Apple Ripeness Identification Using Deep Learning

Bingjie Xiao

A project report submitted to the Auckland University of Technology

in the requirements for the degree of

Master of Computer and Information Sciences (MCIS)

2019

School of Engineering, Computer and Mathematical Sciences

Abstract

Deep learning technologies aid fruit recognition, allowing a computer to detect a fruit and find its ripeness automatically. Apple ripeness identification refers to such type of patter classifications. In this study, the ripeness of apples will be detected with deep learning and convolutional neural network will be employed. The goal of our experiment is to verify the capability of deep learning in apple orchard to reduce human labour. The experiment consists of 4 parts: data preprocessing, object detection, classification, as well as evaluation.

The major innovations of the present study refer to the multi-class recognition of target detection as well as the application of transfer learning models. Our study is considered the first experiment in the classifying maturity in fruit identification. The major contribution here is the study on the characteristics of the learning objectives that the classifier can balance and then achieve the optimal recognition result, i.e., each position can precisely identify the position of the apple in the picture as well as the category of the apple in this position. Transfer learning means to adopt the optimized network model in the experiment. We have optimized the network, and then balanced the detector trained by the model more to achieve multi-class identification.

Keywords: Faster R-CNN, Groung Truth, Object Detection.

Table of Contents

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: *Bingjie Xiao* Date: 17 Oct 2019

Acknowledgment

This research work was completed as the part of the Master of Computer and Information Sciences (MCIS) course at the School of Computer and Mathematical Sciences (SCMS) in the Faculty of Design and Creative Technologies (DCT), Auckland University of Technology (AUT) in New Zealand. I would like to deeply thank my parents for the financial support they provided during my entire time of academic study in Auckland.

My deepest thanks are to my primary supervisor Dr. Wei Qi Yan who has provided me with much appreciated technological guidance and support. I believe that I could not be able to achieve my master's degree without his invaluable help and supervision. In addition, I would like to appreciate the administrators in our school for their support and guidance through the MCIS in the past years.

Bingjie Xiao

Auckland, New Zealand

October 2019

Chapter 1

Introduction

In the present chapter, the background and motivation of the present study will be primarily presented. Besides, the research questions of this study will be answered. Furthermore, the research objectives will be identified.

1.1 Background

Computer vision refers to an interdisciplinary subject that conducts advanced semantic understanding of digital images or videos. Computer vision aims to exploit machine intelligence to view and achieve visual semantics of visual cortex in our human brain.

Recently, deep learning has acted as a vital tool for deep neural networks-based computer vision (Kendall, 2019). Deep learning technologies first find semantic objects from visual data and then export the output of pattern recognition, which has shown to outperform human visual system.

Computer vison covers 4 major tasks (e.g., pattern classification, detection, semantic segmentation, as well as scene understanding). Pattern classification will interpret the label of a class to which the object appears in an image (e.g., person, building, street, as well as vehicle). For object detection, the target of interests in an image or video will be found. All types of vehicles, pedestrians, traffic signs, and traffic lights are visual objects worth noting. Image segmentation will outline the vehicles and roads in the field of view (FoV), requiring image semantic segmentation as a support to outline the foreground objects in the image. Scene understanding aims to label road name, store name, etc.

An image is considered a two-dimensional grid filled up with a colour pixel in respective cell of the grid (Saurabh, 2017), and each image can be considered a matrix consisting of pixel colour intensity. Computer vision processes the images by analysing the matrices. Our human mind cannot rapidly deal with the image data due to considerable computations. Nevertheless, a computer is capable of identifying a visual object in an image and finding the similarity quickly.

Feature extraction refers to the fundamental problem of digital image processing. Various attributes (e.g., colour, texture, edge, motif, and histogram) of each image contribute to feature extraction. Feature extraction can be conducted for object detection and recognition. Object detection, as one of the primary tasks of computer vision, is critical to image understanding. In this study, the basic knowledge of object detection is to be presented.

1.2 Rationale

The motivation of this study lies in the identification of the fruits based on visual object recognition. A myriad of computational models have been built to identify fruits based on deep learning (Mohamud and Gopalakrishnan, 2018). The trends and patterns for fruit recognition have been surveyed. Based on images for mobile applications, a method capable of recognizing fruits faster due to an effective fruit recognition network has been proposed (Ziliang et al., 2019).

Ripe apples have been taken as a research object. Object detection is performed to detect apples in images and then classify the maturity. The apple ripeness identification will contribute to orchards since the machine is capable of automatically identifying the quality of the fruits rapidly.

This study focuses on the ripeness identification of apples with deep learning. The apples are split into 3 levels (unripe, ripe, and overripe). The success of this study will lead robotics to pick up an apple quickly.

1.3 Research Questions

This study aims to identify the ripeness of apples. Apples are primarily classified into 3 types (unripe, ripe, and overripe). Thus, the research question is raised that:

"Which method can be adopted to detect apples, and how to identify the maturity of the apples in an image?"

The core of this topic is the way to detect visual objects and develop the optimal methods for apple detection. Given the requirement of this research question, the corresponding experiment should locate the apples in an image, classify them and then label them.

For deep learning, object detection does not directly require the features (e.g., colour and shape). The first step of object recognition refers to how to accurately locate the position of apples in an image; how to reduce errors between predicted and real positions of apples is another problem to be considered in this study.

1.4 Contributions

This study primarily discusses apple detection and maturity analysis. The apples will be presented in the form of images and then marked with a rectangle. The experiment covers 4 parts (data preprocessing, object detection, classification, and evaluation).

Deep learning, i.e., deep neural network, refers to a type of machine learning. The concept of deep learning originates from the study of artificial neural networks. A multilayer perceptron with multiple hidden layers has been known as the early type of artificial neural networks. Deep learning aims to build a neural network that simulates our human brain. It imitates the mechanism of our human mind to interpret data (e.g., images, sounds, and texts). Faster R-CNN acts as the object detection method to detect apples through deep learning.

All the images in the dataset are collected by mobile cameras and then preprocessed. There are 3 datasets, covering over 10,000 of images which are employed to train Faster R-CNN network.

In this study, MATLAB and its toolboxes are adopted for the experimentation. Not only will the experiment process be explained, but also deep learning models will be analysed.

1.5 Objectives of the Research

Each apple can be classified into ripe, unripe and overripe. Different images for maturity of apples will be captured to build the training dataset. After data preprocessing, labels are assigned to automatically annotate apples in images with deep learning. There are 4 steps in visual object detection: data preprocessing, data labelling, object detector training, and model evaluation.

1.6 Report Structure

The first chapter of this study will present the research background. How computer vision associated with deep learning is explained. Besides, the research goal will be provided in the present chapter. The research question and research objectives are elucidated.

In the second chapter, literature review is depicted, and research methods for the experiments are recommended. All the algorithms and approaches applied in object detection are referenced. The understanding of the apple classification is summarised.

The third chapter will elucidate the experiments, e.g., how to collect data and detect the apples in the images. Which network is taken in the experiment, and how this study can be carried out will be discussed in this chapter.

The final 2 chapters present the results and evaluations of the experiments. We will explain why the results look like this and how to improve this study. An overall summary about this study will be summarized.

Chapter 2
Literature Review

Given the requirements of object detection, this literature review chapter will introduce the vital research methods that can help achieve apple ripeness identification. Data preprocessing, object detection, classification, and evaluation methods from other papers will be reviewed in this chapter.

2.1 Related Work

Multiple methods have been used in apple recognition. The first method finds edges, corners, colours or other signatures as extracting features from an image to help classify objects. Though our human brain is proficient in classifying visual objects, which feature will be taken major effect in human brain processing remains unclear. Conventional machine learning methods have often been performed by extracting a range of features from digital images. In fact, by long-time experimentation and analysis, machine learning algorithms are subsequently used to identify visual objects in the images based on these extracted features.

The second method still aims to extract features that help identify visual objects in the image. However, instead of using feature extraction, automated programs have been adopted for learning process. The salient features will be extracted from the raw image data. Artificial neural networks (especially deep neural networks) are trained to use considerable images. A deep neural network may have more layers of neurons in the end-to-end manner, in which each layer is connected to the next layer and can learn higher-level features of the input images.

2.2 Our Experiment Design

Machine learning can fall into 3 types (unsupervised learning, supervised learning, as well as intensive learning). Unsupervised learning directly contributes to data analysis. The learning algorithms usually obtain features or knowledge from the given samples. There has been no training process in the unsupervised learning. Clustering is a representative of unsupervised learning.

In this study, supervised learning and unsupervised learning are mixed. There are 1,000 images, covering various maturity of apples in the dataset. Given the requirements of this study, 3 types are predefined, and the 1,000 samples are classified into 3 types. The clustering algorithm has classified the 1,000 apple images together and counted the accuracy to ensure the apple images with the identical maturity in the same cluster.

Supervised learning is adopted to train a model with a labelled dataset and conduct pattern classification. In this study, we should collect images of 3 types of apples and manually classify each apple in one image (e.g., mature, unmatured, as well as postmature). Subsequently, we can build a model in deep learning and adopt this model to predict unknown maturity of apples.

Classification algorithm conducts predictions in line with the samples from the dataset to which it belongs to. Clustering algorithm ensures that the samples of the identical class are similar, and that the samples of different classes are distinct.

Object detection refers to a method locating the objects in an image. Object detection is more important than object recognition. By object detection, multiple objects can be detected from one image. Object detection is achieved using multiple methods from convolutional neural networks and others.

2.3 Unsupervised Learning

Unsupervised learning has been applied for object analysis in image processing (Chen, & Wang, 2018). The hierarchical clustering and partition clustering suitable for multi-dimensional intersections analyse are introduced. Partition clustering algorithm builds K clustering groups covering N objects. K denotes the number of groups and the input parameter of algorithms. Iterative computations are conducted to optimize the model by determining the initial partition. The number of clusters should be determined, and each object should be assigned to the closest object.

2.4 R-CNN, Fast R-CNN and Faster R-CNN

Liu et al. (2017) elucidated how object detection is achieved using Faster Region-Convolutional Neural Network (Faster R-CNN). Faster R-CNN starts from R-CNN, and the development undergoes Fast R-CNN.

R-CNN adopts selective search algorithm to extract region proposals from the top of the image to the bottom. The input of the CNN is scaled using each region proposal. Convolutional layers do not require a fixed size of images, whereas fully connected layers require a fixed size input. Support Vector Machine (SVM) is adopted to classify the features. The classified region proposals will use bounding box regression to achieve border regression and will generate the predictive coordinates. R-CNN calculates the convolution for respective region proposal. Nevertheless, R-CNN covers with multiple stages of trainings, so the steps are cumbersome and time consuming.

Fast R-CNN also uses selective search algorithm to extract region proposals, and a whole picture is imported into CNN. Softmax and L1 loss function are employed to be trained together with bounding box regression. Fast R-CNN normalizes the images and introduces the proposed box into the feature map outputted by the final convolution layer. It will be not necessary to repeat the calculation and store large data since the Fast R-CNN should only extract features and suggest areas one time. Unlike R-CNN, ROI (Region of Interest) pooling layer is introduced to the Fast R-CNN in the last convolutional layer. The sizes of the input images are different, leading to the various feature maps. Thus, a fully connected layer cannot directly be employed for classification. Fast R-CNN adopts ROI and pooling layer for each region.

Fast R-CNN remains time-consuming for searching region proposals, which Faster R-CNN solves this problem using region proposal network (RPN). Faster R-CNN imports the entire image into CNN and generates the feature maps. The maps will be inputted into the region proposal network to yield the feature maps of the bounding boxes. Subsequently, a classifier is used to determine which class the objects belong to, and the bounding box positions are adjusted with a regression.

2.5 Object Detection

Bodla's team introduced non-maximum suppression to achieve object detection (Bodla, Singh, Chellappa, & Davis, 2017). The work used basic dataset (e.g., PASCAL VOC 2007 and MS-COCO). All the detection boxes were sorted by the scores. A predefined threshold was adopted to suppress the overlapping with the detection box that achieves the maximum score. The suppress process is recursive on all boxes. There will be a miss if the location of the object overlaps the predefined position. In this work, a Soft-NMS continuous function is adopted for decaying scores of all the objects, so the detection box can overlap with the detection box. Soft-NMS function enhances mean average precision by changing NMS algorithm, and no more parameters are required to be changed. Soft-NMS can be efficient since it is identical to conventional NMS methods, and extra training is not required to be achieved during implementation.

Byeon and Kwak (2017) compared the performance of Faster R-CNN-based object detection and that of ACF-based object detection. Faster R-CNN object detector can find the candidate area of detectable objects and extract feature vectors from each region proposal. ACF

object detector computes several channels and then incorporates them into smoothed channels to yield low resolution.

Decision tree and boost algorithm will be adopted to separate the objects and backgrounds. One method adopts aggregate channel as a block to extract features via several channels. The limitation of this method lies in running multiple detectors simultaneously through filters. With this method, object detector based on Faster R-CNN exhibits better performance, and the precision is higher than ACF object detector.

A method using Fruits-360 dataset has been proposed for fruit recognition (Ziliang, Yan, & Tianbao, 2019). Fruits-360 dataset covers 81 classes and 55244 images in total. With the use of thinner factor, the size of vanilla network can be reduced to perform depth separable convolution neural network. By adapting the depth of the convolution neural network, the cost of computation can be diminished, the robustness can be enhanced, and the performance of the model can be enhanced. This work also uses transfer learning to solve overfitting problem and reduce the long training time.

