Visual Object Detection for Tree Leaves Based on Deep Learning

Lei Wang

A project report submitted to the Auckland University of Technology

in partial fulfillment of the requirements for the degree of

Master of Computer and Information Sciences (MCIS)

2020

School of Engineering, Computer & Mathematical Sciences

Abstract

New Zealand is an affluent country in plant resources, thus it has great value and significance to carry out visual object detection for tree leaves through using digital images. In this project, five local tree leaves are collected as our dataset, and two models, namely, Faster R-CNN and YOLOv5, representing two-stage and one-stage algorithms, are respectively employed to conduct object detection test for tree leaves. Our results show that YOLOv5 model is obviously superior to the Faster R-CNN in both training speed and detection speed. The difference between these two methods is not significant in comparison of mAP, but the YOLOv5 model is a bit superior. We conclude that YOLOv5 method has excellent speed and accuracy.

Keywords: YOLOv5, Faster R-CNN, Leaf Detection

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.

Signature: Lei Wang Date: 23 October 2020

Acknowledgment

I would also like to express my deepest gratitude to my supervisor Dr Wei Qi Yan. In this study, he not only provided me with professional knowledge and prudential guidance, but also assisted me to enrich my learning experience. I believe I could hardly complete my study without Dr Yan’s supervision and instructions. My most sincere thanks are for him.

Auckland, New Zealand

October 2020

Chapter 1
Introduction

This chapter is composed of five parts. In the first part, we introduce the background and motivations of this project, the second part includes the research question we are interested, followed by the contributions, objectives, and structure of this report.

1.1 Background and Motivation

New Zealand is one of the most picturesque countries in the world. The forest resources are very abundant. There are a variety of trees growing everywhere, many of which are unique to New Zealand (McGlone, Buitenwerf & Richardson,2016). The research on leaf classification has been done for decades, and plant image recognition has become an interdisciplinary subject in the field of plant taxonomy and computer vision (Sun, Liu, Wang, & Zhang,2017)。

Since 2006, a large number of papers on deep neural networks have been published, especially in 2012, when Hinton research group participated in the ImageNet (Deng et al., 2009) image recognition competition for the first time and won the first prize through AlexNet (Alom et al., 2018). Since then, neural networks have attracted extensive attention. Deep learning applies a multi-layer computing model to learn abstract data representations and discover complex structures in big data. At present, this technology has been successfully employed to a variety of classification problems including computer vision.

Supervised machine learning and the use of neural networks are fundamental to the application of machine learning to many biological problems. For example, deep learning has recently attained impressive performance on a variety of prediction tasks, such as species identification（Soltis, Nelson, Zare & Meineke, 2020).

The research of plant automatic taxonomy has made great progress, but with the continuous expansion of its application in various fields, it still can not fully meet the needs of reality (Goeau, Bonne & Joly, 2016).

Image classification technology is a traditional basic and important research method in the field of computer vision. However, compared with traditional image classification, object detection is obviously more in line with the practical needs, because it is often impossible to have only one object in a certain scene in reality. The requirement of object detection becomes more complex, which is further based on image classification. All targets and the positions in the image need to be predicted to provide a complete and correct understanding of the image (Kapur, 2017).

Object detection is one of the hot directions in computer vision and digital image processing, it is to solve one of the basic tasks of computer vision field, which is widely used in robot navigation, intelligent video surveillance, industrial testing, and many other fields, through computer vision to reduce the consumption of the human capital, has important practical significance (Zou, Shi, Guo & Ye, 2019).Therefore, visual object detection has become a research hotspot in theory and application in recent years. It is an important branch of image processing and computer vision, as well as the core part of intelligent monitoring system. Due to the extensive application of deep learning, object detection algorithms have been explored rapidly. However, visual objects often have distinct poses and are often blocked. Considering the complexity and diversity of the scene, this is still a challenging topic, and there are many potential and space to be improved.

Object detection algorithms can be roughly divided into two categories. The first one is R-CNN family algorithm based on regional proposal, whose representative network is R-CNN (Girshick, Donahue, Darrell & Malik, 2014), Fast R-CNN (Girshick, 2015), SPPNet (He, Zhang, Ren & Sun, 2015), Faster R-CNN (Ren, He, Girshick & Sun, 2015), FPN(Lin et al., 2017), and Mask-RCNN (He, Gkioxari, Dollar & Girshick, 2017).They are two-stage, which requires the algorithm to generate the candidate box of the object, namely the position of the object, and then do the classification and regression of the candidate box.

One stage detection algorithm, which does not need the region proposal phase, can directly generate the category probability and position coordinate value of the object. Its representative network is the YOLO series, including YOLO (Redmon, Divvala, Girshick & Farhadi, 2016), YOLOv2 (Redmon & Farhadi, 2017), YOLOv3 (Redmon & Farhadi, 2018),YOLOv4(Bochkovskiy, Wang & Liao, 2020), YOLOv5, SSD (Liu et al., 2016) and RetinaNet (Lin, Goyal, Girshick, He & Dollar, 2017).

In the past few years, most of the high-precision object detection algorithms (such as R-CNN, Fast R-CNN, Faster R-CNN) often fail to meet the real-time requirements of the industry for visual object detection due to its slow speed. At present, one stage algorithm represented by YOLO came out and won wide recognition with its excellent performance of speed and accuracy. Up to September 2020, YOLO algorithm has gone through five versions of iteration, and has made great progress in speed and accuracy.

There have been a lot of research work on tree leaf classification. Compared with traditional image classification, visual object detection though digital images is obviously more in line with practical needs, because it is often impossible to have only one object in a certain scene in reality. The requirements for object detection are more complex, requiring algorithms not only to verify what an object is, but also to determine where the object is in the image. There are relatively little research work on leaf object detection, but more focus on automatic driving, video monitoring, mechanical processing, intelligent robot and other fields. Therefore, in this project, we endeavor to explore the performance of deep learning frameworks for tree leaf detection.

On the basis of extensive research on object detection algorithms over the past decades, this project adopts two representative object detection algorithms based on deep learning, namely, YOLOv5 algorithm based on one-stage idea and Faster R-CNN algorithm based on two-stage idea, and applies them to the leaf data set collected locally in New Zealand.

1.2 Research Questions

The research questions of the present report are:

(1) How to implement two object detection models that can recognize leaves and utilize the local leaf data set in New Zealand to detect the effect?

(2) Throughout comparative analysis, how to get the different characteristics and advantages and disadvantages of the models?

Therefore, the core problem to be studied in this report is to compare and analyze the performance difference of the one-stage algorithm model represented by YOLOv5 and the two-stage algorithm model represented by Faster R-CNN in identifying tree leaves, and analyze the advantages and disadvantages of the two models respectively.

1.3 Contribution

Image classification is to take the image as input, and then the probability of belonging to a certain category is output, so as to determine which category the specified image belongs to. Object detection outputs the coordinate position and category of the object. A decimal between 0 and 1.00 at the top of the rectangle in the image represents the confidence that the object is a class, and the probability that the current class is a real class. Usually before training, a threshold (usually 0.5 by default) is given and used to filter out incorrect rectangular boxes (Zhao, Zheng, Xu & Wu, 2019) .

There has been a great deal of literature on leaf classification for decades. However, with the emergence of various new models based on deep learning, computer vision has gradually shifted from simple image classification to more complex object detection, object tracking, semantic segmentation and instance segmentation. In view of the three-dimensional shape of leaves in object detection, leaf images observed from different angles of view may be completely different, so object detection is much more difficult than image classification. In addition, different from traditional leaf recognition research, this project is not only to identify and classify single leaf pictures, but also to realize the recognition of multiple types of leaves in videos.

Compared with image classification, object detection is to classify and locate the objects with variable number. For the object detection task, there may be some problems that make the object detection problem more difficult. For example, (1) there may be multiple objects of different classes and the number of objects is uncertain. (2) Object scale, for example, visual objects of different sizes. (3) External environment interference, such as the change of illumination, the existence of occlusion and the quality of pictures. The existence of these problems makes object detection worth studying and discussing.

In this project, we innovatively trained two different models representing one-stage and two-stage, respectively, using native leaves as the dataset. The advantages and disadvantages of different models were compared and analyzed through actual detection of leaves in the videos.

1.4 Objectives of This Report

This report analyzes the development history and representative algorithms of object detection from two development paths of one-stage and two-stage. Two algorithms representing different algorithm ideas are selected respectively, and leaves collected by ourselves are taken as detection objects for testing. Finally, by comparing the differences of performance indexes between different models, the model is proved to be more suitable for the specific task of leaf object detection.

1.5 Structure of This Report

The structure of this report is described as follows:

- In Chapter 2, we will conduct a literature review and discuss the relevant studies of leaf classification and object detection. Meanwhile, the development history and characteristics of the model Faster R-CNN and YOLO used in this experiment will be introduced in detail.
- In Chapter 3, we will introduce the research methods. Experimental design will be presented in this Chapter, including the deployment environment, data preparation, training model. We also look at the working principles of Faster R-CNN and YOLO in detail. In addition, we will introduce the various performance indicators and their meanings used to compare their performance differences.
- In Chapter 4, we will compare the test results of two different models through figures and tables. We will compare the strengths and weaknesses as well as limitations of these two models. Resultant comparisons will be presented in this chapter.
- In Chapter 5, we will summarize and analyze the experimental results.

- We will draw the conclusion and state our future work in Chapter 6.

Chapter 2
Literature Review

This chapter will start from leaf classification to develop the methods of visual object detection. The development of object detection algorithm is grouped into two different ideas, one-stage and two-stage. The former has a clear advantage in speed, but the latter tends to be more accurate. In the process of development, they are constantly making up for the shortcomings. Along the development of these two different ideas, new and more advanced algorithms emerging from each of them are introduced in this chapter.

2.1 Introduction

Plants play an important role in human life. It is of great significance to set up a database for all kinds of plants, and to use artificial intelligence to identify plants automatically. Research on the classification of plants has been going on for decades. In plant species identification, leaves play an important role compared with other parts such as flowers, seeds and stems (Backes, Casanova & Bruno, 2009). With the application of deep learning technology in the field of object detection, computer vision has gradually developed from the most basic image classification to more complex fields such as object detection, semantic segmentation and motion detection. Deep learning has achieved remarkable success in computer vision tasks. They can achieve optimal performance in various tasks such as image classification, object detection or semantic segmentation (Schmarje, Santarossa, Schroder & Koch, 2020).

There are a lot of literatures on leaf classification, and the research on object detection is in full swing in recent years. But there have been relatively few previous studies of leaf detection.

This chapter will begin with the classification of leaves. When discussing the methods and history of object detection, we will start from the traditional methods, then to the development of object detection methods based on deep learning and their latest achievements, including one-stage and two-stage ideas and their respective representative algorithms.

2.2 The Development of Leaf classification

The image classification of leaves also experienced two stages: traditional method and deep learning method.

The traditional steps of image classification include preprocessing, feature extraction and classification using machine learning classifier. Feature extraction includes the extraction of shape features, texture features and color features from plant leaf images (Lee, Kim & Hong, 2015).

As early as 1993, Guyer et al extracted 17 leaf shape features and classified 40 species of plants (Guyer, Miles, Gaultney, & Schreiber, 1993).In 1998, Im et al. used layered polygons to approximate leaf shapes and applied this method to the classification of a variety of maple trees (Im, Nishida, & Kunii,1998).

In 2000, Oide and Ninomiya used leaf shapes as inputs to neural networks and classified soybean leaves using a Hopfield network and a simple perceptor (Oide, & Ninomiya 2000).

In 2001, Soderkvist used the geometric features of leaves to classify 15 Swedish trees using BP feed-forward neural network. The data set of 15 leaves used in the experiment later became a standard data set -- Swedish leaves data set. Many researchers used this data to test their algorithm.

Peng and Huang used probabilistic neural networks as classifiers to recognize plant leaf images, which had higher recognition accuracy than BP neural networks (Peng & Huang, 2008).

Sun and his colleagues presented the first data set of plant images collected by mobile phones in natural Settings (BJFU100 data set), which includes images of 100 ornamental plants on the campus of Beijing Forestry University. In addition, they designed a 26-layer deep learning model consisting of 8 residual construction primitives. The recognition rate of this model on BJFU100 data set reaches 91.78%, indicating that deep learning is a promising technology for smart forestry (Sun, Liu, Wang & Zhang, 2017).

In 2012, deep convolutional neural network showed excellent performance in ilSVRC-2012 large-scale image classification task. The model received training of more than one million images, and the test error rate of the top 5 in the 1000 categories reached 15.3%. It almost halves the error rate of the best competing methods. This success led to a revolution in computer vision. This advance improves the feasibility of deep learning applications to solve complex practical problems (Krizhevsky, Sutskever, & Hinton, 2012).

In 2016, Zhang and Huai used CNN to identify the leaves of a self-expanding dataset based on PlantNet. The accuracy of SVM classifier and Softmax classifier was 91.11% and 90.90% respectively under simple background and 31.78% and 34.38% respectively under complex background (Zhang, & Huai, 2016) .

In 2017, three deep learning networks GoogLeNet, AlexNet and VGGNet were used to identify plant species on LifeCLEF 2015 data set (Joly et al., 2015), with an overall accuracy of 80% for the best model (Ghazi, Yanikoglu & Aptoula, 2017).

2.3 The History of Traditional Object Detection

Image classification and object detection are important research methods in computer vision. These technologies help machines understand and recognize real-time objects and their environments. In essence, object detection is also an image classification technique. In addition to classification, this technique can also identify the location of object instances from a large number of predefined categories in natural images. Object detection is one of the most basic and challenging tasks in computer vision (Zou, Shi, Guo & Ye, 2019).

From the past nearly two decades of development, the object detection algorithm of natural images can be roughly divided into the period based on traditional manual features before 2013 and the period based on deep learning after 2013.In terms of technology development, object detection has gone through many milestones such as bounding box regression, multi-references boxes (anchors), hard example mining and focusing, and multi-scale and multi-port detection (Zou, Shi, Guo & Ye, 2019).

