Currency Recognition Using Deep Learning

Qian Zhang

A thesis submitted to Auckland University of Technology
in partial fulfillment of the requirements for the degree of
Master of Computer and Information Sciences (MCIS)

2018

School of Engineering, Computer and Mathematical Sciences

Abstract

Currency is an indispensable part of our daily life. However, how to identify true and fake currencies has become the most important issue at present. If we use a computer for currency recognition, it will greatly improve the accuracy of recognition and reduce people's workload effectively.

In recent years, deep learning has become the most popular research direction. It mainly trains a dataset through deep neural networks. There are many different models that can be used in this research project. Throughout these models, accuracy of currency recognition can be improved. Obviously, such research methods are in line with our expectations.

In this thesis, we mainly use Single Shot MultiBox Detector (SSD) model based on deep learning as the framework, employ Convolutional Neural Network (CNN) model to extract the features of paper currency, so that we can much accurately recognize the denomination of the currency, both front and back. Our main contributions are: (1) through using CNN and SSD, the average accuracy of currency recognition is up to 96.6%; (2) in order to ensure the recognition, we selected two models for comparisons. One is MobileNet and the other is faster R-CNN. However, we found from the experimental results that, in general, CNN is much suitable for our currency identification requirements. When a currency is tilted or moved, its denomination and front/back side can still be identified.

Keywords: Currency Recognition, Currency Classification, Deep Learning, Deep Neural Network (DNN), Convolution Neural Network (CNN), Single Shot MultiBox Detector(SSD), Data Augmentation, MobileNet Model, Faster R-CNN.

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: 张倩   Date: 15 August 2018

Acknowledgment

This research work was completed as a part of the Master of Computer and Information Sciences (MCIS) course at the School of Engineering, Computer and Mathematical Sciences (SECMS) in the Faculty of Design and Creative Technologies (DCT) at the Auckland University of Technology (AUT) in New Zealand. At this moment, I would like to thank my parents for their support; my friends for their encouragement during my study in New Zealand.

Among them, I would like to especially thank my supervisor Dr Wei Qi Yan for providing me with a lot of help during the learning process. He can patiently explain to me the questions I have asked, which makes me much confident to continue my research work. In addition, I would like to thank my teachers and administrators of this school for their help in the past.

Qian Zhang

Auckland, New Zealand

15 August 2018

Chapter 1
Introduction

The first chapter consists of five parts. The first part mainly introduces the background of research content and the motivation of this research project. Currency recognition is a practical application and can improve our work efficiency. The research results expected to be achieved will be introduced in the fourth section. The fifth part will be introduced for the structure of this thesis.

1.1 Background and Motivation

Nowadays, there are many recognition methods applied to image recognition, face recognition, license plate recognition and behavior recognition, etc. Currency is the main medium for circulation, different countries have their currency characteristics. However, there would be more and more counterfeit currency in terms of currency development. Counterfeit currency may seriously affect the interests of these countries. Therefore, how to apply recognition technology to the authenticity of money has become the hottest topic and important task at present.

In the past, people could only recognize the authenticity of money through naked eyes, but observation ability of our human eyes is limited, and our eyes have difficulties to distinguish the truth money without borrowing new technologies. Although UV recognition technology is already in use, with the development of counterfeiting technology, this technology is not enough to help people to identify the counterfeit currency with more advanced fraud techniques.

But now, based on image recognition, researchers analyze and compare different viewpoints on their specific identification methods by analyzing currency color and specific data of currency, then find out their specific identification methods (Wu, Chen, Fan, & Phong, 2001). Currency recognition method for data argumentation through using color analysis, image enhancement and so on expands the dataset.

Deep learning is equivalent to deep neural networks (Deng, & Yu, 2014). First, it needs a set of big data. By analyzing training dataset, the accuracy of currency recognition could be continuously improved and our expectations for experimental results could be achieved. Convolutional neural network (CNN) (Hoo-Chang, Roth, Gao, Lu, Xu, Nogues, & Summers, 2016) plays a key role in the recognition process and can improve the accuracy of the overall training through using CNN model. We use CNN as a feature extractor under the framework of Single Shot Multi Box Detector (SSD) model (Dai, Li, He, & Sun, 2016).

In the process of currency recognition, we first need to consider whether the size of a dataset is sufficient, because our data collection was from the images by splitting the video into a single frame, but in the process it may have distortion or blurring may occur, so it is necessary to edit the images and make the image clearer, which also contributes to the accuracy after training.

In the process of deep learning, overfitting is also prone to occur. It is easy to make the

training process much complicated, increase training difficulty and training time, and the training also allows us to study drop technology and avoid overfitting.

1.2   Research Question

Through reviewing the introduction, we know that the research content of this thesis is to classify and identify currencies. Before our start of research, our analysis of other parties can help us better to understand the process of currency recognition and assist us to carry out this project. So the research questions in this thesis are:

What computer technology support do we need to use for currency recognition?

Before we begin the study of major issues, we also need to solve some questions:

What is currency recognition?
Which algorithms could be used for studying currency recognition?
Which algorithm is the best for the currency recognition?

Our main research content is currency recognition, we need to evaluate the algorithms and techniques used in our research process and choose the best algorithms and techniques to achieve our expected goal on currency recognition.

1.3   Contributions

The main argument in this thesis is realization of event-based currency recognition. According to the procedure proposed in this thesis and perform currency recognition step by step, it includes identifying the denomination of currency and the front and back side of currency. In this thesis, the following points are proposed: (1) data collection and processing, (2) data augmentation, (3) currency feature extraction, (4) currency identification, (5) resultant analysis of currency recognition. The specifics of these methods used in our experiments will be explained in Chapter 4 of this thesis.

This thesis also introduced the specific algorithm and process of currency recognition. In Chapter 2, we will also study and compare different detection methods as well as introduce various methods on details.

This thesis will focus on the process of data training using Convolutional Neural

Networks (CNN). We will use MATLAB for currency recognition.

The overall contribution to our research could be divided into the following points: (1) currency recognition based on deep learning, (2) how the CNN training model can meet the needs of our data collection, (3) how to use the CNN model as the basic algorithm that needs to be used when the feature extractor. For currency recognition, our research work is based on deep learning, our research results can meet the needs of current development.

1.4   Objective of This Thesis

First of all, this thesis is about currency recognition, but the process of recognition is divided into two parts: detection and identification, the specific details will be introduced in this thesis.

In order to realize real-time currency recognition, including currency denominations and front and back sides, we need to find clearly identifiable parts during feature extraction and then classify the currency to support the detection and identification.

For this research project, we use multiple methods to conduct experiments and compare these methods to find an optimal algorithm. Therefore, the content of method comparisons will also be detailed in this thesis.

1.5   Structure of This Thesis

This thesis is structured as follows:

In Chapter 2, we will introduce the literature in detail and study the literature on currency recognition and detection in multiple aspects. The first is the study of past currency recognition literature, the methods and conclusions they used when conducting currency recognition, how to perform data augmentation and feature extraction of data, the different algorithms used in recognition and classification as well as the benefits of experiments based on deep learning.

In Chapter 3, we mainly introduce the research methods we use, including how to collect data. In order to meet the needs, we also augmented the data so as to increase the data volume. In addition, the basic workflow of this model and the specific methods are also introduced.

In Chapter 4, we will complete the implementation of currency recognition. The results of the experiment will also be presented in the form of tables and icons and will be described in detail. At the same time, the limitations of this project will be solved in this part.

In Chapter 5, a comparison of the different methods for currency recognition is proposed. We will compare the results obtained in the fourth chapter with the experimental results. Finally, the conclusions and future work will be introduced in Chapter 6.

Chapter 2
Literature
Review

Through the in-depth analysis of past research issues and theoretical reviews, in this thesis we mainly analyze the currency recognition in videos. Through using the research of past literatures, we can provide a better idea for currency recognition, and we will study it from the aspects of recognition and classification. This chapter will introduce a variety of methods of currency recognition.

2.1   Introduction

With the continuous advancement of research and development of artificial intelligence (AI), deep learning has become the mainstream research (Bharkad, 2013) which is based on how to analyze and train various neural network models.

The prime mission we need to consider when we would like to carry out currency recognition is what steps we need to be figured out as a whole when conducting currency recognition. In addition to the initial step of data collection and augmentation, the final step is the model and algorithm we will use when we implement currency recognition. In order to recognize a paper currency, we need to classify and detect front and back sides of the currency (Gunaratna, Kodikara & Premaratne, 2008), we chose to use Convolutional Neural Network (CNN) for extracting currency characteristics.

The advantage of using deep learning is that the dataset is more demanding than the general methods. We need to use a large number of datasets to support our model for deep convolution operation and ensure that the recognition accuracy is improved. At the same time, the robustness is guaranteed, which is very demanding on the degree of training model itself. Since the development of deep learning, there are still many unexplored and unexploited areas that need to be studied, but it is known that the effects of convolutional neural networks and deep learning on detection and recognition currently meet our development needs (Simonyan, K & Zisserman, 2014).

2.2 Currency Detection and Recognition

In the process of currency detection, it is necessary to firstly determine from which aspect to start testing. In 2001, it was proposed to perform currency detection based on edge information of the currency. By recording the anchor lines of currency patterns in the dataset, combining them into a template as well as mixing the currency datasets and templates that need to be detected into in the processing, we observe the degree and difference in order to perform currency detection (Fan, Wu, Micco, Chen & Phong, 2001).

In 2000, a method of secure data encoding was used to detect the authenticity of money (Witschorik, 2000). The magnetic code in the currency needs to be stored as a database, and the currency was detected by comparing the magnetic code. Of course, this method requires a large amount of data information. The magnetic code is updated frequently to support the final requirements of currency detection (Witschorik, 2000).

Ensemble Neural Network (ENN) was proposed for currency recognition through compressing the grayscales of different types of images, using each pixel as an output, inputting it into the neural network for image preprocessing, and increasing the data. A diversity of sets reduces the error rate of a single network by detecting individual neurons (Debnath, Ahmed, Shahjahan & Murase, 2010).

Currency detection based on compressed gradation was proposed in 2008. Based on the ANN environment, noise was removed from the background according to a special linear transformation function without affecting image features. The original grayscale range mapping was set between 0 and 125, the edge detection method was used to provide better robustness for detection. A three-layer backpropagation neural network was proposed, which is effectively detected by different classifications (Gunaratna, Kodikara & Premaratne, 2008). Using MATLAB for currency detection is also a feasible method. In 2008, through using MATLAB for currency detection, we first need to use HSV color space to extract deposit the color of currency in a dataset, and use the archived color information to detect other colors (Yadav, Patil, Karhen & Patil, 2014).

In 2009, a new method of recognition currencies was proposed by using Hidden Markov Model (HMM). This method is no longer limited to the color and size of currency. It could be randomly modeled by using the texture of currency. The model itself is a very powerful stochastic modeling tool, so that damage of currency surface can be avoided. The problem of a dirt of paper currency could not be recognized, this method needs to

preprocess the data, which greatly reduces the amount of calculations in currency recognition (Hassanpour, Farahabadi, 2009).

In 2003, grayscale linear transformation of images was proposed to complete currency recognition, the difference is that they used the width of edges on the top of currency as the feature. Different edge pixels are input into the neural network, a method will force the calculation of pure grayscale pixels faster and could also intuitively identify the original images. The precondition is that the recognition currency has good robustness (Zhang, Jiang, Duan & Bian, 2003).

In 1996, features were extracted from parallel strips in a currency. Based on a neural network, a method called BANK was used. Based on multilayer perceptron, its classifier mainly presents pyramid structure. The structure requirements of BANK itself are relatively simple, but the needs of multilayer perceptron are relatively high (Frosini, Gori & Priami, 1996). The existing currency recognition has certain limitations because in 2011 a method of currency recognition for visual impairment was proposed. Based on Speeded Up Robust Features (SURF) as the basic framework (Hasanuzzaman, Yang & Tian, 2011), currency is partially occluded during data collection, the image is rotated, and the light is changed, the visual point transformation was processed to ensure the experimental method. The framework of SURF can also basically meet their requirements for experiments.

In 1995, the use of genetic algorithms (GA) for currency recognition was proposed (Takeda & Omatu, 1995). The neural network has the function of learning. The information obtained by occluding the money was used as a gene, and the original data has been compared, some of the data was subject to selection, crossover and mutation, the neural network can fulfill the optimization for a short time. The optimization can achieve the role of identifying a currency (Ren, 2017). At the same time, using GA for optimization, the effect of neural networks is comparable.

In 2013, a method of combining Markov chain with local binary image was proposed. After the data had been collected, the processed image was used as a dataset, Gaussian function is used as a classifier activation function of the neural network (Bharkad, 2013). The first three layers of the neural network are working as a classifier, which can improve the accuracy of currency recognition. The advantage of using the Gaussian Function is that when the center distance is close to infinity, the activation is close to zero, and it could be applied to all hidden and output layers.

Regarding currency recognition that training was also conducted using Ensemble Neural Network (ENN) in 2010. The reduced currency image was converted to a

grayscale value, which is compressed within the required range, and compressed pixels are input into the neural network. The use of ENN is mainly to identify highly noisy or old images existing in the dataset of currency image while ENN has smaller number of errors than a single network in classification (Debnath, Ahmed, Shahjahan & Murase, 2010).

The ultimate goal of currency recognition is to identify counterfeit currency. In 2010, Support Vector Machine (SVM) was proposed for currency recognition. Divided a currency into different regions, each with its own kernel, linear weighting through the kernel learning forms different matrices and is learned by using semi-definite programming (SDP) so as to obtain the optimal weights. The use of multiple SVMs as a classifier has been verified better (Yeh, Su & Lee, 2011).

An integrated framework was proposed for a network using convolution based classification, localization and testing (Sermanet, Eigen, Zhang, Mathieu, Fergus & LeCun, 2013). Through learning object boundaries, a novel deep learning was introduced, instead of suppressing the bounding box to increase detection confidence. Furthermore, a new technology was established for inspection and a feature extractor was created from the best model, which is called OverFeat (Sermanet, et al. 2013).

