and accurate models. In particular, for complex tasks such as object detection, models need to be able to identify and locate objects of varying sizes, shapes, and orientations in different environments. Large training datasets can help models encounter a wide range of object instances, backgrounds, and lighting conditions, enabling them to learn powerful features and generalize well to new data. Training on large and varied datasets is crucial for improving the accuracy and generalization ability of deep learning models.

A prime example is the ImageNet Large Scale Visual Recognition Challenge (ILSVRC) dataset, which consists of more than 1.2 million images spanning 1,000 object categories, and has significantly advanced the state-of-the-art techniques for object recognition. The increased diversity and volume of data enable models to learn complex features and subtle patterns in the data, leading to improved performance on unseen data. Moreover, large datasets mitigate the risk of overfitting, where models become too specialized to the training data and fail to generalize to new data. Therefore, collecting and using large and diverse training datasets is essential for achieving high performance in object detection tasks. Compared to small datasets, large datasets provide more samples to cover different appearances, poses, and lighting variations of target categories, making the model more robust. Moreover, large datasets can effectively alleviate the overfitting phenomenon and improve the model's generalization ability (Fu, Nguyen & Yan, 2022).

The purpose of data augmentation techniques is to artificially increase the size and diversity of the training dataset, thereby improving the performance and generalization of the deep learning models. The underlying principle is to introduce variations in the training data that mimic real-world scenarios, which the model is expected to encounter during deployment. For instance, geometric transformations help to simulate different viewpoints, orientations, and positions of the objects in the images, which can help the model learn more robust and invariant features. Similarly, color space transformations help to simulate changes in lighting conditions and atmospheric effects that can affect the appearance of the objects in the images. By introducing such variations, the model can learn to recognize and adapt to these variations during training, leading to better performance on unseen data. Image filtering techniques help to smooth out noise and reduce image artifacts, thereby improving the clarity and quality of the images. This can help the model learn more relevant and discriminative features, leading to better classification and detection accuracy.

In this thesis, geometric transformations are considered as a means of data augmentation due to the large number of kiwifruit labels present on each image in the dataset, which vary in size, ripeness, lighting conditions, and other factors (Xiao, Nguyen & Yan, 2021; Wang & Yan, 2021; Song et al., 2022). In practical hardware deployment, the method of video capture may differ, and could involve either ground-based tracked robot video capture vehicles or aerial unmanned aerial vehicle video capture, resulting in different angles of kiwifruit in the input video. Therefore, to improve the model's robustness and enable high performance in kiwifruit videos captured at different angles, we horizontally and vertically flipped the 855 original images in the training set to expand the dataset. Figure 4.4 shows the original images and their flipped versions after the data augmentation process. Following this process, the number of images in the training set increased to 1514.



(a) Original image

(b) Horizontal image

(c) Vertical image

Figure 4.4: Examples of original image and augmented images

4.1.5 Data Splitting

In the field of deep learning, it is a common practice to divide a dataset into subsets for the purpose of training, validation, and testing in order to evaluate model performance on unseen data and prevent overfitting. The training set is typically used to train the model, while the validation set is used to optimize hyperparameters and monitor model performance during training. Finally, the testing set is used to assess the overall performance of the model on previously unseen data.

In this thesis, we also performed data splitting and used 1883 images for our dataset.

Specifically, 1514 images (80%) were used for training, 246 images (13%) for validation, and 123 images (7%) for testing. We trained the kiwifruit detection model on the training set, optimized the model using the validation set, and evaluated the final performance of the model on the testing set. Figure 4.5 provides a clear summary of the dataset split. This approach allowed us to obtain reliable and accurate results while avoiding overfitting to the training data.

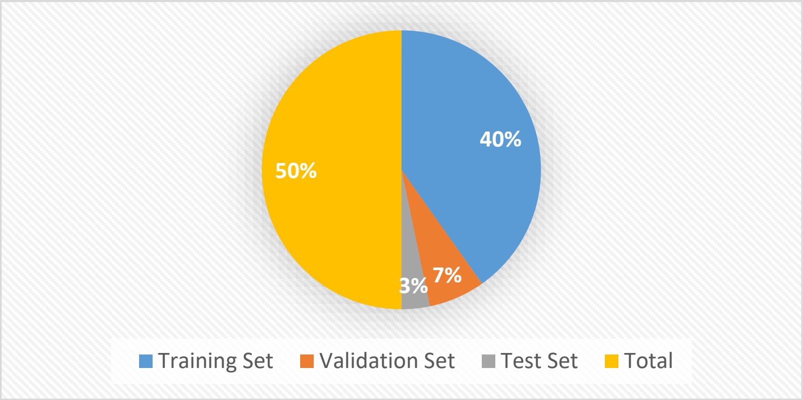
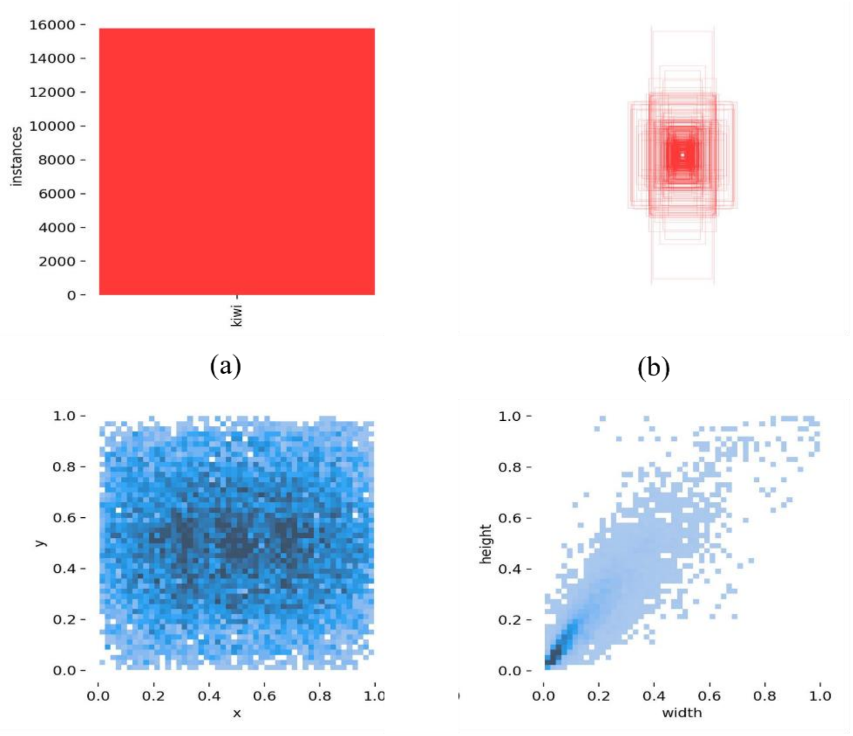


Figure 4.5: The pie chart of data splitting

Figure 4.6 illustrates the label information of the training set, where Figure 4.6 (a) shows the number of training samples for the kiwifruit class. Since only kiwifruit is considered in this study, the bar chart displays the quantity of kiwifruit class in the training set. Figure 4.6 (b) presents the size and quantity of the ground truth labels, and the distribution plot indicates a reasonable distribution of kiwifruit sizes in the annotated training set, primarily featuring smaller fruit volumes found in actual orchards, resulting in a denser distribution of smaller labels. Figure 4.6 (c) displays the position of the ground truth labels center relative to the entire image, and the distribution plot indicates that the distribution of labels in the training set meets the experimental requirements, with a distribution that is centrally clustered, indicating that the labels are uniformly distributed in the training set and mostly located at the center of the images. Figure 4.6 (d) presents the aspect ratio of the objects relative to the entire image, and the distribution plot reveals that the aspect ratio of kiwifruit in the training set is reasonably distributed, with a low number of over-deformed images with excessive width or height.



(a) (b)

(c) (d)

Figure 4.6: The graphical representation of label information in the training set.

Through the data preparation and augmentation work, we have successfully created a specialized image dataset for kiwifruit detection. This dataset consists of a training set, a validation set, and a test set, with 1,514, 246, and 123 images, respectively. These datasets will serve as the foundation for our kiwifruit detection and tracking model. In our research design, we will adopt deep learning techniques and machine learning algorithms to establish a kiwifruit detectiong, tracking and counting model. This model will predict the yield of kiwifruit by analysing the characteristics and properties of kiwifruit images. To improve the accuracy and robustness of the model, we will use a variety of deep learning algorithms and improve the model's performance by hyperparameter tuning and model architecture refinement. In the following experiments, we will train, validate, and test the model and evaluate its performance using various evaluation metrics. Through these experiments, we will validate the feasibility of our research design and provide a new and efficient solution for kiwifruit yield prediction.

4.2 Performance of the Kiwifruit Detection Module

4.2.1 Experimental Environment and Parameter Setup

In this thesis, we first trained an improved YOLOv8 network model for kiwifruit detection based on the pre-trained YOLOv8s model using 1514 kiwifruit images. The experimental environment for the kiwifruit detection module training is described in Table 4.1. The experiments were conducted on Google Colaboratory platform using Python 3.8.16, PyTorch 1.13, CUDA version 11.6, and Tesla T4 GPU with a memory capacity of 16GB. Google Colaboratory provides a free cloud computing platform with GPU support, which allows for efficient training of deep learning models. PyTorch is a widely used deep learning framework that provides efficient implementations of neural network models. The CUDA version 11.6 is used to leverage the power of GPU computation and accelerate the training and inference process. The Tesla T4 GPU is known for its high performance and is well-suited for deep learning applications.

Table 4.1: Experimental environment

Experiment platform	Python	Pytorch	CUDA	GPU
Google Colaboratory	vision 3.8.16	vision 1.13	vision 11.6	Tesla T4 (15110MiB)

In Table 4.2, we describe the hyperparameter settings of the improved YOLOv8 algorithm used in the kiwifruit detection module. The table provides detailed information on various hyperparameter values adjusted during the training process, such as learning rate, batch size, and momentum. The table also lists the values of the number of iterations, warmup epochs, and steps to decrease the learning rate. Specifically, the model was trained for 150 epochs with a batch size of 8 and an image size of 640. The optimizer used was SGD with a learning rate of 0.01, momentum of 0.937, and weight decay of 0.001. The warmup epochs, momentum, and bias learning rate were set to 3.0, 0.8, and 0.1, respectively. The confidence threshold and IoU threshold for non-maximum suppression were set to 0.001 and 0.7, respectively, with a maximum detection number of 3000. Other important hyperparameters included the box, class, and objectness scale, which were set to 7.5, 0.5, and 1.5, respectively. Additionally, data augmentation was applied during training with various parameters, such as hue, saturation, and brightness, and mosaic was used to combine multiple images. Hyperparameters play a crucial role in the performance of the algorithm and fine-tuning them can lead to better detection accuracy. Therefore, by using these hyperparameters, the improved YOLOv8 algorithm can achieve high detection accuracy for kiwifruit in various complex scenarios, providing a solid foundation for kiwifruit yield prediction models.

4.2.2 Experimental Results

In order to verify whether the improved YOLOv8 model has better kiwifruit detection performance than other existing YOLO models, we conducted a comparative experiment in the same experimental environment using the same dataset and parameters. We trained YOLOv4, YOLOv5, YOLOv6, YOLOv7, and the unimproved YOLOv8 model separately using different comparison models to train the dataset for a total of 150 epochs. During the training process, we used the same hyperparameters, including a batch size of 8, image size of 640, and optimizer of SGD. To avoid overfitting, we implemented an early stopping strategy where the training would stop if the model did not show significant improvement after 50 epochs. We also recorded the loss values and validation accuracy during the training process for subsequent analysis and comparison. After training was completed, we tested and validated the trained comparison models on a total of 123 and 246 images, respectively.