Early object detection algorithms are mostly based on manual features. Due to the lack of effective image feature expression methods before the birth of deep learning, people have to design more diversified detection algorithms to make up for the defects of manual feature expression ability.

In 2001, Viola and Jones published a cross-era paper on CVPR, and later generations called the face detection algorithm in the paper as Viola-Jones (VJ) detector (Viola & Jones, 2001).VJ detector realizes the real-time face detection for the first time with very limited computing resources, and its speed is tens or even hundreds of times faster than the detection algorithm in the same period, which greatly promotes the commercialization process of face detection application. The idea of VJ detectors has profoundly influenced the development of the object detection field for at least 10 years. The VJ detector adopts the most traditional and conservative object detection method -- sliding window detection, which is to traverse every scale and every pixel position in the image and judge whether the current window is a face target one by one. The cost of this approach is huge computation (Viola, & Jones, 2004).

Another representative algorithm is the HOG pedestrian detector (Dala & Triggs, 2005). HOG feature was first proposed to solve the pedestrian detection problem. HOG detector follows the original multi-scale pyramid and sliding window idea for detection. In order to detect objects of different sizes, the size of detector window is usually fixed, and the image is scaled successively to build a multi-scale image pyramid. In order to give consideration to speed and performance, the classifier adopted by HOG detector is usually linear classifier or cascade decision classifier (Zhu, Yeh, Cheng & Avidan, 2006).

In the postprocessing of the algorithm, the DPM algorithm adopts the two methods of bounding box regression and context information integration to further improve the detection accuracy (Girshick, Felzenszwalb & Mcallester, 2011).Among them, the main function of bounding box regression is to integrate the base filter and the bounding box corresponding to the component filter and obtain the final precise bounding box coordinates by using linear least square regression. The purpose of context information integration is to use global information to readjust the detection results. In essence, context information reflects the joint prior probability density distribution of various categories of targets in the image, that is, which categories of targets are likely to appear at the same time, and which categories of targets are unlikely to appear at the same time (Girshick, 2012).

Although the object detection model based on deep learning has far surpassed DPM in accuracy in recent years, many ideas in DPM are still important today, such as mixed model, hard sample mining, bounding box regression, and the use of context information.

Up to now, these methods have deeply influenced the development of object detection field.

2.4 Visual Object Detection Based on Deep Learning

In 2012, a variety of artificial neural network design methods have emerged to replace the traditional neural network.

After the CNN achieved great success in the 2012 Imagenet classification task (Hinton, Krizhevsky & Sutskever, 2012), Girshick and others seized the opportunity to break the deadlock and first proposed the Region Based Convolutional Neural Networks (R-CNN) in 2014. Since then, the field of object detection has developed at an unprecedented speed (Girshick, Donahue, Darrell & Malik, 2014).

With the deepening of CNN layers, the network's abstract ability, anti translation ability and anti scale change ability become stronger and stronger. For the image classification task, this is beneficial, but for the detection task, it brings another problem: the accurate position of the object bounding box is becoming more and more difficult to obtain. Therefore, if we want the detection algorithm to obtain stronger translation invariance and scale invariance, we must sacrifice the sensitivity of features in the position and scale changes of the object bounding box to a certain extent. On the contrary, if we want to get more accurate bounding box localization results, we must make some compromises on translation invariance and scale invariance. (Dai & R-FCN, 2016)。 This forced people to give up the detection scheme based on feature map and sliding window, turning their attention to find more accurate object candidate detection algorithm.

At present, deep learning methods in the field of object detection are mainly divided into two development directions: two-stage based and one-stage based. The former refers to the algorithm generating a series of candidate frames as samples, and then classifying the samples through CNN; the latter does not need to generate candidate frames, and directly transforms the problem of object border location into regression problem. The performance of the two methods is also different. The former is superior in detection accuracy and positioning accuracy, while the latter is superior in algorithm speed. The development of these two ideas and their representative algorithms are described below.

2.4.1. Two Stages Algorithm

In the early development process of deep learning technology, researches were mainly carried out around classification problems. This is because the unique structural output of neural networks combines probability statistics and classification problems to provide an intuitive and easy way of thinking. Although many researchers are also working on integrating other areas such as object detection and deep learning, they have not achieved much success. This situation was not solved until the emergence of R-CNN algorithm.

2.4.1.1 R-CNN

R-CNN algorithm was proposed for the first time in 2014, and its algorithm structure also became the classic structure of subsequent two-stage (Girshick, Donahue, Darrell & Malik, 2014).The R-CNN algorithm evaluates the feature similarity of adjacent image subblocks by Selective Search algorithm. By scoring the similar image regions after the merging, the candidate box of the region of interest is selected as the sample and input into the CNN structure. The corresponding feature vector is formed by the positive and negative sample features composed of network learning candidate box and calibration box. Then support vector machine design classifier to classify the feature vectors.

Finally, the localization of object detection is achieved by border regression operation. Although R-CNN algorithm has achieved 50% performance improvement compared with the traditional object detection algorithm, it also has defects: the positive and negative sample candidate areas of the training network are generated by the traditional algorithm, which limits the speed of the algorithm. In addition, CNN does feature extraction for each generated candidate region respectively, which will lead to a large number of repeated operations and further restrict the performance of the algorithm.

2.4.1.2 SPP-Net

In 2015, SPP-NET algorithm was proposed for the repetitive operation of convolutional neural networks (He, Zhang, Ren & Sun, 2015).This algorithm adds a space pyramid pooling structure between the convolutive layer and the full connection layer, and optimizes the method of R-CNN algorithm to cut and scale each candidate region before the input of CNN to make the image sub-block size consistent. This spatial pyramid pooling structure effectively avoids the problems of image object clipping and shape distortion caused by image region clipping and scale operation in R-CNN algorithm.

More importantly, it solves the problem of extracting repeated features from images by CNN, thus greatly improving the speed of generating candidate boxes and saving calculation cost. However, just like R-CNN algorithm, when the image size of training data is inconsistent, the ROI of the candidate box will be greatly enhanced, and BP cannot be used to efficiently update the weight.

2.4.1.3 Fast R-CNN

Girshick proposed a Fast R-CNN algorithm in 2015 in order to improve SPP-NET algorithm (Girshick, 2015).This algorithm designs a pooling layer structure of ROI pooling, which effectively solves the operation that R-CNN algorithm must crop and scale the image areas to the same size. The idea of multi-task loss function is proposed. Gradient can be transmitted directly through ROI Pooling layer. But it still does not get rid of the problem of generating positive and negative sample candidate box in the selective search algorithm.

2.4.1.4 Faster R-CNN

In order to solve the defects of Fast R-CNN algorithm, Faster R-CNN algorithm was proposed in 2015 (Ren, He, Girshick & Sun, 2015). Region Proposal Network (RPN) that assists in generating samples was designed. Its advantage is that the whole network process can share the feature information extracted by CNN, which saves the calculation cost, solves the problem that Fast R-CNN algorithm is slow in generating positive and negative sample candidate boxes, and avoids the decrease of algorithm accuracy caused by too much extraction of candidate boxes. RPN network can generate multi-size candidate boxes in the convolution feature map of fixed size, resulting in the inconsistency between the size of the variable object and the fixed receptive field. This is a shortcoming of the Faster R-CNN algorithm.

2.4.1.5 MR-CNN

The MR-CNN algorithm was proposed by Gidaris and Komodakis in 2015.The algorithm decomposed the detection problem into classification and location problem (Gidaris & Komodakis, 2015).

The classification problem is composed of Multi-Region CNN Model and Semantic Segmentation-Aware CNN Model. The candidate box of the former is obtained by Selective Search. For each sample area, 10 areas are extracted separately and then spliced, which forces the network to capture different aspects of the object. After the entire image is input into the Activation Maps module, it is output through a series of convolution operation feature map. In this part of the network, various classic network structures can be used, such as AlexNet, VGG16, etc.

As for positioning, the algorithm adopts three sample Bounding correction methods for precise positioning, namely, Bbox Regression, Iterative localization and Bounding box voting (Liu, Du, Tian & Wen, 2019).

2.4.1.6 HyperNet

HyperNet algorithm is a variant algorithm with excellent performance proposed by Tsinghua University in 2016.The main improvement of HyperNet lies in the collection of multi-layer Feature maps to obtain multi-scale Hyper Feature, which has the advantages of multi-level abstraction, appropriate resolution and calculation timeliness. Compared with Faster R-CNN, HyperNet is better at dealing with small objects. It has more advantages in high IOU (Kong, Yao, Chen & Sun, 2016).

2.4.1.7 CRAFT

The first stage of R-CNN series algorithm is to generate object proposals, and the second stage is to classify the object proposals. In 2016, the craft algorithm proposed by the Institute of automation of Chinese Academy of Sciences improved the two stages of fast R-CNN (Yang, Yan, Lei & Li, 2016). For the generated object proposals stage, a binary fast R-CNN classifier is added after the RPN to further filter the RPN generated proposals, leaving some high-quality proposals; for the object proposals classification in the second stage, the original classifier is cascaded with N binary classifiers (excluding background class) for more precise object detection.

2.4.1.8 R-FCN

With the emergence of full convolutional network, R-FCN algorithm was proposed in 2016 (Dai, Li, He & Sun, 2016). R-FCN is improved on the framework of Faster R-CNN. This algorithm proposes the idea of Position-Sensitive Score Maps to solve the position-sensitivity problem of object detection. It makes full use of full convolutional network to reduce the total computation, which enables feature sharing to be realized in the whole network. However, it does not take into account the global information and semantic information of the regional proposal.

2.4.1.9 MS-CNN

MS-CNN was proposed by Cai in 2016.The basic idea is to propose a multi-scale convolutional neural network. In view of the different advantages of feature maps at different levels, for example, feature maps at lower levels are relatively good at detecting small objects due to their small sensing fields. However, the higher level can ignore a lot of noise because it takes into account the information of large perception field, so it is more accurate to detect large objects. Considering this feature, the author designed detectors of different scales for different layers on the feature map for the first time. At the same time, the deconvolution layer of feature map is used to replace the upsampling of input image, which improves the speed and accuracy (Cai, Fan, Feris & Vasconcelos, 2016).

2.4.1.10 PVANet

At the end of 2016, the Intel Graphics technology team proposed a lightweight network PVANet, achieving remarkable results (Kim, Hong, Roh, Cheon & Park, 2016). Network is based on basic design principles: more layers with less channels. In addition, the author uses “c. ReLU” and Inception structure to reduce the redundancy of the network.

2.4.1.11 FPN

In 2017, Lin proposed the FPN algorithm, which uses feature maps of different layers to predict objects of different sizes (Ghiasi, Lin & Le, 2019). Most of the original object detection algorithms only use the deep features for prediction, and the low-level feature semantic information is less, but the object position information is accurate.

The high-level feature semantic information is rich, but the object location is rough. In addition, although some algorithms adopt multi-scale feature fusion, they generally adopt the fused features for prediction, while the difference of FPN algorithm lies in that the prediction is carried out independently at different feature layers, and the deep features are used for up-sampling and low-level features for fusion.

2.4.1.12 Mask R-CNN

Mask R-CNN algorithm was proposed by Kaiming He, which achieved excellent recognition effect (He, Gkioxari, Dollar & Girshick, 2017). Mask R-CNN is an extended form of Faster R-CNN. The original idea of Mask R - CNN is to add another branch to Faster R-CNN to add an output, namely object Mask, which means that the original two tasks (classification + regression) will be changed to three tasks (classification + regression + segmentation).Combining a binary Mask with the classification and bounding box from Faster R-CNN produces a surprisingly precise image separation effect.

2.4.1.13 Fast R-CNN

The Fast R-CNN algorithm was proposed by Carnegie Mellon University in 2017, which introduced Generative Adversarial Networks (GAN) into object detection problems, mainly focusing on solving occlusion and deformation problems (Wang, Shrivastava & Gupta, 2017). The author designed two GAN: ASDN and ASTN, corresponding to occlusion and deformation respectively. ASDN and ASTN provide two kinds of different changes. By combining these two kinds of deformation (ASDN output is the input of ASTN), the detector can be trained more robust.

2.4.1.14 CoupleNet

In view of the problem that R-FCN algorithm fails to consider the global information and semantic information of regional proposal, The Institute of Automation of the Chinese

Academy of Sciences proposed CoupleNet algorithm in 2017, which introduced the global and semantic information of proposal on the basis of the original R-FCN, and improved the detection accuracy by combining local, global and semantic information (Zhu et al., 2017).

2.4.1.15 MegDet

Research on CNN-based object detection has been making continuous progress. From R-CNN to Fast/Faster R-CNN and then Mask R-CNN, the main improvement points are all in the new network architecture, new paradigm or new loss function design. However, the key factor of mini-batch size in the training has not been fully studied. This also makes it impossible for the existing deep learning framework to train the object detection model of large Mini-batch, while the object detection algorithm of small Mini-batch often introduces unstable gradients, inaccurate BN layer statistics, imbalance of positive and negative sample proportion and excessively long training time. Therefore, In December 2017, Peng proposed MegDet, a large Mini-batch object detection algorithm (Peng et al.,2018).

The MegDet algorithm can use a training network much larger than the previous Mini-batch size (for example, increasing from 16 to 256), which can also efficiently utilize the combined training of multiple GPUs (up to 128 were used in the experiment of the paper) and greatly reduce the training time, such as from 33 hours to just 4 hours. At the same time, the algorithm can achieve higher accuracy.

2.4.1.16 Light-Head R-CNN

In 2017, light-head RCNN was proposed, mainly discussing how R-CNN balances accuracy and speed in object detection (Li et al., 2017).Light Head R-CNN is the combination of Faster R-CNN and R-FCN.The detector can achieve the optimal tradeoff between speed and accuracy.If it is based on the ResNet101 network, it performs better than Mask RCNN and RetinaNet.