A spatial approach was created that can train multiple locations of an image at the same time, somewhat similar to classification training. Because the model is based on convolution, ownership is shared across all locations. The main difference from localization tasks is that when no object exists, it must predict the background class. Traditionally, negative examples were originally used for training. Afterwards, the most compelling negative errors are added to the training set in the bundle delivery. There may be a potential mismatch between them when independent bootstraps train complex and negative sample datasets. In addition, the need to adjust the size of the bootstrap ensures that the training is not excessive to adapt to a small part. The dynamic training was performed as negative training through an aspect that is computationally expensive but makes the program simpler. Because the feature was original for the training purpose, fine-tuning was conducted (Sermanet, et al. 2013).

In 2003, a real-time currency detection method was proposed for currency classification (Feng, Bo & Long, 2003). Taking the image on currency as a feature of currency detection while taking the currency size of different denominations as one of the extracted features, this information was put into the Kohonen network for training. This method still had the advantages of high speed and high recognition accuracy.

2.3 Deep Learning

Deep learning is a neural network consisting of multiple layers that can learn multiple images as the dataset. It has been mainly used in the fields of speech, vision and image processing. Deep learning is to estimate the weights of each layer through backpropagation algorithm. The processing effect of different layers is different. Although the process is much complicated, it has been successfully applied to computer vision and image processing (Yann, Le, Cun, Yoshua, Bengio & Geoffrey Hinton, 2015).

In artificial neural networks (ANN), supervised learning (SL) and unsupervised learning (UL) are all associated closely. In 2015, a neural network system based on Max-Pooling Convolution has been proposed, which could be used for image recognition, object detection is classified according to different datasets. Through deep learning, each neuron could be activated, but the complexity will be increased as the levels go up (Schmidhuber, 2015).

In 2013, the problem of stochastic gradient descent (SGD) in deep learning has been investigated. When the gradient drops dramatically, the previous set of random parameters will also be altered (Sutskever, Martens, Dahl & Hinton, 2013). Hessian-free is optimized and may slowly return to the previously expected performance. The initialization state of the network is very important for the deep neural network. If the initialization network is not good, it will affect the depth of overall operation and operation. However, when the initialization is adjusted before the start, for the deep neural network, the second-order operation is not required so that the target also could be achieved. Because of rapid development of deep learning, unsupervised learning was proposed in 2011. The purpose is to demonstrate multiple modalities of deep learning, audio and video become representatives of the training modality. In the pure audio data training, the classifier in which it was located is evaluated and a test is launched. It turns out that deep learning is effective for audio and video (Ngiam, Khosla, Kim, Nam, Lee & Ng, 2011).

In 2012, a formalized method was proposed (Krizhevsky, Sutskever & Hinton, 2012). An image having 1.3 million pixels was trained in five different classes through five layers of convolutional layers, some had a maxpool layer at the end which includes two fully connected layers. In order to ensure training speed and accuracy, the entire deep learning training was carried out with efficient GPU support. At the same time, different weights were used to the experiments in order to avoid the problem of overfitting during

training. To prove the importance of image recognition, the increasing depth was experimented in 2014 (Simonyan, K & Zisserman, 2014).

The parameters of this basic framework were modified. In each layer, a 3×3 convolution filter is used. In the experiment of this neural network, a weight for 19 layers is used for image classification. It turns out that the deeper the depth, the higher the accuracy.

Compared to traditional machine learning methods, the performance of deep learning is much better at solving practical problems. In the case of more and more big data, deep learning is needed to analyse and learn in a large amount of comfortable data, which is to combine deep learning and big data technologies. The levels of complexity and layers in deep learning are for this special data, but with more data, the more layers of a neural network, the simpler it becomes, which is one of the reasons why deep learning has become a mainstreaming technology in computing. At the same time, the applications of deep learning are much wider than that of machine learning (Najafabadi, Villanustre, Khoshgoftaar, Seliya, Wald & Muharemagic, 2015).

In 2013, in order to study the applicable neighbourhood of deep learning, it was specifically identified for speech technology. The collected data is trained by automatic learning. This method can reduce human workload. The accuracy of automatic learning is not necessarily higher than that of manual, but automatic learning can basically meet the requirements. For deep learning, the greater the amount of data, the higher the accuracy of learning. For this reason, the typical end-to-end learning is still very effective. At the same time, acoustic or speech understanding has been studied through deep learning (Deng, Huang, Yao, Yu, Seide & Gong, 2013).

In 2009, the application of deep learning architecture with regard to AI was proposed because deep learning is based on multilevel nonlinear operations, multilevel models and complex formulations in neural networks; each level of the framework represents different abstract functions that meets the functional requirements of AI for complex levels of abstraction (Bengio, 2009). Deep neural networks and the unsupervised learning algorithm were the best upgrades to the prior art at the time. In 2015 (Van Merriënboer, Bahdanau, Dumoulin, Serdyuk, Warde-Farley, Chorowski & Bengio, 2015), two Python frameworks, Blocks and Fuel were put forward to train large neural networks. Among them, A block a linear algebra translator that attains metadata to Theano's symbolic calculation graph and provides a wide range of utilities to assist in training networks such as algorithms, records, insights and more. Fuel, a program that provides a standard format for data, allows for multiple types of preprocessing, permit users to easily iterate in the

large datasets.

Multilayer threshold networks without feedbacks are proposed as early in 1978 (Bobrowski, 1978). It is an algorithm that uses binary operations. The proposed algorithm is similar to the form of perceptual classifiers. The threshold network will tend to be stable. In fact, this is the predecessor of deep learning. Given the input vector, it is then classified, and finally the amount of computation increases and the network is stable.

Compared to traditional machine learning or shallow learning with a single layer, it is difficult to ask for in-depth research of complex structures. However, deep learning can naturally process this information because deep learning has a clear layer and classification structure as there are multilinear processors. The processing operation is performed and the output of each layer is known as the input of the next layer to know the highest layer. Unsupervised learning could be effectively utilized with a large number of unlabelled training sets; also, it can be RBM or denoising autoencoder-based pretraining approaches (Yu & Deng, 2011).

The main method of deep learning is to use stochastic gradient descent methods (SGDs) during training. In 2011, a more sophisticated optimization method (Le, QV, Ngiam, Coates, Lahiri, Prochnow & Ng, 2011) was proposed for use with limited memory. Conjugate gradient (CG) and line search could speed up the operation. Throughout the use of locally connected networks and convolutional neural networks, it is concluded that deep learning can also achieve good results without using the algorithms that are trained.

In 2013, a network of sinister squadrons (Y. Sun, X. Wang and X. Tang, 2013) was proposed to estimate the location of key points on the face (Sun, Wang & Tang, 2013). The roll machine network belongs to deep learning, which has a deep structure. Such a method can minimize the portions in the case of partial occlusion without affecting the overall effect. This played a key role in the study of more difficult samples. After a large number of experiments, it was found that this method is desirable in terms of accuracy and reliability of detection.

In terms of identification, the basic network is limited, and specific improvements have been made to speech recognition using neural networks (Waibel, 1989). The important issue was time and expansion. The neural network needs to be represented at the right time. A large amount of human knowledge was needed in the coding process. The time problem was solved by developing a time-delay neural network. After training, the network was modularized for abstract training and a more complex network structure.

In 2015, using a deep learning framework was proposed to conduct experiments (Zhao, Ouyang, Li & Wang, 2015). The deep neural network has the ability to simulate the saliency of objects in the detected image. After integrating all the prerequisites, the deep neural network was used for training; in order to ensure a good initialization, different pretraining models were also used for testing. It is confirmed that in the complex background, the deep neural network under the deep learning framework can meet the experimental needs.

2.4 Deep Neural Network (DNN)

Deep Neural Network (DNN) is a hierarchical classifier, which consists of a convolutional layer and a maxpooling layer. The input of each layer is the output of the previous layer, the pixels of the input image are mapped to the fully connected layer. The reason why DNN is faster is that it does not need to make predictions, and it is based on unsupervised learning methods (Maas, Hannun, & Ng, 2013). The purpose of Multi-Column DNN (MCDNN) for different DNNs is to further improve the performance of recognition and reduce sensitivity to changes of lamination during the identification process. The full link layer was combined into a one-dimensional vector based on the output of the last convolutional layer (Cireşan, Giusti, Gambardella, & Schmidhuber, 2013). The final layer must be a fully connected layer for the DNN model. In the last layer, SoftMax was chosen as the activation function. This is to explain the problem of each neuron being activated and for which particular input image each neuron belongs to (CireşAn, Meier, Masci, & Schmidhuber, 2012).

The problem with DNN is that it is difficult to be optimized. In addition to this problem, DNN is still a good extractor. In 2012, there were many hidden layers in DNN itself. There are multiple flexible DNN parameters on each layer of the unit (Dahl, Yu, Deng, & Acero, 2012).

The key reason for modeling DNN is that it has a complex relationship between input and output, it has a highly nonlinear relationship, in order to avoid over-fitting while modeling without accidental specific examples. The problem is that a large number of datasets are used for training. The problem of reducing overfitting can also be solved by using weight penalties or early stopping. On the problem of voice recognition, DNN can form a good acoustic model, which can be trained through a number of different shallow neural networks with hidden elements (Hinton, Deng, Yu, Dahl, Mohamed, Jaitly, & Kingsbury, 2012). For shallow neural network, it is easy to be trained after learned DNN;

but if the DNN is recognized, a hidden layer will appear. This hidden layer will be added to the first hidden layer and SoftMax output. Between the unit layers, after the completion of the network, the network will be trained again. This is mainly due to making the whole network distinguishable. When the hidden layer reaches a certain number, it will be propagated back to the DNN. By using such model and methods, the effect is good in pretraining, while it ensures accuracy (Seide, Li, Chen, & Yu, 2011).

2.5   Single Shot MultiBox Detector (SSD)

Single Shot MultiBox Detector (SSD) is a large frame model for object detection. It generates bounding boxes from feature maps on different layers, the output of these bounding boxes is obtained. It will form different bounding boxes according to different classifiers; after the classification, we could determine what the object is. It is also used in real-time detection, but it is faster than Faster R-CNN and ResNets. However, SSD not only guarantees the speed of detection, but also ensures the accuracy of detection.

With regard to SSD, the bounding box is found to form a default box, which is distributed at different location within the target image. Deep neural networks combine feature maps of different resolutions in order to detect images of different sizes more quickly. In the detection process, the SSD eliminates the need for subsequent feature extraction after the pixel problem. It saves the calculation results of different stages in each layer, which shows that the SSD is easy to be trained and can also get high accuracy (Liu, Anguelov, Erhan, Szegedy, Reed, Fu & Berg, 2016). MultiBox in SSD requires a Region Proposal Network (RPN) for prediction to determine the bounding box that is not related to the target classification (Huang, Rathod, Sun, Zhu, Korattikara, Fathi & Murphy, 2017).

Figure 2.1 The architecture of SSD

The SSD is implemented through a single feedforward reel network when making prediction bounding boxes. Along with SSD, the simpler VGG-16 is usually chosen. As shown in Figure 2.1, the overall framework of SSD is to propose a bounding box by outputting the selected convolutional layer. At each selected position, the mesh is split for in each layer. The grid gives the offer score, and the bounding box corresponds to the fully connected classification processor, which is used to eliminate the proposal for the bounding box; finally, the remaining bounding boxes are merged by using non-maximum suppression (Granger, Kiran, & Blais-Morin, 2017).

The main part of SSD is the default box. For the feature map cell, each point is fixed. It is mainly used to analyze the offsets between the box and the default boxes, the score of each box is needed to be calculated. For example, when there are d boxes at a point, there are n classes to be classified. Each of the n classes corresponds to a score. For the box, there are 4 for offsets, then $(c + 4) \times d$ filters for all feature map cells are in the feature map. If the size of a feature map is $r \times s$, the final output will be $(c + 4) \times d \times r$

16

$\times s$.

For SSD, its objective function comes from MultiBox, and it could be used to deal with multiple target classifications. When the i-th default box and the j-th ground truth box in category p match each other, it is represented by $X^p_{ij} = 1$; if they can not match each other, it is represented by $X^p_{ij} = 0$. If there is a match between the j-th ground truth box and multiple default boxes, then it is expressed as $\sum_i x^p_{ij} \geq 1$. So, the loss function of SSD is shown as Eq. (2.1).

$$L(x, c, l, g) = \frac{1}{n} \left(L_{\text{conf}}(x, c) + \alpha L_{\text{loc}}(x, l, g) \right) \qquad (2.1)$$

2.5.1 Default Boxes

A fast text detector is faster and more accurate than a competitive method in dealing with text localization (Liao, Shi, Bai, Wang & Liu, 2017). In addition, when combined with a text recognizer, it will be significantly better than the most advanced methods in word recognition and end-to-end text recognition tasks. Text boxes can directly output the coordinates of the word bounding box of multiple network layers, by jointly predicting the existence of text and offsetting coordinates to the default box (Liu, Anguelov, Erhan, Szegedy, Reed & Berg, 2016).

Among them, the text block diagram layer is a key component. The text presence and bounding box are predicted according to the input feature map, the classification score and offset are outputted to the associated default box by using convolution at the map position (Liao et al. 2017). According to the training phase based on the box overlap, ground-truth word boxes can match to default boxes followed by the matching scheme (Liu et al. 2016). Each map location is connected to multiple default boxes of different sizes and effectively divides words by using different scales and aspect ratios. Therefore, the design of default boxes could be more than task specified (Liao et al. 2017).