Figure 4.7 depicts the ground truth of the labeled kiwifruit results on the validation dataset, which were carefully curated to serve as a benchmark for assessing the performance of our detection model. As illustrated, the kiwifruits exhibit significant variations in size, shape, orientation, and environmental context, posing a challenging task for detection.

Table 4.2: The influence of the number of moving objects on the accuracy

Parameters	Value	Parameters	Value
task	detect	visualize	false
mode	train	boxes	true
model	yolov8s.pt	dynamic	false
data	data.yaml	simplify	false
epoch	150	workspace	4
patience	50	nms	false
batch	8	lr0	0.01
imgsz	640	lrf	0.01
save	true	momentum	0.937
save_period	-1	weight_decay	0.001
workers	2	warmup_epochs	3.0
optimizer	SGD	warmup_momentum	0.8
verbose	true	warmup_bias_lr	0.1
seed	0	box	7.5
deterministic	true	cls	0.5
single_cls	false	dfl	1.5
image_weights	false	fl_gamma	0.0
rect	false	label_smoothing	0.0
cos_lr	false	nbs	64
close_mosaic	10	hsv_h	0.015
resume	false	hsv_s	0.7
min_memory	false	hsv_v	0.4
overlap_mask	true	degrees	0.0
mask_ratio	4	translate	0.1
dropout	false	scale	0.5
val	true	shear	0.0
split	val	perspective	0.0
conf	0.001	flipud	0.0
iou	0.7	fliplr	0.5
show	false	mosaic	1.0
hide_labels	false	mixup	0.0
hide_conf	false	copy_paste	0.0
vid_stride	1	cfg	null
line_thickness	3	v5loader	false

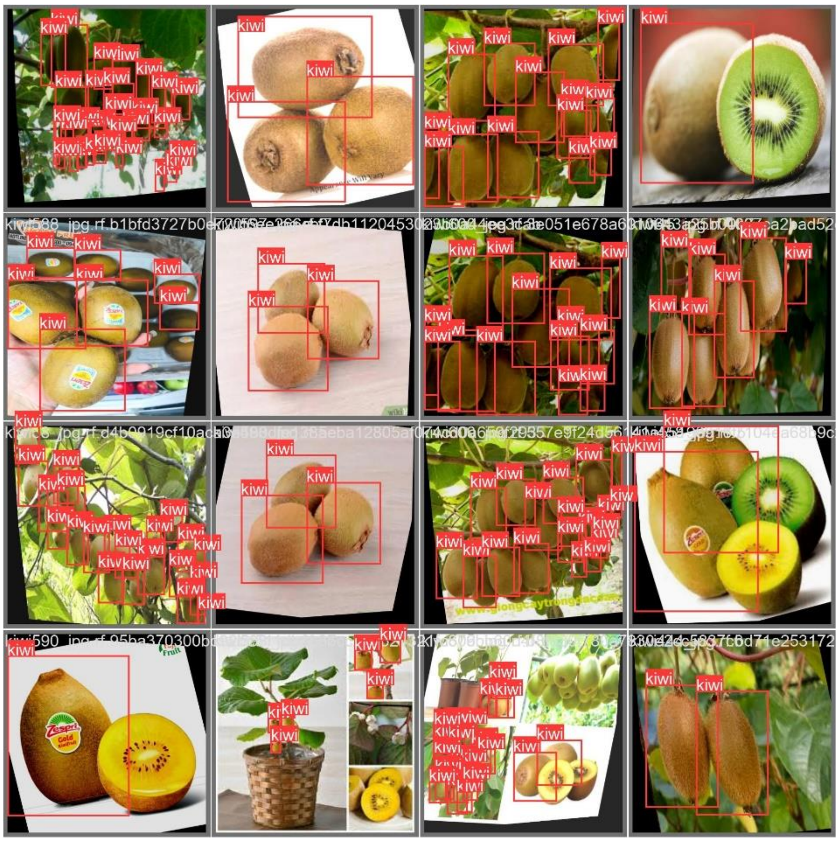


Figure 4.7: The labelled results on the validation set.

Nevertheless, in Figure 4.8, we presented the detection results of our proposed model on the same kiwifruit validation dataset. The proposed model exhibited high accuracy in detecting most of the kiwifruits in different scenes, including those of varying sizes and overlapping with one another, as evidenced by the high degree of overlap between the predicted bounding boxes and the ground truth annotations. Additionally, the proposed model demonstrated remarkable generalization ability, detecting kiwifruits of diverse sizes and orientations. A comparison between Figures 4.1 and Figure 4.2 underscores the efficacy of our proposed model in detecting kiwifruits in real-world scenarios.

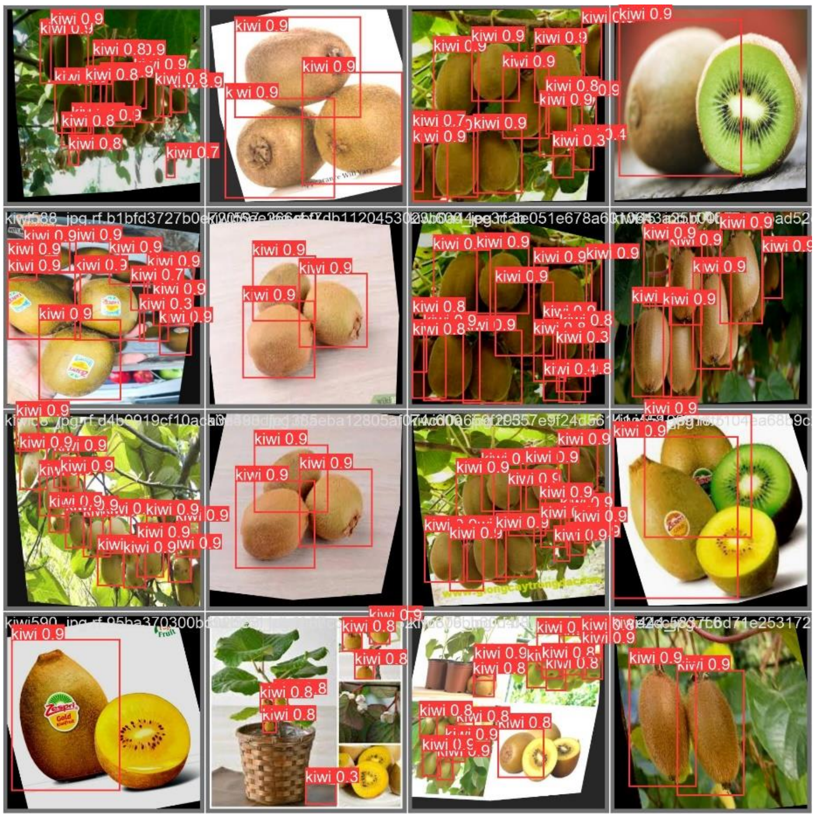


Figure 4.8: The prediction results of our proposed model on the validation set.

Upon completion of 150 training epochs, we assessed the performance of our proposed kiwifruit detection model by plotting its precision-recall (PR) curve, depicted in Figure 4.9. The PR curve depicts the trade-off between precision and recall at various confidence thresholds. Recall reflects the proportion of true positive samples accurately detected by the model, while precision signifies the ratio of true positive samples among the detected results. The area under the curve (AUC) in the PR curve reflects the performance of the model, which was 0.956 in our case. We noticed that the precision of our model was quite high when the recall was low, which was due to the high proportion of true positives in the detected results. However, as recall increased, precision gradually declined due to an increase in the number of false positives detected by the model. Additionally, we evaluated the model's performance using the balance point on the PR curve, which represents the point where the recall and precision are equal. Our model showed a high balance point, indicating that it achieved a good balance between recall and precision and accurately detected positive samples. In conclusion, our kiwifruit detection module showed high performance on the PR curve after training, accurately detecting positive samples.

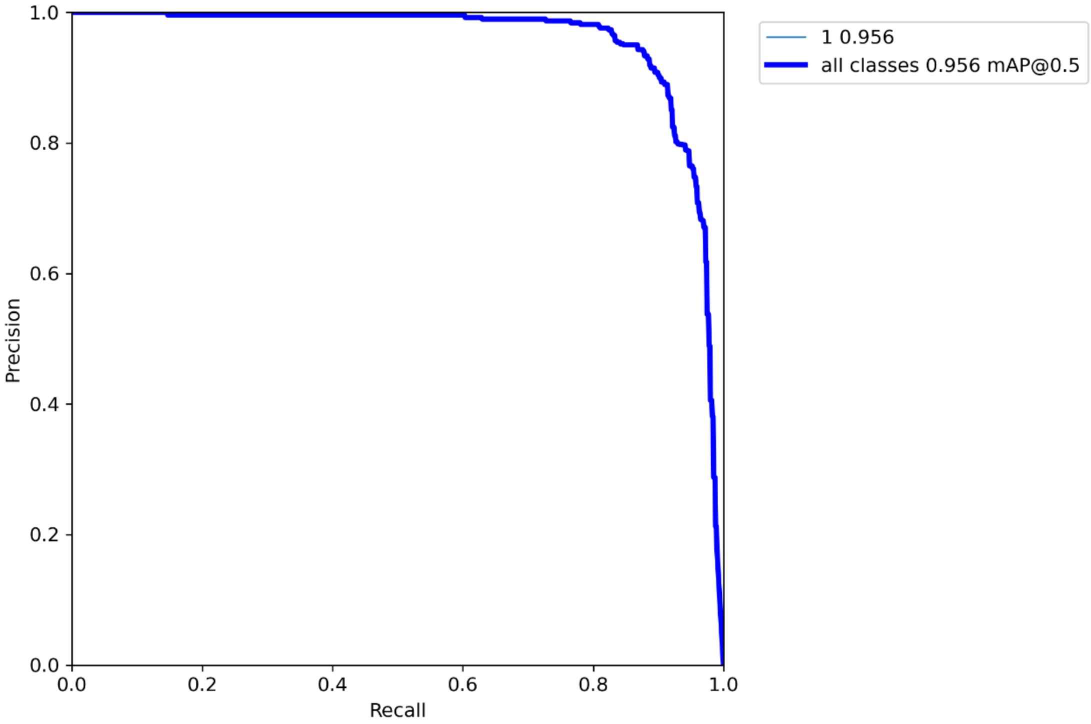


Figure 4.9: Precision-recall curve of our proposed kiwifruit detection module after 150 epochs of training

Figure 4.10 showcases three loss functions and four evaluation metrics. The loss function is composed of bounding box regression loss, objectness loss, and classification loss. The bounding box loss measures the model's accuracy in locating the object's center and coverage of predicted bounding boxes. It comprises position offset and scale change, where position offset refers to the deviation between predicted and ground truth bounding boxes, and scale change indicates the scale ratio between predicted and ground truth bounding boxes. Objectness loss measures the likelihood of an object existing in a proposed region of interest. Each predicted bounding box has an objectness score, indicating the presence of an object. The objectness loss is based on the binary cross-entropy loss function, where a predicted bounding box containing the true target should have an objectness score close to 1 and vice versa. The classification loss remains zero since the data contains only one category.

Figure 4.10 displays four evaluation metrics: Precision (P), Recall (R), mAP_0.5, and mAP_0.5:0.95. Precision measures bbox prediction accuracy, and Recall measures the accuracy of true bbox predictions. mAP_0.5 is the average precision at an IoU threshold of 0.5, while mAP_0.5:0.95 is the average mAP at different IoU thresholds ranging from 0.5 to 0.95 with a step size of 0.05. The validation data shows a rapid decrease in box and objectness losses after approximately 100 epochs of training. The values of four evaluation metrics tend to stabilize after around 80 epochs of training, as shown in the figure.