2.4.2. One Stage Algorithm

In view of the existence of RPN structure, the two-stage method represented by R-CNN algorithm, although the detection accuracy is getting higher and higher, its speed encounters a bottleneck, which makes it difficult to meet the real-time requirements of some scenes. Therefore, a one-stage object detection algorithm based on regression method appears. The one-stage algorithm can give the category and location information directly through the trunk network without using the RPN network. This algorithm is faster. However, the accuracy is slightly lower than the two-stage object detection algorithm.

2.4.2.1 OverFeat

OverFeat algorithm was proposed by Yann Lecun's team at New York University in 2013. The algorithm uses sliding Windows and rule blocks to generate candidate boxes, and then uses multi-scale sliding Windows to increase detection results to solve the problem of complex shapes and different sizes of image objects. Finally, it uses convolutional neural network and regression model to categorize and locate objects (Sermanet et al., 2013).

OverFeat algorithm makes full use of the feature extraction function of CNN. The features extracted in the classification process are also used for various tasks such as positioning and detection. Different tasks can be realized only by changing the last few layers of the network, instead of training the parameters of the whole network from the beginning. This algorithm solved the three computer vision tasks of classification, positioning and detection together for the first time, and won the champion of ILSVRC2013 Task 3 (classification + positioning) in the same year, but it was soon replaced by R-CNN algorithm in the same period.

2.4.2.2 G-CNN

In 2016, Najibi et al. from the University of Maryland proposed the G-CNN algorithm. G-CNN is a kind of object detection algorithm independent of proposal algorithms. The G-CNN algorithm models the object detection problem as finding a path from a fixed grid to a dense object box. This strategy eliminates the generation stage of candidate boxes and reduces the number of candidate boxes that need to be processed, making object detection faster (Najibi, Rastegari & Davis, 2016).

2.4.2.3 YOLO

In 2015, Yolo algorithm proposed by Joseph Redmon of Washington University inherits the idea of regression in overfeat algorithm, and its speed can reach 45 frames per second.， YOLO algorithm is based on the global information of the image to make predictions. Its overall structure is simple. The input image is reconstructed to a fixed size of 448×448 pixels, and the image is divided into 7×7 grid area. The feature training is extracted by CNN to directly predict the border coordinates in each grid and the confidence of each category.

P-ReLU activation function is adopted during the training. However, it also has problems such as inaccurate positioning and unsatisfactory recall rate, poor detection effect for very close objects and very small objects, and relatively weak generalization ability (Redmon, Divvala, Girshick & Farhadi, 2016).

After improvement, YOLOv2 (Redmon & Farhadi, 2017) and YOLOv3 (Redmon & Farhadi, 2018) algorithms were proposed on CVPR 2017 and were nominated for the best paper, focusing on solving the poor recall rate and positioning accuracy. It uses Darknet-19 as a feature extraction network and adds Batch Normalization as pretreatment. The original YOLO uses the full connection layer to directly predict the coordinates of the bounding box, while YOLOv2 refers to the idea of Faster R-CNN, introduces anchor mechanism, and calculates a better anchor template in the training set through k-means clustering.

The operation of Anchor Boxes is used in the convolutional layer to improve the prediction of the candidate box. Meanwhile, the positioning method with strong constraints is adopted to greatly improve the recall rate of the algorithm. Combining with the image fine grain feature, the shallow feature and the deep feature are connected, which is helpful to the detection of the small size target.

In 2020, Alexey Bochkovskiy released YOLO V4 (Bochkovskiy, Wang & Liao, H. Y. M. 2020).YOLO V4 is a significant update to the YOLO family, with an increase in average accuracy (AP) and frame rate accuracy (FPS) of 10% and 12%, respectively, on COCO data sets. While computer vision practitioners are working on YOLO V4, On June 25, 2020, Ultralytics released YOLOV5 on Github, which has the same performance as YOLO V4 and is faster at reasoning. YOLO V5 does perform very well in object detection, especially the reasoning speed of YOLO V5s model 140FPS is amazing. The authors of YOLO V5 have yet to publish a paper.

2.4.2.4 SSD

Aiming at the poor positioning accuracy of the initial YOLO v1 algorithm, Liu et al proposed SSD algorithm in 2016 and combined the regression idea of YOLO with the Anchor box mechanism of Faster R-CNN (Liu et al., 2016). It adopts the border regression of local feature map of multi-scale region in every position of the whole image, which keeps the fast speed of YOLO algorithm and ensures that the border positioning effect is similar to that of Faster R-CNN. However, due to its use of multi-level feature classification, it is difficult to detect small objects, and the receptive field of the last convolutional layer becomes large, making the features of small objects not obvious.

2.4.2.5 R-SSD

Traditional SSD has two defects: (1) feature maps of different layers are used as independent input of classification network, so it is easy for the same object to be detected by boxes of different sizes simultaneously; (2) The detection effect of small size objects is relatively poor.

In 2017, Seoul National University proposed R-SSD algorithm to solve the above two defects (Jeong, Park & Kwak, 2017).On the one hand, it uses the classification network to add feature map connections between different layers to reduce the appearance of repetitive boxes. On the other hand, increase the number of feature map in Feature Pyramid so that it can detect more small size objects. Although a little slower than the traditional SSD algorithm, mAP is higher. The author did not replace the original VGG main network with ResNet, but improved the effect of the original SSD algorithm by improving the feature fusion method to make full use of the features.

2.4.2.6 DSSD

In order to solve the problem that SSD algorithm is difficult to detect small objects, Fu et al proposed DSSD algorithm in 2017. This algorithm changes the basic network of SSD algorithm from VGG-16 to ResNet-101, thus enhancing the ability of network feature extraction (Fu, Liu, Ranga, Tyagi & Berg, 2017).

DSSD algorithm has two special structures: Prediction module and Deconvolution module. The former takes advantage of improving the performance of each subtask to improve accuracy and prevent gradients from flowing directly into the ResNet main network. The latter has added three Batch Normalization layers and three 3×3 convolution layers, among which the convolutional layer ACTS as a buffer to prevent too severe gradient effect on the main network and ensure the stability of the network. One of the biggest improvements of DSSD compared with SSD is that DSSD has greatly improved the detection degree of small objects. However, the detection speed of DSSD is much slower than that of SSD, which is largely due to the deep introduction of RESnet-101.

2.4.2.7 DSOD

In 2017, Fudan University proposed DSOD algorithm. The focus is not on the mAP, but from another perspective to show that the difference between a fine-tune pre-trained model and a new detection model can actually be very small (Shen et al., 2017). DSOD can start training data from scratch, does not require a pre-training model, and can be as effective as the fine-tune model. There are three reasons for the need to retrain the model:

(1) the pre training model is generally trained on the classified image data set such as ImageNet, and it is not necessarily suitable to migrate to the data of the detection model.

(2) the structure of the pre training model is fixed, so it is more troublesome if you need to modify it.

(3) The training target of the pre training classification network is generally inconsistent with the detection target, so the pre training model is not necessarily the optimal choice for the detection algorithm.

2.4.2.8 RON

In 2017, Tsinghua University proposed RON algorithm (Kong et al., 2017).RON is an end-to-end full convolutional network, which focuses on two problems: multi-scale object detection and hard case mining .Part of the network structure of RON algorithm is similar to DSOD (multi-layer prediction and cross-layer feature fusion). Reverse connection is to provide the network with more semantic information. Specifically, the feature map of the lower layer is merged with the feature map of the upper layer after the resolution is enlarged through deconvolution to achieve the purpose of cross-layer feature fusion.

With the in-depth research of deep learning in computer vision, more and more new theories and methods appear. The algorithms of two-stages and the algorithm of one-stage based on the regression idea both learn from each other and constantly fuse, and both have achieved good results.

Chapter 3
Methodology

The main content of this chapter is to clearly articulate research methods, which satisfy the objectives of this report. The chapter primarily covers the details of the principle of YOLOv5 and Faster R-CNN algorithm, the environment deployment, dataset, the process of training model and performance indexes.

YOLO, a representative of one-stage algorithms, and Faster R-CNN, a member of two-stage algorithms, were taken into account in this project to detect local leaves and then compare and analyze the performance differences between them. This chapter will firstly introduce the working principle of YOLO and Faster R-CNN, then bring in the specific process of environment deployment, dataset preparation and model training in turn, and finally explicate various performance in the field of object detection, which are also the criteria for evaluating the performance of the proposed models.

3.1 Working principle and structure analysis of Yolo

In the official code of YOLOv5, four versions of object detection network are given, namely YOLOv5s, YOLOv5m, YOLOv5l and YOLOv5x. The YOLOv5s model was used in this project. YOLOv5s network is the network with the minimum depth and the minimum width of feature map in YOLOv5 series. The YOLOv5s is the fastest, but the AP is also the least accurate. However, this model is also a good choice if the detection is focused on larger targets and less complex scenarios, which are more speed oriented. The other three networks of YOLOv5 series are based on YOLOv5s to continuously deepen and widen the network, and then the AP value is also continuously improved, but the speed will become slower and slower.

The structure of YOLOv5 is very similar to that of YOLOv4, for example, they have a similar network structure, both use CSPDarknet53 (Cross Stage Partial Network) as Backbone. PANET (Path Aggregation Network) and SPP (Space Pyramid Pooling) were used as Neck.

3.1.1 Input end

The input of YOLOv5 adopts the same way of mosaic data enhancement as YOLOv4. Mosaic refers to the method of CutMix data enhancement proposed by the end of 2019 (Yun et al., 2019). However, CutMix only uses two images for splicing, while mosaic data enhancement uses four images, which are randomly scaled, randomly cropped and randomly arranged. In addition, YOLOv5 is also optimized in terms of adaptive image scaling.

3.1.2 Backbone

The function of backbone is to aggregate and combine different fine-grained images to form a convolutional neural network of image features. The backbone part of YOLOv5s adopts CSPNet, the full name of which is “cross stage partial network”. CSPNet solves the problem of repeated gradient information in other large CNN framework backbone. The change of gradient is integrated into the feature map from the beginning to the end, so the parameters and FLOPS values of the model are reduced, which not only ensures the reasoning speed and accuracy, but also reduces the model size.

The difference between YOLOv5 and YOLOv4 is that only the backbone network in YOLOv4 uses CSP structure, while YOLO5 designs two CSP structures. Taken YOLOv5s network as an example, it is "CSP1_X" structure applied to the Backbone, while another "CSP2_X" structure is applied to the Neck to strengthen the ability of network characteristic fusion.

3.1.3 Neck

Neck is a series of network layers that mix and combine image features and transfer them to the prediction layer, also known as Head. The “Neck” part of YOLOv5s adopts PANET(Wang, Liew, Zou, Zhou & Feng, 2019). Neck is mainly used to generate feature pyramids. The feature pyramid can enhance the detection of objects with different scales, so it can recognize the same object with different sizes and scales. In the research of YOLOv4, PANET is considered to be the most suitable feature fusion network for YOLO (Bochkovskiy, Wang & Liao, H. Y. M. 2020). Therefore, both YOLOv5 and YOLOv4 use PANET as a neck to aggregate features.

3.1.4 Head

The Head of the model is mainly used for final inspection. It applies anchor box to feature map and generates final output vector with class probability, object score and bounding box. In the YOLOv5 model, the model head is the same as the previous versions of YOLOv3 and YOLOv4. These heads with different scaling scales are used to detect objects of different sizes. Each head has a total of (80 classes + 1 probability + 4 coordinates) * 3 anchor frames, a total of 255 channels.

3.1.5 Loss function, activation function and optimization function

The loss function of object detection task is generally composed of classification loss and bounding box recurrence loss. "GIOU_Loss" is used as the loss function of the bounding box in YOLOv5. In the post-processing of object detection, NMS operation is usually needed for filtering many object frames. In yolov5, weighted NMS is used.

The selection of activation function is very important for deep learning network. In YOLOv5, the activation function Leaky ReLU is used in the middle / hidden layer, and the activation function Sigmoid is used in the final detection layer.

The author of YOLOv5 provides us with two optimization functions Adam and SGD, and preset the training parameters matching with them. The default is SGD. If we are training for a smaller custom data set, Adam is a better choice. Adam is used in this project

3.2 Analysis of the working principle of Faster R-CNN

After the accumulation of R-CNN and Fast RCNN, Girshick proposed the Faster R-CNN algorithm in 2016. Structurally, Faster R-CNN integrates feature extraction, proposal extraction, bounding box regression and classification into one network, which greatly improves the overall performance, especially the detection speed.

The main implementation steps of Faster R-CNN are as follows. The first step is feature extraction. Faster R-CNN first extracts the feature map of the candidate image. The feature map is Shared for subsequent RPN (Region Proposal Network) layer and fully Connection layer.

The second step is to enter RPN (Region Proposal Network). RPN network is used to generate regional candidate image blocks. This layer determines through “softmax” that anchors belong to the foreground or background and uses bounding box regression to correct them to get precise proposals.

The third step is ROI Pooling. This layer collects the input feature map and candidate object areas. After synthesizing the information, the feature map of the object area is extracted and sent to the following full connection layer to determine the object category.

The fourth step is Classification. The object area feature map is used to calculate the category of the object area, and the boundary box regression is used to obtain the final precise location of the detection box.

Therefore, we also see that the biggest highlight of Fast R-CNN is that it proposes an effective method to locate the object area, which greatly reduces the time consumption of convolution calculation, so the speed has been greatly improved.

3.3 Environment deployment

Since the model training and verification need a lot of computing power, we use Google Colab platform for training and verification. Colab is a free deep learning cloud platform provided by Google based on Jupiter notebook. It provides a free Tesla P100 GPU for deep learning researchers. The calculation environment is as follows:

Table 1 Colab environment configuration

Name	Configuration
GPU	Tesla P100
Yolov5 framework	Pytorch
Faster R-CNN framework	Tensorflow
Language	Python3.6
Operation platform	Colab（Linux）

The computing power of data set production and annotation is small, so we use local environment to make data set faster and more convenient. After that, the data set is put into the Colab platform for training and verification. In the local Windows environment, we use Lableme software to label the collected images. The specific configuration and environment are shown in the following table:

Table 2 Local configuration and environment for data set processing

Name	Configuration
CPU	Intel(R) Core™ i5-7300U
Annotation tool	Labelme v4.5.6
Language	Python v3.7
Operating system	windows 10

3.4 Data set preparation

The basic dataset is five types of tree leaves collected in the park of Auckland: (1) Magnolia grandiflora, (2) Boehmeria nivea, (3) Clausena lansium, (4) Euphoria longan, (5) Hibiscus.