Within the SSD framework, how to design the best tiling is also an open question, and the design of the default box is highly task-specific (Li, Chen, Chen, Dai & He, 2017). During the training phase, the ground truth box matches the default box based on the box overlapping, following the matching scheme (Du, Fu & Wang, 2016). Each map location is associated with multiple default boxes of different aspect ratios. They effectively divided skeleton video clips by using scaling and aspect ratio. Unlike general objects, skeleton boxes tend to have large aspect ratios. Therefore, a "high" default box includes a large aspect ratio, including six aspect ratios (Li, et al, 2017). They used the same loss function proposed by Du et al. Let x be the matching indication matrix (Du, Fu & Wang,

2016), c be the confidence, l be the predicted position, and g be the true position. Specifically, for the i-th default box and the j-th base fact, $x_{ij} = 1$ indicates a match and $x_{ij} = 0$. The loss function is defined as Eq. (2.2),

$$L(x, c, l, g) = \frac{1}{n} (L_{\text{conf}}(x, c) + \alpha L_{\text{loc}}(x, l, g)) \tag{2.2}$$

where N is the number of default boxes that match the ground truth box, and α is set to 1.0.

2.5.2 Loss Function

In order to solve the classification problem and avoid the loss of application data, the loss function will have an impact on a specific depth model and will also be robust to the classifier (Janocha, & Czarnecki, 2017). The loss function based on the deep neural network classifier has an effective solution to the problem of probability. From a linear model to a nonlinear model, the loss function is the optimal choice. The Softmax loss function has been used in signal proceeding that could be trained in depth models (Wen, Zhang, Li, & Qiao, 2016). For face recognition, the loss function could be used for training. The central loss function could be easily optimized in CNN, and the training of deep features could be achieved. Using the central loss function, an object information could be quickly captured.

How to define the loss function of CNN has great significance for learning the discriminative features of contour detection. The clustering process converts the binary classification problem into multiple types of problems, which could be solved by minimizing the SoftMax function and the loss function used in the standard CNN. Among them, the binary classification problem is defined as whether the predicted image block belongs to an outline or a non-contour. The multiclass problem is whether the predicted image block belongs to each shape class or negative class (Ko, Kim, & Jun, 2005). For contour detection, misclassification between shape categories is negligible, and contour patches are classified as backgrounds as an error or vice versa. So the SoftMax function equally penalizes the loss of each class, which is not suitable for learning the discriminative features between the contour patch and the background patch (Reponen, Huuskonen, & Mihalic, 2008).

Therefore, based on the observation of the losses for positive versus negative in the training, a new objective function was defined, that is combined an extra loss for contour versus non-contour with the SoftMax loss. The loss is shared between each shape class and is named as positive shared loss (Li, & Yang, 2003). At the same time, additional

losses, led to better regularization, can be in better contour feature learning. Contour detection is conducted by using classifiers (Shen, Wang, Wang, Bai, & Zhang, 2015) and a standard non-maximum suppression scheme (Dollár, & Zitnick, 2013), computationally and efficiently structured forests are used as the deep features for contour and non-contour classification. The most advanced results are based on the Berkeley Segmentation Dataset and Benchmark (BSDS500) (Arbelaez, Maire, Fowlkes, & Malik, 2011).

Therefor,e two new algorithms were proposed. The first is that the method of optimizing the exponential loss using parallel update under the premise that the number of features is not too large. This method usually converges faster than the sequential update method. The second method is a parallel update method for logistic losses. Based on the unified processing of exponential and logical loss functions, a new convergence is derived, two algorithms (Basu, & Ebrahimi, 1991) were simultaneously presented and proved. Then, sequential-update algorithms for the two loss functions are described and analyzed. The similarity between the enhancement and logistic regression loss functions was calculated (Rriedman, Hastie, & Tibshirani, 2000).

2.6 Convolution Neural Network (CNN)

In 2013, it was proposed how CNN became more powerful (Sainath, Mohamed, Kingsbury & Ramabhadran, 2013). Another type of spectral correlation that exists in neural networks could be used to reduce spectral changes. Using a CNN model for large vocabulary presentations, includes 2 convolutional layers, 4 fully connected layers, and 3 policy layers; it will be found that CNN, in this case, can meet the prerequest. This shows that under premise of the large model, CNN is more stable and its performance is much better.

CNN is a backpropagating neural network with 2D weight kernel. The image input into the CNN is generated by using two kinds of ROIs, one is averaging and subsampling, the other is feature extraction based on texture. It is then arranged and finally entered into the CNN model. In 1996, CNN was used to conduct mammograms to investigate whether ROI can classify normal tissues (Sahiner, Chan, Petrick, Wei, Helvie, Adler & Goodsitt, 1996). In this process, Receiver Operating Characteristic (ROC) was used to evaluate the accuracy of CNN for classification accuracy. After CNN parameter integration, the results could be detected using ROC. The results show that CNN can detect and classify the content of X-ray photos.

Against CNN in 2015, MatConvNet was proposed to help CNN implement open source toolbox, the environment for deep integration was implemented in MATLAB. MatConvNet can help the CNN framework to be more straightforward, making it easier to perform under the operations using MATLAB functions, while providing filters for convolution operations. Using MATLAB codes makes it easy for CNN to present the framework. This method could be used to train large datasets on computers with both CPU and GPU configurations (Vedaldi & Lenc, 2015).

In 2014 (Huang, Qiao, & Tang, 2014), the use of CNN for high performance computing to solve the problem in distinguishing text information from background in complex texts was proposed. CNN outliers could be distinguished from the overall textual information, including background clusters and objects in the background, while using Maximally Stable Extremal Regions (MSERs) and sliding window methods. MSERs can reduce the number of windows being scanned, and the sliding window can distinguish the connections of multiple characters in the detected text. In this way, strong robustness could be achieved.

In the process of image recognition, the accuracy of image recognition will be affected by camera noise, saturation and image compression. In 2014 (Xu, Ren, Liu & Jia, 2014), a deep convolutional neural network was proposed. Deep convolutional neural networks can capture these outliers and also reference a separable value. The structure performs a deconvolution operation on robustness. The combination of this method has good performance on the problem of image convolution.

In speech recognition, CNN model and the hybrid NNHMM framework were combined to perform speech recognition in 2012. Speech datasets were used to improve recognition performance. A pair of local filters and a maxpool layer are added at the bottom of the neural networks. By using this method, the speech spectral changes are normalized, the CNN model could be well recognized in the speech. CNN can achieve a relative error of more than 10%, and the result is already able to meet current needs (Abdel-Hamid, Mohamed, Jiang & Penn, 2012).

At present, the development of CNN is still very rapid, its usability is still very high, but the development of the medical field is still relatively slow. In 2016, CNN was proposed for two specific problems lymph node (LN) detection and interstitial lung disease (ILD), through five cross-validation methods (Hoo-Chang, Roth, Gao, Lu, Xu, Nogues & Summers, 2016). It was found that the CNN model has a higher performance for the classification of medical imaging.

In the process of image recognition or detection, most of the models still need manually labeling for the datasets. Although the accuracy of manual marking is relatively high, it wastes a lot of time. Therefore, a CNN model for predicting the depth of single-view images under the framework of unsupervised learning is proposed (Garg, BG, Carneiro & Reid, 2016). This method was implemented by training the network through using means of automatic coding. In order to ensure the accuracy of the automatic marking, an image with a significant displacement between various views is used for inverse distortion during training so as to reconstruct the image. By applying such a method, the marking time could be saved, and the sensor is not required to be corrected.

2.6.1 Convolution Layer

In 2011, convolutional layer was trained by using the size and number of datasets and the connection parameters (Ciresan, Meier, Masci, Maria Gambardella & Schmidhuber, 2011). The effective displacement is performed on the input image, and the filter is adjusted according to the size of kernel, and the filter is learned under the framework of the convolution layers.

The convolution layer is to input the information to the next layer after passing the convolution operation. The convolutional layer has relationship with the size of the input imaging data. More convolutional layers, more accurate the classification of features is, and the deep convolutional neural network is more accurate for identifying problems. For the difference of filtering different images, the edge of target object could be detected by using the feature mapping method in the convolution process. The images of different types of filters after convolution are also different.

For the convolution layer, as the first layer of CNN model, it is mainly through each layer to receive the characteristics of the input image, this feature needs to be used within a range of perception (Abdel-Hamid, Deng & Yu, 2013). Each neuron is used to share the same weight information, receive the frequency offset of different inputs, and then perform a convolution operation in the convolutional layer to input the result into the next layer. When the convolutional layer calculates the resolution to be activated, there will usually be some lower resolution parts, which will be represented by a pooling layer. The pool function calculates that the activated data is applied to the neurons and then generated from the same feature map in the convolution layer.

2.6.2 Feature Map Layer

In 2007, in the study of handwriting recognition (Lauer, Suen & Bloch, 2007), a feature processor was used to extract features from raw data to generate feature vectors. The eigenvectors are only extracted from the original data.

In 2011 (Masci, Meier, Cireşan & Schmidhuber, 2011), the method of feature extraction using the autoencoder was proposed. It was verified by using cross-explanation. This is an unsupervised layering method. Using maxpooling layer with filters, pretraining the CNN's initialization network can make the accuracy higher, and the obtained feature map is more accurate in the case of classification.

In neural networks, when signature recognition is performed, because the signature itself has a problem of high line similarity, in this case, the method based on geometric features is selected, and the feature of signature image could be employed to achieve matching (Huang & Yan, 1997).

2.6.3 Bounding Box

User-provided object bounding boxes serve as a simple and popular interactive paradigm for many existing interactive image segmentation frameworks that tend to use bounding boxes to exclude content outside of the target object (Greig, Porteous & Seheult, 1989). The bounding boxes are used to enhance topological priority to avoid the solution from lavishly shrinking (Lempitsky, Kohli, Rother & Sharp, 2009). The bounding box is the most natural examples for user interaction that have been studied and implemented. Because it is very straightforward for the users that could be completed within only two times of clicking the mouse (Blake, Rother, Brown, Perez & Torr, 2004). However, the user-specified bounding box limits the attention of segmentation to its interior, the property is difficult to be merged and formalized; that means, the bounding box based segmentation is difficult to access each partial frame of the boundary (Boykov & Jolly, 2001). In the past, it was normal to reduce the energy by constraining a group, which was too loose for the bounding box.

A suitable bounding box segmentation was proposed in a new and tight enough way, which is a new segmentation framework that enables higher levels of tension (Blake, Rother, Brown, Perez & Torr, 2004). The proposed framework used global optimization techniques such as convex continuous optimization and graphical cutting to avoid the trap of local curve evolution (Lempitsky, Kohli, Rother & Sharp, 2009). This framework is

segmentation based, the energy is minimized, so that the edge could be processed uniformly.

2.7 Multilayer Perception

Multilayer Perceptron (MLP) is a type of neural networks that consists of input layers and nodes which compute one or more hidden and output layers (Gardner & Dorling, 1998). Among them, a nonlinear function is needed to calculate the output, and the backpropagation algorithm could be used to train the classification and regression of the MLP. MLPs are feedforward neural networks, which are composed of several layers of nodes through a one-way connection; the model is trained by using means of backpropagation. The basic component of MLP is the input vector **X** composed of N dimension and the output vector is M dimension (Zanaty, 2012).

Multilayer perception (MLP) plays a supervisory role in the recognition of classification applications. They are also divided into classification and regression (Murtagh, 1991). Multilayer perception could be used instead of traditional methods. Set **x** to be a vector in binary; o is the scalar output; **w** is the weight. Then the perceptron algorithm is Eq. (2.3).

$$o = \sum_j w_j \, x_j \tag{2.3}$$

Let θ be some threshold. If $o \geq \theta$, when we would have threshold $o < \theta$ for the given input, is incorrectly category. We therefore seek to modify the weights and the threshold.

In order to reduce the occurrence of misclassification, so set $\theta \leftarrow \theta + 1$.

When $x_j = 0$, then no changes w_j. If $x_j = 1$, then set the $w_j \leftarrow w_j - 1$ in order to mitigate the impact on weights.

When the output is less than the given threshold, this means that the value should be set larger when the value is initially entered. This method achieves multiple updates. The updated weights and thresholds are:

$$o = \sum_j w_j \, x_j \tag{2.4}$$

$$\Delta\theta = - (t_p - o_p) = -\delta_p \tag{2.5}$$

$$\Delta w_i = (t_p - o_p) \, x_{pi} = \delta_p x_{pi}. \tag{2.6}$$

where θ changes in the threshold for pattern p and w_i updates weights for pattern p.

2.7.1 Object Classification

In 1992 (Guo, & Gelfand, 1992), trees are classified based on nonlinear features. An

effective pruning algorithm for trees was also used, which could be compared on the basis of waveform recognition and character recognition. This classification method can reduce error rate; the training time is shorter to meet the demand.

The key issue in visual classification is to identify the settings of related classes. In 2009, a large number of characterizations of classified objects were proposed (Gehler, & Nowozin, 2009) for setting up multiple classification items, and conducting kernel learning and comprehensive evaluation of these characterization and classification items. Such an approach could be used for classification using the expected combination of methods and can effectively deal with related problems. In 2008 (Epshtein, & Ullman, 2005), a method was proposed to automatically perform object classification for layered training. According to the fragment of the information, the fragment is decomposed into the best components of the classification object. Learning of the entire structure is accomplished by using the training samples. This method is more informative than the one without hierarchical feature extraction.

In 2014, Yoon Kim (Kim, Y., 2014) experimented CNN-based classification tasks by using pre-training. For a simple CNN model, it has two characteristics for less parameter adjustment and static vector, the accuracy is satisfactory. Overall performance could be improved by making small adjustments to specific vectors. In the case of only one layer of convolution, the need for classification can also be achieved.

For target detection, the first thing is to perform target detection (Javed & Shah, 2002). The target detection of moving objects cannot be recognized at one time because of speed, bounding box, etc., hence object motion is needed. Recurrent Motion Image (RMI) is used to provide vectors for feature extraction and calculate the target object of repeated motion.

In 2011 (Song, Chen, Huang, Hua, & Yan, 2011), the output of one task was proposed as another task. This iterative method performs object detection between each other.