Based on Faster R-CNN, a hybrid detector was proposed for partially occluded object detection (Hus et al., 2018). This work emphasizes the implication of the network depth. However, even if they used a network with the depth from sixteen to thirty, the accuracy would not be further enhanced. Exploding gradients presents the accuracy because it hampers convergence at the beginning of the training processing. Normalization can converge the ten layers network using stochastic gradient descent and backpropagation. With the increase in the depth of the network, degradation problem occurs, and the accuracy is down-regulated. Reducing the number of network layers properly will not adversely affect the accuracy of the model. In contrast, excessive network layers can cause more errors. Using multiple part detector should solve the problem of partial occlusion.

A classification based on 22 layers GooLeNet and ILSVRC 2014 was employed to incorporate selective research model into high predictions object bounding (Szegedy, Liu, Jia, Sermanet, Reed, Anguelov, Erhan, Vanhoucke, & Rabinovich, 2015). A deep neural network Inception for computer vision was employed to deepen the network. ILSVRC 2014 classification model sets bounding boxes around the objects and identifies overlapping of the bounding box. The bounding boxes should be over 50% to be counted as feasible. In the

inception architecture, all the convolutions use rectified linear activation. The parameters exert minor impact on the inception architecture.

The object detection methods primarily stress the difference between objects and background; they cannot achieve high performance using the low-level cues (Wang et al., 2016). Object detection methods cannot represent advanced semantic features. The output of convolutional neural network is rough without any distinct boundary. Segmentation is employed to solve this problem. The framework of segmentation exploits advanced semantic characteristics and outputs scores for respective region. The significance of each image is assessed using the scores of each area. The areas are split with the edge retention method, the significance of the map is yielded with clear boundaries. Segmentation can split objects into parts, whereas the entire object is difficult to highlight.

Single Shot Multi-Box Detector (SSD) facilitates the real-time object detection and enhances the portability of the model (Dong, et al., 2019). The Batch Normalization (BN) layer of MobileNet has been adopted to compress the network. This model is capable of predicting the default boxes via feature maps on multiple layers. The default boxes are employed to classify and regress the bounding boxes. The SSD model is capable of predicting the convolutional output from each layer and achieving the mean average precision. To ensure the speed of real-time detection, the SSD method omits the target frames randomly.

The visual data collected from real world can be blurry, rotated, and jittered with noises (Liu et al., 2018). Noise and invalid data can adversely affect the result of object detection. In this work, an image degradation model was created based on YOLO. A mathematical model is built to generate standard degraded image datasets, which is used to train YOLO object detector.

2.6 Data Augmentation

Sufficient labelled data significantly impact classification. Data augmentation can resolve this problem. Insufficient and ill-posed data affect combatting overfitting. In fact, numerous datasets are expensive. A deep adversarial method has been developed to formulate data augmentation and supervised generative adversarial network (GAN). With a loss function for the discriminator of GAN to classify the real images and multiple “fake classes”, this fine-grained classification method is referred as 2k loss and contrast to k+1 loss with GANs. The augmented data generated should be discriminative in the classification as well.

2.7 Previous Work

Object detection is a critical research area for computer vision. Object detection methods fall into conventional machine learning algorithms and deep learning algorithms. Deep learning algorithms (e.g., R-CNN, Fast R-CNN, Faster R-CNN, and SPP-Net) are also classified into 2 types. One aims to generate and classify a region proposal network, and the other is to generate and classify the region proposal area simultaneously using YOLO and SSD, etc. How those networks are built is listed in Table 2.1.

Faster R-CNN refers to Region-Based Convolution Neural Network exhibiting high detection efficiency and high accuracy. R-CNN object detection method adopts a sliding window to scan the whole picture and then detect them respectively. If the detector detects objects, it will be recorded; otherwise, the detector will ignore the region without objects until all the region proposal areas have been detected. Accordingly, R-CNN object detector does not apply to practical application. The process of conducting independent scans by various regions in one image and recording the region of interests is known as multi-stage object detection.

The convolution process convolutes an image into feature maps that retain the position information of the original objects. Fast R-CNN repeats the original multi-stage method, whereas it compresses the steps into one. Fast R-CNN added feature maps with the “Selective Search” method for detection. The region of interest (ROI) search is not fixed. All the images are uniformly connected to the full connected layer of the identical dimension based on the Spatial Pyramid Pooling. The multiple loss functions exploit regression during the training process and then optimize the results. Faster R-CNN method inherits the Fast R-CNN. The only and critical change refers to the object position prediction mechanism. The original selective search method has too much computational redundancy in the process. Thus, the RPN (Region Proposal Network) is generated and acts as the mainstream object detection method in the R-CNN family.

Networks	Explanations
R-CNN	Searching possible candidate boxes as soon as possible based on color, edge, etc. Counting each image block once. Applying SVM for classification. Regression only service for the final classifier.
Fast R-CNN	Using regression algorithm for end-to-end training. Using ROI pooling layer to yield vector features. Using multiple task loss for CNN training.
Faster R-CNN	Using RPN to locate the region area directly. Using anchors to outputs 2,000 proposals. Linking outputs to ROI pooling layers.
YOLO	Viewing the entire image while testing for prediction. Using a single network for forecasting.
SSD	Scaling different aspect ratios and the position of each feature map to discretize the bounding boxes in the output space into a set of default boxes. Giving scores each type existing in each default box and adjusting the box for a better object bounding box.

Table 2.1 Comparisons of different networks.

The SSD model can be significantly affected by the size of the bounding box; it performs poorly on small targets detection. After multiple layers convolution, less information is left for small targets. Though increasing the size of the input images can facilitate the detection of small objects, convolution problem remains and affects the detection models.

Compared with the RCNN series object detection model, YOLO network exhibits lower accuracy in identifying the position. The prediction of 2 boxes in each grid reduces multiple times detection of the identical target. The region proposal method displays more overlapping.

According to the existing work, we select Faster R-CNN object detector for this study. After the convolution operation, the efficiency of training is enhanced.

Chapter 3
Methodology

This chapter primarily elucidates the experiment design and how the methods are conducive to the apple ripeness identification. The whole process of apple recognition is illustrated in this chapter. Besides, the experimental tools are also detailed in this chapter.

3.1 Experiment Design

Object detection refers to a problem of using neural networks to locate objects. This suggests that we should not only determine whether there is an apple in an image, but also mark the object. Positioning implies to determine the specific location of the apple in an image.

On the whole, the classification problem has only one large target. Nevertheless, in the object detection problem, one image can involve multiple objects with different types in a single image. Thus, image classification can help learn apple recognition, while positioning helps find an apple position.

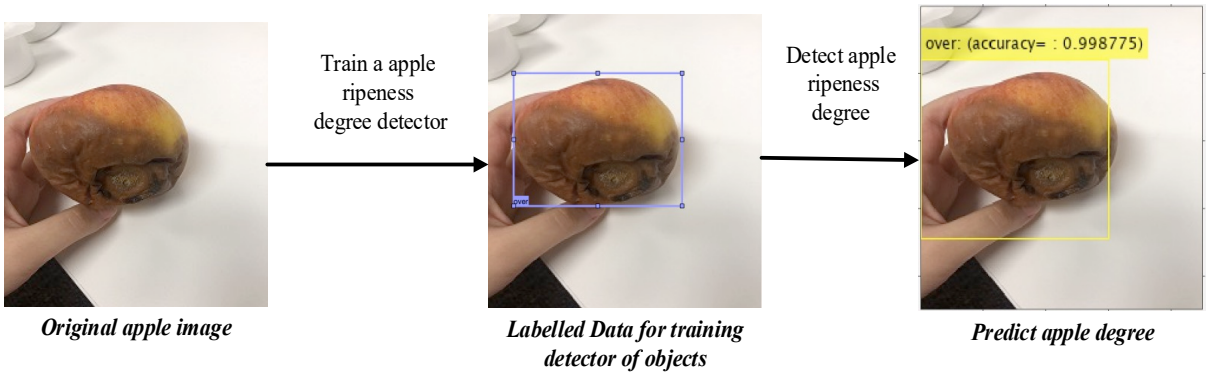


Figure3.1 The process of apple ripeness detection.

In Figure 3.1, we should detect an apple in the image and draw a bounding box around it. This often involves 2 operations, i.e., to predict the type of target and to draw a box around the target, which is termed as data preprocessing. After all images are marked, we use MATLAB to train a Faster R-CNN detector for apple detection. The diagram of the whole entire process is given in Figure 3.1.

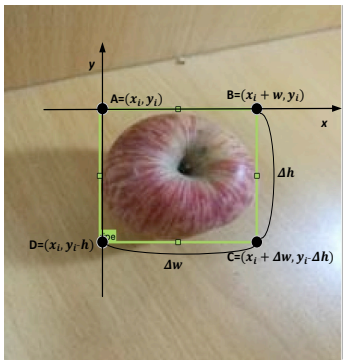


Figure3.2 A bounding box.

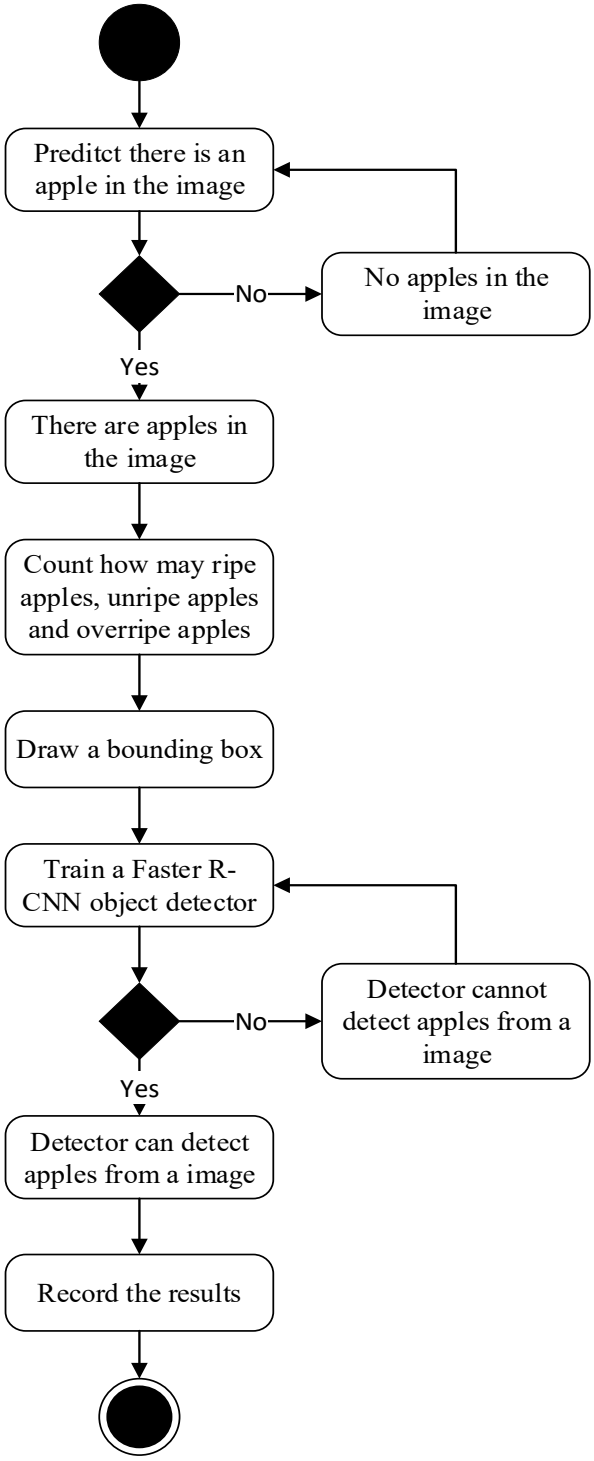


Figure3.3 The whole procedure of this study.

3.1.1 Define Goal

A single image acts as a bounding box. To locate the image, we should build a neural network and output the bounding box. To be specific, the neural network outputs 4 points, which are

denoted as A, B, C and D respectively. These 4 points represent the bounding box of the detected object. In the bounding box, w_i denotes the width, and h_i is the height of the detected objects. The neural network is capable of detecting the target object by outputting the (x_i, y_i) coordinates of the 4 points on the picture.

Thus, the training set covers not only the object labels to be predicted by the neural networks, but also 4 corner points of the bounding box. The supervised learning algorithm is employed to output a classification label and the coordinates of 4 points.