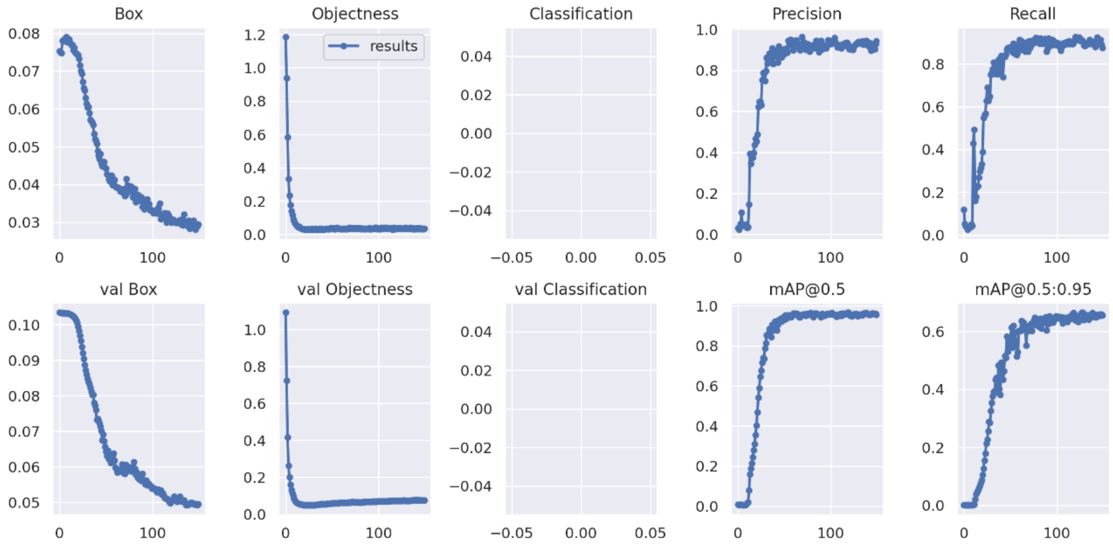


Figure 4.10: Plots of box loss, objectness loss, precision, recall and mAP over the training epochs for the training and validation set

Table 4.3 presents the comparison results of various YOLO models on our dataset, including their training epochs, model sizes, precision, recall, mAP@0.5, and mAP@[.5:.95]. Our proposed model achieved the highest precision of 0.934, recall of 0.911, mAP@0.5 of 0.942, and mAP@[.5:.95] of 0.677 among all the models, demonstrating its superior performance over other YOLO models. Specifically, compared with YOLOv4, our proposed model showed a 7.3% improvement in precision, 9.8% improvement in recall, 8.8% improvement in mAP@0.5, and 16.4% improvement in mAP@[.5:.95]. Compared with YOLOv5, our model achieved a 4.2% increase in precision, 7.8% increase in recall, 6.9% increase in mAP@0.5, and 9.2% increase in mAP@[.5:.95]. Compared with YOLOv6, our model showed a 3% increase in precision, 3.5% increase in recall, 3.6% increase in mAP@0.5, and 6.8% increase in mAP@[.5:.95]. Compared with YOLOv7, our model still outperformed in precision, recall, mAP@0.5, and mAP@[.5:.95] with 1.7%, 1.4%, 2.5%, and 2.8% improvements, respectively. Although the original YOLOv8 model demonstrated good performance with a precision of 0.921, recall of 0.905, and mAP of 0.921 at IoU threshold of 0.5 and mAP of 0.658 at IoU threshold of 0.95, our proposed model outperformed YOLOv8 in all evaluation metrics. YOLOv4, YOLOv5, and YOLOv6 models exhibited relatively lower performance, especially in mAP. Overall, our proposed model showed outstanding performance in detecting kiwifruits, and the high-performance kiwifruit detection module provided a reliable foundation for our kiwifruit counting model. By using this detection module, we can effectively detect and locate kiwifruits and pass them to the counting model for processing. Due to the high performance and reliability of the kiwifruit detection module, our kiwifruit counting model can accurately identify and count kiwifruits, thus improving the reliability and efficiency of the entire system, meeting the demand for efficient and intelligent modern agriculture production (Yan, 2019).

Table 4.3: The performance comparison of different YOLO models on our dataset

Model	Epoch	Size	Precision	Recall	mAP@0.5	mAP@[.5:.95]
YOLOv4	150	640	0.861	0.813	0.854	0.513
YOLOv5	150	640	0.892	0.833	0.873	0.585
YOLOv6	150	640	0.904	0.876	0.906	0.609
YOLOv7	150	640	0.917	0.897	0.917	0.649
YOLOv8	150	640	0.921	0.905	0.921	0.658
Our proposed	150	640	0.934	0.911	0.942	0.677

Furthermore, in order to evaluate the effectiveness of the inserted modules in our improved model, we conducted ablation experiments on YOLOv8 as the baseline, investigating the impact of channel attention module (CAM), spatial attention module (SAM), and Focal-EIoU loss on the performance of the original YOLOv8 model. CBAM is a combination of CAM and SAM, which integrates channel and spatial attention mechanisms into a single module. CAM and SAM aim to model the interdependence between channels and spatial positions in the feature map. CAM calculates channel attention by capturing global background information for all spatial positions, while SAM simulates spatial attention by focusing on the region with the most informative content in the feature map. Focal-EIoU loss is a novel loss function designed to address the class imbalance problem in object detection tasks, which achieves better performance than commonly used cross-entropy loss and focal loss.

Table 4.4: The ablation studies on our dataset

Models	SAM	CAM	Focal-EIoU	mAP@0.5	mAP@[.5:.95]
YOLOv8 (baseline)				0.921	0.658
+SAM	✓			0.927	0.660
+CAM		✓		0.924	0.658
+CBAM	✓	✓		0.933	0.661
+ Focal-EIoU			✓	0.937	0.671
+CBAM+ Focal-EIoU **(Our proposed)**	✓	✓	✓	**0.956**	**0.677**

Our ablation experimental results demonstrate that the inserted CBAM module combining SAM and CAM, and the improved Focal-EIoU loss function can effectively enhance the performance of the YOLOv8 model on kiwifruit detection tasks. As shown in Table 4.4, we conducted ablation experiments on our collected dataset. The baseline model of YOLOv8 achieved a performance of 0.921 and 0.658 on mAP@0.5 and mAP@[.5:.95] metrics, respectively. After incorporating the SAM attention mechanism, the model's mAP@0.5 metric slightly improved to 0.927, and the mAP@[.5:.95] also showed a small increase. With the addition of the CAM attention mechanism, the model's mAP@0.5 metric showed a slight improvement compared to the baseline model, but there was no improvement in mAP@[.5:.95]. However, with the addition of the CBAM attention mechanism, the model's performance improved significantly, achieving 0.933 and 0.661 on mAP@0.5 and mAP@[.5:.95] metrics, respectively. By adding the Focal-EIoU, the model's mAP@0.5 metric showed a significant improvement, reaching a performance of 0.937. Finally, with the addition of both CBAM and Focal-EIoU, the model showed a significant improvement in all metrics, achieving the best performance of 0.956 and 0.677 on mAP@0.5 and mAP@[.5:.95], respectively. Therefore, the ablation experiments demonstrate that CBAM and Focal-EIoU have a significant effect on improving the performance of kiwifruit detection models.

4.3 Performance of Kiwifruit Counting Module

The performance of the kiwifruit counting module is a crucial aspect of our proposed kiwifruit yield prediction model, which comprises detection and tracking modules. Previous experiments have already demonstrated the effectiveness of the detection module and achieved excellent detection results. However, the accuracy of the counting module is also vital to the overall performance of the model.

In this section, we will present the results of the kiwifruit tracking and counting modules and evaluate their performance. The tracking module tracks the detected kiwifruits over a period of time and calculates the final yield using a counter. The tracking module aims to address the challenges posed by kiwifruit occlusion and natural factors such as wind and animal interference that cause trajectory interruptions. To evaluate the performance of our kiwifruit counting module, we conducted experiments on a kiwifruit orchard video dataset that we collected. This dataset includes videos captured by ground cameras and conveyor belt videos of kiwifruit orchards, with varying illumination conditions, camera angles, and kiwifruit growth stages.

Our proposed yield prediction model's performance is illustrated in Figures 4.11~ Figure 4.13. Overall, the model demonstrates good detection, tracking, and counting of kiwifruits. Figure 4.11 presents the prediction result of the yield prediction model in the first frame of a real orchard video. The total number of kiwifruits is displayed in the top-left corner of the image, while the ID, class name, and confidence level of each target are shown in the top-left corner of each anchor box. In this video frame, a large number of kiwifruits are overlapping, and some of them are only partially visible, with only around 10% of their surface area visible.

Additionally, the different absolute distances of kiwifruits from the camera result in significant differences in their sizes. Moreover, the presence of leaves with different colors in the video scene may also cause prediction errors in the model. However, after manually counting the visible kiwifruits, the result was 20, consistent with the prediction of the model. This video frame demonstrates the excellent performance of our proposed model in detecting and counting highly overlapping targets in complex backgrounds.

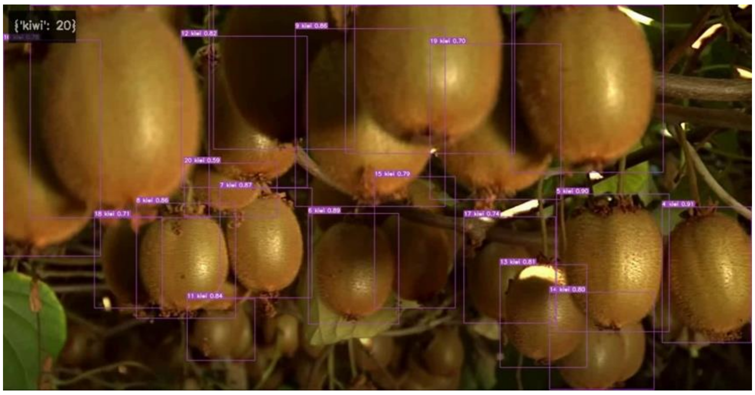


Figure 4.11: The performance of kiwifruit detection and counting in the 1st frame of an orchard video.

Figure 4.12 shows the prediction results of the yield prediction model in the 65th frame of a kiwifruit sorting conveyor video, where the total count of kiwifruit is displayed in the top left corner of the image, and the ID, category name, and confidence of each target are shown in the top left corner of each anchor box. The rolling conveyor constantly changes the position and status of kiwifruit during the sorting and transportation process, and the kiwifruit on the conveyor are sometimes obstructed by the arms of sorting personnel, causing the model to lose targets. However, our proposed kiwifruit counting model still maintains good performance in the face of these challenges.

Figure 4.13 displays the prediction results of our model in the 109th frame of the kiwifruit orchard video, where the kiwifruits are mainly concentrated in the central area, while there are numerous leaves and branches with similar colors in other areas of the image. In addition, the kiwifruit in the upper part of the images are only partially visible. Nevertheless, our proposed model successfully avoids erroneous prediction anchor boxes caused by the high similarity of leaves and branches. Furthermore, this image also demonstrates the accuracy of our model in predicting incomplete kiwifruit, which is consistent with the actual situation in production.