2.7.2 Logistic Regression

The main purpose of logistic regression is to minimize the errors between the training data and the prediction data, which gives an output value of 0 or 1. What the classifier could achieve is an image with n_x dimension vector $\mathbf{x}$ and the label of $\mathbf{y}$. It is estimated that this image is the probability of the image $\hat{y}$, the probability is expressed as Eq. (2.7).

$$\hat{y} = p\,(y = 1\,|\,x),\, 0 \le \hat{y} \le 1 \tag{2.7}$$

When there is a large number of images, in order to display the probability, it is

necessary to use a linear fitting method to complete the classifier by finding the law. Specifying an n_x-dimension vector **w** and a vector **b** as parameters, the probabilities obtained are expressed as Eq. (2.8).

$$\hat{\mathbf{y}} = \mathbf{w}^T X + \mathbf{b} \tag{2.8}$$

Logistic Regression was employed to develop models from cluster training data and validated them by combining external data from multiple centers to develop clinical predictive models. Variable selection sometimes leads to a significant relative bias in estimating the predicted effect, but this has little effect on the performance of the model in the simulation (Peduzzi, Concato, Kemper, Holford & Feinstein, 1996).

Multi-level logistic regression (Peduzzi, 1996) has been employed to ovarian tumor classification. The influence of the parameter estimation and the prediction performance of the Logistic multilevel regression model were also studied. Models were built in different numbers of samples from source groups with a degree of clustering and tested the predictive performance of the model. Among them, the LMER function was used to fit a multilevel logistic regression model using Laplace approximation (Bates, 2007) and RMS package was used for model evaluation. Finally, the number of EPVs determines the bias in the parameter estimates of the predictive models developed using multilevel logistic regression in cluster data and the resulting predictive performance (discrimination and calibration) of the model in external data (Peduzzi, 1996) (Ng, & Jordan, 2002).

2.7.3 Localization

Experiments are conducted based on the ImageNet ILSVRC 2013 dataset and advanced results on positioning and inspection tasks. The size and position of the images will be significantly changed when an image of the ImageNet dataset is selected to contain the majority of the images to be roughly centered. Positions are applied to a sliding window and in multiple scales using ConvNet (Sermanet, et al. 2013). However, many viewing windows may contain fully identifiable parts of the object but not the entire object, although there are good classifications, positioning and detection are not good. The training system can generate a distribution of categories for each window and produce a prediction of the position and size of the bounding box of the object relative to the window (Sermanet, et al. 2013).

ConvNets have been proposed for detection and location using multiple scales sliding windows. As early as the early 1990s, multiple strings, faces and hands were proposed (Matan, Baird, Bromley, Burges, Denker, Jackel & Thompson, 1992). Recently,

ConvNets in natural images, face detection and pedestrian detection show the most advanced performance (Garcia & Delakis, 2004). By combining multiple location predictions, the detection could be performed without training the background samples, and a time-consuming and complex bootstrap training process can be avoided (Sermanet, et al. 2013).

The classifier and regression network were employed at all locations and scales to generate object bounding box predictions. It is only necessary to recalculate the final regression layer after computing the classification network so that the output of the Softmax layer of class at each location provides an object of class in the corresponding field of view.

In the second step, regression training is performed after completing the reliability configuration for each boundary frame. The regression network is a merged feature map input from the fifth layer, with two fully-linked and hidden layers, respectively. The resulting output layer that ultimately specifies the edge of the bounding box has 4 units, with (3×3) copies as well as the Δx and Δy displacement.

In the third step, after increasing the resolution of the first step prediction, the ratio of the predicted object related to each window position is regressed, the bounding boxes are merged and accumulated to a small number of objects.

In the fourth step, the network is applied to the Imagenet 2012 validation set based on localization criteria. In the end, they achieved a lower error rate by combining multiscale and multiview methods with regression predictions for all spatial locations at both scales (Sermanet, et al. 2013).

2.7.4 Overfitting

Deep neural networks are currently a suitable research method for solving big data problem, but the most serious problem existing in deep neural networks is the overfitting in the process of model training. In fact, the speed of training itself is slow, it is difficult to solve the problem of overfitting through multiple predictions. Dropout is a technology that can fix overfitting problems at present (Srivastava, Hinton, Krizhevsky, Sutskever & Salakhutdinov, 2014). In order to prevent too many common connections, some units are randomly discarded during training. Using smaller weights to predict the results of these sparse networks, this method can reduce the occurrence of overfitting.

Since the discarded units are independent, the fixed probability of each unit is preserved, and this fixed probability is independent on other units. Usually, the fixed

probability is chosen to be set to 0.5 because most network tasks have an optimal value; but for the unit to be input, the fixed probability is closer to 1.0.

In order to resolve the problem of common adaptation between these units, some are hidden by random discarding. Units are the best way to avoid overfitting, and dropout has been used in many areas, such as object classification, face recognition, character recognition, data analysis, and more.

For overfitting problems, corrections can also be made through both reverse training and early stop training (Caruana, Lawrence & Giles, 2001). The reverse training method is used to avoid the overfitting problem, mainly because there are great differences in the overfitting between different model regions. Overfitting can be avoided by reverse training.

Avoiding stopping early training for large networks is due to early stopping training can protect low-non-linear regions (Caruana, Lawrence & Giles, 2001). If it is assumed that the phenomenon of overfitting occurs globally, then it needs to be solved by weight attenuation. The following equations can be used to generate a multinomial fit as shown in Eq. (2.9):

$$y = \begin{cases} -\cos(x) + v & 0 \le x < \pi \\ \cos\big(3(x-\pi)\big) + v & \pi \le x \le 2\pi \end{cases} \tag{2.9}$$

MLP changes the size of parameters by using means of backpropagation to improve the overfitting problem in highly nonlinear regions. For conjugate gradient training, it is much inclined to train a scale network. It is also found that the backpropagation network is more effective than the conjugate gradient network in terms of nonlinear regions with different degrees because the function of backpropagation tends to make learning smoother, while the function of conjugate gradient learns complex features.

2.8 Data Augmentation

In the process of using neural networks for training, it is easy to cause overfitting problems, especially when conducting recognition studies, increasing the number of data sets, expanding the number of training data sets, deforming the data and increasing the amount of data by rotating (Perez, & Wang, 2017). Data enhancement can help improve the accuracy of the task and could be applied to more complex research issues. Data enhancement is also a method that has the function of optimizing data. In 1999 (Meng & Van Dyk, 1999), a method was proposed to modify the parameters and enhance the amount of data. The data was enhanced by using spatial model, and the data could be

modified directly to quickly mix the Markov chain.

Data augmentation is the formation of new samples by transforming training samples in order to improve the stability and robustness of the classifier. It is now proposed that small conversions could be performed for each sample, optimized on the trust area and linear operations (Fawzi, Samulowitz, Turaga & Frossard, 2016). After data augmentation, the samples are integrated into the input neural network to perform random gradient descent operations in neural networks.

Generally, data augmentation can improve the performance of neural networks. For the neural networks, the sensitivity is poor. In data enhancement, color projection and lens distortion are selected. The most common data augmentation is to convert the color and generate a new color sample map by randomly converting the color of the target image (Wu, Yan, Shan, Dang, & Sun, 2015). The method of lens distortion is true. The degree of distortion and the type of distortion are randomly generated, which can ensure the quality of samples after data augmentation. Specified to the application, in order to reduce the degree of influence of overfitting on the final accuracy, the sample is subjected to data augmentation. All images are cropped, and the cropping method is selected randomly. After cropping, an image can also be rotated. The rotation angle of the image is much abundant; finally, the pixels of the image are randomly modified (Karpathy, Toderici, Shetty, Leung, Sukthankar, & Fei-Fei, 2014).

When using CNN for discriminative spectro-temporal patterns, this model is well suited for classifying environmental sounds, but research has been an obstacle in this area. By analyzing the results after data augmentation for each type of audio, it could be seen that data augmentation can also improve the accuracy of the model.

2.9 MobileNet Model

For mobile and embedded deep learning, the use of streamlined model of MobileNet for recognition is a new approach based on deep neural networks that performs separable convolution operations (Howard, Zhu, Chen, Kalenichenko, Wang, Weyand, & Adam, 2017). It solves the problems by utilizing the standard volume integration into depthwise convolution and pointwise convolution. The depthwise separable convolutions are with width multiplier α and resolution multiplier ρ shown as Eq. (2.10).

$$D = D_k \times D_k \times \alpha M \times \rho D_F \times \rho D_F + \alpha M \times \alpha N \times \rho D_F \times \rho D_F \qquad (2.10)$$

where $\rho \in (0, 1]$ which is typically set implicitly, $\rho = 1$ is the baseline MobileNet and ρ

< 1 is the reduced MobileNets (Angin, Campbell, Kounavis, & Liao, 1998). Resolution multiplier has the effect of reducing computational cost by ρ^2.

(a)   (b)

Figure 2.2 Standard Convolution and Depthwise Convolution Structure

For standard convolution, its relationship with batchnorm and ReLu (Wang, Xia, Tang, Li, Yao, Cheng, & Yang, 2016) is shown in Figure 2.2(a). Regarding MobileNet, there will be a 3 × 3 deep convolutional layers and a 1 × 1 pointwise convolution layer. After each convolutional layer, there will be a batchnorm layer and a ReLu (Maas, Hannun, & Ng, 2013) shown in Figure 2.2(b).

2.10 Faster R-CNN

For the Faster R-CNN, in the process of progress, although the detection time is shortened, but also some bottlenecks have been encountered. In 2015 (Ren, He, Girshick, & Sun, 2015), during the detection process of Faster R-CNN, a Region Proposal Network (RPN) was introduced, which can share the characteristics of the detected object. RPN is a completed convolutional network that could be used to predict the location of an object being detected. Because RPN and Faster R-CNN could be alternately optimized at runtime, the two models could be run simultaneously, and this will improve detection speed (Tian, Huang, He, He, & Qiao, 2016). The main CNN with multiple convolutional

layers will take the completed image as input and no longer use CNN for each R-CNN. The detection method of Region of Interests (RoI) is distinguished by using the selection method based on the feature map. RoI is used to reduce the size of the feature map. By such method, the region with fixed height and width is obtained and set to super parameters.
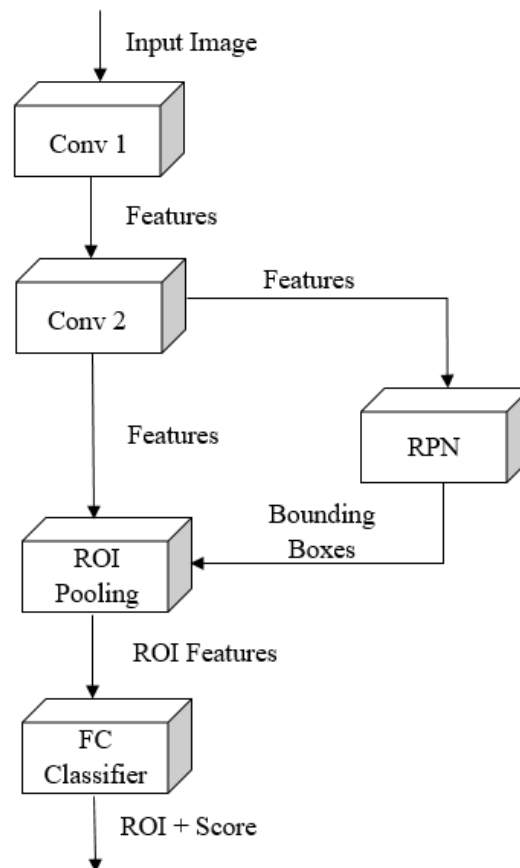

Figure 2.3 The architecture of Faster R-CNN

The basic framework of Faster R-CNN is shown in Figure 2.3. It uses a regional network (RPN). The RPN can generate a number of bounding boxes (ROI) to propose these ROI proposals as a proposal for classification. Features are aggregated by sharing (Granger, Kiran, & Blais-Morin, 2017).

Faster R-CNN is based on a training network to ensure that the input image is not deformed, thus obtain a feature map. RPN is divided into two different paths. A 1×1 convolution uses a linear activation function for the regression operation (Girshick, Donahue, Darrell, & Malik, 2016). Each bounding box has four vertices, and these four points are set to (dw, dy, dx, dh).

The other is to distinguish the entire area and determine whether the anchor of each area contains the target. The specific operation method is to set a threshold and select a

probability ratio greater than the threshold range, wherein a portion smaller than the minimum threshold is discarded. For example, if the threshold is set to 0.3, then the portion which is smaller than 0.3 is discarded. The part between them is background, and the one which is larger than 0.7 will be a positive example.

Faster RCNN is used for object detection, which is updated on the basis of fast R-CNN (Girshick, 2015). Faster R-CNN improves training speed and accuracy, and it can classify target objects much efficiently. Face detection based on deep learning is a hot topic now. Face detection using faster R-CNN is a further experiment in this study (Sun, Wu, & Hoi, 2018).

Faster R-CNN is based on the framework of Fast R-CNN, feature concatenation, hard negative mining, multi-scale training, model pre-training, and proper calibration of key parameters. The extracted feature maps are screened and classified, and then entered into the proposal layer for analysis, and then entered into the classification layer for classification detection. Because of the RPN layer, accuracy of this detection is also improved.

In the current study of currency classification, fitness classification is a method used to determine whether a currency could be recycled or replaced. The current technology is usually a study that is carried out after envisioning the denomination and direction of the currency, which is actually pre-classifying the currency. Aiming at this problem, a fitness-classification method based on deep learning is proposed (Pham, Nguyen, Kim, Park, & Park, 2018). This method can extract and classify features by using image sensor and CNN regardless of the currency of denomination and direction.

The Regional Proposal Network (RPN) takes an image of arbitrary size as an input image, proposes the image and stores the object according to the proposal. In this process, the full volume network is used for modelling, because the final is to share the faster R-CNN network, a shared network model is established. As shown in Figure 2.4, the single location marked will be shared in the fully connected layer by sliding the window during mini-network operation. Because it is necessary to make a prediction proposal for the k regions at the same time, there will be $4k$ output in the regression layer, and the coordinates of the box are of the length of k; the output for the classification layer has the number of $2k$, which is equivalent to parameterizing k anchors, and each anchor will slide the window.