3.1.2 Object Detection Using Faster R-CNN

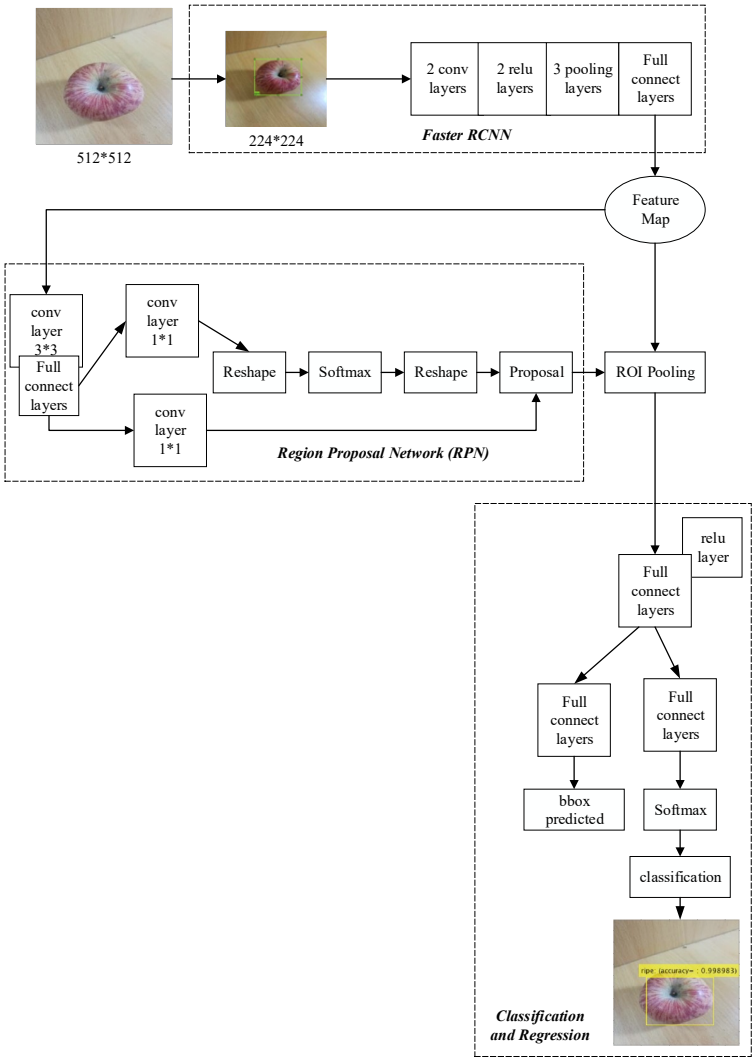


Figure3.4 Apple detection using Faster R-CNN.

RPN network and Fast R-CNN share the identical CNN, the input value can be considered feature maps, and the output can be a plurality of candidate regions. Obviously, the Faster R-CNN consists of 4 parts:

(1) Convolutional layer

The input is the whole image, and the output refers to the extracted features, termed as feature maps. Faster R-CNN first supports inputting images of any size. Before accessing into the network, the image is normalized and scaled. For instance, the short side of the image can be set no more than 224, and the long side of the image is set no more than 224; m×n is assumed as 224×224.

(2) RPN network (Region Proposal Network)

This network is adopted to replace the previous search selective in R-CNN. The input is an image. RPN aims to use the convolutional neural network to yield region proposal directly. The method aims to slide a window over the last convolution layer for border regression to get multi-scale anchor boxes.

The RPN network also refers to a fully-convolutional network (FCN), which can be trained by end-to-end process in deep learning for the tasks to yield the predicted anchor boxes with boundaries and the scores of visual object.

(3) ROI pooling.

The mapped area is split into sections of the identical size, and the number of sections is identical to the dimension of the output. ROI pooling allows the network to yield fixed-size corresponding feature maps from the boxes with various sizes.

(4) Classification and regression.

By outputting the precise location of the candidate region in the image, the candidate region can be predicted. RPN and Fast R-CNN share the identical feature. During the first iteration, the model, obtained using ImageNet, initializes the parameters of the convolution layer in RPN and Fast R-CNN network, which is achieved with the shared convolution layer parameters of Fast R-CNN from the second iteration of RPN training. The shared convolutional layer parameters in RPN only correspond to those of the convolutional layer and other layers that are not shared by fine-tuning. When Fast-RCNN is being trained, the convolutional layer parameters are shared with the RPN unchanged, and only the parameters representing the layers will not be shared.

3.2 Data Description and Preprocessing

3.2.1 Original Data

The following set shows typical images of dataset.

(1) Single apple in one image.
(2) Multiple apples with one mature one in an image.
(3) Multiple apples having strong interference in one image.

Figure3.5 Original data samples captured by mobile phone camera.

3.2.2 Labelled Data

In this study, MATLAB Image Labeller is applied to locate apples in images. The rectangle with the label represents our marked bounding box. The area in the bounding box refers to the Region of Interest (ROI). After all ROI areas are labelled, the images will be stored for future use.

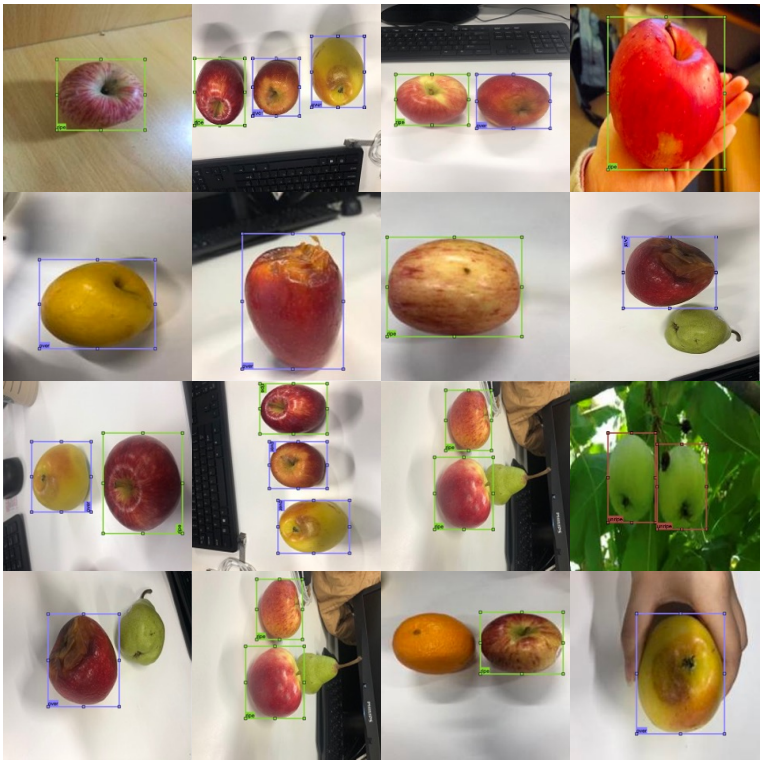


Figure3.6 Labelled data for experiments.

3.2.3 Data Augmentation

A feasible neural network model requires considerable training data. Thus, rising the number of training images can improve the precision of a deep learning model. Data augmentation costs less, and it is easier to acquire novel data using exist images by flipping, panning, or rotating. Two methods can be applied for data augmentation:

(1) Offline data augmentation can process visual data directly, and the enhancement factor helps enrich original datasets.

(2) Online data augmentation more applies to large datasets training with machine learning frameworks, which can be optimized using GPUs.

Figure3.7 Image samples after data augmentation.

In the experiment associated with apple ripeness identification, offline data augmentation is selected. There are 2 type of datasets in this study, one with data augmentation method, and the other without data augmentation. 3 types of data augmentation methods are used.

(1) Zooming

The image can be enlarged or shrunk. The original dataset is scaled down, and the size is modified to 224×224 and 512×512.

(2) Rotating

Rotation refers to a clockwise or anticlockwise rotation. Each experimental image is rotated six times anticlockwise at an angle of 10 degrees.

(3) Noising

Overfitting usually occurs when neural networks learn high-frequency features since low-frequency features are easy to learn. To eliminate high-frequency features, noisy data are randomly added. Gaussian noise and salt-and-pepper noise are employed to achieve data augmentation.

3.2.4 Interference of Image Data

Figure3.8 Image samples with noises.

The model is not tested with degraded images, and the existing Figure3. 9models do not apply to randomly captured images. However, the images collected using mobile phone cameras in practice have excessive problems.

(1) The captured images can be blurred due to camera shaky.
(2) The objects are insufficiently clear for the overlapped apples or obstructions.
(3) The images may exhibit poor quality due to image resizing or compression.

After the first-round training, invalid data is excluded, including:

(1) Bounding box covers the image, so the ROI area cannot be marked.
(2) Different bounding boxes have excessive overlapping area.
(3) The expected ROI areas are hidden, and the features are inconspicuous.

3.2.5 Quantity and Quality of Image Data

Due to the experimental requirements, 3 datasets are established with more than 10,000 images.

Our apple images are captured indoor with high quality. The ripe apples are provided from supermarkets until they gradually turn overripe. During this procedure, the numbers of each image class is calculated twice, and the accumulated value is larger than the real number of images. Nevertheless, the total number of the experimental images is stable.

Unripe apple images are captured outdoors. Due to the conditions (e.g., lighting and weather), the collection of such experimental pictures is disturbed, resulting in numerous invalid data. The visual objects in the picture are occluded sometimes, covered or hooked; they cannot be presented clearly. Besides, there is more than one object on one image, and the number of labels will be noticeably larger than the total number of experimental images.

Category	Ripe	Overripe	Unripe	Total
Number of Images	144	111	92	280
Number of Labels	552	452	409	1413

Table3. 1 The Dataset I.

Category	Ripe	Overripe	Unripe	Total
Number of Images	1325	1083	1040	2880
Number of Labels	5416	4475	3976	12967

Table3. 1 The Dataset II.

Category	Ripe	Overripe	Unripe	Total
Number of Images	4149	3713		7064
Number of Labels	12564	10812		23376

Table3. 3 The Dataset III.

Dataset I and Dataset II refer to randomly datasets used to test whether the quantity of images affect the results. Dataset III has been enhanced with data augmentation.

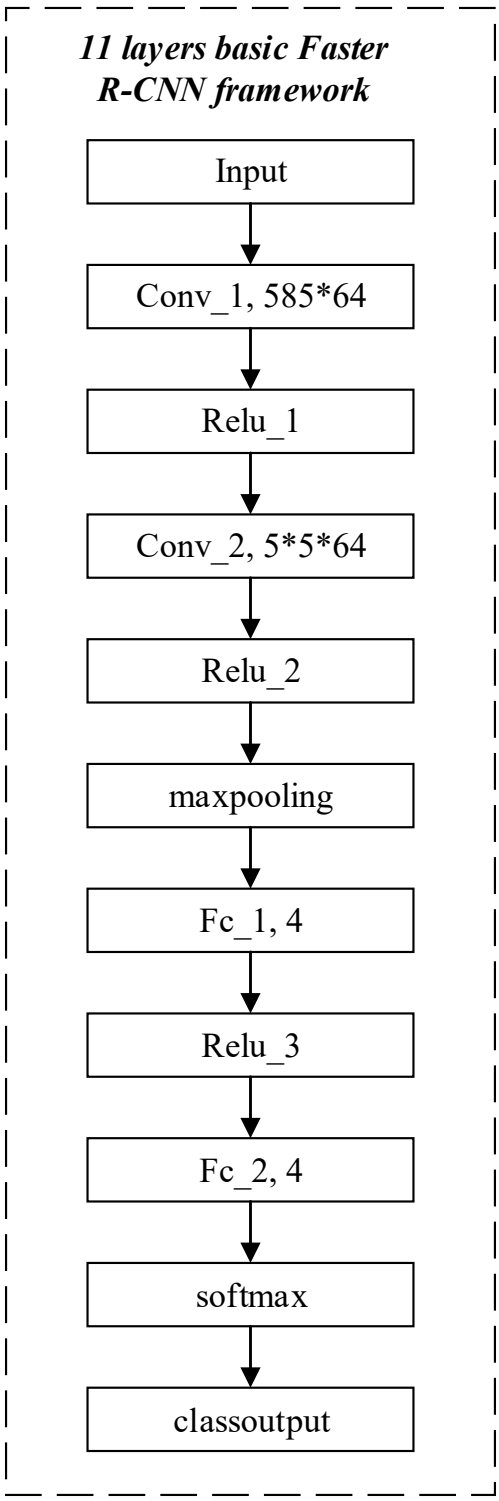


Figure 3.9 A sketch of simple network.

Subsequent analysis will be conducted for the large interference of the unripe apple images. This type of data is temporarily deleted during the third round of experimentation in Dataset III. In this study, all of 3 datasets are randomly split into training set and test set. All the datasets are resized to a specific size.

3.3 Neural Networks

3.3.1 11-layer Faster R-CNN Network

A Faster R-CNN network is used for apple ripeness recognition. First, an image dataset is inputted and passed to a convolutional neural network, and then a feature map of the image returns. Subsequently, the RPN returns the proposed object and the scores for ROI pooling layer to reduce all proposals into the identical size. Lastly, the proposal will be passed to fully connected layers using a softmax layer and a linear regression layer to classify and output the bounding box of the object.