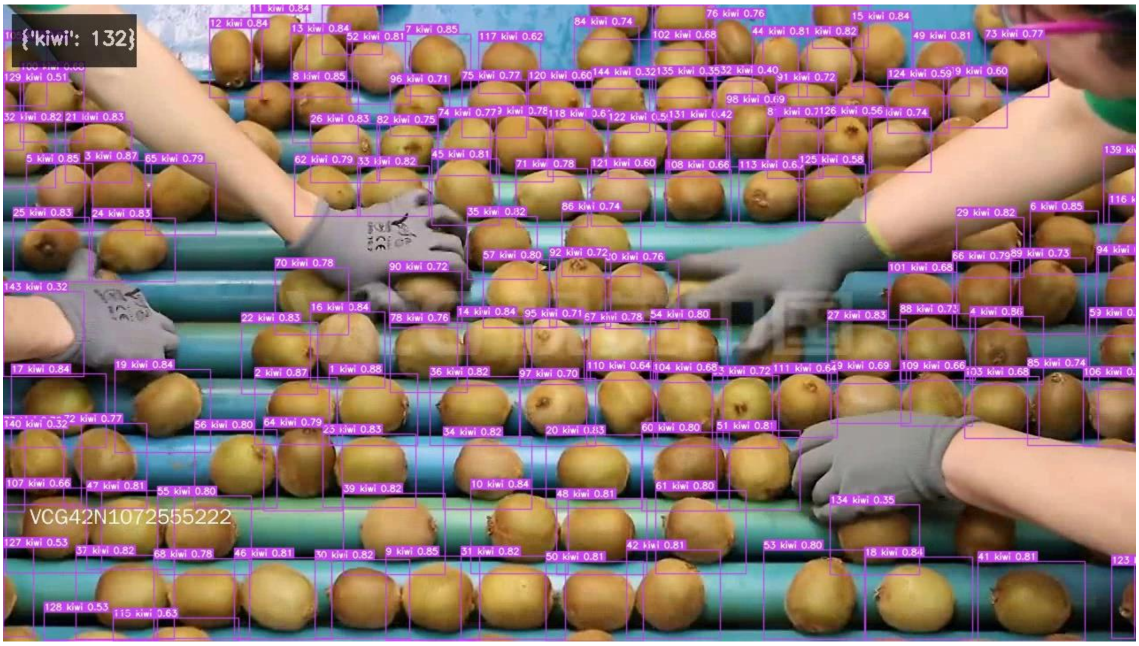


Figure 4.12: The performance of kiwifruit detection and counting in the 65th frame of a sorting conveyor video.

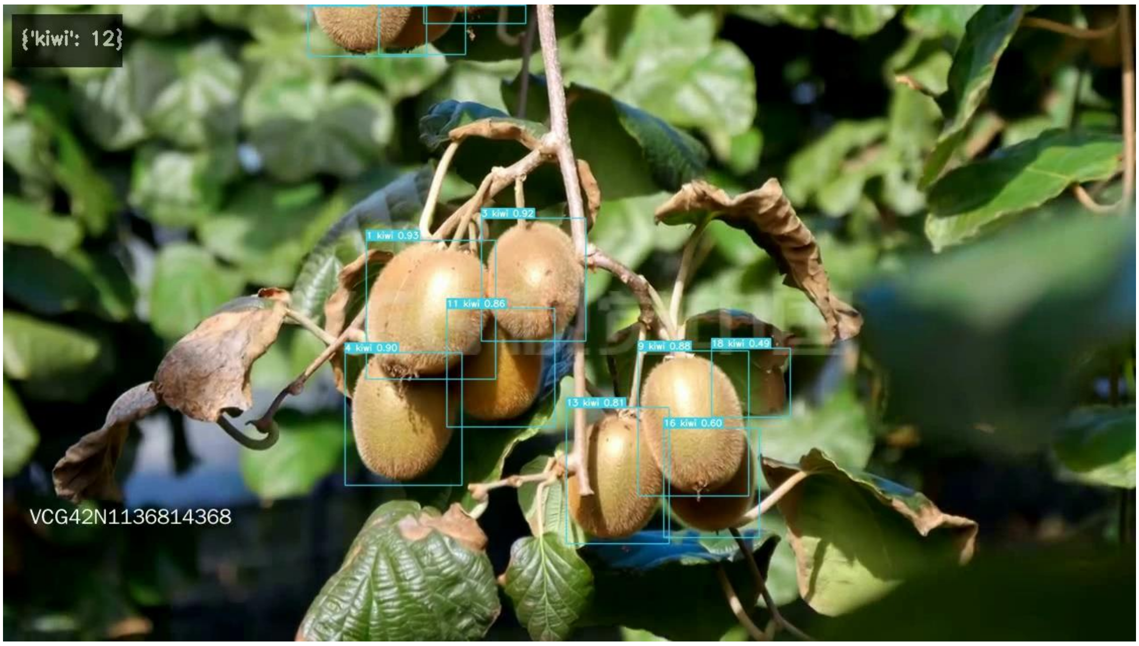


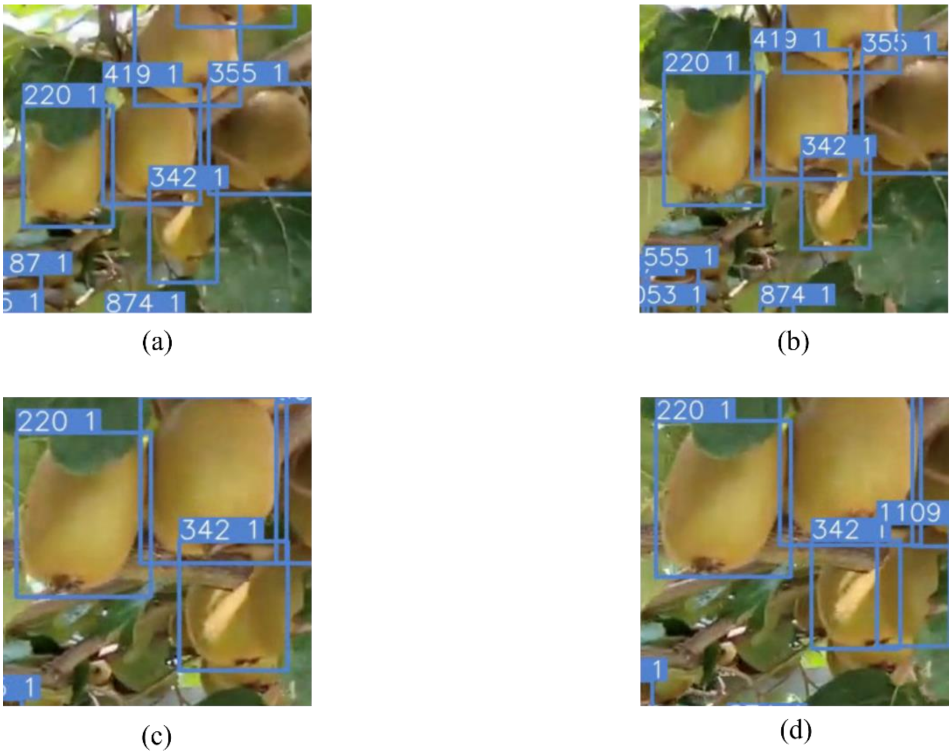
Figure 4.13: The performance of kiwifruit detection and counting in the 109th frame of an orchard video.

In this thesis, we evaluated the performance of the proposed tracking algorithm by conducting experiments on real kiwifruit orchard videos and recording tracking results under various conditions. We presented four sets of comparative tracking results in the thesis, each containing four images. These comparative results included tracking performance under different conditions, such as tracking of overlapping fruits, tracking interruptions, re-labeling of fruits after tracking interruptions caused by branch occlusions, and re-labeling of lost kiwifruits.

Specifically, we observed that in the kiwifruit orchard videos, due to potential overlapping of fruits, the tracking algorithm needs to accurately differentiate between different fruits and assign unique IDs to them. Figure 4.14 illustrates the ID assignment of occluded fruits in motion due to overlapping. The four images in the figure show the motion states of kiwifruits in the video from distant to close regions in chronological order. Despite the change in the occlusion status of the kiwifruit with ID 342, the ID remains unchanged, and no target loss or ID reassignment occurs.

Furthermore, during long-term tracking, some fruits may move or disappear in the scene, resulting in fruit loss and re-appearance. In our experiments, we observed such cases and re-labeled the lost kiwifruits using the algorithm, ensuring the integrity and stability of tracking. Figure 4.15 presents an example of tracking lost and re-appearing kiwifruits. Kiwifruit with ID 1004 in Figure 4.15 (a) gradually becomes occluded until it completely disappears in Figure 4.15 (c), but reappears and is tracked with the same ID as the video frames continue, indicating that the ID of the kiwifruit remains unchanged despite the temporary disappearance that caused tracking target loss.

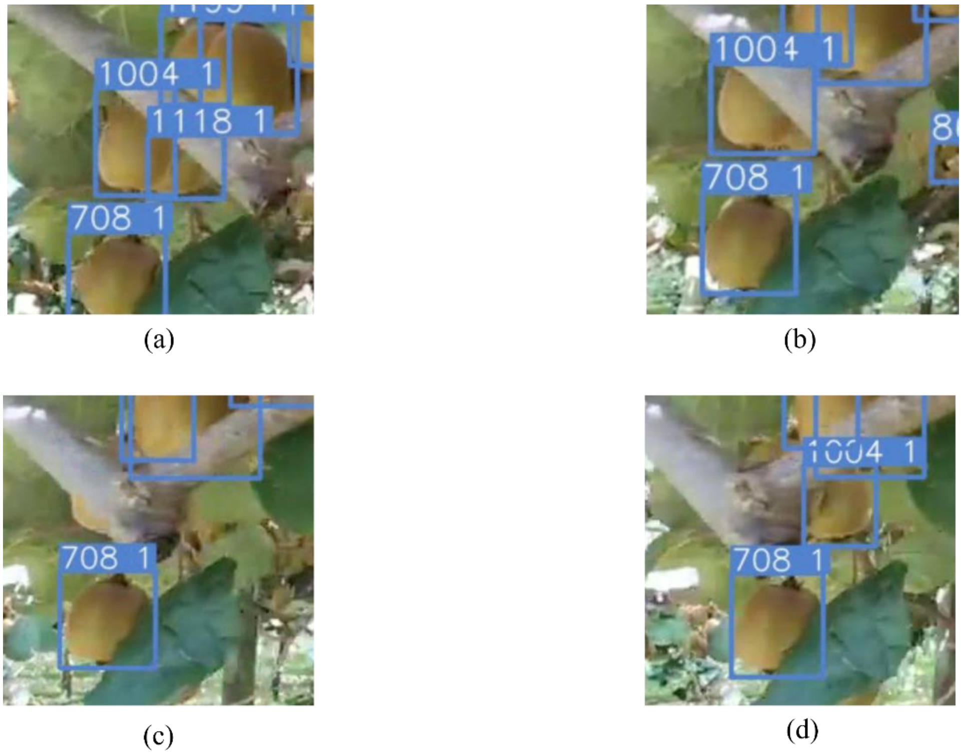
In addition, due to the complex branching structure of fruit trees in the orchard, tracking interruptions may occur due to occlusion by tree branches during the tracking process. To address this issue, our algorithm is able to re-label the interrupted fruit after the occlusion is resolved, ensuring tracking continuity and accuracy. Figure 4.16 presents an example of re-tracking and re-labeling of kiwifruits that were lost and reappeared due to occlusion by tree branches and leaves. Kiwifruit with ID 1664 in Figure 4.16 (a) is lost in Figure 4.16 (b) due to occlusion by tree leaves, but is re-tracked and labeled with the same ID in Figure 4.16 (c) and Figure 4.16 (d). Kiwifruit with ID 1047 in Figure 4.16 (a), Figure 4.16 (b), and Figure 4.16 (c) is lost in Figure 4.16 (d) due to occlusion by tree branches, but is re-detected and tracked in the subsequent tracking process. Partial re-labeling of kiwifruits also occurred in Figure 4.17, where kiwifruits with IDs 2171 and 2168 in Figure 4.17 (a) were not detected and labeled due to extensive occlusion by tree leaves in Figure 4.17 (b), but the lost kiwifruit with ID 2168 reappeared and was re-labeled in Figure 4.17 (c), indicating the robustness of our algorithm in handling occlusion challenges in complex environments.



(a) (b) (c) (d)

Figure 4.14: The example of obscured kiwifruit tracking.