3.4.1 Data set acquisition

We took pictures of the collected data in various states, including changing the distance, changing different angles and so on. The shooting background includes natural background and white paper background. The shooting includes different types of single leaf and combination of different leaves. A total of eight groups of video data were captured, each video duration was 30 seconds – 80 seconds, and the recording frame rate was 60fps, as shown in Figure 3.1.

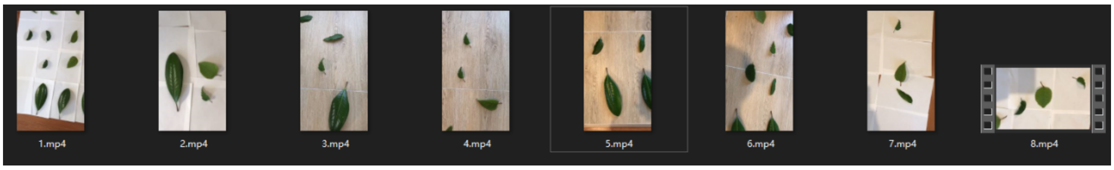


Figure 3.1 Leaf data collected

After that, we sparse frames from the video to obtain 419 images, some of which are shown in Figure 3.2.

Figure 3.2 Video frames extracted from a video

3.4.2 Data set annotation

We use labelme, a data annotation software, to label the obtained images. The method of labeling is to label each category by manually selecting a rectangle box. In the process of labeling, different types of leaf abbreviations are represented by the combination of different numbers and initials of leaf names. 1 -'m.g. 'stands for Magnolia grandiflora, 2 -'b.n.' stands for Boehmeria nivea, 3 -'c.l. 'stands for Clausena lansium, 4 -'e.l.' stands for Euphoria longan, and 5 -'h. 'stands for Hibiscus.

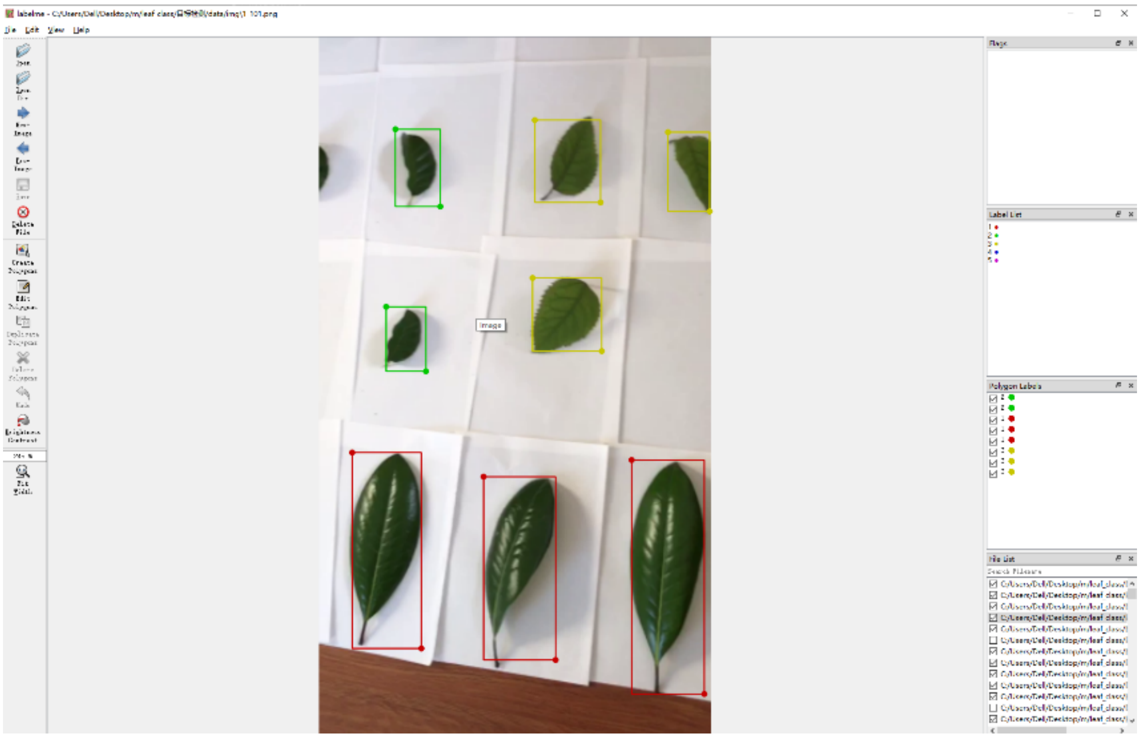


Figure 3.3 Data annotation through Lableme software

The annotated data is saved as a JSON file, which contains the name of each data and the location and category information of each rectangle box. Yolov5 and faster R-CNN have different requirements for annotation files, which need to be converted according to different requirements.

(1) YOLOv5

Yolov5 requires each image label to be stored in a TXT format file, the specific contents include: category serial number, horizontal coordinate of the upper left corner of the rectangle box, vertical coordinate of the upper left corner of the rectangle box, width of the rectangle box, length of the rectangle box.Each value is the ratio of the edge of the rectangle to the entire image size.

(2) Faster R-CNN

Faster R-CNN requires all labels to be stored in one TXT format filev. Specific content includes: image position and name, horizontal coordinate of upper left corner of rectangle box, vertical coordinate of upper left corner of rectangle box, width of rectangle box, length of rectangle box, category of the rectangle box.vEach value is the actual size of the pixels of the rectangle in the image.

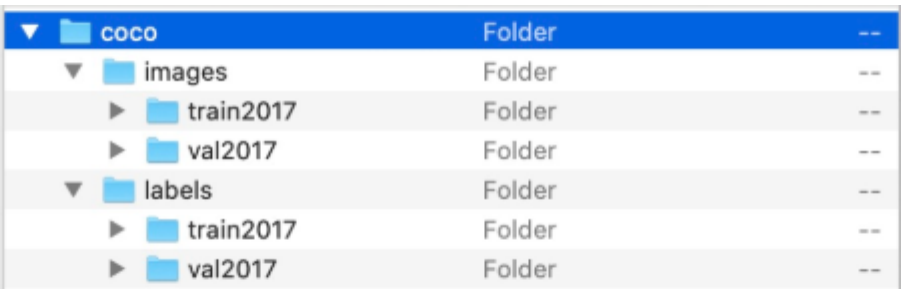
3.4.3 Data Augmentation

Given the relatively small number of overall datasets, data augmentation is necessary. The specific operations include flipping, zooming in, zooming out, clipping and combining. The 419 images collected were expanded 4 times to 1676 images. Then the data is divided into training set and verification set according to the ratio of 8:2. The final training set has 1340 pictures and 336 verification sets.

3.5 Training Model

3.5.1 Yolo v5 model

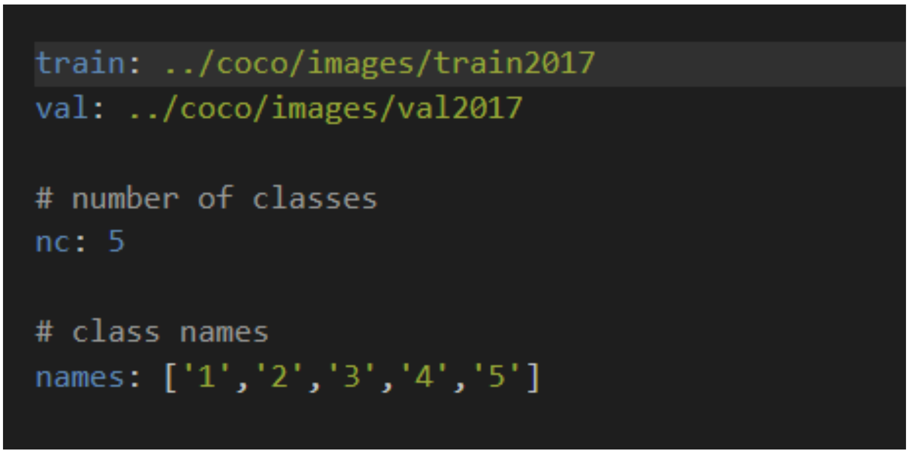
Firstly, we put the data in a specified location, as shown in Figure 3.4.



```
▼ coco                Folder   --
    ▼ images          Folder   --
        ▶ train2017   Folder   --
        ▶ val2017     Folder   --
    ▼ labels          Folder   --
        ▶ train2017   Folder   --
        ▶ val2017     Folder   --
```

Figure 3.4 Data storage directory format

Then we configure the parameter file to determine the training data path and validation data path, as shown in Figure 3.5.



```
train: ../coco/images/train2017
val: ../coco/images/val2017

# number of classes
nc: 5

# class names
names: ['1','2','3','4','5']
```

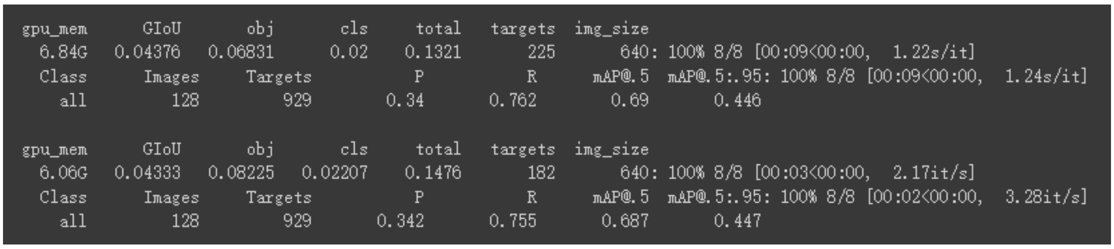
Figure 3.5 YOLOv5 parameter configuration

After the model and data are ready, we set the hyperparameters as shown in Table 3.1.

Table 3.1 hyperparameter configuration

Name	value
optimizer	SGD
Initial learning rate	0.001
Batch_size	8
Image size for input	640x480 pixels
Batch size	40

After setting the hyperparameters,   we start the training process. A part of the training process is shown in Figure 3.6.



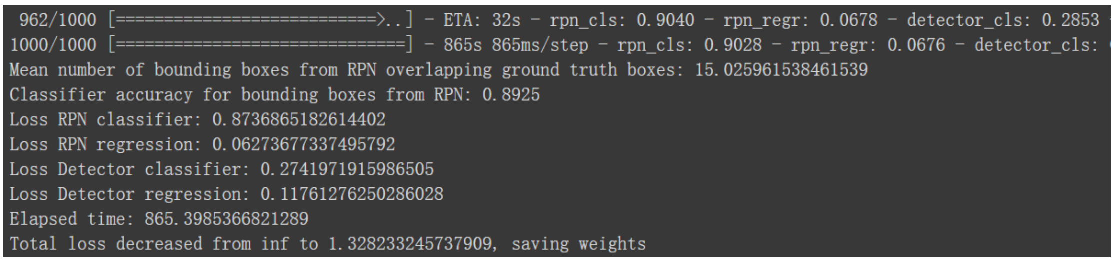
```
gpu_mem      GIoU       obj       cls     total   targets  img_size
  6.84G   0.04376   0.06831      0.02    0.1321       225       640: 100% 8/8 [00:09<00:00,   1.22s/it]
  Class    Images   Targets         P         R   mAP@.5  mAP@.5:.95: 100% 8/8 [00:09<00:00,   1.24s/it]
    all       128       929      0.34     0.762      0.69      0.446

gpu_mem      GIoU       obj       cls     total   targets  img_size
  6.06G   0.04333   0.08225   0.02207    0.1476       182       640: 100% 8/8 [00:03<00:00,   2.17it/s]
  Class    Images   Targets         P         R   mAP@.5  mAP@.5:.95: 100% 8/8 [00:02<00:00,   3.28it/s]
    all       128       929     0.342     0.755     0.687      0.447
```

Figure 3.6 Part of the training process of Yolo v5

3.5.2 Faster R-CNN model

We set the data according to the same configuration of YOLOv5. After construct the network model, we configure the training file, and start training after setting the hyperparameters. The training process is shown in Figure 3.7.



```
 962/1000 [===========================>..] - ETA: 32s - rpn_cls: 0.9040 - rpn_regr: 0.0678 - detector_cls: 0.2853
1000/1000 [==============================] - 865s 865ms/step - rpn_cls: 0.9028 - rpn_regr: 0.0676 - detector_cls:
Mean number of bounding boxes from RPN overlapping ground truth boxes: 15.025961538461539
Classifier accuracy for bounding boxes from RPN: 0.8925
Loss RPN classifier: 0.8736865182614402
Loss RPN regression: 0.06273677337495792
Loss Detector classifier: 0.2741971915986505
Loss Detector regression: 0.11761276250286028
Elapsed time: 865.3985366821289
Total loss decreased from inf to 1.328233245737909, saving weights
```

Figure 3.7 A part of the training process of Faster R-CNN

3.6 Evaluation Methods

3.6.1 Confusion Matrix

The confusion matrix describes the classification accuracy of a classifier. Assuming that there are only two categories of classification objectives, positive and negative, the meanings of TP, FP, FN and TN are as follows:

(1) True positions (TP): the number of positive cases correctly divided, that is, the number of instances that are actually positive cases and are classified as positive cases by the classifier;

(2) False positions (FP): the number of instances wrongly divided into positive cases, that is, the number of instances that are actually negative but are classified as positive cases by the classifier;

(3) False negatives (FN): the number of instances wrongly divided into negative cases, that is, the number of instances that are actually positive but classified as negative by the classifier;

(4) True negatives (TN): the number of cases correctly divided into negative cases, that is, the number of instances that are actually negative cases and are classified as negative cases by the classifier.

where *P* stands for precision, and the calculation formula is the actual number of positive samples in the predicted samples / the number of all positive samples, that is,

$$precision = TP / (TP + FP) \tag{3.1}$$

where *R* stands for recall, and the calculation formula is the actual number of positive samples in the forecast samples / the number of predicted samples, i.e.,

$$recall = TP / (TP + FN) \qquad (3.2)$$

3.6.2 Precision-recall curve

By changing the recognition threshold, the system can recognize the first k images in turn. The change of the threshold value will cause the change of precision and recall values. The PR curve is given as follow.