3.3.2 ResNet-50

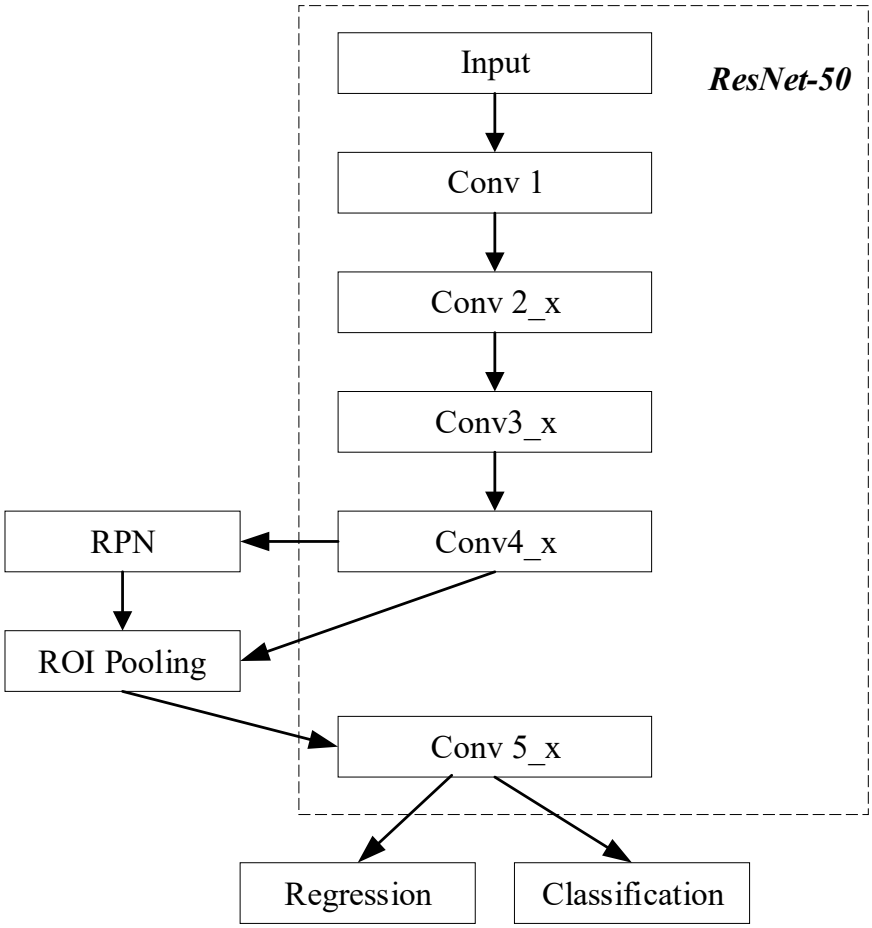


Figure 3.10 How Faster R-CNN works with ResNet-50.

There are 4 groups of blocks in ResNet-50, and each group covers 4, 6 or 3 blocks. There are 3 convolutional layers in each block of ResNet-50. ResNet-50 is a type of residual network, allowing the network to be as deep as possible. Theoretically, the accuracy decreases with the

rise in the depth of the network. ResNet-50 provides 2 options (identity mapping and residual mapping) to solve this problem.

How ResNet-50 uses shortcut connection method to achieve the residual network is shown in Figure 3.11. Identity mapping refers to the curve, and residual mapping refers to the part of the curve.

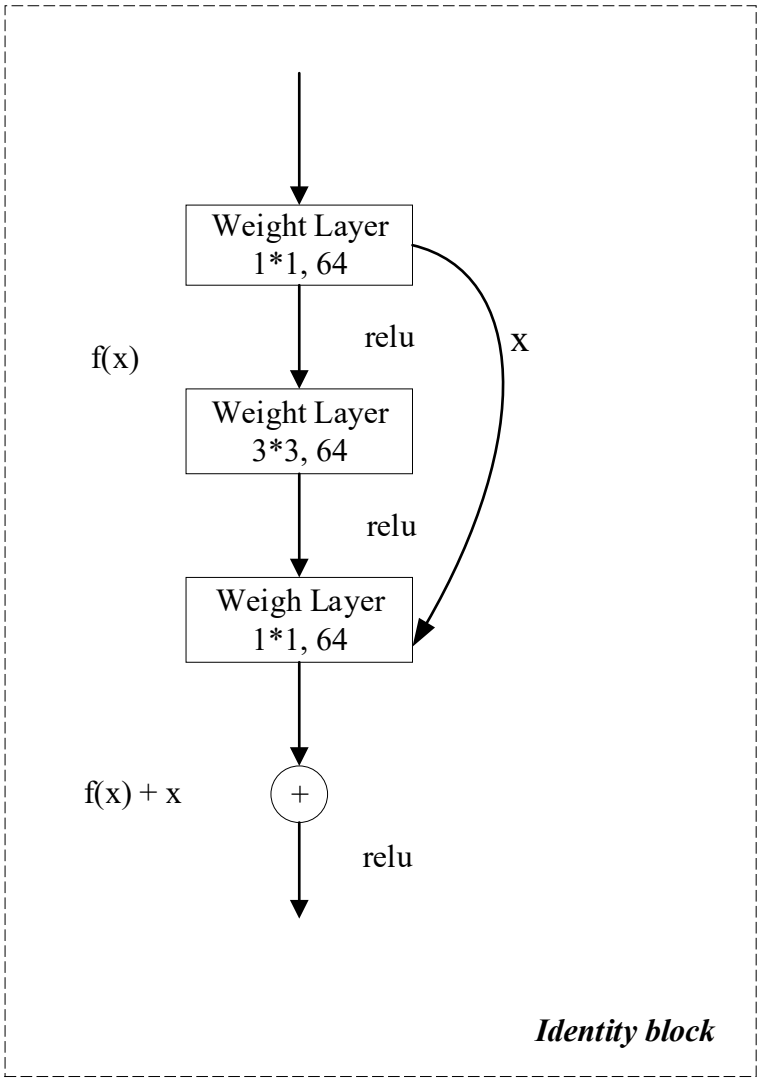


Figure 3.11 The identify block of ResNet-50.

If the network has reached an optimal and the network continues being deepened, the residual mapping will be pushed as 0. Thus, the identity mapping will keep the network in a theoretically optimal state. The performance of ResNet-50 will not decrease as the depth rises.

Identity mapping refers to x in Figure 3.11, residual mapping refers to $f(x)$, the residual value refers to $f(x) = f(x) + x$.

The whole structure of ResNet-50 refers to a building of blocks. The identify clock is also termed as the bottleneck design, and the purpose is to down-regulate the number of parameters for better performance. The first 1×1convolution reduces the 256D channel to 64D and finally returns to 1×1 convolution.

Shortcut connection $f(x)$ is added in line with the numbers of channel dimension. For a different channel, ResNet-50 creates a basic convolutional block. Identity block has the same input and output dimensions which can be used for concatenation. Convolutional block is with different input and output dimensions, and it cannot be connected in series. The role of convolutional block is to modify the dimension of the feature vectors. Convolutional neural network contributes to convert the image into a feature map. With the rise in the depth of networks, the output channels rise. The dimensions should be modified before entering the identity block.

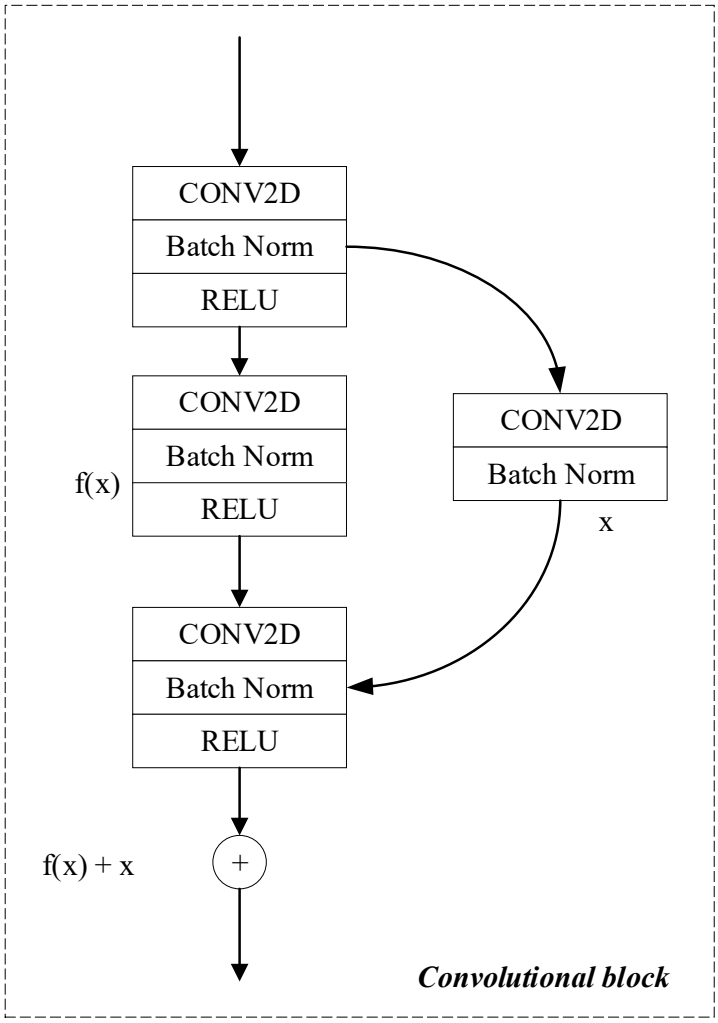


Figure 3.12 The convolutional block of ResNet-50.

3.3.3 ResNet-50 Transfer Learning

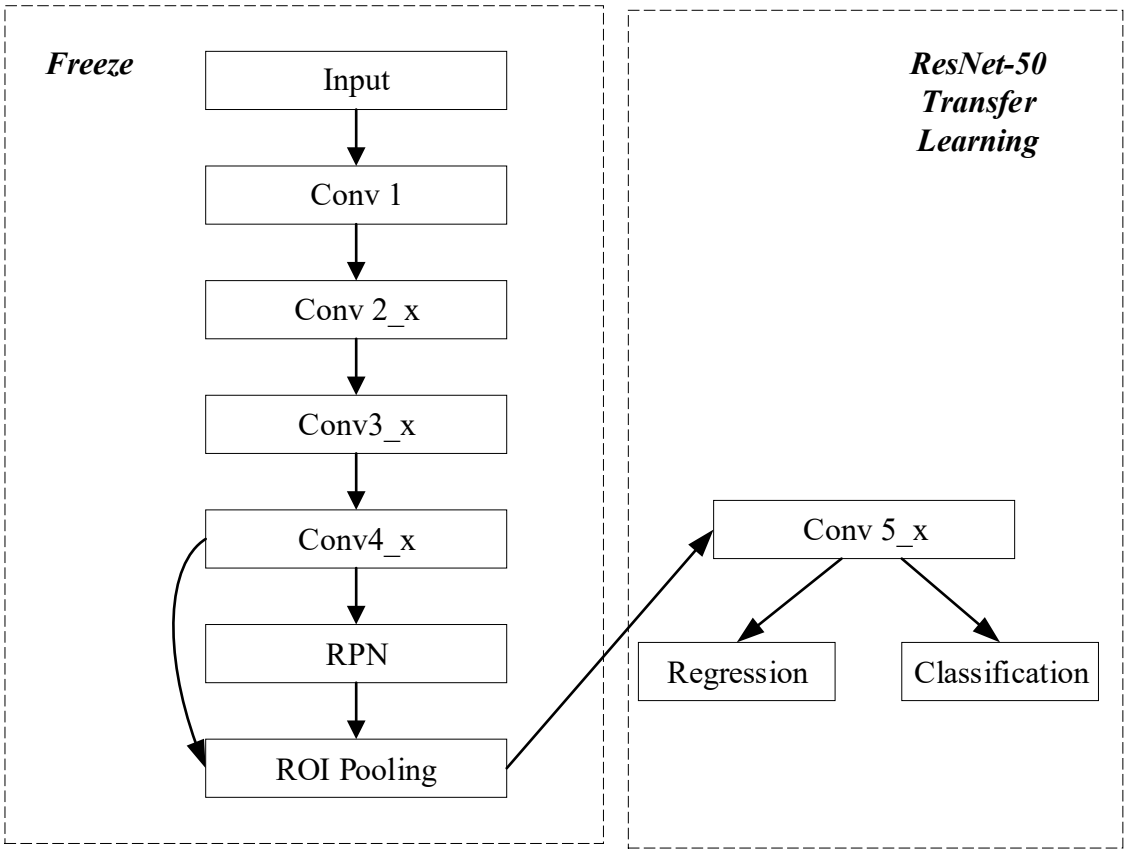


Figure 3.13 The diagram of transfer learning using ResNet-50.

For the case where the dataset is small, it is unlikely to train a large neural network with millions of parameters. A larger model requires more data otherwise the overfitting problem cannot be avoided. Transfer learning is suitable for applying the robust feature extraction ability of large-scale neural networks.

The 3 datasets exhibit low data volume and low data similarity. In this case, fine-tuning for the network is suitable. Freezing most of the convolutional layer near the input partial of the pre-training model, as well as training partial convolutional layer close to the output and the fully connected layer are likely to be achieved.

In the experiment, the initial 4 blocks of ResNer-50 are frozen, and the remaining blocks are trained again. We truncate the last 3 layers of ResNet-50 and replace it with the new softmax layer associated with apple ripeness identification problem. Also, novel regression layers are added, and the parameters of RPN network are reset.