The centre point is discussed to use 3 scales and 3 aspect ratios, yielding $k = 9$ anchors at each sliding position. This method is mainly used to translate and not change the function of the anchor when it is translated and calculated. When an RPN is to be trained,

each anchor needs a binary tag, which assigns two anchors: (a) the anchor/anchors with the highest Intersectionover-Union (IoU) overlap with a ground-truth box, or (b) an anchor that has an IoU overlap higher than 0.7 with any ground-truth box. If the IoU is lower, then it needs to be defined as a negative label with a value of 0.3. What is not accurate or harmful is the meaningless training goal (Ren, He, Girshick, & Sun, 2015).
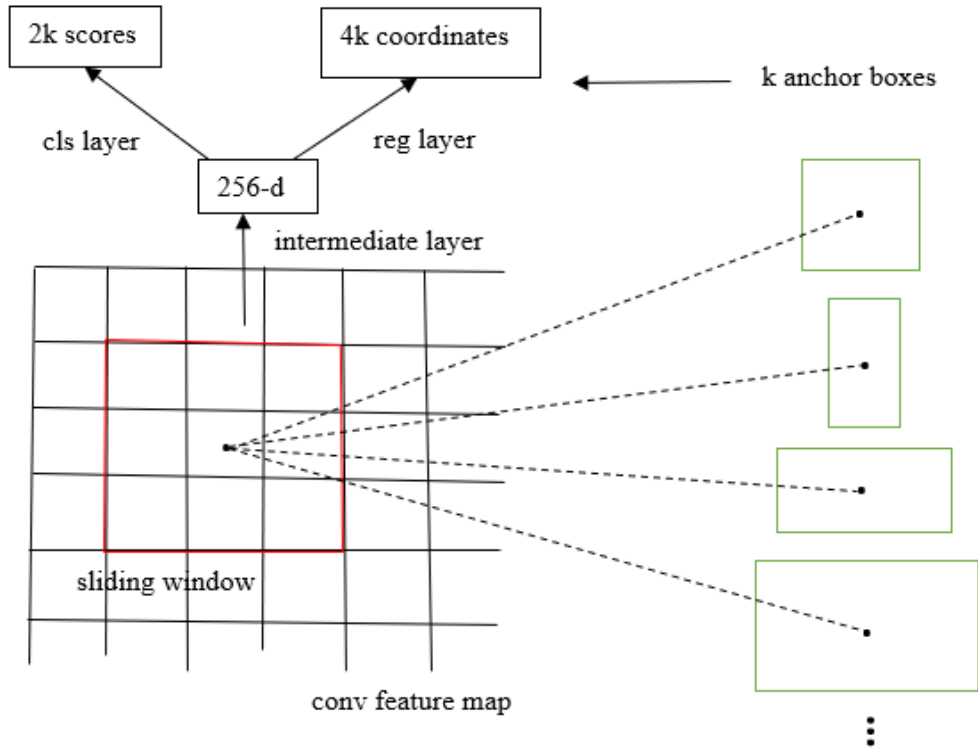

Figure 2.4 Region Proposal Network (RPN)

Faster R-CNN can be used for pedestrian detection. The use of RPN for regional proposals has a good effect on the study of pedestrian detection (Zhang, Lin, Liang, & He, 2016). The multiple heights of pedestrian average aspect ratios with the original RPN are compared to find that unsuitable heights have an impact on accuracy, which reduces the accuracy of detection. In order to meet the needs of different pixels, different scales of anchors for research are used, because of multi-scale detection objects, it is much appropriate to use multiscale anchors.

According to the RPN proposal, a fixed-scale feature will be generated and then it can be better trained. Using RPN can help the model to perform subsequent feature extraction (Lin, Dollár, Girshick, He, Hariharan, & Belongie, 2017).

$$L\left(\{p_i\}, \{t_i\}\right) = \frac{1}{N_{cls}} \sum_i L_{cls}\left(p_i, p_i^*\right) + \lambda \frac{1}{N_{reg}} \sum_i p_i^* L_{reg}\left(t_i, t_i^*\right) \qquad (2.11)$$

The loss function of RPN are shown in Eq. (2.11), where i is predicted as an object in

the anchors. The ground-truth label p_i^* is 1 if the anchor is positive and 0 if the anchor is negative. The four vectors of the bounding box are represented by t_i. t_i^* represents the relationship between the ground-truth box and the anchor. L_{cls} indicates the log loss of classification, L_{cls} is divided into two categories, one is an object and the other is no object. Regarding the regression loss function, R is a robust loss function, and p_i^*, L_{reg} is expressed as the activation regression function of this function (Zhang, Lin, Liang, & He, 2016).

2.11 Summary

This chapter is mainly about analysis of the past literatures. For the currency recognition we will conduct, we surveyed from deep learning, detection, classification and recognition, and also studied the training model that may be used. Analytical methods and learning experiences will be adopted for our project from the literature.

Chapter 3
Methodology

In this chapter we mainly introduce the specific methods applied to this research project, how to achieve the requirements of currency recognition, describe and introduce the details from each step. Through these details, we can understand our entire research more clearly.

3.1 Introduction

Throughout the study of literature, we have a basic concept of how to carry out currency recognition. From the aspect of data collection, we need to start from how to quickly obtain high-quality clear currency images to build the dataset and how to ensure the clarity of currency images. There is a need to ensure that the location information of the currency is as rich as possible to support our idea about currency recognition.

To satisfy the research needs for the volume of data, it is not enough to acquire the morally valid images from videos; in this case, we need to find a way to guarantee the data quality while increasing the amount of data and ensure that the quality of the images does not decrease. For this purpose, it is standard practice to enhance the data collection. Its primary objective is to help expand the amount of data and improve the data through different steps and obtain new data images. Such data images can be different. The aspect of the original data map is modified to complete the dataset.

We finally chose to use the SSD model as the framework for currency recognition. We chose Convolutional Neural Network (CNN) to remove noises because there are a lot of noisy currency images in the data collection, the convolutional layer through CNN can better help us to meet the requirements of currency recognition.

3.2 Research Design

Our main research content is currency recognition. Before we start, we need to make a design of the research project from the initial collection of currency images to the training dataset, and then the experimental results. Before we formally start our project, we need the basic ideas for specific content of each part.
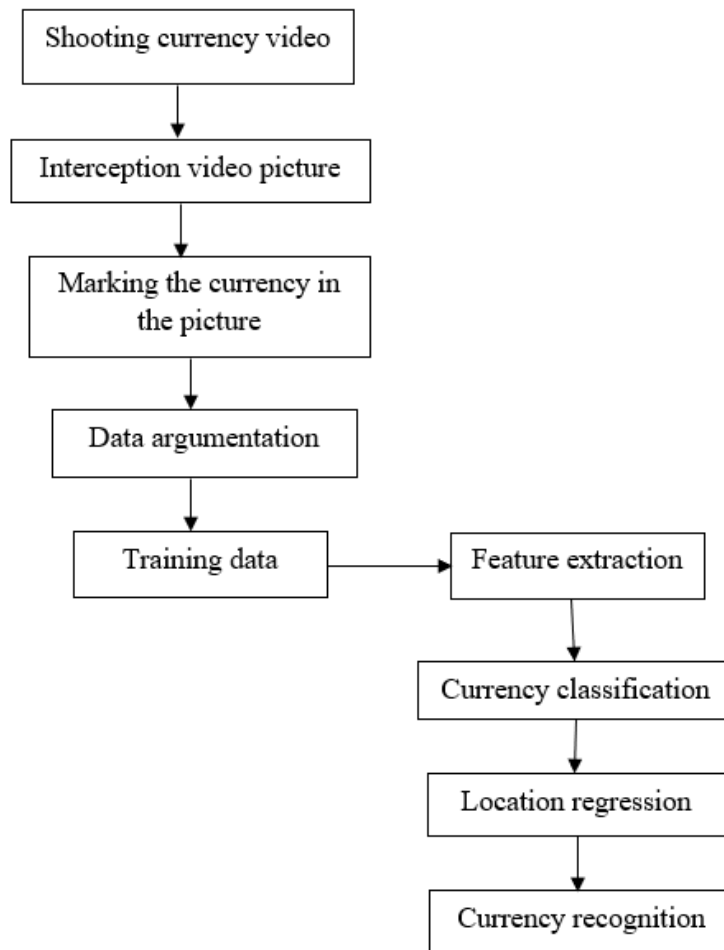

Figure 3.1 The steps of currency recognition

As shown in Figure 3.1, this flowchart is the basic concept of the specific research process. Regarding currency recognition, collecting datasets is the most basic action; we acquired a single frame of the video to get the data image of the currency. Before training the data, the data image needs to be filtered, the image that meets the experimental requirements is selected as a dataset, and then to conduct the data argumentation using the marked images so as to increase the size of the dataset. After fulfilled the work, then we send them to the MLP layer for currency classification, finally we complete the currency recognition.

3.3 Data Collection

3.3.1 Getting Dataset

Before the start of this project, we thought a lot of different ways for data collection, such as using a picture of a currency downloaded from the Internet as a dataset or using a printed currency to make a video shot as the dataset.

However, we found through experimentation, these two methods are not workable, because the first method is difficult to meet the requirement that we need to move the currency at different angles during the recognition process; in the second method, the dataset was the printed money pictures, currency details lost; if we use such a dataset, our research results will be less rigorous.

In order to enhance our research results, we finally chose to use real money as our data source. We have chosen 5NZD, 10NZD, and 20NZD as the monetary denomination for currency recognition. The currency of each denomination will have front and back sides. We first shot videos of each side of these three denominations. During the video shooting process, we need to make sure that the currency is flat and the currency could be fully displayed in the videos. At the same time, we moved the currency forward, backward, left, right, and bevel, which enriched the content of the dataset. We also need to ensure that when we are shooting video, there is enough luminance around it, because we expect to capture the details of currency more clearly. We put these videos into six folders, we named the folder as 5NZD-F, it means that this folder is the front side of 5 New Zealand dollars as shown in Figure 3.2.

After obtained the videos of the currency, we need to edit the video and secure each frame that the currency is clear and complete in each picture, there are not any parts out of the boxes. In this case, we get 50 pictures of each denomination, so we have obtained a total of 300 pictures as a dataset, the resolution of each picture is 1280×720 as shown in Figure 3.3.
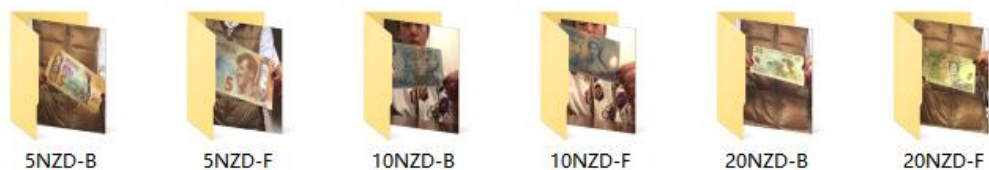

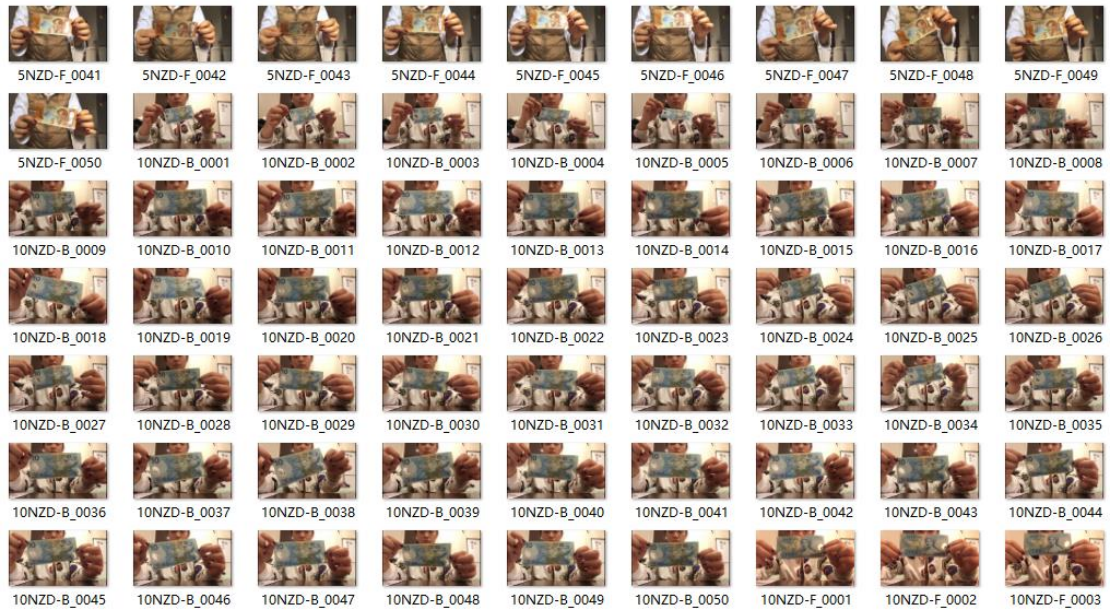Figure 3.2 The six folders of the currency video

Figure 3.3 The dataset of the currency

3.3.2 Marking Data

In the above data collection, we know that in order to reduce the workload of manual labeling, we have selected a total of 300 clear currency pictures as the dataset.

To highlight the benefits of manual tagging (Elhofi, & Helaly, 2015), we did not use the parallel rectangles we used in the past to perform inspection tasks. Instead, we chose the quadrilateral shown in Figure 3.4 to mark the position of the currency. Its biggest advantage is that it can accurately mark not only the position of the currency, but also the state of the currency.