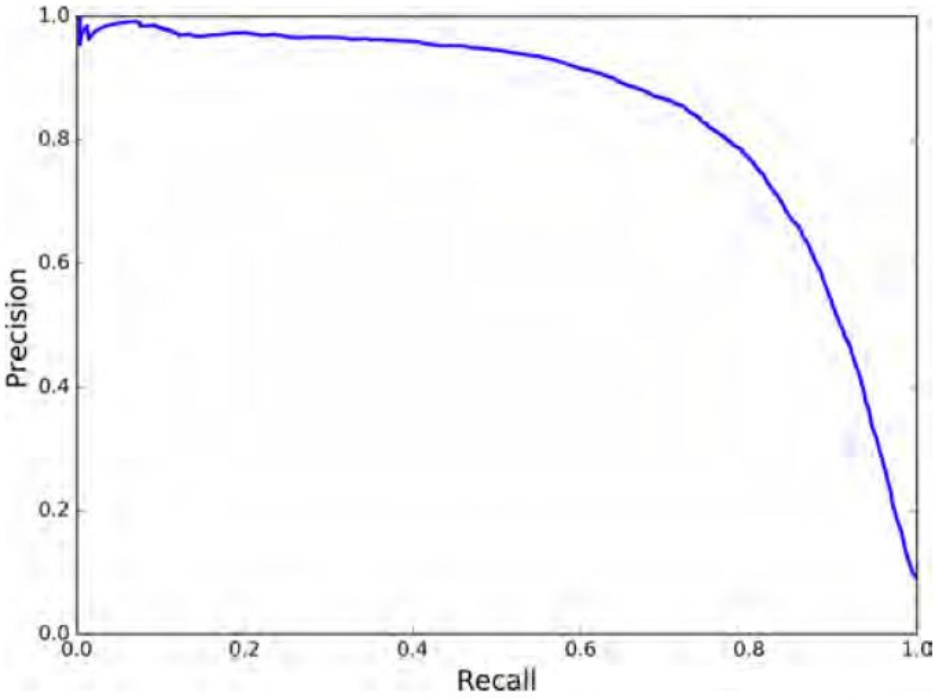


Figure 3.8 Precision-Recall curve

In the multiclass classification task, we need to know the precision and recall corresponding to top-1 to top-N (*n* is the number of all test samples). Obviously, with more and more samples selected, recall will be higher and higher, and precision will show a downward trend. The change of recognition threshold enables the system to recognize the first *k* images in turn. The change of threshold value will lead to the change of precision and recall value, and then the curve can be obtained.

If the performance of a classifier is better, it should have the following performance: while the recall value increases, the precision value remains at a high level. The classifier with poor performance may lose a lot of precision values in exchange for the improvement of recall value.

3.6.3 AP and mAP

AP means average precision, which is the area under the precision recall curve. Generally speaking, the better the classifier is, the higher the AP value is.

mAP is the mean average precision, which is the average AP value of multiple categories of AP. The size of mAP must be in the interval [0,1], the larger the better. This index is a key index to measure the detection accuracy in object detection.。

3.6.4 IoU

The value of IOU can be understood as the coincidence degree between the box predicted by the model and the box marked in the original picture. The calculation method is the intersection of detection result and ground truth divided by their union, which is the detection accuracy. This quantity, also known as the Jaccard index, was first proposed by Paul Jaccard in the early 20th century (Berman & Blaschko, 2017).

3.6.5 FPS

In addition to the detection accuracy index mAP, another important performance index of object detection algorithm is speed. Only with fast speed can real-time detection be realized, which is extremely important for some application scenarios. A common measure of speed is FPS (frame per second), the number of images that can be processed per second. The comparison between FPS values needs to be based on the same hardware condition. In addition, the time required to process an image can also be used to evaluate the detection speed. The shorter the time, the faster the detection speed.

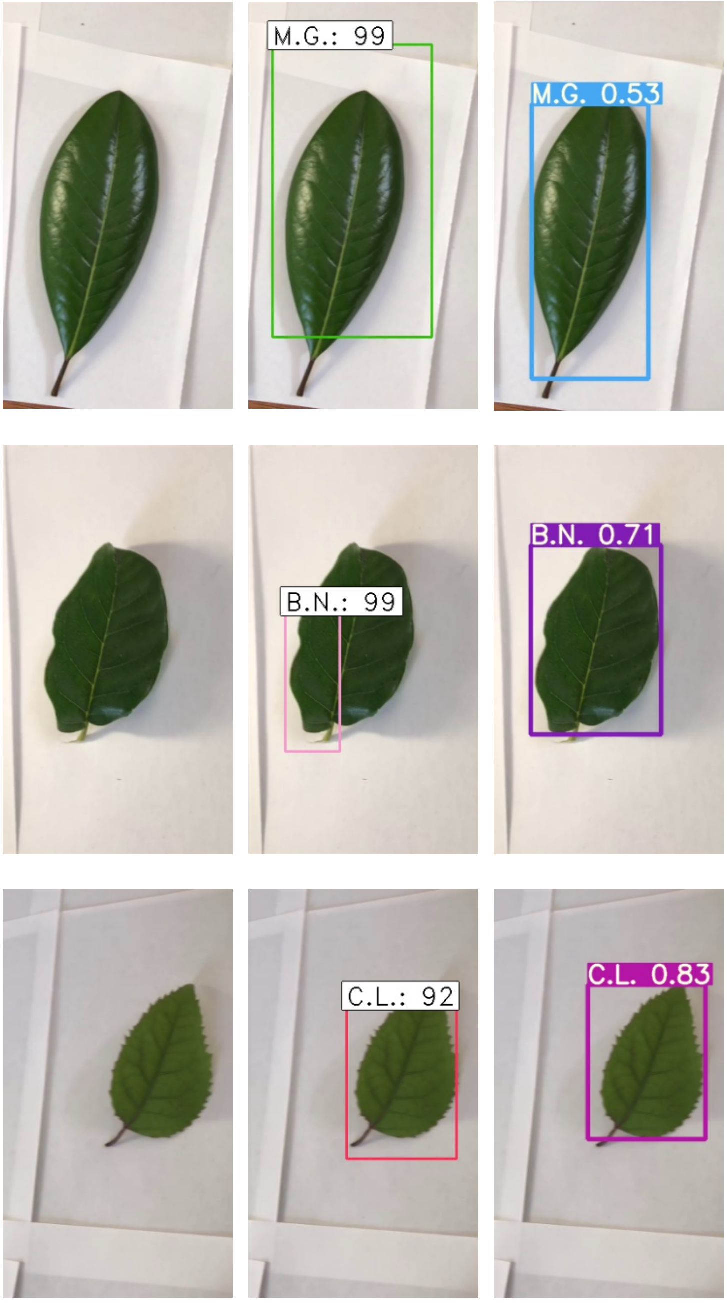
Chapter 4
Results

The main content of this chapter is to compare the object detection results of the two models and analyze the differences in the training model and the reasons

4.1 Comparison of object detection results

Our data contains different backgrounds, different numbers and different types of leaves. In order to better compare the results of the two models, it is necessary to test in different scenarios. The left picture of each line in this section is the original image, the middle picture is the result of Faster R-CNN model test, and the right picture is the result of YOLOv5s model test.

4.1.1 Test in single leaf scene



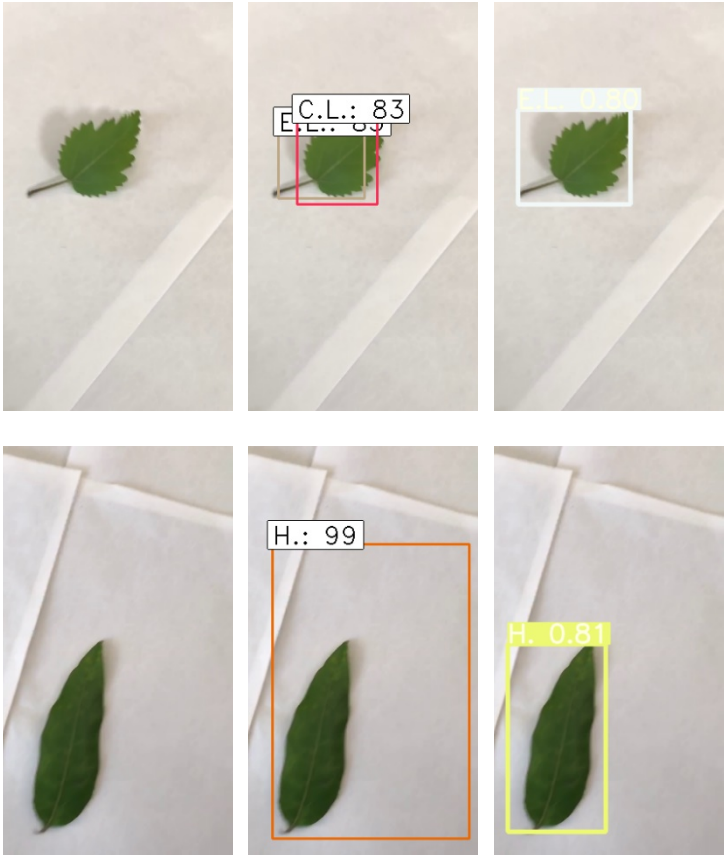


Figure 4.1 The results of Faster R-CNN and Yolov5 for five kinds of leaves

In the single-leaf scenario, we see that YOLOv5 is able to accurately identify objects, and the box selection of object boxes is more appropriate. Meanwhile, Faster R-CNN is able to identify the majority accurately. Based on the EL leaf, two results were exported. In addition, when the model Faster R-CNN identifies the "BN leaf", it only recognizes a part of the leaf, and the identification box is too large when it identifies the leaf H.Therefore, the Faster R-CNN model is not very accurate in the selection of recognition box.

4.1.2 Test in a small number of mixed leaves

The sequence of motion pictures is as same as Figure 4.1. The left is the original picture, the middle is the test result of the Faster R-CNN model, and the right is the test result of the YOLOv5 model.

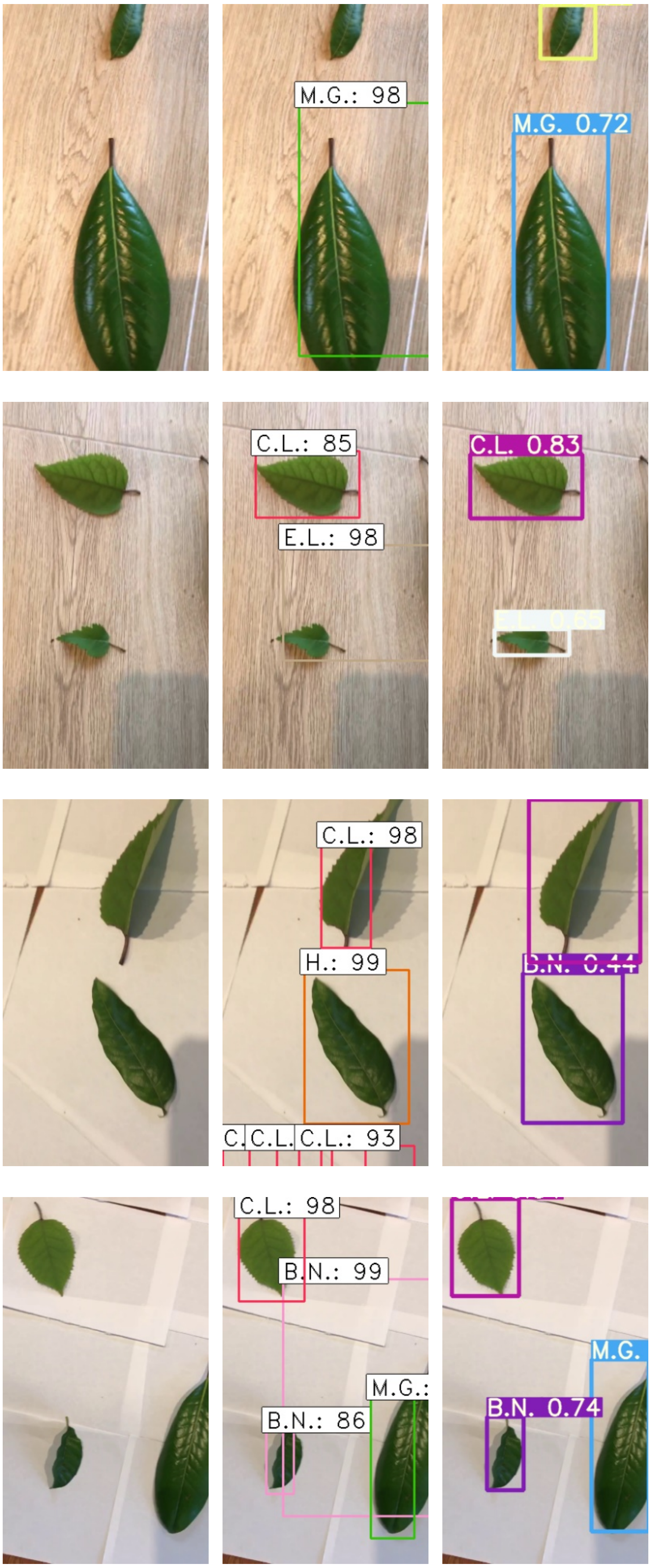


Figure 4.2 The results of different leaf species

In Figure 4.2, we see that the Faster R-CNN model is able to accurately identify most objects, but it also missed some individual leaves. At the same time, there are some cases where the box is too large or too small.