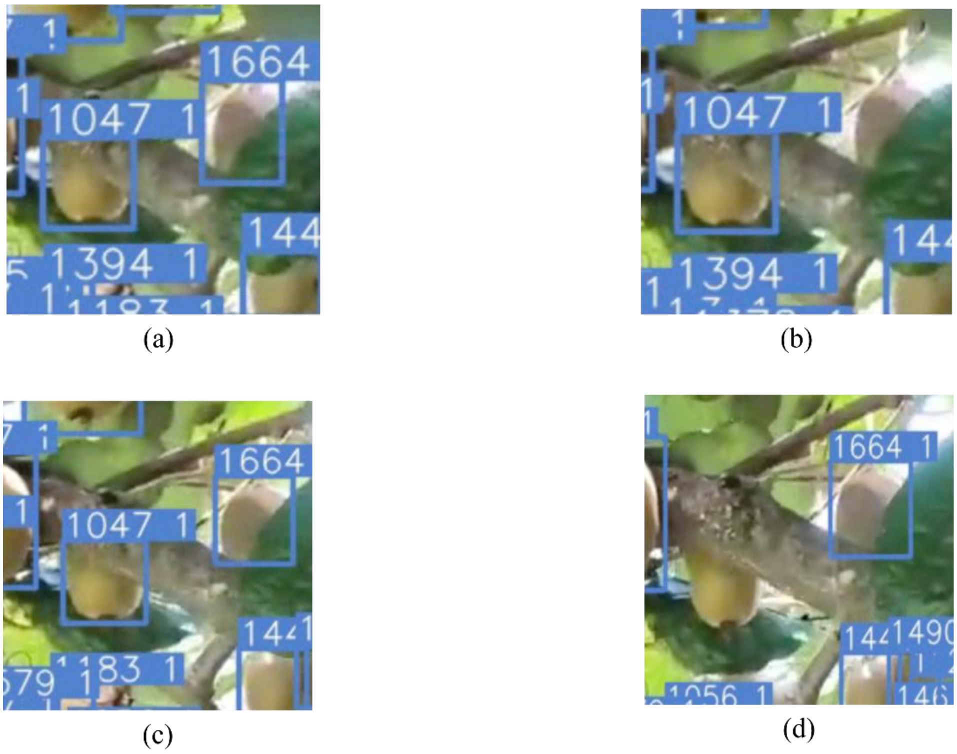
In light of the evidence presented, it is reasonable to conclude that the effectiveness and reliability of the proposed tracking algorithm in handling various challenges encountered in kiwifruit orchard videos, including overlapping fruits, tracking interruptions, and occlusions by tree branches and leaves, and ensuring accurate and continuous tracking of kiwifruits. These findings contribute to the advancement of fruit tracking research in agricultural applications and have practical implications for improving kiwifruit management and harvesting operations.



(a) (b)

(c) (d)

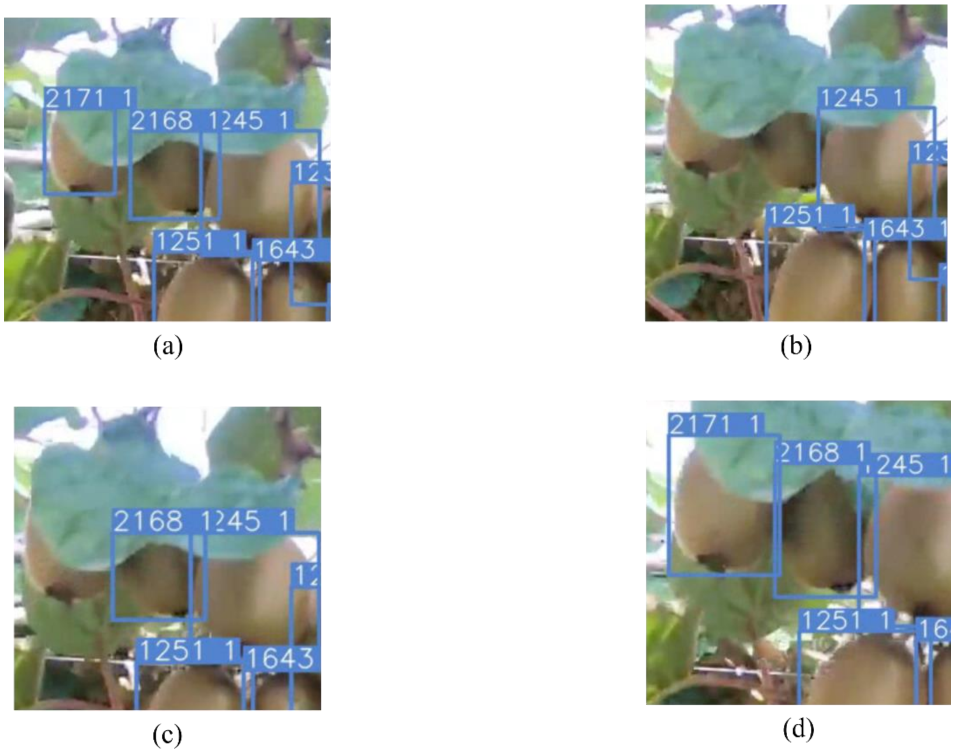
Figure 4.15: The example of kiwifruit tracking with interruptions.



(a) (b)

(c) (d)

Figure 4.16: The Example of kiwifruit tracking with re-labeling after occlusion by tree branch.



(a) (b) (c) (d)

Figure 4.17: The example of re-labeling lost kiwifruit tracking.

To evaluate the performance of our kiwifruit detection, tracking and counting model, we manually calculated the actual yield of 20 video segments and compared it with the yield generated by our model. Table 4.5 compares the performance of the kiwifruit yield prediction model with the ground truth results obtained by manual counting of the 20 video segments. The ground truth column displays the actual kiwifruit yield in each video, while the model count column shows the predicted yield generated by our kiwifruit counting module. The count error column represents the absolute difference between the predicted yield and the actual yield, while the error rate column represents the percentage of the count error to the actual yield. Overall, the count error ranged from 0.13% to 0.38%, with an average count error of only 0.22%.

This result suggests that the model is robust and can accurately predict kiwifruit yield, regardless of lighting, camera angles, and kiwifruit growth stages. The count accuracy column shows the accuracy of the model in detecting kiwifruit in the video. It is defined as the proportion of the correctly detected kiwifruit to the total number of kiwifruits in the video. The average count accuracy was 0.78, indicating that the model correctly detected 78% of the kiwifruit in the video. The ACP of the prediction model is the average count accuracy of the kiwifruit counting module. It is a measure of the overall accuracy of the model in predicting kiwifruit yield in all videos.

Table 4.5: The performance comparison of predicted and ground truth kiwifruit yields in 20 video clips.

Video ID	Ground Truth	Model Count	Counting Error	Error Rate	Counting Precision
1	485	577	+92	0.190	0.810
2	1074	1306	+232	0.216	0.784
3	712	843	+131	0.184	0.816
4	441	521	+80	0.181	0.819
5	1379	1691	+312	0.226	0.774
6	293	348	+55	0.188	0.812
7	491	587	+96	0.196	0.804
8	313	369	+56	0.179	0.821
9	1098	1349	+251	0.229	0.771
10	604	721	+117	0.194	0.806
11	746	962	+216	0.290	0.710
12	1243	1405	+162	0.130	0.870
13	224	308	+84	0.375	0.625
14	1009	1271	+262	0.260	0.740
15	2361	2570	+209	0.089	0.911
16	1810	2253	+443	0.245	0.755
17	322	409	+87	0.270	0.730
18	1080	1304	+224	0.207	0.793
19	611	794	+183	0.300	0.700
20	1317	1608	+291	0.221	0.779
ACP of Prediction Model = 0.782					

The ACP of the prediction model was 0.782, indicating that the model had high accuracy in predicting kiwifruit yield. Our kiwifruit counting module achieved high accuracy in predicting final yield while demonstrating robustness to changes in lighting, camera angles, and kiwifruit growth stages. Moreover, even in challenging situations, such as obstructions and interruptions caused by wind and animal interference, the module was able to accurately track kiwifruit. These results demonstrate the effectiveness of our kiwifruit yield prediction model in real kiwifruit planting scenarios.

Chapter 5
Analysis and Discussions

In this chapter, we conduct an analysis and discussion of the experimental results obtained from our proposed kiwifruit detection model. We compare the performance of different models and provide insights into their strengths and weaknesses.

5.1 Analysis

In this thesis, we present a novel kiwifruit yield prediction model based on detection, which comprises two main components: a detection module and a tracking module. By integrating the enhanced detection module with a Kalman filter-based tracking module, we attained precise kiwifruit counting in orchard videos, exhibiting remarkable performance in both detection and tracking accuracy. In this analysis section, we conduct a thorough assessment of the proposed model, encompassing the experimental outcomes of the detection module and tracking module.

In this chapter, we provide a comprehensive evaluation of the proposed detection-based kiwifruit yield prediction model that consists of two modules: the detection module and the tracking module. Specifically, we focus on analysing the experimental results of the detection module. We conducted qualitative research to validate the effectiveness of our proposed detection model for detecting small overlapping kiwifruits in complex backgrounds.

Additionally, we conducted comparative experiments to evaluate the performance of several popular one-stage object detection algorithms. A comparison of the performance of our proposed kiwifruit detection model with other state-of-the-art detection models is presented in Figure 5.1, in terms of precision, recall, mAP@0.5, and mAP@[.5:.95]. The results demonstrate that our proposed improved detection model outperforms all other models in the YOLO family, achieving a mean average precision at intersection over union 0.5 of 95.6%, which is an improvement over previous model.

Furthermore, we conducted ablation experiments to study the effectiveness of different components of our improved detection module, including channel attention mechanism, spatial attention mechanism, and CBAM, and validated the performance of our proposed Focal-EIoU loss function. Our ablation experiments show that both channel and spatial attention mechanisms contribute to the performance improvement, with the spatial attention mechanism having a more significant impact. Moreover, the overall performance improvement from CBAM is higher than that of individual channel and spatial attention mechanisms. Finally, our ablation experiments confirm that the combination of CBAM and the proposed Focal-EIoU loss function provides a significant performance improvement for our kiwifruit detection module.

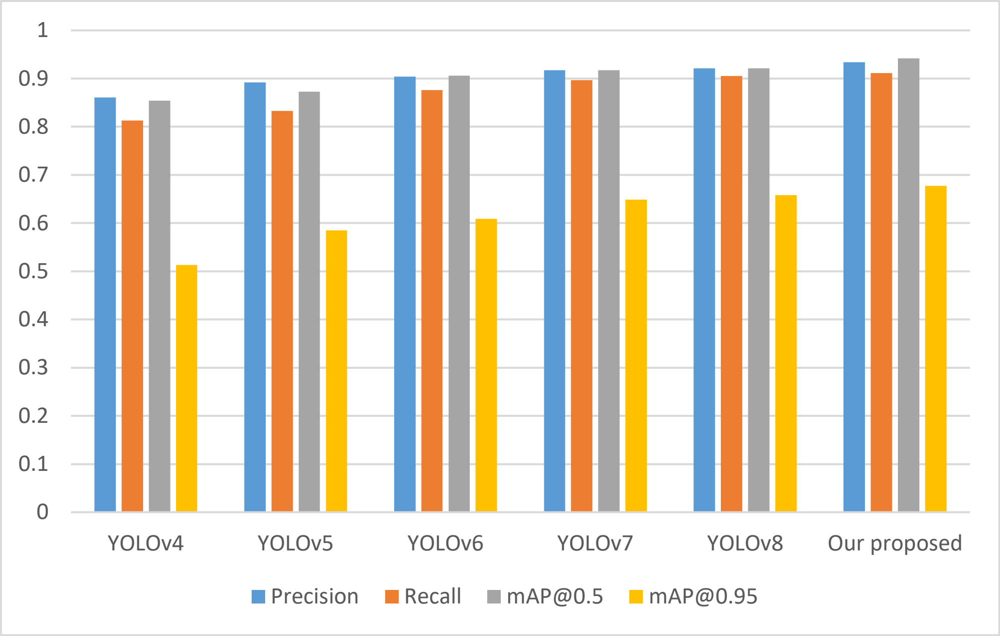


Figure 5.1: The bar chart of comparison of performance metrics for kiwifruit detection module.