For manually labeling, we chose to use the MATLAB GUI for marking (Chen, & Zhang, 2006). The manually marking process starts from the upper left corner of the currency image and marks the four corners of the currency image in a clockwise direction. After the markup is over, we can drag the marker box to make adjustments, which will make the markup of the entire currency image more accurate. At the same time, it avoids the problem that the mark box cannot accurately mark the currency in the image because of the influence of surrounding environment.
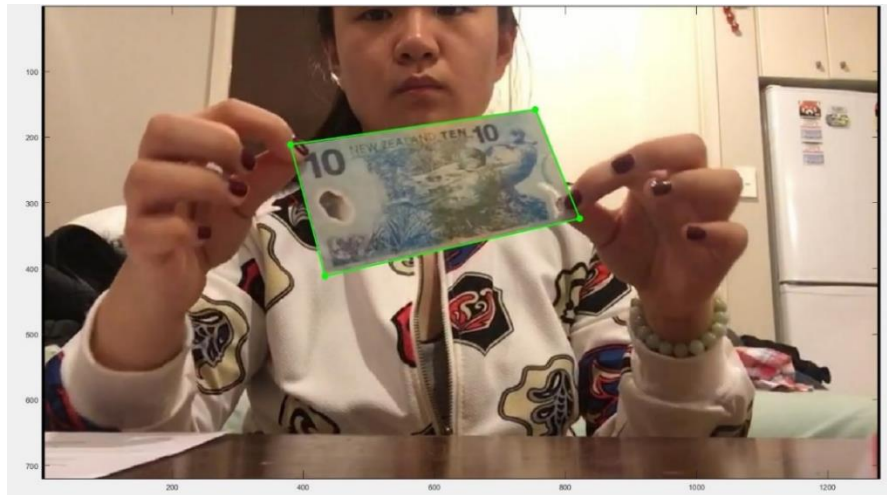
Figure 3.4 Using the quadrilateral to conduct the manual marking

3.3.3 Data Argumentation

So far we know that after the data collection, we have carried out the procedure of data manually marking; we grouped all the collected data into six categories, we obtained a total of 300 valid currency raw images for the purpose of recognition. But this amount of data is insufficient, it is difficult to support the training of deep learning. Therefore, in order to achieve better training of the data, we have fulfilled data argumentation on the original data and generated new data; the overall data volume is increased by using this method. In the process of data argumentation, we have five steps (Ronneberger, Fischer, & Brox, 2015) as shown in Figure 3.5.
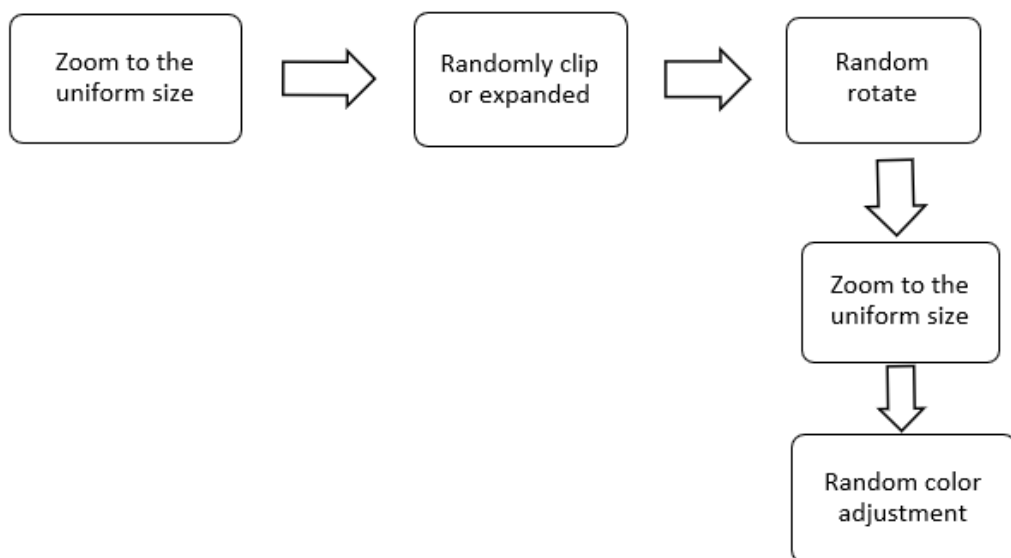

Figure 3.5 The steps of data argumentation

- *Zoom to the uniform size*. We know that during the data collection process, the collected data will be different in size due to pixelization problem and distance problem, but we need to scale the image size in the dataset to a uniform for currency training. Therefore, we need to scale the dataset to a size that can accommodate model training without affecting currency recognition. Usually the size of video input is Full HD (1920 × 1080), HD (1280 × 720), qHD (960 × 540), nHD (640 × 360), etc. We chose to normalize the data into nHD 640 × 360 as the input size of the data for the convenience of training.

- *Randomly clip or expand*. In currency recognition, we need to identify the currency in different locations to ensure the rigor of our research, but for the original dataset we collected, the diversity of the location and size of the currency is not enough, so random clipping or expansion is required to change currency position and size information. We first developed an area ratio, which means that the size of currency images as a portion of the entire screen. When the ratio in the data is greater than the original scale, we use random cropping to change the ratio and modify the currency position. When the data ratio is smaller than the original scale, we increase the original image in a random extension, fill the extension with black, and replace the currency position.

- *Randomly rotate.* In the video shooting, though we also adjusted the currency at different position, the angle of currency recognition is not sufficient. So in order to enrich the position information of the currency and increase the currency to scan the image from different angles, we choose to randomly rotate the original currency image and increase the angles of the currency image, such as rotating 180º, 90 º clockwise and 90 º counterclockwise.

- *Zoom to the uniform size*. After the random cropping or scaling or random rotation of the currency image, the size of the images has been modified again; because the training requires uniform size data, we need to scale the dataset again and make sure the size of the image is even.

- *Random color adjustment*. We know that the videos were captured in various environments; therefore, the angles will have a bias in the color due to various lighting conditions, we choose to convert the three primary colors RBG (red, blue, green) in the image to HSV (hue, saturation and color value) (Cucchiara, Grana, Piccardi, Prati, & Sirotti, 2001); we randomly adjust the saturation, hue and color values of the image to make the colors of the images are basically the same.

After the data augmentation, each of the original video frames can obtain 25 currency enhanced images. In other words, our original 300 raw data will receive 300 × 25 images in the dataset after the argumentation; the number of datasets will be expanded from the previous 300 images to 7,500, thus greatly improve the efficiency and integrity. The dataset after data argumentation can make our research experiments much accurate. In this way, we can enrich the position, size and angle of the currency, and adjust the color of the currency image to make our research work much comprehensive. Figure 3.6 is an enhanced effect of a raw data in the dataset after data argumentation.

Figure 3.6 Data argumentation

In the data enhancement, in addition to color adjustment, in remaining steps we adjust the position of the marked quadrangles. Therefore, the main difficulty in data enhancement lies in calculating the relevant parameters of the quadrilateral after transformation. Now let's discuss how quads are calculated under various transformations.

Let quadrilateral position matrix be shown as Eq. (3.1).

$$\text{p} = \begin{bmatrix} x1 & x2 & x3 & x4 \\ y1 & y2 & y3 & y4 \end{bmatrix} \tag{3.1}$$

For scaling transforms S (s_x, s_y), where s_x represents the horizontal scale and s_y is the vertical scale as shown in Eq. (3.2).

$$P' = \begin{bmatrix} s_x & \\ & s_y \end{bmatrix} P \tag{3.2}$$

For rotation transforms R (θ, w, h), where θ is the angle rotated counterclockwise, where w and h are the width and length of the image, the center of rotation is the centroid of the image. In the rotation transformation, we have Eq. (3.3).

$$P' = \begin{bmatrix} \cos\theta & -\sin\theta \\ \sin\theta & \cos\theta \end{bmatrix} \left(P - \begin{bmatrix} w/2 \\ h/2 \end{bmatrix} I \right) + \begin{bmatrix} w/2 \\ h/2 \end{bmatrix} I \qquad (3.3)$$

Among them

$$\mathbf{I} = [1, 1, 1, 1].$$

For the translation transform $T(x_0, y_0)$, where x_0 and y_0 are the translation distance between the horizontal and vertical directions, respectively; then the transformation matrix is shown as Eq. (3.4).

$$\mathbf{p'} = \mathbf{p} + \begin{bmatrix} x0 \\ y0 \end{bmatrix} I \qquad (3.4)$$

After the derivation of the quadrilateral related parameters is completed, the entire process of data enhancement could be realized.

3.4 SSD Model

We chose to use the SSD model for currency testing. First, we know that SSD is a single-stage model (Sun, Liang, Wang, & Tang, 2015), and the image will be extracted through CNN (Farfade, Saberian, & Li, 2015), extracting features (Cohn, Zlochower, Lien, &Kanade, 1998) will use two MLP (Multilayer Perceptron) to classify and locate targets as shown in Figure 3.7. For the detection task of currency recognition, we chose to set the six convolution layers of the CNN model. We will first input the currency image into the CNN model, filter through convolution layers, then determine the bounding box, input the data of the conv layer into the MLP layer for positioning and classification, and complete the basic detection.
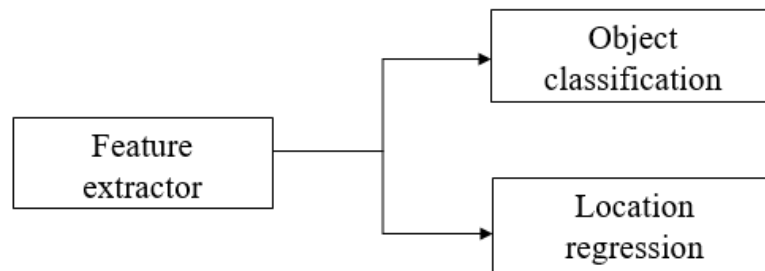

Figure 3.7 SSD model

- Input process layer

The image is stored in the computer as a matrix of three RGB color components, where the value of each element is an integer from 0 to 255. However, the DNN network model is a continuous model, we need to convert the discrete color values into continuous floating-point values for storage, we need to normalize them into the range of [-1, 1].

- Convolution layer: conv_01 to conv_06

A standard convolutional layer consists of a convolution operation and a pooling operation. In the convolution layer, we set the input matrix as **X**, the convolution template as **W**, the bias matrix as **b**, and the output of the convolutional layer is shown in Eq. (3.5).

$$Y = \text{maxpool} (\alpha (\mathbf{W} * \mathbf{X} + \mathbf{b})) \tag{3.5}$$

where '*' represents a convolution operation, $\sigma (\cdot)$ is a nonlinear activation function, and maxpool($\cdot$) represents the maximum pooling operation.

We know that convolution is a weighted sum of a small part of the input matrix according to the weights of the template. Because the method is more concise and requires fewer parameters than the summation of the entire matrix, it could be calculated.

The nature of the convolution is a linear transformation, after it continues to be superimposed, it is still a linear process. We use the nonlinear activation function to nonlinearized the results, which allows the neural network to be nonlinearly fitted. For the activation function, we know that there is ReLU (Severyn, & Moschitti, 2015), but this activation function may have problems with insufficient node training, we chose to use LeakyReLU (Springenberg, Dosovitskiy, Brox, & Riedmiller, 2014) as an activation function. The activation function of LeakyReLU (Kuo, 2016) is expressed as Eq. (3.6).

$$\text{LeakyReLU} (x) = \begin{cases} x & x > 0 \\ 0.1x & x \geq 0 \end{cases} \tag{3.6}$$

Our goal of using maxpooling is to filter the characteristics of the convolutional extraction, choose only the main part of the feature. Its operation is to divide the matrix into several grids, and the elements in each grid take the maximum value as output. After the convolution layer is superimposed, it can effectively extract high-level features of the image.

- Flatten: dropout layer

The result of the convolutional input is a matrix. To be able to use the result as input for subsequent MLPs, the matrix needs to be expanded to vectors. In order to prevent the overfitting of the model (Hawkins, 2004), we chose to use the dropout strategy, which refers to output randomly after disabling some nodes. This will increase the fault tolerance of the model, but some node failures will not affect the overall effect of this model.

- Classification MLP layer

In the fully connected layer, a vector **x** is input, and the output is another vector **y**. If we set the weight matrix to **W** and the offset vector to **b**, then the fully connected layer could be represented as Eq. (3.7).

$$y = \sigma (\mathbf{W} \cdot \mathbf{x} + \mathbf{b}) \tag{3.7}$$

where $\mathbf{W} \cdot \mathbf{x}$ is a multiplication of matrix and vector, $\sigma(\cdot)$ is a nonlinear activation function. We have already explained that LeakyReLU (Xing, Ma, & Yang, 2016) is used in the above.

Multilayer perceptron (MLP) refers to a network consisting of fully connected layers. We chose to use a two-layer full-connection layer to form a classification MLP. For classification problems, the probability vector is generally used as the output result. That is, the output of a sample is a classification vector as Eq. (3.8).

$$\mathbf{p} = [p_1, \ldots, p_n] \tag{3.8}$$

where n represents the total number of classifications, p_k represents the probability that the sample belongs to the k-th classification. In order to measure the gap between the model output and the actual classification of the sample, cross-entropy is generally used as loss function. Let the actual classification vector of the sample be Eq. (3.9).

$$\mathbf{q} = [q_1, \ldots, q_n] \tag{3.9}$$

Let the sample actually belong to the j-th classification, then there is Eq. (3.10).

$$q_k = \begin{cases} 1 & k = j \\ 0 & k \neq j \end{cases} \tag{3.10}$$

Thus, the model classification loss function is Eq. (3.11).

$$L_{\text{cls}} = - \sum_{k=1}^{n} q_k \ log_2 \ p_k \tag{3.11}$$

The smaller the loss function value, the more accurate the classification.

- Location MLP layer: local output

The structure of the positioning MLP layer is similar to the classification of the MLP layer and is also composed of two fully connected layers. The main difference lies in the coding of the output results and the design of the loss function.