For YOLOv5, even a partial leaf is able to be recognized, and the box size is appropriate. But one of them was wrongly identified, and the leaf H was wrongly identified as the leaf BN.

4.1.3 The Results in Complex Scenarios with Many Leaves

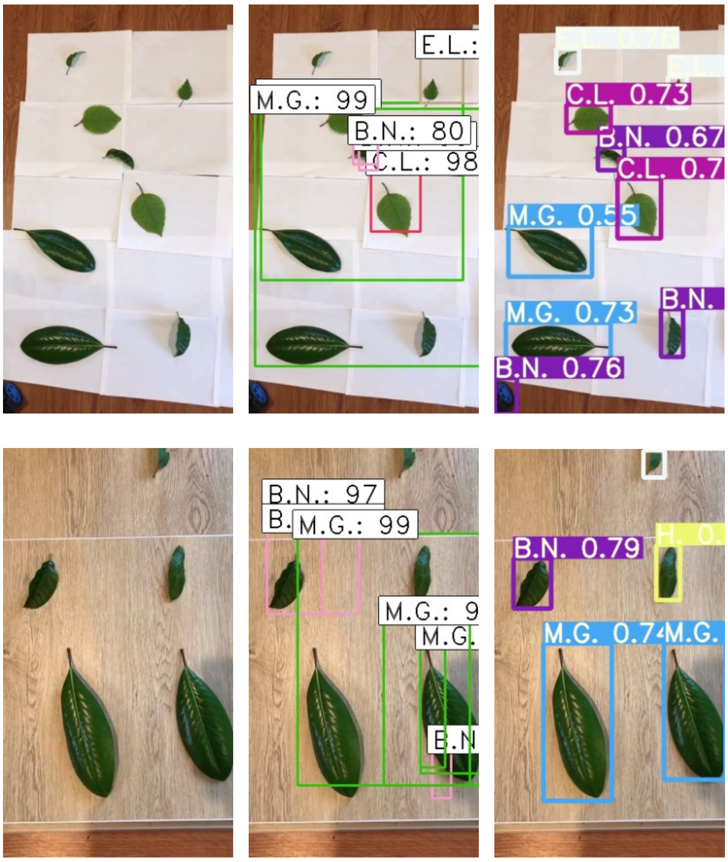


Figure 4.3 The test results of five kinds of leaves were included

In the complex scene where all the five kinds of leaves are contained, the YOLOv5 model can identify almost all the leaves accurately. However, the toy wheel in the lower left corner of the first set of pictures is mistakenly identified as a leaf. The Faster R-CNN model can only identify part of the leaves in this scene, and the box selection is not very accurate. But it also doesn't recognize the toy wheel in the lower left corner of the first set of images as a leaf.

4.2 Comparative analysis of the performance of the two models

4.2.1 Training loss

The appropriate loss function is one of the most important steps to ensuring that the model works as expected. The main function of object detection is to locate and identify objects, while the function of loss function is to make positioning more accurate and recognition accuracy higher.

Training Loss reflects the overall situation of whether different models are suitable for the selection of loss function and whether the setting of hyperparameters is reasonable. Below, we will make a comparative analysis of the results of classification loss, regression loss and overall loss of the two algorithms.

(1) Classification loss

In Figure 4.4, in terms of classification task, both algorithms can decline steadily, but YOLOv5 declines more steadily and continues to decline, while Faster R-CNN declines slowly on the whole.

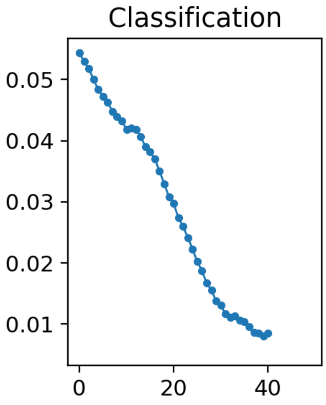


Figure 4.4 Classification loss training curve of yolov5 model

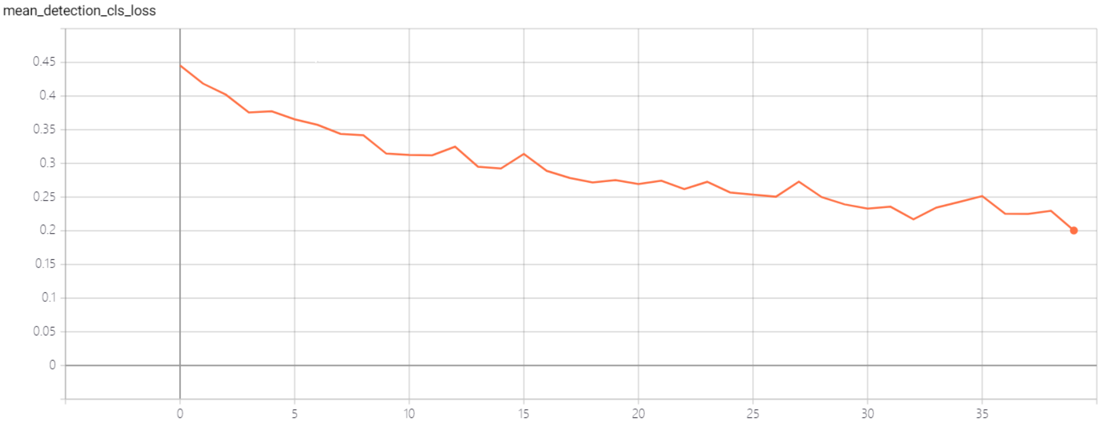


Figure 4.5 The training loss curve of Faster R-CNN model for classification

（2）Regression loss

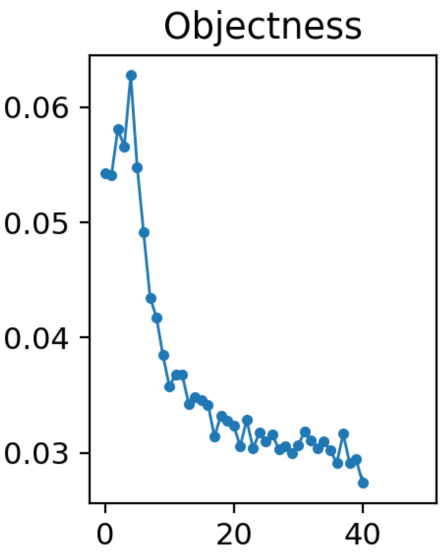


Figure 4.6 The training loss curve of YOLOv5 model for regression

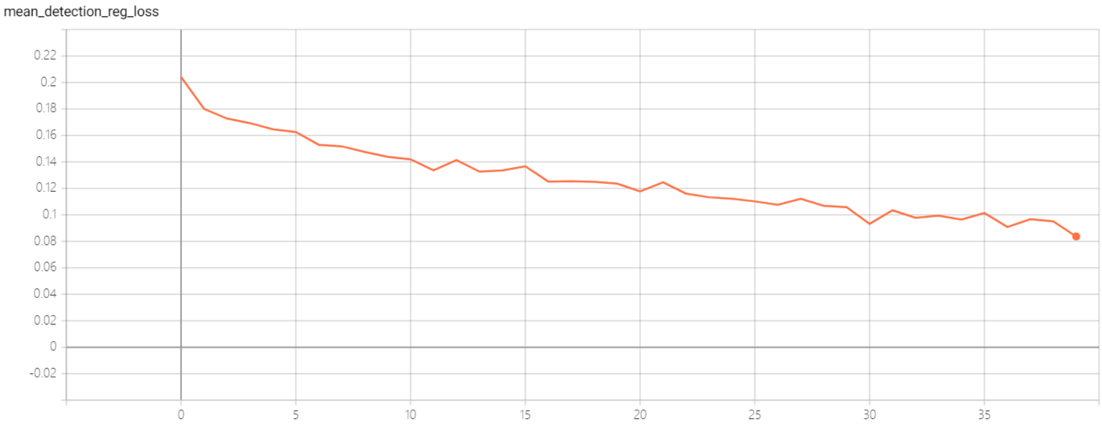


Figure 4.7 The training loss curve of Faster R-CNN model for regression

In terms of regression loss, the two algorithms can also maintain the trend decline. YOLOv5 cuts off rapidly at the beginning, but tends to be stable in the later stage, and the whole process presents a small amplitude of shock, while Faster R-CNN drops more steadily and have a small amplitude of shock.

(3) Total loss

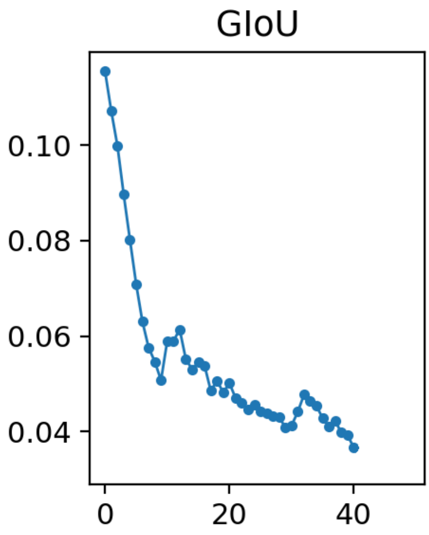


Figure 4.8 Total loss training curve of yolov5 model

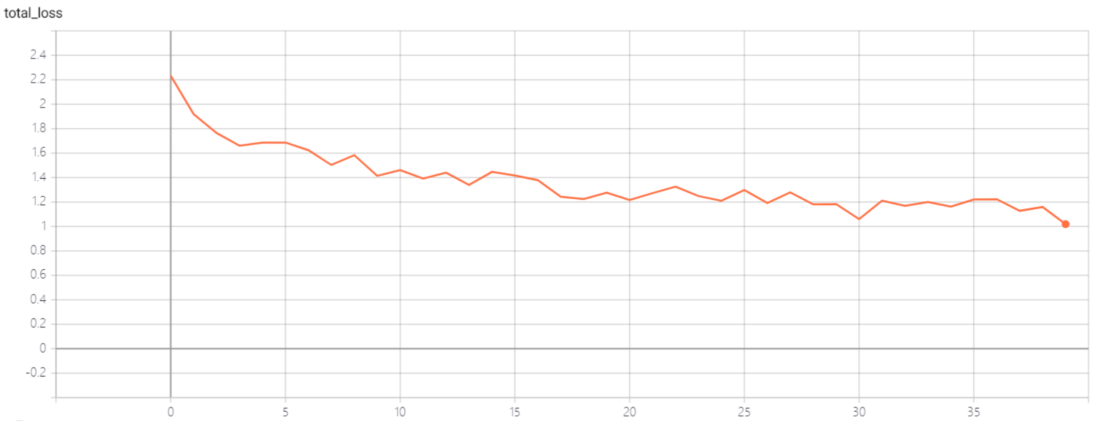


Figure 4.9 Total loss training curve of Faster R-CNN model

As for the total loss curve, YOLOv5 decreases Faster in the early stage and then tends to slow down, while Faster R-CNN also decreases slowly and steadily in the whole process.

4.2.2 Comparison of Deep Learning Models in Speed

The network structure of the model determines the training speed and execution speed of the model as well as the memory usage. The training speed, execution speed and memory usage of the two models are shown in the following table. In Table 4.1, YOLOv5 has obvious advantages in speed and memory consumption. Compared with Faster R-CNN, YOLOv5 is nearly 32 times faster in training speed, nearly 39 times faster in execution speed, and nearly 8 times smaller in memory occupancy.

Table 4.1 Comparison of speed and memory consumption of the two models

Type	Yolov5	Faster R-CNN
Training speed	26 ms/step	814ms/step
Execution speed	0.011	0.432s
Memory usage	14MB	109MB

4.2.3 Comparison of model mAP

Based on the IoU threshold value of 0.5, the mAP results obtained by the two models are shown in the figure below. It can be seen that both methods can keep the accuracy increasing gradually, but Yolov5 starts to increase from 0, while Faster R-CNN starts to increase from about 0.8, indicating that it increases more slowly. The primary reason why the initial accuracy of Faster R-CNN is so high is that the model is a two-stage algorithm. In the RPN candidate box training in the first stage, there are many candidate boxes in each column, and the boxes that are not objects are classified as negative classes, resulting in a high accuracy even if all the results are negative classes.

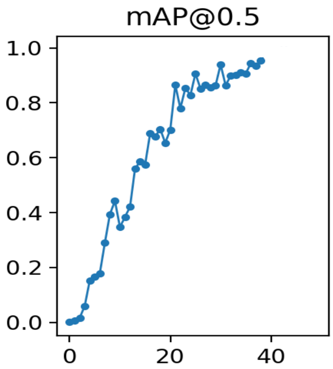


Figure 4.10 mAP training curve of yolov5 model

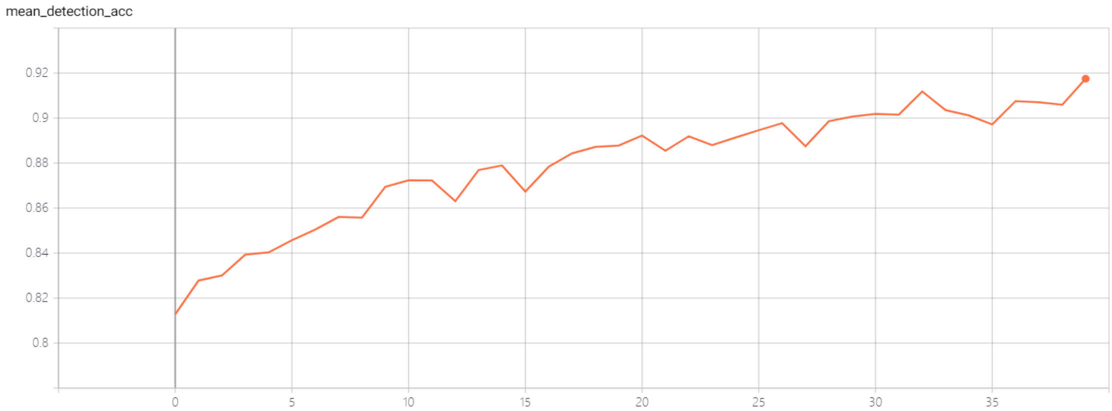


Figure 4.11 mAP training curve of Faster R-CNN model

After the model training, we selected the one with the lowest total loss as the optimal model to be saved and carried out verification set verification, as shown in Figure 4.12. Finally, it was calculated that the mAP of yolov5mAP was 0.932 and the mAP of Faster R-CNN was 0.918.