3.3.4 GoogLeNet

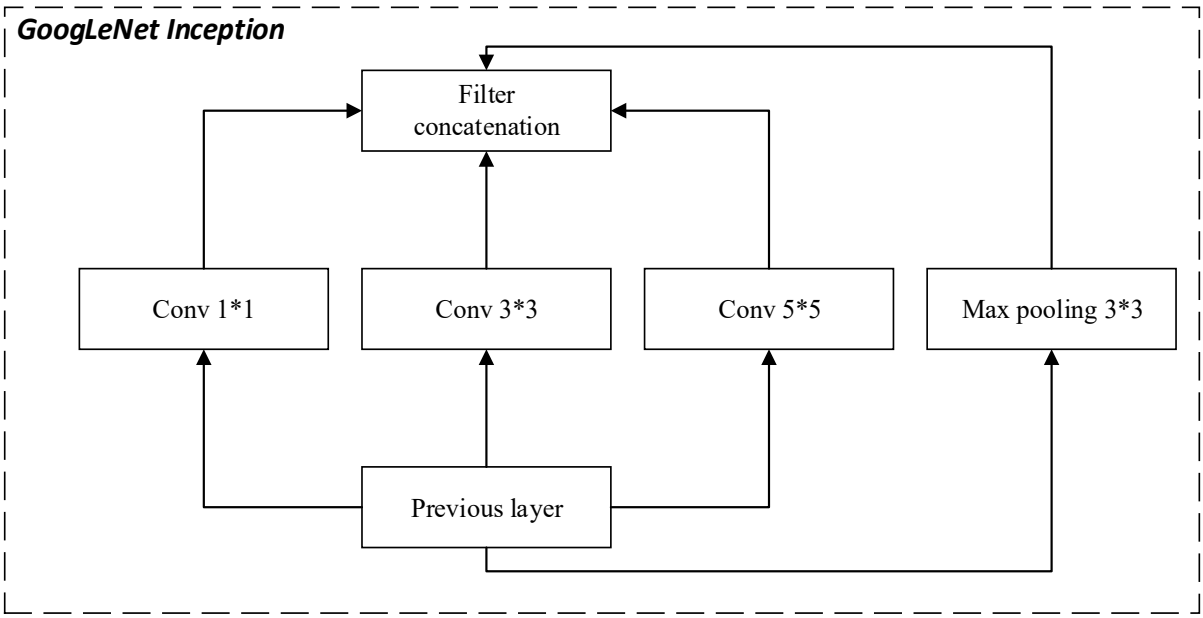


Figure 3.1 The diagram of GoogLeNet.

It is known that the safest way to get a high-quality network model is to increase the depth (layers) or width (number of neurons) of the model. However, defects will occur. Excessive parameters will result in overfitting with the quantity limitation of training dataset. A larger network can cause computational complexity, and it is difficult to grain. The backward gradient will be easy to varnish and hard to optimize if the network gets deeper.

A sparse connection model is recommended to convert a fully connected layer or a general convolution layer. Dataset I and Dataset II are nonuniform sparse data, and the conventional sparse connection model cannot break the network and improve the learning ability.

GoogLeNet can maintain the sparseness of the network structure and the highly utilize dense matrices. GoogLeNet is a 22-layer network using inception structure, being easy to add and modify layers. Average pooling layer replaces the fully connected layers in the end of the network. To avoid the gradient varnishing, the network additionally adds 2 auxiliary softmax layers for the forward conduction gradient. There are 2 auxiliary classifiers in GoogLeNet. Classifiers are capable of adding smaller weights to the final classification, i.e., the model is better for parallel operations and equivalent to model integration. GoogLeNet provides additional regularization.

3.4 Bounding Boxes

There should be 3 types of boxes in object detection for analysing the objects and detection results. As shown in Figure 3.15, gTruth box, anchor box and predicted box represent the marked result, the detected result, and the predicted result, respectively.

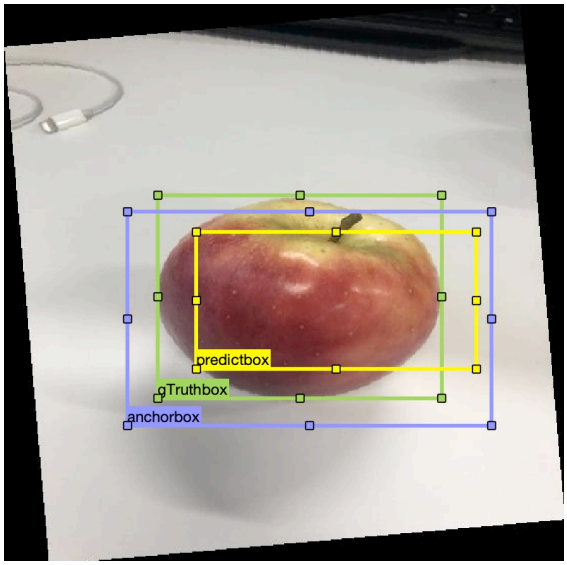


Figure 3.2 Boxes of Ground Truth labelling sample.

Ground truth boxes are labelled manually, which are the location of the object in an image. The original image and the corresponding ground truth are shown like this. Likewise, all images in the training dataset are labelled. The original images are sent to train the model, and the apple detection algorithm gives prediction results with bounding boxes if the confidence is greater than a threshold.

On the whole, the anchor box in an image is not generated by a training dataset, whereas it is actively designed by program. Anchor box is usually distributed across the image. The essence of the anchor box is the sliding window which is traversing the image. A cell will represent multiple anchor boxes. The quantity of anchor box is also considered set at the initialization stage. For instance, if we cut an image in to 3×3 grids and set 2 anchor boxes, those 2 boxes will pass all the grids and achieve predications.

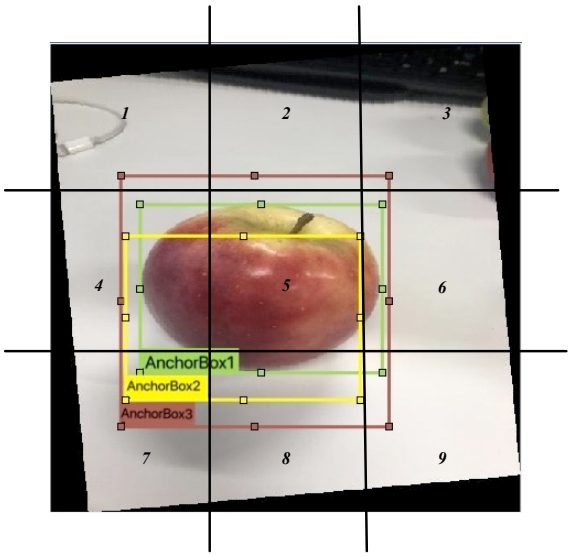


Figure 3.3 A sample of anchor box.

c = 1 is defined as there is an apple in the image, while c = 0 denotes that there is not an apple. Thus, c_1, c_2, c_3 represent 3 types of ripe apples, overripe apples, and unripe apples, respectively. The input is the whole image and the output is a target label T. The target label T in the anchor box means a vector.

$$T = \begin{bmatrix} c \\ x \\ y \\ w \\ h \\ c1 \\ c2 \\ c3 \end{bmatrix} \tag{3.1}$$

The detector model is to find a loss regression that fits the following mapping,

$$(A_1, B_1, C_1, D_1) = (A_2, B_2, C_2, D_2) \approx (A, B, C, D) \tag{3.2}$$

The predicted box is generated based on the prediction of the anchor boxes. Non-maximum suppression (NMS) allows each box to retain only one prediction bounding box. NMS arranges all the predicted probabilities from high to low. Each box only retains the one with the maximum probability. There will be only one with the maximal probability of respective box prediction value.

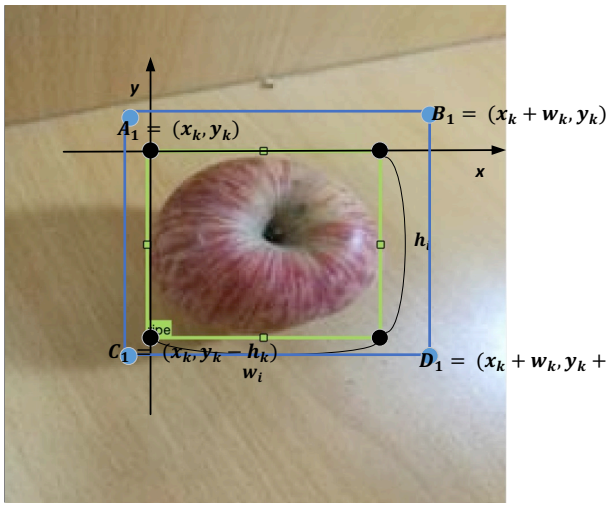


Figure 3.4 A sample of anchor box predicted box.

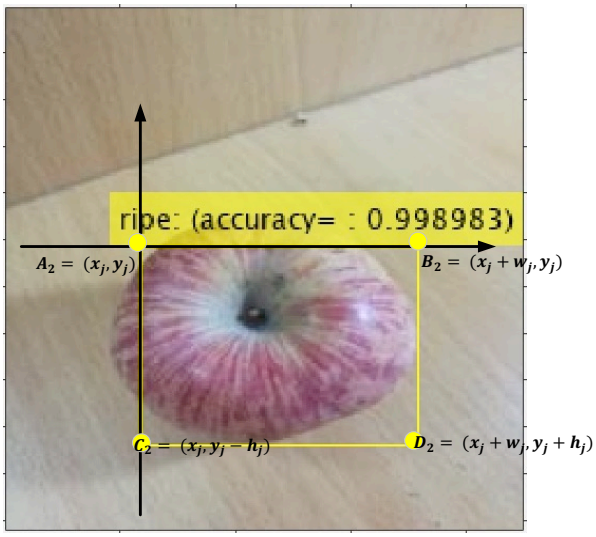


Figure 3.5 A sample of non-maximum suppression (NMS) predicted box.

In Figure 3.17, A_1, B_1, C_1 and D_1 denote the predicted anchor box using detector. The offset is as the following:

$$\Delta x = (x_i - x_k)/w_k \tag{3.2}$$

$$\Delta y = (y_i - y_k)/h_k \tag{3.3}$$

$$\Delta w = \log(w_i/w_k) \tag{3.4}$$

$$\Delta h = \log(h_i/h_k) \tag{3.5}$$

where x_k and y_k denote the midpoint of the predicted apple location, h_k is the number of vertically divided anchors, w_k is the number of horizontally divided anchors. x_i, y_i, w_i and h_i are the values in Figure 3.18. Δx, Δy, Δw and Δh should be the required pan value and scale scaling value calculated by using region proposal. In fact, w_i and h_i may be larger or smaller than w_j and h_j, whereas the actual range (0,1) reveals that the predicted result is the time of anchor box width. The loss function makes Δx, Δy, Δw and Δh minimum.

$$Loss = \sum_{\Delta}^{N}((\Delta x \quad \Delta y \quad \Delta w \quad \Delta h) - \begin{pmatrix} x_j & y_j & w_j & h_j \end{pmatrix}^T \emptyset_5(x_k \quad y_k \quad w_k \quad h_k)) \tag{3.6}$$

The size of height is not associated with the anchor box, the ground truth box, or the predicted box. Thus, each box can be larger or smaller than the other boxes. Coordinate values of width and height are not absolute values since ROI pooling layer resizes each anchor box to the identical size.

3.5 Evaluation Methods

A well-trained target detection model gives extensive predictions, whereas most of the predictions are with low confidence scores, the results will consider predictions where the confidence reaches over a threshold. Intersection over Union, accuracy, precision and recall are all available for object detection model assessment.

3.5.1 IOU

Intersection of Union (IOU) refers to an area between the overlapping area (intersection) and the ground truth bounding box. The IOU ratio indicates the intersection area divided by the union area.

Accuracy reveals to the ability of the classifier to determine the entire sample. By modifying the confidence threshold, a prediction box can vary between positive and negative. In general, all predictions above the threshold (defined by box value and class classification) are positive, while the below is negative.

.

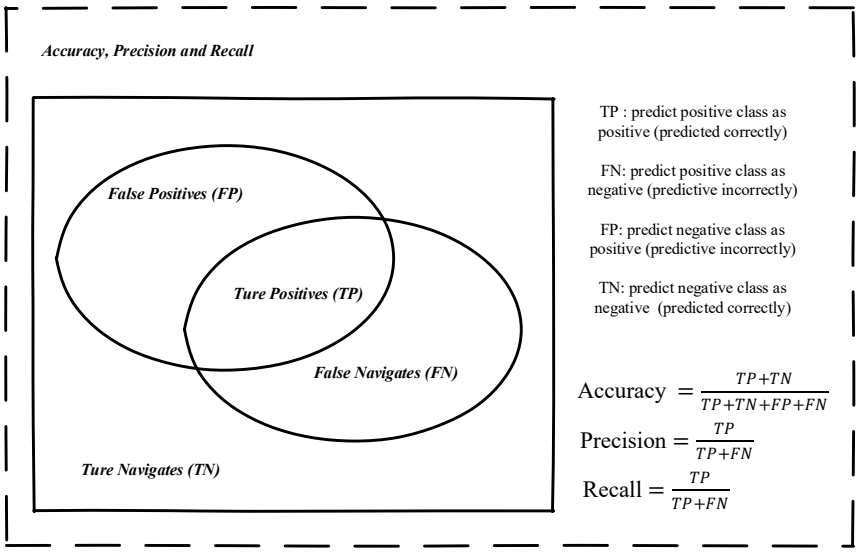


Figure 3.6 Anchor box value sample.

3.5.2 Precision

The precision of a certain type of object refers to the ratio between the number of objects detected accurately, and the number of objects detected precisely. To calculate recall, the number of negative predictions should be calculated. The calculation can be difficult since the model does not count each negative part of the object.

For each image, the ground truth data provides the real number of objects in each type of images. It is easy to calculate the IOU ratio of each positive prediction box and ground truth box. The maximum IOU ratio is considered the nearest ground truth. The threshold can be used for calculating the number of True Positives (TP) and the number of False Positives (FP) for respective type in an image. True Negatives are more difficult to calculate, whereas False Negatives, the objects that the model missed can be calculated.