Positioning the output code of the MLP layer can have a variety of designs, we design two kinds of coding as shown below as Eq. (3.12).

$$\begin{aligned} \text{box}_1 &= [x_1, y_1, x_2, y_2, x_3, y_3, x_4, y_4] \\ \text{box}_2 &= [x_1/w, y_1/ h, x_2/ w, y_2/h, x_3/ w, y_3/h, x_4/w, y_4/h] \end{aligned} \tag{3.12}$$

Among them, $x_1, y_1, x_2, y_2, x_3, y_3, x_4,$ and y_4 are the coordinates of the four vertices of the quadrilateral, where box_1 uses the original coordinates as the encoding and box_2 normalizes the original coordinates to the range of $[0, 1] \times [0, 1]$.

The model positioning loss function is Eq. (3.13).

$$L_{\text{loc}} = \sum_{k=1}^{8} f(l_k - \hat{l}_k) \tag{3.13}$$

where l is the position vector predicted by the model, $\hat{l}$ is the vector of the actual position of the note, and the smooth 1-norm is taken as f, so we see Eq. (3.14).

$$f(x) = \begin{cases} |x| - 0.5 & x > 1 \\ 0.5x^2 & x \leq 1 \end{cases} \tag{3.14}$$

The smooth 1-norm can avoid the problem of 1-norm derivative discontinuity and 2-norm gradient explosion problem.

- Prediction layer predict

In the prediction layer, we first integrate the two kinds of output results of MLP classification and positioning, then we can get the output results again. The output results are $[x_1, y_1, x_2, y_2, x_3, y_3, cls, conf]$, where $x_1, y_1, x_2, y_2, x_3, y_3, x_4, y_4$ are the vertex coordinates of the quadrilateral, cls is used to represent the maximum value of the classification probability. We use this maximum value as the final classification result, expressed as Eq. (3.15).

$$p_{cls} = \max \{p_1, \ldots, p_n\} \tag{3.15}$$

The $conf$ indicates the confidence that the sample belongs to the classification, its calculation method is Eq. (3.16).

$$conf = \frac{\exp(p_{cls})}{\sum_{k=1}^{n} \exp(p_k)} \tag{3.16}$$

- Verification layer valid

After completing the previous series of work, we need to verify the correct rate of the model. The verification process is considered in two aspects. One is whether the classification is correct. To meet this requirement, the classification is as same as the classification of our samples, it is classified correctly. The other is whether or not the positioning is correct. Generally, we use $IoU > 0.5$ as a criterion for judgment. The IoU calculation method is Eq. (3.17).

$$IoU = \frac{S_{P \cap G}}{S_{P \cup G}} = \frac{S_{P \cap G}}{S_P + S_G - S_{P \cap G}} \tag{3.17}$$

Among them, S_P (predict) indicates the area of the prediction result, S_G (ground truth) indicates the actual position, IoU is the area of the intersection of the two, and the area of the union of the two.

For traditional rectangular mark boxes, IoU calculations are relatively simple. Let the coordinates of the upper left and lower right corners of the predictive bounding box be x_1, y_1, x_2, y_2, and the actual bounding box is x_3, y_3, x_4, y_4, so it is shown as Eq. (3.18).

$$\begin{aligned} S_P &= (x_2 - x_1)(y_2 - y_1) \\ S_G &= (x_4 - x_3)(y_4 - y_3) \\ S_{P \cap G} &= \max(0, \min(x_2, x_4) - \max(x_1, x_3)) \\ &\quad \times \max(0, \min(y_2, y_4) - \max(y_1, y_3)) \end{aligned} \tag{3.18}$$

For a quadrilateral, though the area of quadrilateral itself is relatively simple to calculate, the area of the intersecting part is much complex and is not suitable for constructing a DNN network structure. Therefore, we use a quadrangular enclosing

rectangle to approximate *IoU*.

- Training layer train

We know that the original intention of using DNN model training is to use the value of the minimum loss function based on the gradient optimization algorithm. For currency detection, the loss function is composed of two parts that can ultimately be expressed as Eq. (3.19).

$$L = L_{cls} + wL_{loc} \tag{3.19}$$

This algorithm represents a weighted summation of the classified and positioned loss functions, where w is the given weight. In our training, we will use $w \in \{1, 10\}$.

Let θ be a parameter in the model, then its formula in training is updated as the Eq. (3.20).

$$\theta \leftarrow \theta - \lambda \frac{\partial L}{\partial \theta} \tag{3.20}$$

where λ is the learning rate in the model and $\partial L/\partial \theta$ represents the partial derivative of loss function $L(\cdot)$ with respect to parameter θ. Gradient calculations are derived from back-propagation algorithm based on our model structure.

As the data training progresses, the training layer continually updates the parameters of each layer, so we see the connection of the training layer. At the same time, each layer uses a random initial parameter and connects to the initial layer, we save the updated parameters to the save layer on the hard disk. Finally, the obtained data is input to the valid layer, and it is verified with the trained dataset so as to determine whether the dataset is fully trained.

In this thesis, currency detection, as a subtask of currency identification, needs to be divided into two parts in training; that is, positioning and classification in MLP, the overall learning framework is TensorFlow. Specifically, we split it into convolutional layers and fully connected layers.

3.5 Summary

In this section, we introduce how to collect data for currency recognition. By using data argumentation to increase the size of the dataset, we can improve the content of our research quality. How to use the CNN model for training is the part of our main contribution. Throughout introduction of this chapter, we can clearly understand the methods and algorithms. In the next chapter, we will introduce the experimental results and summarize the results.

Chapter 4

Results

This chapter mainly introduces the specific methods and implementations of currency recognition. The steps to implement currency recognition will be introduced on details. The experimental configuration and data arrangement for the experiment will be described in this chapter. We evaluate the results based on the experiments.

4.1 Introduction

The main part of currency recognition is still based on image recognition. In order to get our ultimate goal of currency recognition, we need to acquire images in the process of data collection. The currency image needs to be as flat as possible. Currency in the images needs to be kept intact, and the angular position of the image needs to be different. These requirements are all to make our currency recognition much accurate. We have fulfilled this requirement through using data argumentation.

The basic configuration is a prerequisite for currency recognition and we conduct better research after understanding the configuration.

When the training of networks is completed, we have obtained the outputs. For the currency, our research results are mainly based on the accuracy of four models, the training and the verification as well as the classification and positioning. The analysis leads to a model with high accuracy. In addition, we also analyzed the accuracy and loss function of the model. At a higher accuracy rate, the data was fully trained, and over-fitting was avoided during the training. It also shows that the CNN model we choose can carry out a good job.

For the result of currency recognition, we need to recognize the front and back side of the currency's nationality and confidence, at the end we will show and explain the results.

4.2 Data Collection and Experimental Environment

The main research purpose of this thesis is to implement the method of detecting and recognizing denomination and front as well as back sides of currency before a camera. Therefore, we need to collect data to support our experiments. In order to ensure the success of our experiments, we could not complete the research only from one denomination of currency, so we chose three different denominations of New Zealand dollars having different colors and patterns, which can increase the difficulty of our research.

We use 5NZD, 10NZD, and 20NZD as the sample denomination of the dataset. We obtain data by capturing the videos from the front and back sides of the three kind of currencies. In the first part of Chapter 3, we have detailed the process of data collection. Apart from the clearness of these currencies, it is also necessary to ensure that there are different distances and angles of images in order to meet the accuracy of our research outcomes.

In addition, it is worth mentioning that in order to ensure better training results, we should pay attention to the following points when collecting data:

(1) The picture is clear, the action range is small, and motion blur is avoided.

(2) The position of a currency in the screen is pretty rich, it should not always appear in an area of the screen, the possibility of occurrences in each area of the screen is equal.

(3) The currency is large in size and acquired images are from different distances and views.

(4) The currency is shown in rich shapes and has a better view so as to suit the needs of real applications.

(5) The currency should not bend, because we will use the quadrilateral to mark the position of this currency. If the currency is curved, the outlines could not be described by quads.

(6) The currency should not go beyond the screen, the full appearance of the currency is needed.

From Chapter 3, we also know that after completing the data collection, we have acquired 25 samples for each image. We have a total of 300 images as the original data. Then, we have a total of 7500 images to be as our final dataset.

After collecting the above data, we need to manually label the data (Hochstadt, 1975). In order to reduce the amount of data manually labelled, we extracted 50 frames for each

type of currency. At the same time, in order to give full play to the advantages of self-collecting and labelled data, we use the quadrilateral mark shown in Figure 4.1 to mark the position of the currency compared to the parallel rectangular frame of the object detection task. With the quadrilateral mark, the position of this currency can be described more accurately than the parallel rectangular frame. Not only can the position of the currency in the picture be reflected, but also the shape in which the currency is tilted.

We designed the Matlab GUI program shown in Figure 4.1 to mark the video frame. When marking, we just start from the upper left corner and mark the positions of the four vertices in clockwise direction. After marking the four vertices, we can drag the vertices to correct them.

In order to get accurate mark of the currency position, we chose to use manually marking. This example is given in Figure 4.2, which is the three denominations including the front and back sides.
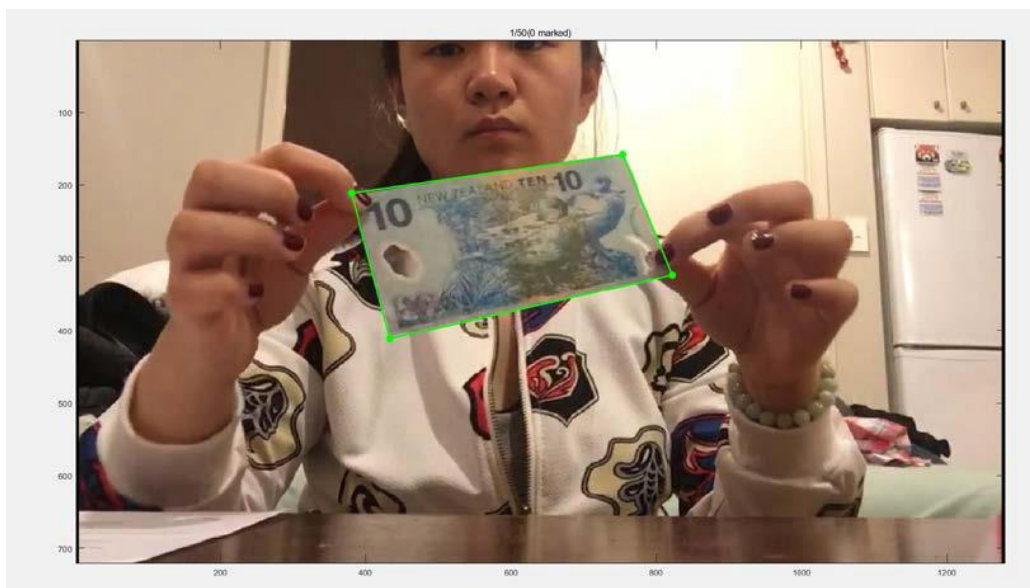

Figure 4.1 Manual marking a currency

Figure 4.2 The sample of manual marking data

The experiment is run on a laptop installed Microsoft Windows 10 Operating System using Intel Core i7 CPU 2.0GHZ. All data tags and data augumentation were carried out in Matlab R2016a; we used programming languge Python for training; we also use the platform TensorFlow as our basic framework for classification.

4.3 Currency Recognition

From Chapter 3, we know that when we design the model, we use two ways of quadrilateral coding, **box1** = [$x_1, y_1, x_2, y_2, x_3, y_3, x_4, y_4$] and **box2** = [$x_1/w, y_1/h, x_2/w, y_2/h, x_3/w, y_3/h, x_4/w, y_4/h$], then we get two different training models. In order to compare and select the optimal model, we set the weights using $w \in \{1, 10\}$ for training.

Another factor in the CNN model is the problem of learning rates (Wei, Xia, Huang, Ni, Dong, Zhao, & Yan, 2014). In the choice of learning rate, if the learning rate is too large, the correct rate will cause large fluctuations when the model converges, which will lead to lower precision. Conversely, when the learning rate is too small, it will directly affect the speed at which the model converges, which will lead to more steps in the model training process and may also cost our time on the model convergence. Assuming that the training time is m, learning rate is expressed as show like Eq. (4.1).

$$\lambda_k = \exp \left(\gamma max - \frac{n-1}{m-1} (\gamma max - \gamma min)\right)^{\ln 10} \tag{4.1}$$

For 7500 samples, we take a batch size of 50 and perform 100 epoch exercises. A total of 7250 units need to be trained. It takes about 31 hours to train a model on a single CPU and 4 models for a total of 16 hours. The training results are shown in the following Table 4.1.

Table 4.1 Results of the different models after training using CNN

model	Mean	train	vaild
box_1_w_01	0.9660	0.9638	0.9507
box_1_w_02	0.9631	0.9634	0.9453
box_2_w_01	0.9335	0.9480	0.8753
box_2_w_02	0.9248	0.9548	0.8447

From Table 4.1, we intuitively see that the accuracy of each model in the four training models is above 90%. From this viewpoint, all four models are fully trained during the training process. There is not overfitting in the training. But the accuracy is based on the verification set, we finally chose *box_1* as the quadrilateral code and weight 1.0 because the overall accuracy of this model is the highest among the four models.

In order to further compare the training process of the four models, we conducted a comparison of the four models from the loss function of the training dataset and the verification dataset, the correct rate, the loss function of the positioning and classification.

From Figure 4.3, we see that the four curve comparisons of Model 1.

- The subfigure at upper left corner of Figure 4.3 shows the loss functions of the training dataset and the verification dataset. When the loss function decreases, the value of the verification does not increase, which indicates that the model does not appear excessively overfitting in training (Sarle, 1996).

- The subfigure at upper right corner of Figure 4.3 shows the accuracy of the validation. We see that the model has gradually stabilized at the end, which shows that the model is fully trained.