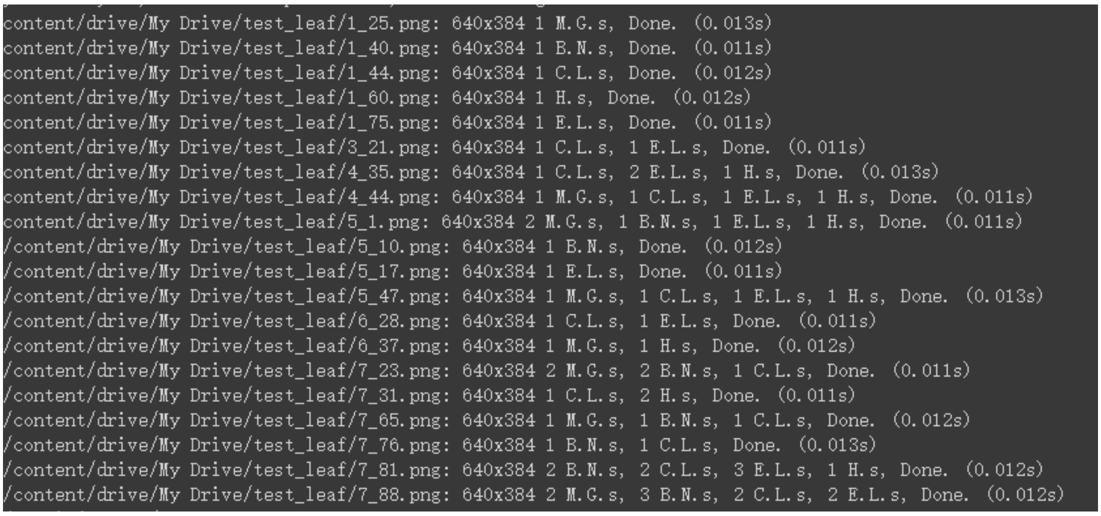
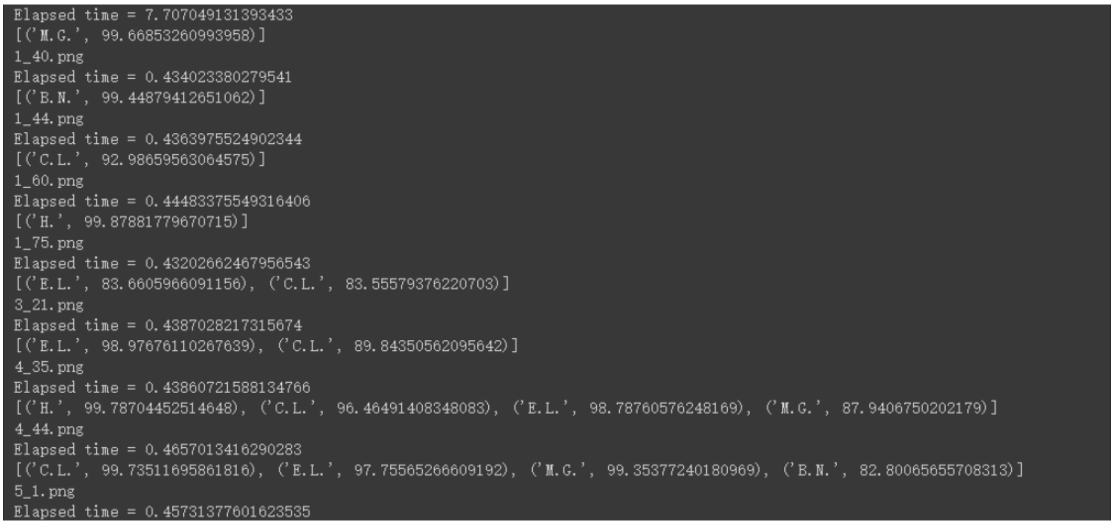

```
content/drive/My Drive/test_leaf/1_25.png: 640x384 1 M.G.s, Done. (0.013s)
content/drive/My Drive/test_leaf/1_40.png: 640x384 1 B.N.s, Done. (0.011s)
content/drive/My Drive/test_leaf/1_44.png: 640x384 1 C.L.s, Done. (0.012s)
content/drive/My Drive/test_leaf/1_60.png: 640x384 1 H.s, Done. (0.012s)
content/drive/My Drive/test_leaf/1_75.png: 640x384 1 E.L.s, Done. (0.011s)
content/drive/My Drive/test_leaf/3_21.png: 640x384 1 C.L.s, 1 E.L.s, Done. (0.011s)
content/drive/My Drive/test_leaf/4_35.png: 640x384 1 C.L.s, 2 E.L.s, 1 H.s, Done. (0.013s)
content/drive/My Drive/test_leaf/4_44.png: 640x384 1 M.G.s, 1 C.L.s, 1 E.L.s, 1 H.s, Done. (0.011s)
content/drive/My Drive/test_leaf/5_1.png: 640x384 2 M.G.s, 1 B.N.s, 1 E.L.s, 1 H.s, Done. (0.011s)
/content/drive/My Drive/test_leaf/5_10.png: 640x384 1 B.N.s, Done. (0.012s)
/content/drive/My Drive/test_leaf/5_17.png: 640x384 1 E.L.s, Done. (0.011s)
/content/drive/My Drive/test_leaf/5_47.png: 640x384 1 M.G.s, 1 C.L.s, 1 E.L.s, 1 H.s, Done. (0.013s)
/content/drive/My Drive/test_leaf/6_28.png: 640x384 1 C.L.s, 1 E.L.s, Done. (0.011s)
/content/drive/My Drive/test_leaf/6_37.png: 640x384 1 M.G.s, 1 H.s, Done. (0.012s)
/content/drive/My Drive/test_leaf/7_23.png: 640x384 2 M.G.s, 2 B.N.s, 1 C.L.s, Done. (0.011s)
/content/drive/My Drive/test_leaf/7_31.png: 640x384 1 C.L.s, 2 H.s, Done. (0.011s)
/content/drive/My Drive/test_leaf/7_65.png: 640x384 1 M.G.s, 1 B.N.s, 1 C.L.s, Done. (0.012s)
/content/drive/My Drive/test_leaf/7_76.png: 640x384 1 B.N.s, 1 C.L.s, Done. (0.013s)
/content/drive/My Drive/test_leaf/7_81.png: 640x384 2 B.N.s, 2 C.L.s, 3 E.L.s, 1 H.s, Done. (0.012s)
/content/drive/My Drive/test_leaf/7_88.png: 640x384 2 M.G.s, 3 B.N.s, 2 C.L.s, 2 E.L.s, Done. (0.012s)
```

Figure 4.12 Screenshot of Yolov5 model



```
Elapsed time = 7.707049131393433
[('M.G.', 99.66853260993958)]
1_40.png
Elapsed time = 0.434023380279541
[('B.N.', 99.44879412651062)]
1_44.png
Elapsed time = 0.4363975524902344
[('C.L.', 92.98659563064575)]
1_60.png
Elapsed time = 0.44483375549316406
[('H.', 99.87881779670715)]
1_75.png
Elapsed time = 0.43202662467956543
[('E.L.', 83.6605966091156), ('C.L.', 83.55579376220703)]
3_21.png
Elapsed time = 0.4387028217315674
[('E.L.', 98.97676110267639), ('C.L.', 89.84350562095642)]
4_35.png
Elapsed time = 0.43860721588134766
[('H.', 99.78704452514648), ('C.L.', 96.46491408348083), ('E.L.', 98.78760576248169), ('M.G.', 87.9406750202179)]
4_44.png
Elapsed time = 0.4657013416290283
[('C.L.', 99.73511695861816), ('E.L.', 97.75565266609192), ('M.G.', 99.35377240180969), ('B.N.', 82.80065655708313)]
5_1.png
Elapsed time = 0.45731377601623535
```

Figure 4.13 Screenshot of Faster R-CNN model

4.3 Limitations of the Research

4.3.1 The data set

The number of data sets in this project is generally small. In addition, compared with the complex background of leaves in nature, the tested scene is relatively simple, so the trained model is difficult to be extended to the natural scene for leaf detection.

4.3.2 The limit of computational power

(1) Limited by the calculation force, batch_size is set to 40, which is relatively small.

(2) Because this project chose to train on the Colab platform, the model could not be trained for a long time due to time constraints, so the model could not be equipped with a larger epoch for more complete training of the model

Chapter 5

Analysis and Discussions

In this chapter, experimental results are analyzed and compared. The comparisons of the results under various conditions will be critically analyzed.

5.1 Analysis

From the experimental results of this project, we see that the YOLOv5 model representing the first-stage algorithm is superior to the Faster R-CNN model representing the two-stage algorithm in most indicators. The differences between the two algorithms in training speed, execution speed, model size, accuracy and other indexes are shown in the following table. In addition, the yolov5 model also performs better in rectangular box regression. Yolo, especially the latest Yolov5 algorithm, takes up less memory and executes faster, so it can be adapted to more devices and scenarios.

Table 5.1 Comparison of experimental results between yolov5 model and Faster R-CNN model

Type	YOLOv5	Faster R-CNN
Training speed	26 ms/step	814ms/step
Execution speed	0.011s	0.432s
Memory usage	14MB	109MB
mAP	0.932	0.918
Total loss	0.032	1.028

5.2 Discussions

Faster R-CNN uses raw image data as input, while YOLOv5 makes Mosaic data augmentation for input data. Each time four training images are read and flipped, scaled, changed color gamut, etc. Data augmentation enables the model to learn to recognize objects in a smaller scope, and its advantage is to enrich the background of the object detection, and the data of four images will be calculated at a time during BN calculation, so that the mini-batch size does not need to be very large, and even a SINGLE GPU can achieve better results. The following figure shows how Mosaic works:

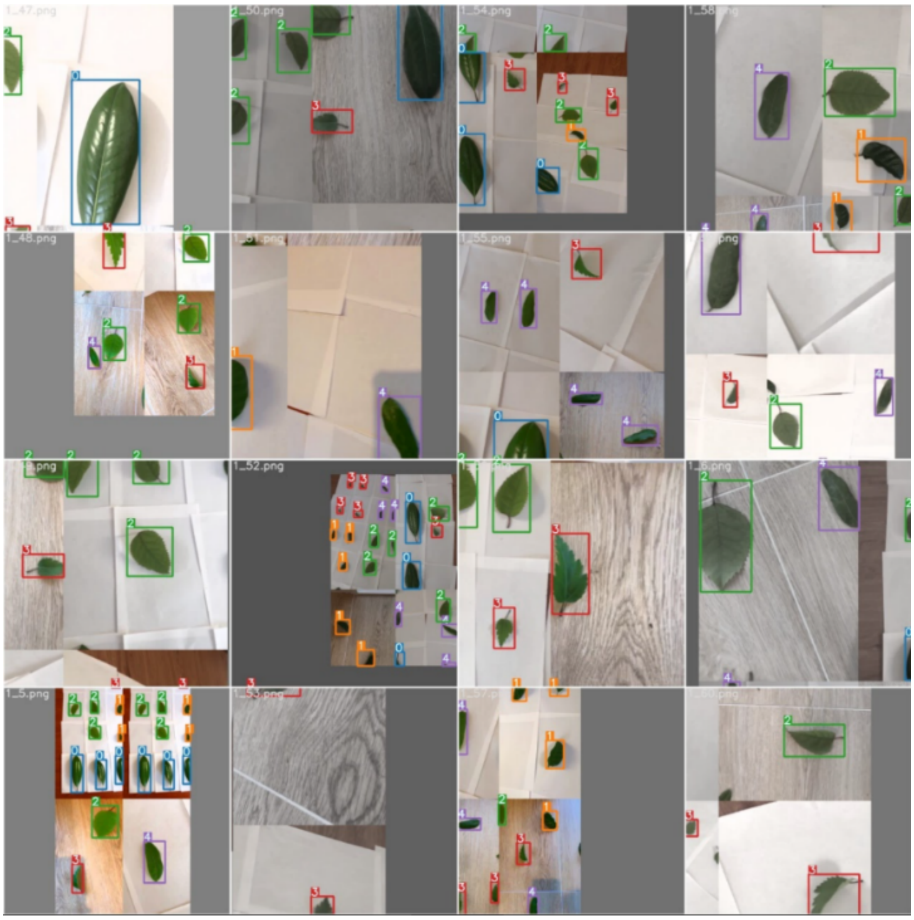


Figure 5.1 The operation of Mosaic data aggregation

In the feature extraction process, YOLOv5 adopts CSP+FPN+PAN structure. CSPNet can effectively enhance the learning ability of CNN, brings a relatively large performance improvement, and also reduce the amount of computation (Wang et al., 2020). In addition, YOLOv5 also adds PAN operation to add high-level semantics to the feature, which is more conducive to classification.

ResNet+ FPN structure is adopted for Faster R-CNN. The accuracy of Faster R-CNN is also good, but its main drawback is that its speed is relatively slow. The reason why the speed is not fast enough is that for each recommended area, ROI Pooling layer will be respectively injected, and then several full connection layers will be used for classification and regression. This process slows down the speed.

Loss calculations are based on objectness score, class probability score, and bounding box regression score in the YOLO model. YOLOv5 adopts GIoU loss as bounding box loss. It calculates the loss of class probabilities and object scores using the balanced_sigmoid_cross_entropy and Logits loss functions. Its advantage is that it can balance the positive and negative samples through the balance parameter, so as to get a better effect.

To sum up, YOLOv5 has made various optimizations in data augmentation, feature extraction, loss function and other aspects, making it excellent in both accuracy and speed, and more suitable for more complex and diverse application scenarios.