The idea of average precision (AP) can be conceptually considered finding the area under the precision-recall (PR) graph. The calculation is approximate by smoothing out the zigzag pattern. For instance, the blue line in Figure 3.20 is the actual average precision achieved by MATLAB and the red line is the smoothing pattern.

.

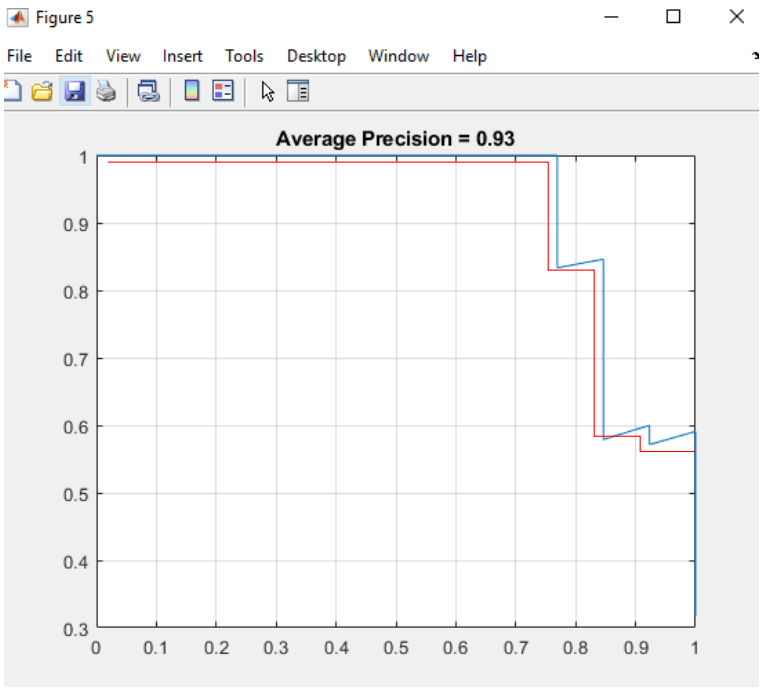


Figure 3.7 Average precision.

AP is computed as the average of maximum precision at these 11 recalls:

$$Average\ Precision = \frac{1}{11}\sum_{\hat{r}} P(\hat{r}) \tag{3.8}$$

This is to find the total area under the red curve. If we average the accuracy of all the objects in the model, the mAP (mean average precision) is generated. For each type, we calculate APs and denote the AP average of all types as mAP. Mean average precision is usually calculated on a dataset.

While the absolute quantification of model output is not easy to explain, mAP can be useful as a relatively good metric. When this metric is being calculated on a popular public dataset, this metric can be easily used to compare existing and novel methods of target detection problems.

However, depending on the distribution of the various classes in the training data, the mAP value may be very high with good training data, while other classes are lower. Thus, we check the AP values of each class when analysing the model results.

Chapter 4

Results Analysis and Discussions

In this chapter, our results of the experiments will be demonstrated based on the design. This chapter will examine the results and discuss the differences between various research methods. A comparison is drawn to discuss the outcomes.

4.1 Experimental Environment

MATLAB and its computer vision toolbox are employed for this study. It is noteworthy that due to the incompatibility of the MATLAB version, datasets made using MATLAB version 2019a can only be used on later versions of MATLAB.

Office Visio accounts for drawing flowcharts and schematics of networks to clarify the experiment processes for gaining insight into complex information.

First, a pretrained object detector is adopted for transfer learning. This method can be faster due to the detector have been trained repeatedly. Second, we create a custom object detector. A simple network architecture is designed to learn the features.

4.2 Results

Apple detection has a limitation of time-consuming and intensive computationally. We have trained our models for several times and the results are listed in Table 4.1 and Table 4.2.

Number of epochs	Number of images	Network	Precision of ripe	Precision of overripe	Precision of unripe
10	280	11 layers network	17%	88%	6%
		ResNet-50	13%	93%	12%
30	2800	11 layers network	37%	47%	19%
		GoogLeNet	32%	54%	21%
		ResNet-50	36%	53%	17%

Table 4. 1 The results using random dataset with low quantity.

Number of epochs	Number of images	Network	Precision of ripe	Precision of overripe
10	1880	ResNet-50	50%	48%
		GoogLeNet	40%	53%
	7064	11 layers network	36%	38%
		ResNet-50 Transfer Learning	66%	63%

Table 4. 2 The results with data augmentation.

4.3 Analysis

Different parameters are set in this study to get a better model.

(1) Batch size.

SGD-based training is generally used in deep learning. In other words, each training considers the batch size. The loss function required for each model is not obtained from one dataset, whereas it is weighted by a set of datasets. The number of datasets in this group is *batchsize*.

(2) Iteration.

One iteration is equal to train *batchsize* samples.

(3) Epoch.

One epoch is equal to train all the samples in the training set. The maximum *batchsize* refers to the total number of samples *N*. The minimum *batchsize* is 1, suggesting that only one round of training has been conducted.

For instance, there are 280 samples in the training Dataset I and *batchsize = 1*, the sample set is trained. Though the first epoch and the 10-th epoch use the same 280 images in the training set, the weights for the model are different. Since the models of different epochs are in different positions in the loss function, the later the training iteration of the model is, the closer it will be to the bottom, the cost of calculations will be lower.

In the first round of training, a dataset consisting of 280 images, which are classified into 3 types, is used. When setting the parameters of options, *minibatch = 1, epoch = 5* are first selected for training, and the learning rate is set as 0.001. After a long-time training, the precision of all types is shown as 0.00. Subsequently, we import a sample of test dataset, we get an accuracy rate, whereas this model cannot classify the apple ripeness.

Figure 4.1 The result with epoch as 5.

(a) Single apple (b) Multiple apples

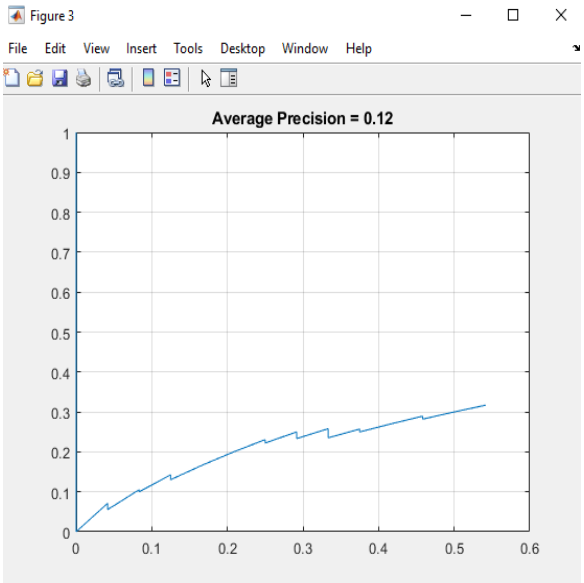
Figure 4.2 The results with learning rate lower than 0.0001.

Due to the limitations of outdoor capturing conditions, unripe apple images have caused numerous problems, so apples are deleted from the unripe apple pictures, and the detector is retrained. To further enhance the quality of the model, the data are augmented, and then the ResNet-50 model is used for transfer learning.

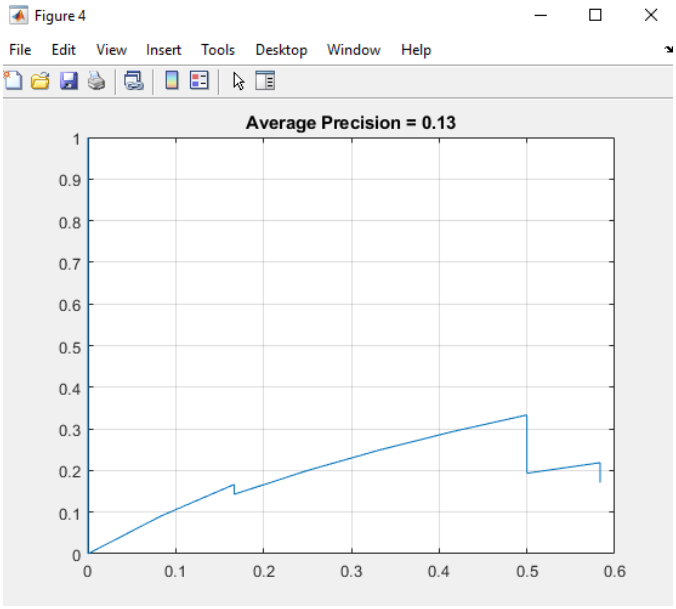
4.4 Discussions

In the initial training, it is suggested that the classification of the detector is very unstable. Ripe and unripe apples are with a very low precision, while the recognition of overripe apple exhibits very high precision. The ratio of our experimental samples is balanced. After the training samples are re-examined, considerable interference is found in Dataset I and Dataset II. Furthermore, 280 images in Dataset I limit the learning ability of computer in object detection since the data is small without noticeable features for recognition.

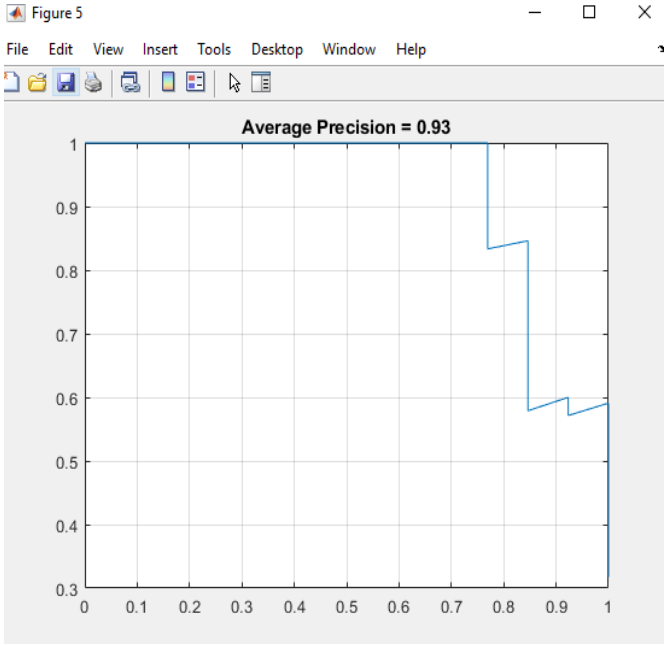
When the proposed model classifies the collected samples, there has a confidence suggesting the probability of samples is a positive one or negative one. The sample will be divided by selecting the appropriate threshold, usually setting as 50%. If the probability reaches over 50%, it is considered a positive example where less than 50% is a negative example. All samples can be sorted by confidence, the threshold can be selected one by one. When each sample acts as a threshold, the corresponding precision and recall can be calculated, and then the ROC curves can be plotted.



(a) The PR curve of unripe apples Dataset I using ResNet-50.



(b) The PR curve of ripe apples Dataset I using ResNet-50.



(c) The PR curve of overripe apples Dataset I using ResNet-50.

Figure 4.3 The results with ResNet-50 using Dataset I.

According to the proposed method, the recall is incremental. When the threshold point is shifted to the left, the positive example is considered to be more positive; otherwise, it will not

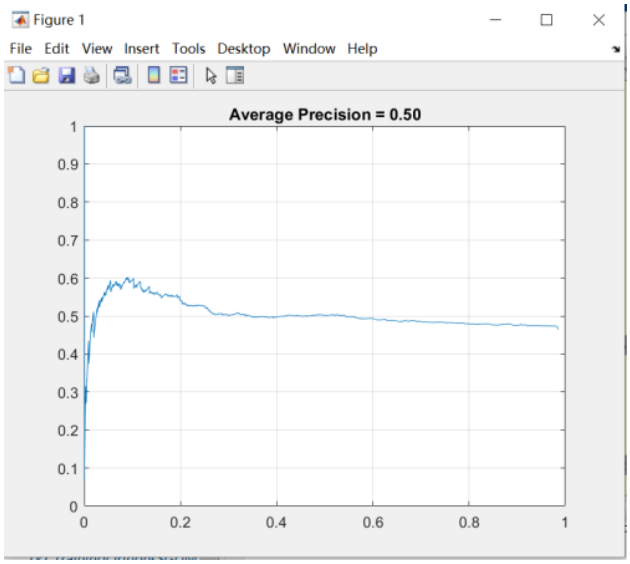
decrease. The precision is not diminished, and the second may be oscillated. Though the positive case is judged to be more positive while the negative case is more positive, the precision will still oscillate, whereas the overall trend should be lower.

In Figure 4.3, it's clearly that the overripe type with higher precision produces a more reasonable PR curve, while the remaining 2 classes with lower precision show another trend.