The proposed kiwifruit yield prediction model in this thesis consists of two modules, namely, the detection module and the tracking module. As mentioned earlier, the detection module is based on an improved YOLOv8 network, achieving an mAP of 95.6% at an IoU threshold of 0.5, demonstrating its effectiveness in accurately detecting kiwifruit in videos. The detection module outputs the position and class information of kiwifruit targets, which are then passed to the tracking module based on the Kalman filter algorithm. The tracking results are matched using the Hungarian algorithm, enabling the tracking of kiwifruit's motion trajectory, and achieving an average counting accuracy of 78.2% after integration with the detection module.

The ability of our method to correctly assign IDs and count obscured targets further demonstrates its effectiveness in kiwifruit detection, tracking and counting. Specifically, the Kalman filter algorithm predicts the position and status of kiwifruit in the next frame based on the current state and historical information. Meanwhile, the Hungarian algorithm completes large-scale target matching and tracks the movement trajectory of kiwifruit between the previous and current frames.

By combining these two algorithms, our tracking module achieves high tracking accuracy and robustness to kiwifruit occlusion and partial visibility. In addition, to evaluate the effectiveness of our proposed tracking module, we conducted tracking performance experiments on 20 videos with different detection accuracy and frame rates. The experimental results demonstrate that our tracking module achieves high tracking accuracy in different scenarios.

5.2 Discussions

In this thesis, we have proposed a kiwifruit yield prediction approach based on an upgraded YOLOv8 network and the Kalman filter algorithm. Our approach consists of two modules, namely detection and tracking, which work collaboratively to detect and track kiwifruits in digital videos and predict their production. The performance of our suggested system in detecting and tracking kiwifruits in videos and forecasting their yield has been assessed.

Our research results have demonstrated that the inclusion of the attention mechanism and intersection over union (IOU) calculation in the YOLOv8 network can significantly improve its detection performance. By utilizing an enhanced YOLOv8 model, we achieve high precision in identifying kiwifruits in videos. The detection module provides the location and class information of the detected kiwifruits, which serve as inputs to the tracking module. The tracking module, employing the Kalman filter algorithm and the Hungarian algorithm, estimates the position and status of the kiwifruit in the next frame based on the current state and historical information. The Hungarian algorithm is utilized to match the predicted results with the actual results, thereby determining the kiwifruit's motion trajectory in the video. Through the integration of these two algorithms, our method achieves high tracking accuracy and robustness, even in situations where kiwifruits are partially visible or occluded by other objects.

To evaluate the performance of our suggested technique, we conducted trials on 20 different videos with varying detection accuracy and frame speeds. The results demonstrate that our technique exhibits good tracking accuracy and is capable of handling diverse conditions. With an average counting accuracy of 78.2%, our approach proves to be useful in estimating kiwifruit yield. Additionally, we compared our suggested detection module with existing approaches such as YOLOv4 and YOLOv5. The experimental results indicate that our method surpasses these approaches in terms of detection accuracy, further highlighting the effectiveness of our proposed method.

In summary, our proposed method has demonstrated its effectiveness in accurately detecting and tracking kiwifruits in videos and predicting their yield. The combination of the improved YOLOv8 network and the Kalman filter algorithm enables us to achieve high accuracy and robustness in various scenarios. The contributions of our work to the field of kiwifruit yield prediction are valuable, and our method holds the potential to be applied in actual kiwifruit production to enhance efficiency and quality. By emphasizing the practical implications of our research, we envision its real-world applications benefiting the kiwifruit industry and contributing to its advancement.

Chapter 6
Conclusion and Future Work

This chapter will provide a summary of the subject and methodology of the current thesis, as well as highlight any limitations or shortcomings encountered during the experiments. Based on these findings, new research directions will be proposed, laying the groundwork for future work.

6.1 Conclusion

In this thesis, we proposed an improved detection-based tracking kiwifruit yield prediction model using a combination of improved YOLOv8, Kalman filtering, and Euclidean distance. Our model aims to accurately predict the yield of kiwifruit crops by using computer vision technology to detect and track fruits, reducing waste in terms of manpower and resources for growers.

To evaluate the performance of our model, we collected a customized kiwifruit image dataset and conducted experiments. The results showed that our model could achieve counting of kiwifruits with high accuracy and low error rates. Specifically, the detection module of our model achieved a mAP of 95.6% at an IoU threshold of 0.5, and when combined with the tracking and counting module, the average counting precision reached 78.2%, demonstrating its effectiveness in predicting kiwifruit yield.

Additionally, we conducted a series of comparative experiments to verify the effectiveness of our proposed model. The results showed that the attention mechanism added to the detection module improved the overall performance of the model. Moreover, the improved loss function had a positive impact on the performance of our model. By adjusting these parameters and improving the algorithm model, we can improve the accuracy of our model.

All in all, our proposed model using the combination of improved YOLOv8, Kalman filtering, and Euclidean distance provides a good solution for accurately predicting kiwifruit crop yields. This model has demonstrated its effectiveness in real-world scenarios and can be easily extended to other crops with similar characteristics. In the future, further research can be conducted to optimize the model's performance and explore its potential applications in other areas.

6.2 Limitations

Although the results of this study are encouraging, it is important to acknowledge and consider the limitations of our research, which may affect the generalizability and applicability of our findings to other environments or populations.

Firstly, the kiwifruit dataset used in this study is limited in scale and variety, which may affect the generalization ability of our model. Despite our efforts to ensure that the dataset covers a range of different environmental conditions and cultivation varieties, the performance of the model may be affected when applied to other datasets with diverse features and varieties.

Secondly, the proposed model heavily relies on computer vision technology, which may be limited by external factors such as diverse lighting conditions or camera angles. Although we attempted to mitigate these factors by augmenting the image dataset in various ways, in the real world, the performance of the model may be affected by the increase in uncontrollable variables.

Thirdly, the proposed model is computationally intensive, requiring high-performance computing resources to achieve detection and tracking. This may limit its applicability on low-resource environments or devices with insufficient computing power.

Finally, although our model achieved high accuracy in kiwifruit detection, tracking and counting, there may be other factors that influence yield that were not considered in this study, such as soil quality, irrigation, and pest management. Future research should focus on incorporating these factors into the model to improve its predictive accuracy.

In summary, these limitations highlight the need for further research and improvement of our proposed model, as well as the limitations and potential challenges that will be encountered when applying computer vision technology to real-world agricultural settings.

6.3 Future Work

In this thesis, we proposed an improved yield prediction model for kiwifruit crops using a combination of improved YOLOv8, Kalman filtering, and Euclidean distance. Our model showed promising results in counting and accurate prediction of kiwifruit yield using computer vision techniques, reducing waste in terms of human and material resources. However, there is still room for improvement, and several future directions can be explored to enhance the performance and applicability of our proposed model.

Firstly, one potential future direction is to extend our model to other crop types. Our proposed model can be adapted to other crops with similar features, such as shape, size, and color, by retraining the model with a customized dataset. The extension of our model to other crops can potentially have a significant impact on agricultural productivity and efficiency, benefiting farmers and consumers alike.

Secondly, collecting and enhancing training datasets can be another important future work. A larger and more diverse dataset can improve the generalization and robustness of our model, making it more suitable for different growing conditions and environments. Additionally, the use of data augmentation techniques, such as rotation, scaling, and translation, can increase the variability of the training data, making the model more capable of handling complex scenarios and unexpected situations (Luo, Yan & Nguyen, 2022).

Thirdly, pruning experiments can be conducted to optimize the model's prediction speed. Pruning is a technique for reducing the size of deep neural networks by removing unnecessary parameters or connections without sacrificing the model's accuracy. By pruning our model, we can potentially reduce its computational complexity and memory footprint, making it more suitable for deployment on resource-constrained devices, such as embedded systems or mobile devices.

Finally, integrating our proposed model with embedded Linux development boards, such as Raspberry Pi, can enable the model to work seamlessly with robots or drones, extending its application scenarios beyond fixed locations. The integration of our model with embedded systems can potentially enable monitoring and autonomous decision-making, improving the efficiency and precision of agricultural operations.

In conclusion, our proposed yield prediction model using a combination of improved YOLOv8, Kalman filtering, and Euclidean distance has shown promising results in counting and accurate prediction of kiwifruit yield. However, further research can be conducted to extend the model to other crop types, improve the training dataset, optimize the model's prediction speed, and integrate the model with embedded systems. These future directions can potentially enhance the performance and applicability of our proposed model and contribute to the advancement of precision agriculture.

References

Al-Sarayreh, M., Reis, M., Yan, W., Klette, R. (2019) A sequential CNN approach for foreign object detection in hyperspectral images. *International Conference on Information, Communications and Signal.*

An, N., & Yan, W. Q. (2021). Multitarget tracking using Siamese neural networks. *ACM Transactions on Multimedia Computing, Communications, and Applications, 17*(2s), 1-16.

An, N. (2020) Anomalies Detection and Tracking Using Siamese Neural Networks. Master's Thesis. Auckland University of Technology, New Zealand.

Antol, S., Agrawal, A., Lu, J., Mitchell, M., Batra, D., Zitnick, C. L., & Parikh, D. (2015). VQA: Visual question answering. *IEEE International Conference on Computer Vision (ICCV).*

Apolo-Apolo, O., Martínez-Guanter, J., Egea, G., Raja, P., & Pérez-Ruiz, M. (2020). Deep learning techniques for estimation of the yield and size of citrus fruits using a UAV. *European Journal of Agronomy*, 115, 126030.

Awate, A., Deshmankar, D., Amrutkar, G., Bagul, U., & Sonavane, S. (2015). Fruit disease detection using color, texture analysis and ANN. *International Conference on Green Computing and Internet of Things (ICGCIoT).*

Barbedo, J. (2019). A review on the use of unmanned aerial vehicles and imaging sensors for monitoring and assessing plant stresses. *Drones, 3*(2), 40.

Bargoti, S., & Underwood, J. P. (2017). Image segmentation for fruit detection and yield estimation in apple orchards. *Journal of Field Robotics, 34*(6), 1039-1060.

Bazame, H. C., Molin, J. P., Althoff, D., & Martello, M. (2021). Detection, classification, and mapping of coffee fruits during harvest with computer vision. *Computers and*

Electronics in Agriculture, 183, 106066.

Bewley, A., Ge, Z., Ott, L., Ramos, F., & Upcroft, B. (2016). Simple online and realtime tracking. *IEEE International Conference on Image Processing (ICIP)*.

Blackman, S. (2004). Multiple hypothesis tracking for multiple target tracking. *IEEE Aerospace and Electronic Systems Magazine, 19*(1), 5-18.

Buzzy, M., Thesma, V., Davoodi, M., & Mohammadpour Velni, J. (2020). Real-time plant leaf counting using deep object detection networks. *Sensors, 20*(23), 6896.

Carion, N., Massa, F., Synnaeve, G., Usunier, N., Kirillov, A., & Zagoruyko, S. (2020). End-to-end object detection with Transformers. *ECCV*, 213-229.

Cecotti, H., Rivera, A., Farhadloo, M., & Pedroza, M. A. (2020). Grape detection with convolutional neural networks. *Expert Systems with Applications*, 159, 113588.

Chen, S. W., Shivakumar, S. S., Dcunha, S., Das, J., Okon, E., Qu, C., . . . Kumar, V. (2017). Counting apples and oranges with deep learning: A data-driven approach. *IEEE Robotics and Automation Letters, 2*(2), 781-788.

Chlingaryan, A., Sukkarieh, S., & Whelan, B. (2018). Machine learning approaches for crop yield prediction and nitrogen status estimation in Precision Agriculture: A Review. *Computers and Electronics in Agriculture*, 151, 61-69.