Chapter 6
Conclusion and Future Work

In this chapter, we will summarize the subject and methods of this project, and envision new research direction according to the result and insufficiency of the experiment, preparing for the future work.

6.1 Conclusion

In the field of leaf recognition, compared with the traditional classification problem, object detection is obviously more in line with the practical needs. However, most of the existing two-stage object detection algorithms with high accuracy are relatively slow and cannot meet the demand of the industry for real-time object detection. After five versions of iteration, the YOLO algorithm, the representative of the one-stage algorithm, has greatly improved in speed and accuracy, especially in terms of accuracy, which has also surpassed some two-stage algorithms.

In this report, YOLOv5 and Faster R-CNN model were used to realize object detection of leaves collected locally in New Zealand. Experimental results show that YOLOv5 algorithm is superior in almost all indicators. Especially, YOLOv5 algorithm is far superior to Faster R-CNN algorithm in terms of speed, memory occupancy and accuracy of object position prediction.

6.2 Future Work

In this project, the number and species of leaves as data sets are relatively small, and in order to ensure the experimental effect, the shapes of leaves of different species selected are significantly different. However, in practical applications, different kinds of leaves with very similar shapes may appear, and the difficulty of identification will be greatly increased. Therefore, it is necessary to train the model in the future with the increase in the number and types of leaves.

The background of this experiment is relatively simple, but in practical application, the background of leaves is often very complex. How to realize object detection with high recognition rate under complex background is the next research direction.

In this experiment, we implement the identification of a single leaf, while in reality, a large number of leaves on a tree tend to gather together with more complex shapes. How to realize the object detection under the real tree leaf scene is the direction of the next research.

References

Alom, M. Z., Taha, T. M., Yakopcic, C., Westberg, S., Sidike, P., Nasrin, M. S., ... & Asari, V. K. (2018). The history began from AlexNet: A comprehensive survey on deep learning approaches. *arXiv preprint arXiv:1803.01164*

Aziz, L., bin Haji Salam, S., & Ayub, S. (2020). Exploring deep learning-based architecture, strategies, applications and current trends in generic object detection: A comprehensive review. *IEEE Access*

Backes, A. R., Casanova, D., & Bruno, O. M. (2009). Plant leaf identification based on volumetric fractal dimension. *International Journal of Pattern Recognition and Artificial Intelligence*, *23*(06), 1145-1160

Berman, M., & Blaschko, M. B. (2017). Optimization of the Jaccard index for image segmentation with the lovász hinge. *CoRR, abs/1705.08790*, *5*

Bochkovskiy, A., Wang, C. Y., & Liao, H. Y. M. (2020). YOLOv4: Optimal speed and accuracy of object detection. *arXiv preprint arXiv:2004.10934*

Cai, Z., Fan, Q., Feris, R. S., & Vasconcelos, N. (2016). A unified multi-scale deep convolutional neural network for fast object detection. In *European Conference on Computer Vision* (pp. 354-370). Springer

Dai, J., Li, Y., He, K., & Sun, J. (2016). R-FCN: Object detection via region-based fully convolutional networks. In *Advances in neural information processing systems* (pp. 379-387)

Dai, K. J., & R-FCN, Y. L. (2016). Object detection via region-based fully convolutional networks. arxiv preprint. In *arXiv preprint*

Dalal, N., & Triggs, B. (2005). Histograms of oriented gradients for human detection. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR'05)* (Vol. 1, pp. 886-893)

Deng, J., Dong, W., Socher, R., Li, L. J., Li, K., & Fei-Fei, L. (2009). ImageNet: A large-scale hierarchical image database. In *IEEE Conference on Computer Vision and Pattern Recognition* (pp. 248-255)

Fu, C. Y., Liu, W., Ranga, A., Tyagi, A., & Berg, A. C. (2017). DSSD: Deconvolutional single shot detector. *arXiv preprint arXiv:1701.06659*

Ghazi, M. M., Yanikoglu, B., & Aptoula, E. (2017). Plant identification using deep neural networks via optimization of transfer learning parameters. *Neurocomputing*, *235*, 228-235

Ghiasi, G., Lin, T. Y., & Le, Q. V. (2019). NAS-FPN: Learning scalable feature pyramid architecture for object detection. In *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition* (pp. 7036-7045)

Gidaris, S., & Komodakis, N. (2015). Object detection via a multi-region and semantic segmentation-aware CNN model. In *Proceedings of IEEE International Conference on Computer Vision* (pp. 1134-1142)

Girshick, R. (2015). Fast R-CNN. In *Proceedings of the IEEE International Conference on Computer Vision* (pp. 1440-1448)

Girshick, R. B. (2012). *From rigid templates to grammars: Object detection with Structured Models*. Chicago, IL, USA: University of Chicago

Girshick, R. B., Felzenszwalb, P. F., & Mcallester, D. A. (2011). Object detection with grammar models. In *Proceedings of Advances in Neural Information Processing Systems* (pp. 442-450)

Girshick, R., Donahue, J., Darrell, T., & Malik, J. (2014). Rich feature hierarchies for accurate object detection and semantic segmentation. In *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition* (pp. 580-587)

Goëau, H., Bonnet, P., & Joly, A. (2016). Plant identification in an open-world (lifeclef 2016)

Guyer, D. E., Miles, G. E., Gaultney, L. D., & Schreiber, M. M. (1993). Application of machine vision to shape analysis in leaf and plant identification. *Transactions of the ASAE (USA)*

He, K., Gkioxari, G., Dollár, P., & Girshick, R. (2017). Mask r-cnn. In *Proceedings of IEEE International Conference on Computer Vision* (pp. 2961-2969)

He, K., Zhang, X., Ren, S., & Sun, J. (2015). Spatial pyramid pooling in deep convolutional networks for visual recognition. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, *37*(9), 1904-1916

Hinton, G. E., Krizhevsky, A., & Sutskever, I. (2012). ImageNet classification with deep convolutional neural networks. In *Proceedings of Advances in Neural Information Processing Stems*, *25*, 1106-1114

Im, C., Nishida, H., & Kunii, T. L. (1998). Recognizing plant species by leaf shapes-a case study of the acer family. In *Proceedings of International Conference on Pattern Recognition (Cat. No. 98EX170)* (Vol. 2, pp. 1171-1173)

Jeong, J., Park, H., & Kwak, N. (2017). Enhancement of SSD by concatenating feature maps for object detection. *arXiv preprint arXiv:1705.09587*

Joly, A., Goëau, H., Glotin, H., Spampinato, C., Bonnet, P., Vellinga, W. P., ... & Müller, H. (2015). LifeCLEF: Multimedia life species identification challenges. In *International Conference of the Cross-Language Evaluation Forum for European Languages* (pp. 462-483). Springer

Kapur, S. (2017). *Computer vision with Python 3: Image classification, object detection, video processing, and more*. Packt.

Kim, K. H., Hong, S., Roh, B., Cheon, Y., & Park, M. (2016). PvaNet: Deep but lightweight neural networks for real-time object detection. *arXiv preprint arXiv:1608.08021*

Kong, T., Sun, F., Yao, A., Liu, H., Lu, M., & Chen, Y. (2017). RON: Reverse connection with objectness prior networks for object detection. In *Proceedings of IEEE*

Conference on Computer Vision and Pattern Recognition (pp. 5936-5944)

Kong, T., Yao, A., Chen, Y., & Sun, F. (2016). HyperNet: Towards accurate region proposal generation and joint object detection. In *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition* (pp. 845-853)

Krizhevsky, A., Sutskever, I., & Hinton, G. E. (2012). ImageNet classification with deep convolutional neural networks. In *Advances in neural information processing systems* (pp. 1097-1105)

Lee, H. H., Kim, J. H., & Hong, K. S. (2015). Mobile-based flower species recognition in the natural environment. *Electronics Letters*, *51*(11), 826-828

Li, Z., Peng, C., Yu, G., Zhang, X., Deng, Y., & Sun, J. (2017). Light-head R-CNN: In defense of two-stage object detector. *arXiv preprint arXiv:1711.07264*

Lin, T. Y., Dollár, P., Girshick, R., He, K., Hariharan, B., & Belongie, S. (2017). Feature pyramid networks for object detection. In *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition* (pp. 2117-2125)

Lin, T. Y., Goyal, P., Girshick, R., He, K., & Dollár, P. (2017). Focal loss for dense object detection. In *Proceedings of IEEE International Conference on Computer Vision* (pp. 2980-2988)

Liu, W., Anguelov, D., Erhan, D., Szegedy, C., Reed, S., Fu, C. Y., & Berg, A. C. (2016). SSD: Single shot multibox detector. In *European Conference on Computer Vision* (pp. 21-37). Springer

Liu, Z., Du, J., Tian, F., & Wen, J. (2019). MR-CNN: A multi-scale region-based convolutional neural network for small traffic sign recognition. *IEEE Access*, *7*, 57120-57128

McGlone, M., Buitenwerf, R., & Richardson, S. (2016). The formation of the oceanic temperate forests of New Zealand. *New Zealand Journal of Botany*, *54*(2), 128–155

Najibi, M., Rastegari, M., & Davis, L. S. (2016). G-CNN: An iterative grid based object detector. In *Proceedings of IEEE Conference on Computer Vision and Pattern*

Recognition (pp. 2369-2377)

Oide, M., & Ninomiya, S. (2000). Discrimination of soybean leaflet shape by neural networks with image input. *Computers and electronics in agriculture*, *29*(1-2), 59-72

Peng, C., Xiao, T., Li, Z., Jiang, Y., Zhang, X., Jia, K., ... & Sun, J. (2018). MegDet: A large mini-batch object detector. In *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition* (pp. 6181-6189)

Peng, H. E., & Huang, L. (2008). Feature extraction and recognition of plant leaf. *Journal of Agricultural Mechanization Research*, *6*, 168-170.

Redmon, J., Divvala, S., Girshick, R., & Farhadi, A. (2016). You Only Look Once: Unified, real-time object detection. In *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition* (pp. 779-788)

Redmon, J., & Farhadi, A. (2017). YOLO9000: Better, faster, stronger. In *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition* (pp. 7263-7271)

Redmon, J., & Farhadi, A. (2018). YOLOv3: An incremental improvement. *arXiv preprint arXiv:1804.02767*.

Ren, S., He, K., Girshick, R., & Sun, J. (2015). Faster R-CNN: Towards real-time object detection with region proposal networks. In *Advances in Neural Information Processing Systems* (pp. 91-99)

Schmarje, L., Santarossa, M., Schröder, S. M., & Koch, R. (2020). A survey on semi-, self-and unsupervised techniques in image classification. *arXiv preprint arXiv:2002.08721*

Sermanet, P., Eigen, D., Zhang, X., Mathieu, M., Fergus, R., & LeCun, Y. (2013). OverFeat: Integrated recognition, localization and detection using convolutional networks. *arXiv preprint arXiv:1312.6229*

Shen, Z., Liu, Z., Li, J., Jiang, Y. G., Chen, Y., & Xue, X. (2017). DSOD: Learning deeply supervised object detectors from scratch. In *Proceedings of IEEE International*

Conference on Computer Vision (pp. 1919-1927)

Söderkvist, O. (2001). Computer vision classification of leaves from Swedish trees. Linköping University.

Soltis, P. S., Nelson, G., Zare, A., & Meineke, E. K. (2020). Plants meet machines: Prospects in machine learning for plant biology. *Applications in Plant Sciences*, *8*(6), 1

Sun, Y., Liu, Y., Wang, G., & Zhang, H. (2017). Deep learning for plant identification in natural environment. *Computational Intelligence and Neuroscience*, *2017*

Viola, P., & Jones, M. (2001). Rapid object detection using a boosted cascade of simple features. In *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition*(Vol. 1, pp. I-I)

Viola, P., & Jones, M. J. (2004). Robust real-time face detection. *International Journal of Computer Vision*, *57*(2), 137-154

Wang, C. Y., Mark Liao, H. Y., Wu, Y. H., Chen, P. Y., Hsieh, J. W., & Yeh, I. H. (2020). CSPNet: A new backbone that can enhance learning capability of CNN. In *Proceedings of IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops* (pp. 390-391)

Wang, K., Liew, J. H., Zou, Y., Zhou, D., & Feng, J. (2019). PANet: Few-shot image semantic segmentation with prototype alignment. In *Proceedings of IEEE International Conference on Computer Vision* (pp. 9197-9206)

Wang, X., Shrivastava, A., & Gupta, A. (2017). A Fast R-CNN: Hard positive generation via adversary for object detection. In *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition* (pp. 2606-2615)

Yang, B., Yan, J., Lei, Z., & Li, S. Z. (2016). Craft objects from images. In *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition* (pp. 6043-6051)

Yun, S., Han, D., Oh, S. J., Chun, S., Choe, J., & Yoo, Y. (2019). Cutmix: Regularization strategy to train strong classifiers with localizable features. In *Proceedings of IEEE*

International Conference on Computer Vision (pp. 6023-6032)

Zhang, S., & Huai, Y. J. (2016). Leaf image recognition based on layered convolutions neural network deep learning. *Journal of Beijing Forestry University*, *38*(9), 108-115

Zhao, Z. Q., Zheng, P., Xu, S. T., & Wu, X. (2019). Object detection with deep learning: A review. *IEEE Transactions on Neural Networks and Learning Systems*, *30*(11), 3212-3232

Zhu, Q., Yeh, M. C., Cheng, K. T., & Avidan, S. (2006). Fast human detection using a cascade of histograms of oriented gradients. In *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition (CVPR'06)* (Vol. 2, pp. 1491-1498)

Zhu, Y., Zhao, C., Wang, J., Zhao, X., Wu, Y., & Lu, H. (2017). CoupleNet: Coupling global structure with local parts for object detection. In *Proceedings of IEEE International Conference on Computer Vision* (pp. 4126-4134)

Zou, Z., Shi, Z., Guo, Y., & Ye, J. (2019). Object detection in 20 years: A survey. *arXiv preprint arXiv:1905.05055*