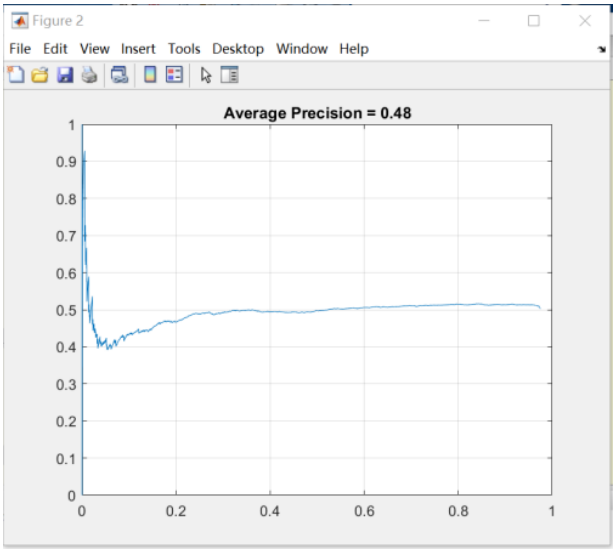
Besides, PR curve will pass the point (0,0). If all samples are judged as negative, then TP = 0, then Precision = Recall = 0, so the PR curve will pass the (0,0) point with the threshold. If the point shift left, the initial precision is very close to 1.00, and the recall is very close to 0.0, so the line rising from (0,0) is possible. If the first few points are negative, the curve will gradually rise from the (0,0) point.

If the number of negative cases is over 1.0, precision will not be 0. Thus, a reasonable PR curve should be the curve initially pulled from (0,0) to (0,1), where the previous predictions are correct and all positive, so the precision will always be 1.

The PR curve can better reflect the performance of the classification when the ratio between positive and negative samples is large. In Figure 4.3, even though the type of ripe and unripe apples is with lower precision, PR curve indicates that there are still rooms for the improvement because the trend of the curve is upward. Thus, increasing the samples of the data may help enhance accuracy.



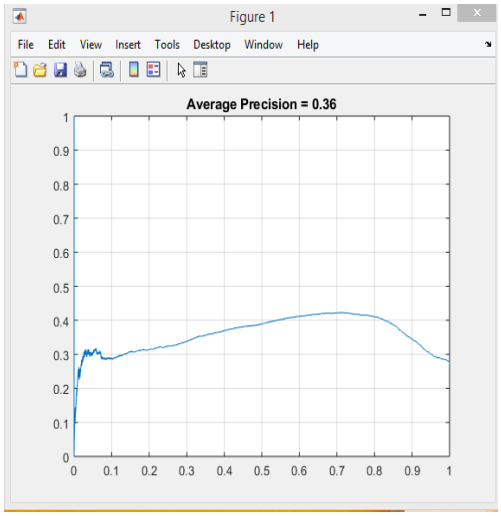
(a) The PR curve of ripe apples Dataset II using ResNet-50.



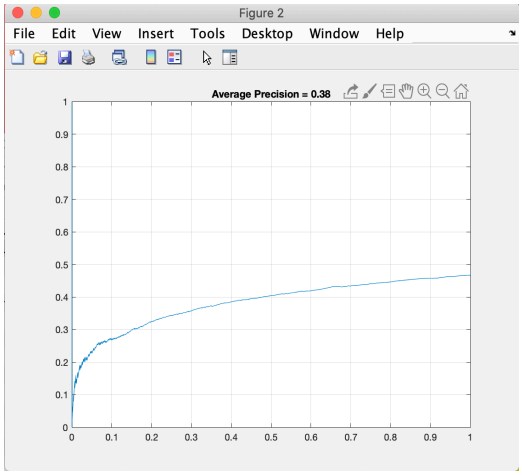
(b) The P-R curve of overripe apples Dataset II using ResNet-50.

Figure 4.4 The results with ResNet-50 using Dataset II.

In the experiment, 1800 are selected from Dataset II including ripe and overripe apple images for 2 classes classification. The training results are presented in Figure 4.4. The results in Figure 4.4 suggest that the data with larger interference is removed, and the average precision level of the entire model is enhanced. This reveals that the quality dataset has an interference with the model learning ability. After removing the data with interference, the detector exhibits relatively stable performance.

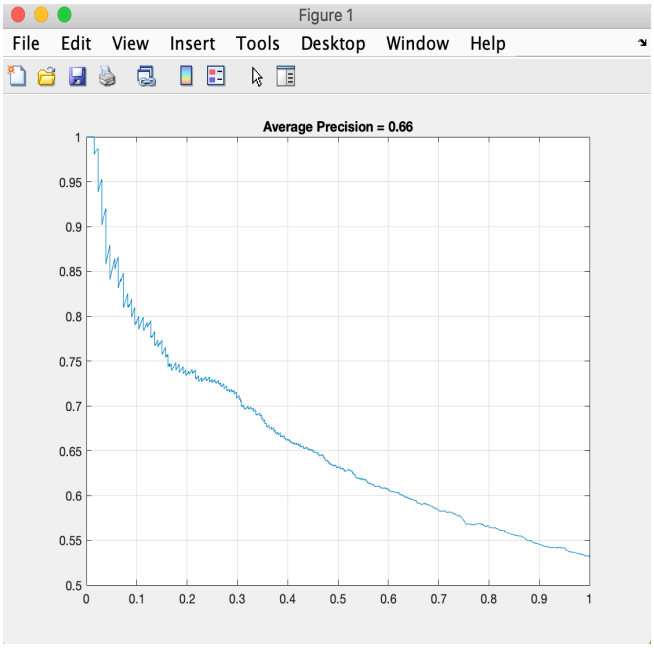


(a) The PR curve of ripe apples Dataset III using 11-layer network.

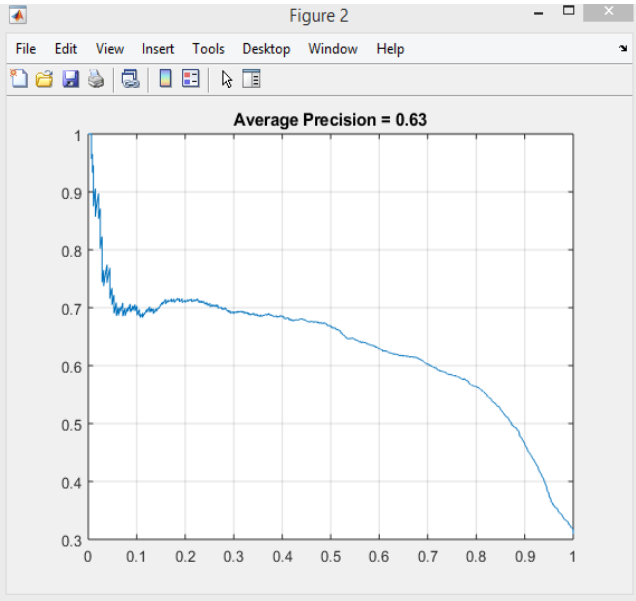


(b) The PR curve of overripe apples Dataset III using 11-layer network.

Figure 4.5 The results with 11-layer network using Dataset III.



(a) The PR curve of ripe apples Dataset III using ResNet-50.



(b) The P-R curve of overripe apples Dataset III using ResNet-50.

Figure 4.6 The results with ResNet-50 using Dataset III.

After data augmentation, we retrain the model. In this study, the 11-layer Faster R-CNN network is reused to compare how the number of convolution layers impact on the detector. In the meantime, we also use a transfer learning based on ResNet-50 to optimize the network.

Dataset III covers more image samples and deletes problematic samples. After the dataset is improved, the performance of the model is enhanced, and the network with less convolution layer cannot achieve the purpose of learning.

The depth of the proposed neural network is critical. Nevertheless, if the network gets deeper, the performance cannot be further enhanced. The degradation problem occurs with the depth increase of the layers of the network. If the network is sufficiently deep, the precision will get saturated and then degraded. As shown in Figure 4.6, the PR curve displays a downward trend. This problem is not caused by the overfitting problem. The depth of the network does not satisfy the requirement of the experiment.

When the model generates lower precision, the model remains with high performance and can label an apple image from the test dataset with high accuracy. Good precision can ensure good accuracy and performance. On the whole, if the precision is not good, it is unlikely to have high accuracy. In contrast, the precision is good, the accuracy may not be good. This indicates that the random error is small, and the detector error is large.

Chapter 5

Conclusion and Future Work

This chapter summarises the whole process of this study. There will be a suggestion on how to optimize the experiment and envision the future work associated with the experimentation.

5.1 Limitations of the Research

Though we do not use mean average precision to assess the performance of the model, the mAP value suggests that this study requires further improvement.

Number of epochs	Number of images	Network	mAP
10	280	11 layers network	37%
		ResNet50	39.33%
10	1880	ResNet-50	49%
		GoogLeNet	46.5%
30	2880	11 layers network	34.33%
		ResNet-50	35.33%
		GoogLeNet	35.67%

Table5. 1 Mean average precision overview.

Given the current mean average precision listed in Table 5.1, we can clearly figure out that the proposed apple detection model is not suitable for the case with multiple types. The results of the precision reveal that the training model is suitable for single type. For the overripe apples, the model works well. Given the valid unripe apple types, the model did not perform well. Even if there are improvement after the data augmentation, the model still require improvement.

The 2D image does not refer to the true shape of the apples because of distortions. The model may not capture the actual shape as human view. The 2D image is only for the apple detection, i.e., there is a space for improvement of the dataset.

5.2 Conclusion

Affects	Parameters
Mini Batch Size	1
Learning Rate	0.001
Epoch	$10 \leq \text{epoch} \leq 30$
Quantity of Data	The more the better
Quality of Data	Data Augmentation
Networks	ResNet-50 Transfer Learning

Table5. 2 The parametric settings related to affects.

2 ways can be adopted to update the parameters of neural networks. The first method is to traverse all datasets to calculate the loss function and the gradient function for each parameter. The other is to calculate the loss function every time, check the data, and then ask the gradient to update the parameters. This method can be fast, whereas the convergence performance is not sufficiently good. To overcome the shortcomings of the 2 methods, minibatch gradient decent, a compromised method is suggested with a small batch of gradients. This method separates the data into several batches and updates the parameters in batches. Since the number of samples in the batch is smaller than the entire data set, the amount of calculations is not relatively large.

The minibatch size can only be set as 1.000 in apple ripeness identification since the hardware for training does not support it. Whether the mini batch size impacts on the experimental results remains unknown. Setting the learning rate as 0.001 is appropriate as a lower learning rate leads to an overload in the training process.

The quality and quantity of the images more noticeably impact the model. Also, the depth of the network affects the model. The current dataset has 10,000 images, whereas a deep learning model requires numerous data. ResNet-50 model is overly deep for the current dataset. Data collection also requires well-characterized images without interference. The resize function is only required for data augmentation since Faster R-CNN does not require the same input size of images.

Data augmentation is to primarily solve the problem of insufficient samples and imbalanced data. Both methods may lead to overfitting problem. Transfer learning may help address the imbalanced problem of the dataset.

On the whole, the classification method takes overall classification accuracy as the learning goal. When the sample is not balanced, the classification model will focus on the class covering more images, thus optimizing the classification. Data augmentation methods may lose the characteristics of the original image, so the optimal suggestion is to resample the training set.

5.3 Future Work

The current experiment has defects in data collection. Due to insufficient data samples and weaknesses in the data collection, the class of unripe apples has not achieved good results. In the meantime, the data distortion, yielded by the resized data size when the data is enhanced, also affects the experimental results. Given all these, the next step is to recreate a high-quality dataset.

After the new data set is created, ResNet-50 network is employed to train and measure the relevance between the depth of network and the size of a dataset. In the meantime, the learning rate can be gradually down-regulated with the iteration of training to achieve the optimal. The current dataset applies to 30 epochs; due to the limitation of data volume, however, the relationship between data volume, number of epochs, and network depth cannot be determined. Thus, our future work is to probe the parameters of the optimized models.

References

Amara, J., Bouaziz, B., & Algergawy, A. (2017). A Deep Learning-based Approach for Banana Leaf Diseases Classification. BTW (Workshops), pp. 79-88.

Bodla, N., Singh, B., Chellappa, R., & Davis, L. S. (2017). Soft-NMS--Improving Object Detection with One Line of Code. In Proceedings of the IEEE international conference on computer vision (pp. 5561-5569).

Buzzelli, M., Belotti, F., & Schettini, R. (2018). Recognition of Edible Vegetables and Fruits for Smart Home Appliances. In IEEE 8th International Conference on Consumer Electronics-Berlin (ICCE-Berlin) (pp. 1-4). IEEE.

Byeon, Y. H., & Kwak, K. C. (2017). A Performance Comparison of Pedestrian Detection using Faster RCNN and ACF. In 6th IIAI International Congress on Advanced Applied Informatics (IIAI-AAI) (pp. 858-863). IEEE.

Cai, Z., & Vasconcelos, N. (2018). Cascade R-CNN: Delving into High Quality Object Detection. In Proceedings of the IEEE conference on computer vision and pattern recognition (pp. 6154-6162).

Cao, C., Wang, B., Zhang, W., Zeng, X., Yan, X., Feng, Z., ... & Wu, Z. (2019). An Improved Faster R-CNN for Small Object Detection. IEEE Access, 7, 106838-106846.

Chen, D., & Wang, H. (2018). Application on Intersection Classification Algorithm Based on Clustering Analysis. In IEEE Annual Computer Software and Applications Conference (COMPSAC) (Vol. 2, pp. 290-297). IEEE.

De Rita, N., Aimar, A., & Delbruck, T. (2019). CNN-based Object Detection on Low Precision Hardware: Racing Car Case Study. In IEEE Intelligent Vehicles Symposium (IV) (pp. 647-652).

Dias, P. A., Tabb, A., & Medeiros, H. (2018). Multispecies Fruit Flower Detection using A

Refined Semantic Segmentation Network. IEEE Robotics and Automation Letters, 3(4), 3003-3010.