Comaniciu, D., & Meer, P. (1999). Mean shift analysis and applications. *IEEE International Conference on Computer Vision*.

De Luna, R. G., Dadios, E. P., Bandala, A. A., & Vicerra, R. R. (2020). Tomato growth stage monitoring for smart farm using deep transfer learning with machine learning-based maturity grading. *AGRIVITA Journal of Agricultural Science, 42*(1).

Deng, J., Dong, W., Socher, R., Li, L., Kai Li, & Li Fei-Fei. (2009). ImageNet: A large-scale hierarchical image database. *IEEE Conference on Computer Vision and Pattern Recognition*.

Dong, C., & Duoqian, M. (2023). Control distance IOU and control distance IOU loss for better bounding box regression. *Pattern Recognition*, 137, 109256.

Dorj, U., Lee, M., & Yun, S. (2017). A yield estimation in citrus orchards via fruit detection and counting using image processing. *Computers and Electronics in Agriculture,* 140, 103-112.

Farkhodov, K., Lee, S., & Kwon, K. (2020). Object tracking using CSRT Tracker and R-CNN. *International Joint Conference on Biomedical Engineering Systems and Technologies*.

Fawakherji, M., Potena, C., Prevedello, I., Pretto, A., Bloisi, D. D., & Nardi, D. (2020). Data augmentation using GANs for crop/weed segmentation in precision farming. *IEEE Conference on Control Technology and Applications (CCTA)*.

Ferentinos, K. P. (2018). Deep learning models for plant disease detection and diagnosis. *Computers and Electronics in Agriculture*, 145, 311-318.

Ferguson, A. R. (2004). 1904—the year that Kiwifruit (Actinidia deliciosa) came to New Zealand. *New Zealand Journal of Crop and Horticultural Science, 32*(1), 3-27.

Fortmann, T., Bar-Shalom, Y., & Scheffe, M. (1983). Sonar tracking of multiple targets using Joint Probabilistic Data Association. *IEEE Journal of Oceanic Engineering, 8*(3), 173-184.

Fountas, S., Espejo-Garcia, B., Kasimati, A., Mylonas, N., & Darra, N. (2020). The future of digital agriculture: Technologies and opportunities. *IT Professional, 22*(1), 24-28.

Fu, L., Gao, F., Wu, J., Li, R., Karkee, M., & Zhang, Q. (2020). Application of consumer RGB-D cameras for fruit detection and localization in field: A critical review. *Computers and Electronics in Agriculture*, 177, 105687.

Fu, Y., Nguyen, M., & Yan, W. Q. (2022). Grading methods for fruit freshness based on deep learning. *SN Computer Science, 3*(4).

Fu, Y. (2020) Fruit Freshness Grading Using Deep Learning. Master's Thesis. Auckland University of Technology, New Zealand.

Gao, F., Fang, W., Sun, X., Wu, Z., Zhao, G., Li, G., . . . Zhang, Q. (2022). A novel apple fruit detection and counting methodology based on deep learning and trunk tracking in modern orchard. *Computers and Electronics in Agriculture*, 197, 107000.

Gao, F., Fu, L., Zhang, X., Majeed, Y., Li, R., Karkee, M., & Zhang, Q. (2020). Multi-class fruit-on-plant detection for Apple in SNAP system using Faster R-CNN. *Computers and Electronics in Agriculture*, 176, 105634.

Gao, F., Yang, T., & Fu, L. (2021). Apple fruit detection and counting based on deep learning and trunk tracking. *ASABE Annual International Virtual Meeting*.

Gene-Mola, J., Vilaplana, V., Rosell-Polo, J. R., Morros, J., Ruiz-Hidalgo, J., & Gregorio, E. (2019). Multi-modal deep learning for fuji apple detection using RGB-D cameras and their radiometric capabilities. *Computers and Electronics in Agriculture,* 162, 689-698.

Girshick, R. (2015). Fast R-CNN. *IEEE International Conference on Computer Vision (ICCV)*.

Girshick, R., Donahue, J., Darrell, T., & Malik, J. (2014). Rich feature hierarchies for accurate object detection and semantic segmentation. *IEEE Conference on Computer Vision and Pattern Recognition*.

Gongal, A., Amatya, S., Karkee, M., Zhang, Q., & Lewis, K. (2015). Sensors and systems for Fruit Detection and localization: A Review. *Computers and Electronics in Agriculture*, 116, 8-19.

Gongal, A., Karkee, M., & Amatya, S. (2018). Apple fruit size estimation using a 3D machine vision system. *Information Processing in Agriculture, 5*(4), 498-503.

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

Gu, Q., Yang, J., Kong, L., Yan, W. Q., & Klette, R. (2017). Embedded and real-time vehicle detection system for challenging on-road scenes. *Optical Engineering, 56*(6), 063102.

Guadagna, P., Fernandes, M., Chen, F., Santamaria, A., Teng, T., Frioni, T., . . . Gatti, M. (2023). Using deep learning for pruning region detection and plant organ segmentation in dormant spur-pruned grapevines. *Precision Agriculture*.

Gulshan, V., Peng, L., Coram, M., Stumpe, M. C., Wu, D., Narayanaswamy, A., . . . Webster, D. R. (2016). Development and validation of a deep learning algorithm for detection of diabetic retinopathy in retinal fundus photographs. *JAMA, 316*(22), 2402.

Hassoon, I. M. (2021). Shape feature extraction techniques for fruits: A Review. *Iraqi Journal of Science*, 2425-2430.

He, L., Fang, W., Zhao, G., Wu, Z., Fu, L., Li, R., . . . Dhupia, J. (2022). Fruit yield prediction and estimation in orchards: A state-of-the-art comprehensive review for both direct and indirect methods. *Computers and Electronics in Agriculture,* 195, 106812.

Horn, B. K., & Schunck, B. G. (1981). Determining optical flow. *Artificial intelligence, 17*(1-3), 185-203.

Hung, C., Xu, Z., & Sukkarieh, S. (2014). Feature learning-based approach for WEED classification using high resolution aerial images from a digital camera mounted on a UAV. *Remote Sensing, 6*(12), 12037-12054.

Hur, J., & Roth, S. (2020). Optical flow estimation in the deep learning age. *Modelling Human Motion*, 119-140.

Itakura, K., Narita, Y., Noaki, S., & Hosoi, F. (2021). Automatic pear and apple detection by videos using deep learning and a Kalman filter. *OSA Continuum, 4*(5), 1688.

Jarvinen, T. D., Choi, D., Heinemann, P., & Baugher, T. A. (2018). Multiple object tracking-by-detection for fruit counting on an apple tree canopy. *ASABE Annual International Meeting (p. 1).*

Julier, S. J., & Uhlmann, J. K. (1997). New extension of the Kalman filter to nonlinear systems. *SPIE Proceedings*.

Kalman, R. E. (1960). A new approach to linear filtering and prediction problems. *Journal of Basic Engineering, 82*(1), 35-45.

Kamath, U., Graham, K. L., & Emara, W. (2022). Bidirectional encoder representations from Transformers (BERT). *Transformers for Machine Learning*, 43-70.

Kamilaris, A., & Prenafeta-Boldú, F. X. (2018). Deep learning in agriculture: A survey. *Computers and Electronics in Agriculture*, 147, 70-90.

Kanimozhi, T., & Latha, K. (2015). An integrated approach to region based image retrieval using firefly algorithm and support vector machine. *Neurocomputing*, 151, 1099-1111.

Khalil, Z., & Abdullaev, S. (2021). Neural Network for grain yield predicting based multispectral satellite imagery: Comparative study. *Procedia Computer Science*, 186, 269-278.

Kindermann, R., & Snell, L. J. (1980). Contemporary mathematics. *Providence, RI: American Math. Society*.

Koirala, A., Walsh, K. B., Wang, Z., & McCarthy, C. (2019). Deep learning for real-time fruit detection and orchard fruit load estimation: Benchmarking of 'Mangoyolo'. *Precision Agriculture, 20*(6), 1107-1135.

Krizhevsky, A., Sutskever, I., & Hinton, G. E. (2017). ImageNet classification with deep convolutional neural networks. *Communications of the ACM, 60*(6), 84-90.

Kschischang, F. R., Frey, B. J., & Loeliger, H. A. (2001). Factor graphs and the sum-product algorithm. *IEEE Transactions on Information Theory, 47*(2), 498-519

Kuhn, H. W. (1955). The Hungarian method for the assignment problem. *Naval Research Logistics Quarterly, 2*(1-2), 83-97.

Lakshmi, S., & Sankaranarayanan, D. (2010). A study of edge detection techniques for segmentation computing approaches. *International Journal of Computer Applications, CASCT (1)*, 35-41.

LeCun, Y., Bengio, Y., & Hinton, G. (2015). Deep learning. *Nature, 521*(7553), 436-444.

Lecun, Y., Bottou, L., Bengio, Y., & Haffner, P. (1998). Gradient-based learning applied to document recognition. *Proceedings of the IEEE, 86*(11), 2278-2324.

Li, F., Zhang, Y., Yan, W., Klette, R. (2016) Adaptive and compressive target tracking based on feature point matching. *International Conference on Pattern Recognition* (ICPR), (pp.2734-2739).

Li, G., Suo, R., Zhao, G., Gao, C., Fu, L., Shi, F., . . . Cui, Y. (2022). Real-time detection of kiwifruit flower and Bud simultaneously in orchard using YOLOv4 for robotic pollination. *Computers and Electronics in Agriculture*, 193, 106641.

Li, X., Geng, H., Zhang, L., Peng, S., Xin, Q., Huang, J., . . . Wang, Y. (2022). Improving maize yield prediction at the county level from 2002 to 2015 in China using a novel deep learning approach. *Computers and Electronics in Agriculture*, 202, 107356.

Li, Y., Al-Sarayreh, M., Irie, K., Hackell, D., Bourdot, G., Reis, M. M., & Ghamkhar, K. (2021). Identification of weeds based on hyperspectral imaging and machine learning. *Frontiers in Plant Science*, 11.

Li, Y., Feng, X., Liu, Y., & Han, X. (2021). Apple quality identification and classification by image processing based on convolutional neural networks. *Scientific Reports, 11*(1).

Li, Y., Huang, H., Xie, Q., Yao, L., & Chen, Q. (2018). Research on a surface defect detection algorithm based on MobileNet-SSD. *Applied Sciences, 8*(9), 1678.

Liu, G., Hou, Z., Liu, H., Liu, J., Zhao, W., & Li, K. (2022). TomatoDet: Anchor-free detector for Tomato Detection. *Frontiers in Plant Science*, 13.

Liu, T., Nguyen, M., Yan, W. (2021) Mobile augmented reality using both front and rear cameras for online shopping of sunglasses. *International Symposium on Geometry and Vision.*

Liu, W., Anguelov, D., Erhan, D., Szegedy, C., Reed, S., Fu, C. Y., & Berg, A. C. (2016). SSD: Single shot multibox detector. *ECCV Conference*.

Liu, X., Zhao, D., Jia, W., Ji, W., & Sun, Y. (2019). A detection method for apple fruits based on color and shape features. *IEEE Access*, 7, 67923-67933.

Liu, X., Yan, W. (2022) Depth estimation of traffic scenes from image sequence using deep learning. *Pacific-Rim Symposium on Image and Video Technology*.

Liu, Z., Yan, W., Yang, B. (2018) Image denoising based on a CNN model. *International Conference on Control, Automation and Robotics.*

Lou, H., Duan, X., Guo, J., Liu, H., Gu, J., Bi, L., & Chen, H. (2023). DC-YOLOv8: Small size object detection algorithm based on camera sensor. *Pre-print.*

Lucas, B. D., & Kanade, T. (1981, August). An iterative image registration technique with an application to stereo vision. *International Joint Conference on Artificial Intelligence* (Vol. 2, pp. 674-679).