Dong, E., Lu, Y., & Du, S. (2019). An Improved SSD Algorithm and Its Mobile Terminal Implementation. In IEEE International Conference on Mechatronics and Automation (ICMA) (pp. 2319-2324). IEEE.

Eaton, A. T. (2017). Fruit Injury Types Recognized in Annual New Hampshire Apple Harvest Evaluations. Extension Specialist, Entomology. UNH Cooperative Extension, 13.

Fachrurrozi, M., Fiqih, A., Saputra, B. R., Algani, R., & Primanita, A. (2017). Content Based Image Retrieval for Multi-objects Fruits Recognition using K-means and K-nearest Neighbour. In 2017 International Conference on Data and Software Engineering (ICoDSE) (pp. 1-6). IEEE.

Fachantidis, A., Partalas, I., Taylor, M. E., & Vlahavas, I. (2015). Transfer Learning with Probabilistic Mapping Selection. Adaptive Behavior, 23(1), 3-19.

Feng, J., Zeng, L., & He, L. (2019). Apple Fruit Recognition Algorithm Based on Multi-Spectral Dynamic Image Analysis. Sensors (Basel), 19(4): 949.

Gongal, A., Amatya, S., Karkee, M., Zhang, Q., & Lewis, K. (2015). Sensors and Systems for Fruit Detection and Localization: A review. Computers and Electronics in Agriculture, 116 (C), pp. 8-19.

Guo, L., Lei, Y., Xing, S., Yan, T., & Li, N. (2018). Deep Convolutional Transfer Learning Network: A New Method for Intelligent Fault diagnosis of Machines with Unlabelled Data. IEEE Transactions on Industrial Electronics, 66(9), 7316-7325.

Huang, Z., Cao, Y., & Wang, T. (2019). Transfer Learning with Efficient Convolutional Neural Networks for Fruit Recognition. In IEEE 3rd Information Technology, Networking, Electronic and Automation Control Conference (ITNEC) (pp. 358-362). IEEE.

Hou, L., Wu, Q., Sun, Q., Yang, H., & Li, P. (2016). Fruit Recognition Based on Convolution Neural Network. In 12th International Conference on Natural Computation, Fuzzy Systems and Knowledge Discovery (ICNC-FSKD) (pp. 18-22). IEEE.

Howlader, M. R., Habiba, U., Faisal, R. H., & Rahman, M. M. (2019). Automatic Recognition

of Guava Leaf Diseases using Deep Convolution Neural Network. In 2019 International Conference on Electrical, Computer and Communication Engineering (ECCE) (pp. 1-5). IEEE.

Hsu, S. C., Huang, C. L., & Chuang, C. H. (2018). Vehicle Detection using Simplified Fast R-CNN. In 2018 International Workshop on Advanced Image Technology (IWAIT) (pp. 1-3). IEEE.

Hsu, S. C., Wang, Y. W., & Huang, C. L. (2018). Human Object Iidentification for Human-robot Interaction by using Fast R-CNN. In 2018 Second IEEE International Conference on Robotic Computing (IRC) (pp. 201-204). IEEE.

Jana, S., Basak, S., & Parekh, R. (2017). Automatic Fruit Recognition from Natural Images using Colour and Texture Features. In 2017 Devices for Integrated Circuit (DevIC) (pp. 620-624). IEEE.

Kendall, A. G. (2019). Geometry and Uncertainty in Deep Learning for Computer Vision (Doctoral dissertation, University of Cambridge).

Kim, J. Y., Vogl, M., & Kim, S. D. (2014). A Code Based Fruit Recognition Method via Image Convertion using Multiple Features. In International Conference on IT Convergence and Security (ICITCS) (pp. 1-4). IEEE.

Islam, M., Dinh, A., Wahid, K., & Bhowmik, K. (2017). Detection of Potato Diseases Using Image Segmentation and Multiclass Support Vector Machine. IEEE 30th Canadian Conference on Electrical and Computer Engineering (CCECE), pp. 1-4.

Ji, W., Zhao, D., Cheng, F., Xu, B., Zhang, Y., & Wang, J. (2012). Automatic Recognition Vision System Guided for Apple Harvesting Robot. Computers & Electrical Engineering. 38 (5), pp. 1186-1195.

Juhnevica-Radenkova, K., Radenkovs, V., & Seglina, D. (2016). Microbiological Changes and Severity of Decay in Apples Stored for a Long-term Under Different Storage Conditions. Zemdirbyste Agriculture, 103(4), pp. 391-396.

Lal, S., Behera, S. K., Sethy, P. K., & Rath, A. K. (2017). Identification and Counting of Mature Apple Fruit Based on BP Feed Forward Neural Network. In 2017 Third International

Conference on Sensing, Signal Processing and Security (ICSSS) (pp. 361-368). IEEE.

Li, G., Ma, Z., & Wang, H. (2011). Image Recognition of Grape Downy Mildew and Grape Powdery Mildew Based on Support Vector Machine. In International Conference on Computer and Computing Technologies in Agriculture (pp. 151-162). Springer, Berlin, Heidelberg.

Liu, B., Zhang, Y., He, D., & Li, Y. (2017). Identification of Apple Leaf Diseases Based on Deep Convolutional Neural Networks. Symmetry, 10(1), pp. 11.

Liu, B., Zhao, W., & Sun, Q. (2017). Study of Object Detection Based on Faster R-CNN. In Chinese Automation Congress (CAC) (pp. 6233-6236). IEEE.

Liu, C., Tao, Y., Liang, J., Li, K., & Chen, Y. (2018). Object Detection Based on YOLO Network. IEEE Information Technology and Mechatronics Engineering Conference (ITOEC). Chongqing, China.

Lu, Y., Yi, S., Zeng, N., Liu, Y., & Zhang, Y. (2017). Identification of Rice Diseases Using Deep Convolutional Neural Networks. Neurocomputing, vol. 267, pp. 378-384.

Manana, M., Tu, C., & Owolawi, P. A. (2018). Preprocessed Faster RCNN for Vehicle Detection. In International Conference on Intelligent and Innovative Computing Applications (ICONIC) (pp. 1-4). IEEE.

Marathe, A., Anirudh, R., Jain, N., Bhatele, A., Thiagarajan, J., Kailkhura, B., ... & Gamblin, T. (2017). Performance modeling Under Resource Constraints using Deep Transfer Learning. In Proceedings of the International Conference for High Performance Computing, Networking, Storage and Analysis (p. 31). ACM.

Mohamud, A. H., & Gopalakrishnan, A. K. (2018). Fruit Feature Recognition Based on Unsupervised Competitive Learning and Backpropagation Algorithms. In 2018 International Conference on Engineering, Applied Sciences, and Technology (ICEAST) (pp. 29-32). IEEE.

Mohanty, S. P., Hughes, D. P., & Salathe, M. (2016). Using Deep Learning for Image-based Plant Disease Detection. Frontiers in Plant Science, vol. 7, pp. 1419.

Murugan, V., Vijaykumar, V. R., & Nidhila, A. (2019). A Deep Learning RCNN Approach for

Vehicle Recognition in Traffic Surveillance System. In International Conference on Communication and Signal Processing (ICCSP) (pp. 0157-0160). IEEE.

Nourmohammadi-Khiarak, J., Mazaheri, S., Moosavi-Tayebi, R., & Noorbakhsh-Devlagh, H. (2018). Object Detection utilizing Modified Auto Encoder and Convolutional Neural Networks. In 2018 Signal Processing: Algorithms, Architectures, Arrangements, and Applications (SPA) (pp. 43-49). IEEE.

Nyarko, E. K., Vidovic, I., Radocia, K., & Cupec, R. (2018). A nearest Neighbor Approach for Fruit Recognition in RGB-D Images Based on Detection of Convex Surfaces. Expert Systems with Applications, 114, pp. 454-466.

Nguyen, D. T., Nguyen, T. N., Kim, H., & Lee, H. J. (2019). A High-Throughput and Power-Efficient FPGA Implementation of YOLO CNN for Object Detection. IEEE Transactions on Very Large-Scale Integration (VLSI) Systems.

Rachmawati, E., Supriana, I., & Khodra, M. L. (2017). Toward a New Approach in Fruit Recognition using Hybrid RGBD Features and Fruit Hierarchy Property. In 2017 4th International Conference on Electrical Engineering, Computer Science and Informatics (EECSI) (pp. 1-6). IEEE.

Rochac, J. F., Zhang, N., Thompson, L., & Oladunni, T. (2019). A Data Augmentation-Assisted Deep Learning Model for High Dimensional and Highly Imbalanced Hyperspectral Imaging Data. IEEE International Conference on Information Science and Technology (ICIST). Hulunbuir, China.

Rochac, J. F., Zhang, N., Xiong, J., Zhong, J., & Oladunni, T. (2019). Data Augmentation for Mixed Spectral Signatures Coupled with Convolutional Neural Networks. IEEE International Conference on Information Science and Technology (ICIST). Hulunbuir, China.

Rzanny, M., Seeland, M., Waldchen, J., & Mader, P. (2017). Acquiring and Preprocessing Leaf Images for Automated Plant Identification: Understanding the Tradeoff Between Effort and Information Gain. BMC. Plant Method, Accesses: 7674.

Scheffler, O., Coetzee, C., & Opara, U. (2018). A Discrete Element Model (DEM) for Predicting Apple Damage During Handling. Biosystems Engineering, 172, pp.29-48.

Song, W., Wang, H., Maguire, P., & Nibouche, O. (2016). Differentiation of organic and non-organic apples using near infrared reflectance spectroscopy—a pattern recognition approach. In 2016 IEEE SENSORS (pp. 1-3). IEEE.

Sun, S., Wu, Q., Jiao, L., Long, Y., He, D., & Song, H. (2018). Recognition of Green Apples Based on Fuzzy Set Theory and Manifold Ranking Algorithm. Optik, 165, pp. 395-407.

Sun, W., & He, Y. (1998). Spatial-chromatic Clustering for Colour Image Compression. IEEE World Congress on Computation Intelligence FUZZ-IEEE, pp. 1601-1604.

Shukla, D., & Desai, A. (2016). Recognition of Fruits using Hybrid Features and Machine Learning. In International Conference on Computing, Analytics and Security Trends (CAST) (pp. 572-577). IEEE.

Szegedy, S., Liu, W., Jia, Y., Sermanet, P., Reed, S., Anguelov, D., Erhan, D., Vanhoucke, V., & Rabinovich, A. (2015). Going Deeper with Convolutions. IEEE Conference on Computer Vision and Pattern Recognition, pp. 1-9.

Teoh, T. T., Chiew, G., Jaddoo, Y., Michael, H., Karunakaran, A., & Goh, Y. J. (2018). Applying RNN and J48 Deep Learning in Android Cyber Security Space for Threat Analysis. In 2018 International Conference on Smart Computing and Electronic Enterprise (ICSCEE) (pp. 1-5). IEEE.

Teoh, T. T., Chiew, G., Franco, E. J., Ng, P. C., Benjamin, M. P., & Goh, Y. J. (2018). Anomaly Detection in Cyber Security Attacks on Networks using MLP Deep Learning. In 2018 International Conference on Smart Computing and Electronic Enterprise (ICSCEE) (pp. 1-5). IEEE.

Thilagavathi, M., & Abirami, S. (2017). Application of Image Processing in Diagnosing Guava Leaf Diseases. International Journal of Scientific Research and Management, 5(7), pp. 5927-5933.

Tu, S., Xue, Y., Zheng, C., Qi, Y., Wan, H., & Mao, L. (2018). Detection of Passion Fruits and Maturity Classification Using Red-Green-Blue Depth Images. Biosystems Engineering, 175, pp. 156-167.

Wang, S., Chen, Z., & Ding, Z. (2019). The Unified Object Detection Framework with

Arbitrary Angle. In 2019 5th International Conference on Big Data and Information Analytics (BigDIA) (pp. 103-107).

Wang, X., Ma, H., & Chen, X. (2016). Salient Object Detection via Fast R-CNN and Low-level Cues. In 2016 IEEE International Conference on Image Processing (ICIP) (pp. 1042-1046). IEEE.

Yan, J., Wang, H., Yan, M., Diao, W., Sun, X., & Li, H. (2019). IOU-adaptive Deformable R-CNN: Make Full Use of IOU for Multi-class Object Detection in Remote Sensing Imagery. Remote Sensing, 11(3), 286.

Yang, C., Hu, Y., Lin, H., Sa, L., Liu, Y. (2017). Overlapped Fruit Recognition for Citrus Harvesting Robot in Natural Scenes. International Conference on Robotics and Automation Engineering (ICRAE). Shanghai, China.

Zhang, X., Qiao, Y., Meng, F., Fan, C., & Zhang, M. (2018). Identification of Maize Leaf Diseases Using Improved Deep Convolutional Neural Networks. IEEE Access, 6.

Zhang, X., Wang, Z., Liu, D., & Ling, Q. (2019). Dada: Deep Adversarial Data Augmentation for Extremely Low Data Regime Classification. In ICASSP 2019-2019 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP) (pp. 2807-2811). IEEE.

Zhou, R., Damerow, L., Sun, Y., & Blanke, M. M. (2012). Using Colour Features of CV. “Gala” Apple Fruits in an Orchard in Image Processing to Predict Yield. Precision Agriculture, 13(5), pp. 568-580.