Luo, Z., Yan, W. Q., & Nguyen, M. (2022). Kayak and sailboat detection based on the improved YOLO with Transformer. *International Conference on Control and Computer Vision*.

Lv, J., Ni, H., Wang, Q., Yang, B., & Xu, L. (2019). A segmentation method of red apple image. *Scientia Horticulturae*, 256, 108615.

Maas, A. L., Hannun, A. Y., & Ng, A. Y. (2013, June). Rectifier nonlinearities improve neural network acoustic models. ICML (Vol. 30, No. 1, p. 3).

Machefer, M., Lemarchand, F., Bonnefond, V., Hitchins, A., & Sidiropoulos, P. (2020). Mask R-CNN refitting strategy for plant counting and sizing in UAV imagery. *Remote Sensing, 12*(18), 3015.

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.

Milioto, A., Lottes, P., & Stachniss, C. (2018). Real-time semantic segmentation of crop and weed for precision agriculture robots leveraging background knowledge in CNNS. *IEEE International Conference on Robotics and Automation (ICRA)*.

Milletari, F., Navab, N., & Ahmadi, S. (2016). V-net: Fully convolutional neural networks for volumetric medical image segmentation. *International Conference on 3D Vision (3DV)*.

Mohanty, S. P., Hughes, D. P., & Salathé, M. (2016). Using deep learning for image-based plant disease detection. *Frontiers in Plant Science*, 7.

Nair, V., & Hinton, G. E. (2010). Rectified linear units improve restricted Boltzmann machines. *International Conference on Machine Learning (ICML-10)* (pp. 807-814).

Nanaa, K., Rizon, M., Rahman, M. N., Ibrahim, Y., & Aziz, A. Z. (2014). Detecting mango fruits by using randomized Hough Transform and backpropagation neural network. *International Conference on Information Visualisation*.

Ngo, T. N., Wu, K., Yang, E., & Lin, T. (2019). A real-time imaging system for multiple honeybee tracking and activity monitoring. *Computers and Electronics in Agriculture*, 163, 104841.

Nguyen, T. T., Vandevoorde, K., Wouters, N., Kayacan, E., De Baerdemaeker, J. G., & Saeys, W. (2016). Detection of red and bicoloured apples on tree with an RGB-D camera. *Biosystems Engineering*, 146, 33-44.

Nguyen, M., Tran, H., Le, R., Yan, W. (2017) A tile-based color picture with hidden QR code for augmented reality and beyond. *ACM Symposium on Virtual Reality Software and Applications.*

Nguyen, M., Le, R., Yan, W. (2017) A personalized stereoscopic 3D gallery with virtual reality technology on smartphone. *International Conference on Image and Vision Computing New Zealand (IVCNZ)*.

Ninomiya, S. (2022). High-throughput field crop phenotyping: Current status and challenges. *Breeding Science, 72*(1), 3-18.

Olaniyi, E. O., Oyedotun, O. K., & Adnan, K. (2016). Intelligent grading system for banana fruit using neural network arbitration. *Journal of Food Process Engineering, 40*(1).

Pan, C., & Yan, W. Q. (2020). Object detection based on saturation of visual perception. *Multimedia Tools and Applications, 79*(27-28), 19925-19944.

Pan, C., Yan, W. (2018) A learning-based positive feedback in salient object detection. *International Conference on Image and Vision Computing New Zealand*.

Pan, C., Liu, J., Yan, W., Zhou, Y. (2021) Salient object detection based on visual perceptual saturation and two-stream hybrid networks. *IEEE Transactions on Image Processing*.

Papadimitriou, C. H., & Steiglitz, K. (1982). Combinatorial Optimization: Algorithms and Complexity. *Prentice Hall*.

Pearl, J. (1998). Bayesian Networks. Los Angeles, CA: Computer Science Dept., *University of California, USA*.

Quan, L., Feng, H., Lv, Y., Wang, Q., Zhang, C., Liu, J., & Yuan, Z. (2019). Maize seedling detection under different growth stages and complex field environments based on an improved Faster R–CNN. *Biosystems Engineering*, 184, 1-23.

Radosavljevic, Z. (2006). A study of a target tracking method using global nearest neighbor algorithm. *Vojnotehnicki Glasnik*, (2), 160-167.

Rahnemoonfar, M., & Sheppard, C. (2017). Deep count: Fruit counting based on deep simulated learning. *Sensors, 17*(4), 905.

Raina, R., Madhavan, A., & Ng, A. Y. (2009). Large-scale deep unsupervised learning using graphics processors. *Annual International Conference on Machine Learning*.

Ramcharan, A., Baranowski, K., McCloskey, P., Ahmed, B., Legg, J., & Hughes, D. P. (2017). Deep learning for image-based cassava disease detection. *Frontiers in Plant Science*, 8.

Redmon, J., & Farhadi, A. (2017). YOLO9000: Better, faster, stronger. *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*.

Redmon, J., Divvala, S., Girshick, R., & Farhadi, A. (2016). You only look once: Unified, real-time object detection. *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*.

Ren, S., He, K., Girshick, R., & Sun, J. (2017). Faster R-CNN: Towards real-time object detection with region proposal networks. *IEEE Transactions on Pattern Analysis and Machine Intelligence, 39*(6), 1137-1149.

Shan, T., & Yan, J. (2021). SCA-Net: A spatial and channel attention network for medical image segmentation. *IEEE Access*, 9, 160926-160937.

Skovsen, S., Dyrmann, M., Mortensen, A., Steen, K., Green, O., Eriksen, J., . . . Karstoft, H. (2017). Estimation of the botanical composition of Clover-Grass Leys from RGB images using data simulation and fully convolutional neural networks. *Sensors, 17*(12), 2930.

Shen, H., Kankanhalli, M., Srinivasan, S., Yan, W. (2004) Mosaic-based view enlargement for moving objects in motion pictures. *IEEE ICME'04*.

Song, Y., Chen, G., & Liu, J. (2019). An improved mask R-CNN with global context modeling for instance segmentation. *Chinese Automation Congress (CAC)*.

Song, Z., Tomasetto, F., Niu, X., Yan, W. Q., Jiang, J., & Li, Y. (2022). Enabling breeding selection for biomass in slash pine using UAV-based imaging. *Plant Phenomics*, 2022.

Sun, Y., Lei, C., Khan, E., Chen, S. S., Tsang, D. C., Ok, Y. S., . . . Li, X. (2017). Nanoscale zero-valent iron for metal/metalloid removal from model hydraulic fracturing wastewater. *Chemosphere*, 176, 315-323.

Sun, Z., Di, L., & Fang, H. (2018). Using long short-term memory recurrent neural network in land cover classification on Landsat and Cropland data layer time series. *International Journal of Remote Sensing, 40*(2), 593-614.

Toda, Y., & Okura, F. (2019). How convolutional neural networks diagnose plant disease. *Plant Phenomics*, 2019, 1-14.

Too, E. C., Yujian, L., Njuki, S., & Yingchun, L. (2019). A comparative study of fine-tuning deep learning models for plant disease identification. *Computers and Electronics in Agriculture*, 161, 272-279.

Urmashev, B., Buribayev, Z., Amirgaliyeva, Z., Ataniyazova, A., Zhassuzak, M., & Turegali, A. (2021). Development of a weed detection system using machine learning and neural network algorithms. *Eastern-European Journal of Enterprise Technologies, 6*(2) 114.

Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., ... & Polosukhin, I. (2017). Attention is all you need. *Advances in Neural Information Processing Systems*, 30.

Vinyals, O., Toshev, A., Bengio, S., & Erhan, D. (2015). Show and tell: A neural image caption generator. *IEEE Conference on Computer Vision and Pattern Recognition (CVPR).*

Wang, J., Kankanhalli, M., Yan, W., Jain, R. (2003) Experiential sampling for video surveillance. *ACM SIGMM International Workshop on Video surveillance* (pp.77-86).

Wan, S., & Goudos, S. (2020). Faster R-CNN for multi-class fruit detection using a robotic vision system. *Computer Networks*, 168, 107036.

Wang, L., & Yan, W. Q. (2021). Tree leaves detection based on deep learning. *Communications in Computer and Information Science*, 26-38.

Wang, Y., Xing, Z., Ma, L., Qu, A., & Xue, J. (2022). Object detection algorithm for lingwu long jujubes based on the improved SSD. *Agriculture, 12*(9), 1456.

Wojke, N., Bewley, A., & Paulus, D. (2017). Simple online and realtime tracking with a Deep Association metric. *IEEE International Conference on Image Processing (ICIP)*.

Woo, S., Park, J., Lee, J., & Kweon, I. S. (2018). CBAM: Convolutional block attention module. *ECCV 2018*, 3-19.

Xia, Y., Nguyen, M., & Yan, W. Q. (2023). A real-time kiwifruit detection based on improved YOLOv7. *Image and Vision Computing*, 48-61.

Xia, Y., Nguyen, M., Yan, W. (2023) Kiwifruit counting using KiwiDetector and KiwiTracker. IntelliSys conference.

Xiao, B., Nguyen, M., Yan, W. (2021) Apple ripeness identification using deep learning. *International Symposium on Geometry and Vision, 53-67*.

Xiao, B., Nguyen, M., Yan, W. (2023) Apple ripeness identification from digital images using transformers. *Multimedia Tools and Applications*, Springer Science and Business Media LLC.

Xiao, B., Nguyen, M., Yan, W. (2023) Fruit ripeness identification using transformers. *Applied Intelligence*, Springer Science and Business Media LLC.

Yan, W. Q. (2019). Introduction to Intelligent Surveillance. *Springer International Publishing*.

Yan, W. Q. (2021). Computational Methods for Deep Learning: Theoretic, practice and applications. *Springer*.

Yan, W. Q. (2023). Computational Methods for Deep Learning – Theory, Algorithms, and Implementations (2nd Edition). *Springer*.

Yan, W., Wang, J., Kankanhalli, M. (2005) Automatic video logo detection and removal. *Multimedia Systems* 10 (5), 379-391.

Yang, Q., Chen, C., Dai, J., Xun, Y., & Bao, G. (2020). Tracking and recognition algorithm for a robot harvesting oscillating apples. *International Journal of Agricultural and Biological Engineering, 13*(5), 163-170.

Yang, T. T., Zhou, S. Y., Xu, A. J., & Yin, J. X. (2020). A method for tree image segmentation combined adaptive mean shifting with image abstraction. *Journal of Information Processing Systems, 16*(6), 1424-1436.

Yosinski, J., Clune, J., Bengio, Y., & Lipson, H. (2014). How transferable are features in deep neural networks? *Advances in Neural Information Processing Systems*, 27.

Yu, Y., Zhang, K., Zhang, D., Yang, L., & Cui, T. (2019). Optimized Faster R-CNN for fruit detection of strawberry harvesting robot. *ASABE Annual International Meeting.*

Zhang, Y., Ren, W., Zhang, Z., Jia, Z., Wang, L., & Tan, T. (2022). Focal and efficient IOU loss for accurate bounding box regression. *Neurocomputing*, 506, 146-157.

Zhao, K. (2021) *Fruit Detection Using CenterNet*. Master's Thesis, Auckland University of Technology, New Zealand.

Zhao, K., Yan, W. (2021) Fruit detection from digital images using CenterNet. *International Symposium on Geometry and Vision, 313-326.*

Zheng, K., Yan, Q., Nand, P. (2017) Video dynamics detection using deep neural networks. *IEEE Transactions on Emerging Topics in Computational Intelligence*.

Zheng, Z., Wang, P., Liu, W., Li, J., Ye, R., & Ren, D. (2020). Distance-IOU loss: Faster and better learning for bounding box regression. *AAAI Conference on Artificial Intelligence, 34*(07), 12993-13000.

Zhu, X., Cheng, D., Zhang, Z., Lin, S., & Dai, J. (2019). An empirical study of spatial attention mechanisms in deep networks. *IEEE/CVF International Conference on Computer Vision (ICCV)*.