- The subfigure at lower left corner of Figure 4.3 indicates the loss function values for classification and positioning. The gradient descent of the loss function of the classification is significantly faster than that of the lost function of the positioning, which indicates that the phenomenon of overfitting of the training process of the classification is lower than that of the positioning.

- The subfigure at lower right corner of Figure 4.3 compares the accuracy of classification and positioning. The accuracy of this classification is significantly higher than that of the positioning. This shows that positioning is more difficult for training than that of classification.
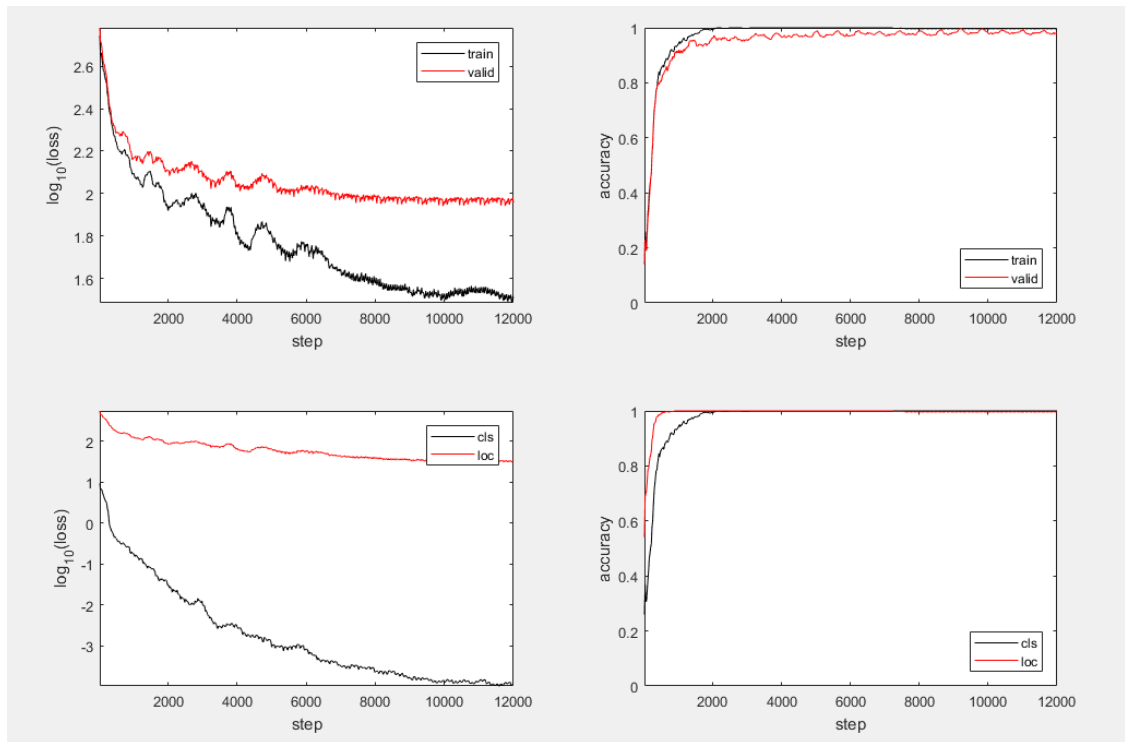
Figure 4.3 *Box*=1, *w*=01 training curve

From Figure 4.4, we see that the four curve comparison of Model 2.

- The comparison of the loss function and the verification shown in subfigure at the upper left corner of Figure 4.4 indicates that there is a small increase in the middle, but the whole trend is descent. The training dataset is directly proportional to the verification set, indicating that there is not overfitting.

- The accuracy rate of the training dataset and the verification dataset in the subfigure at the upper right corner of Figure 4.4 is gradually stabilized, indicating that the Model 2 is also fully trained.

- The comparison of the loss function values and classification is shown in the subfigure at the lower right corner of Figure 4.4, the curve of this classification is generally much faster than the positioning. In the lower right corner for the accuracy comparison, the classification accuracy rate is also faster than the positioning. These two figures show that Model 2 is as same as Model 1, and positioning is more difficult to be trained than that of classification.
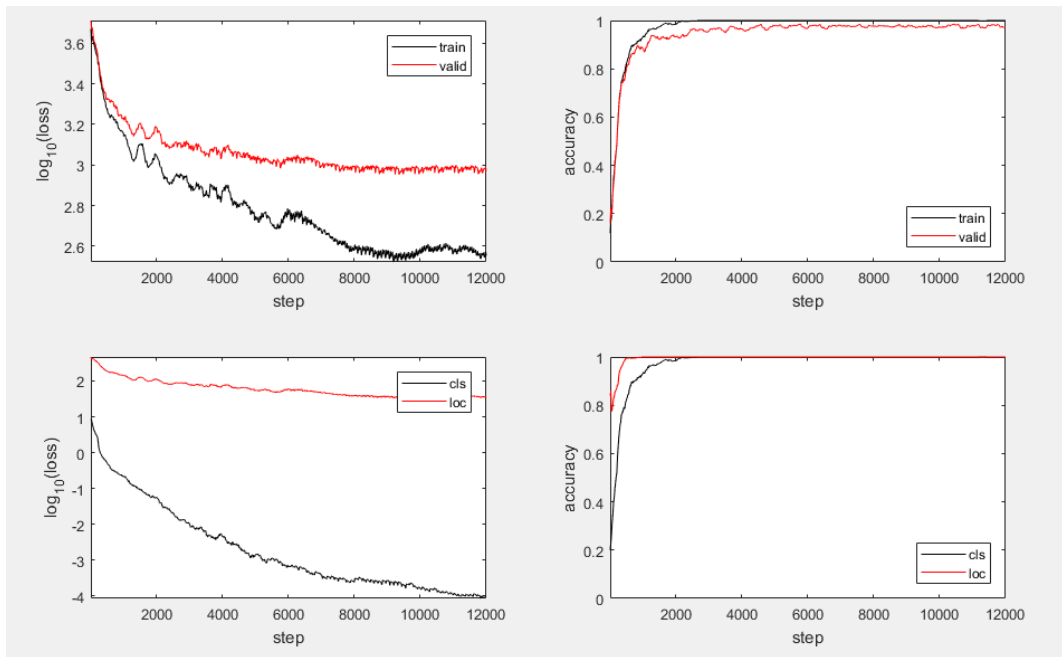
Figure 4.4 *Box*=1, *w*=10 training curve

From Figure 4.5, we see that the four curve comparison of Model 3.

- By comparing the loss function value curves, we find that both the training set and the verification set are descent as a whole, but the verification machine fluctuates greatly, indicating that there is overfitting in Model 3.

- In the upper right corner for the accuracy curve comparison, we see that the accuracy of the Model 3 is not gradually stabilized, the fluctuation is large, which shows that the model 3 is not fully trained.

- From the graph of this loss function in the subfigure at the lower left corner of Figure 4.5, we see that the loss function at the beginning of the partial positioning is lower than that of the classification, the descending speed of the classification is faster than that of positioning. It is difficult to train the classification of Model 3 at the beginning, but the latter part is still much difficult for the training of positioning.

- From the accuracy graph on the lower right corner, the accuracy of classification is faster than the positioning, indicating that positioning is more difficult to be trained.
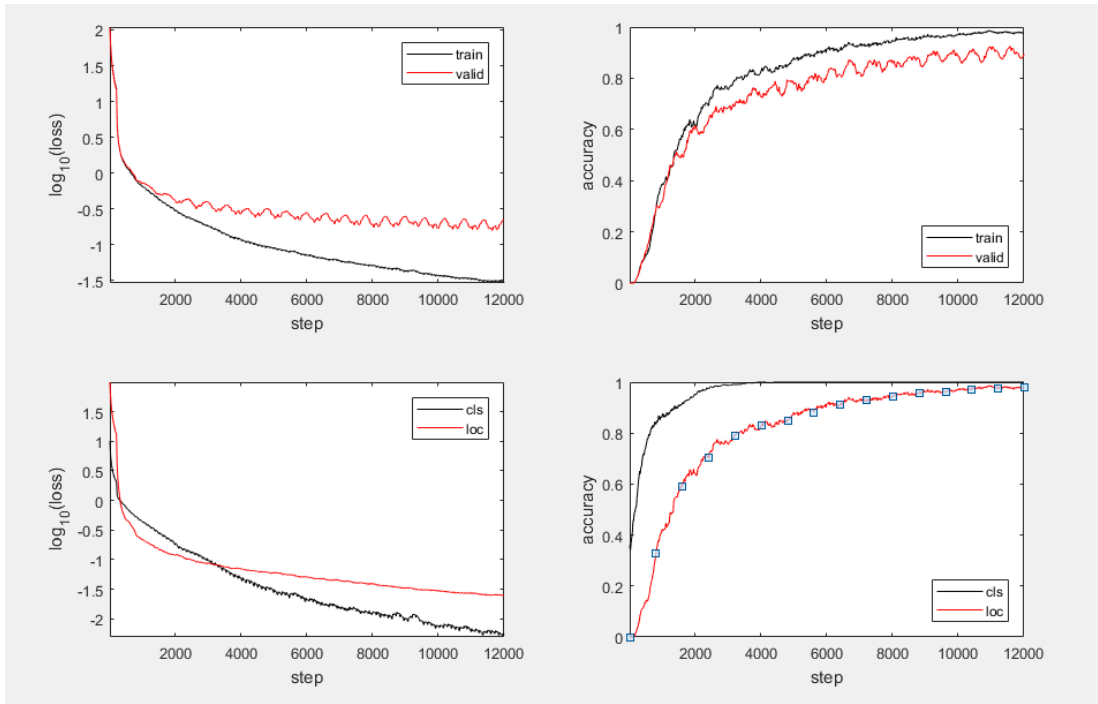
Figure 4.5 *Box*=2, *w*=01 training curve

From Figure 4.6, we see that the four comparisons of Model 4.

- The subfigure at the upper left corner of Figure 4.6 is a comparison of the loss function curves of the training set and the verification set. We see that the loss function of the training set decreases while the loss function of the verification set also descent, which indicates that the model 4 does not overfit during the training process.

- The subfigure at the upper right corner of Figure 4.6, shows the accuracy of the training dataset and the verification dataset. We see from the comparison that the drop is slower, indicating that Model 4 is not well trained in training.

- The subfigure at the lower left corner of Figure 4.5 shows the loss function of the location and classification on the training set of Model 4. From this figure, we see that the positioning loss function of Model 4 falls faster than the classification loss function. This shows that in Model 4, classification is more difficult than positioning, and the actual gap is larger.

- The subfigure at the lower right corner of Figure 4.5 shows the accuracy of the positioning and classification on the training set of Model 4. We see that the accuracy of the classification increases. The speed reading is faster than the positioning, and finally it is gradually stable. It shows that in the final accuracy, the positioning is still more difficult than the classification, but the final result is acceptable.
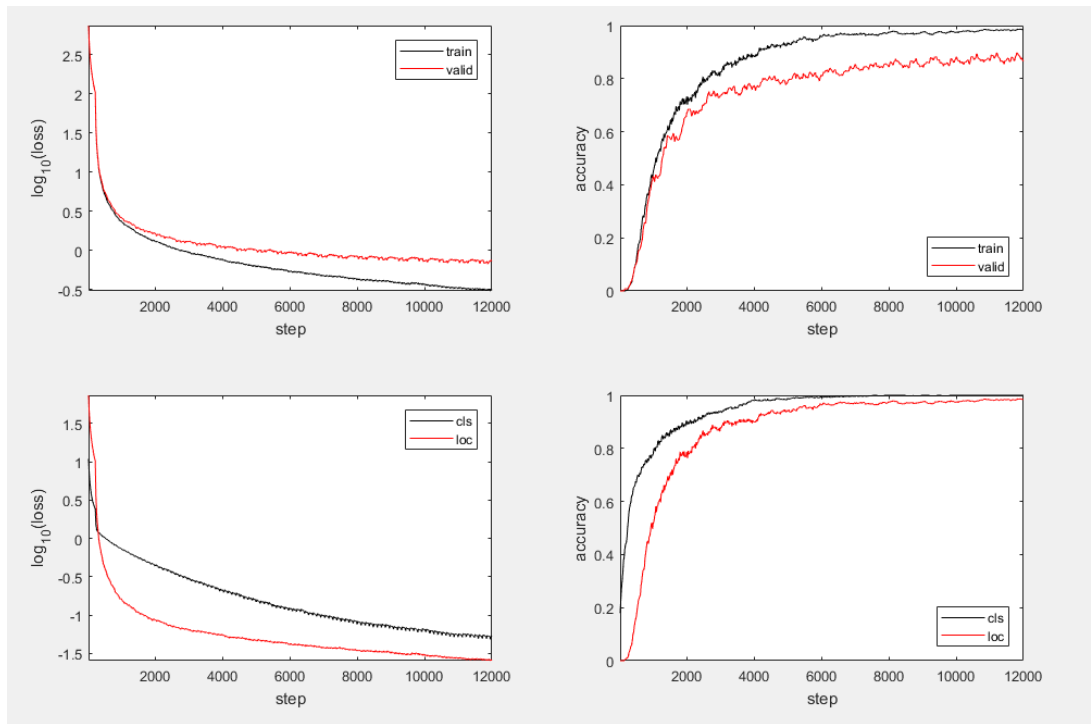
Figure 4.6 *Box*=2, *w*=10 training curve

Finally, we conducted a curve comparison chart for the four models for the accuracy of the verification dataset as shown in Figure 4.7. From the figure, we can see that the accuracy of the verification set of the model is improved when using the quadrilateral coding of *box*_1. Obviously it is faster than the encoding method of *box*_2, and the final accuracy of model of *box*_1 is higher than the model of *box*_2. We also found that the weight has little effect on the accuracy of the model.

Through a series of previous comparisons, we conclude that quadrilateral coding Model 1 has the highest accuracy and the fastest convergence during training. So we chose the most optimal model of the quadrilateral coding of Model 1.
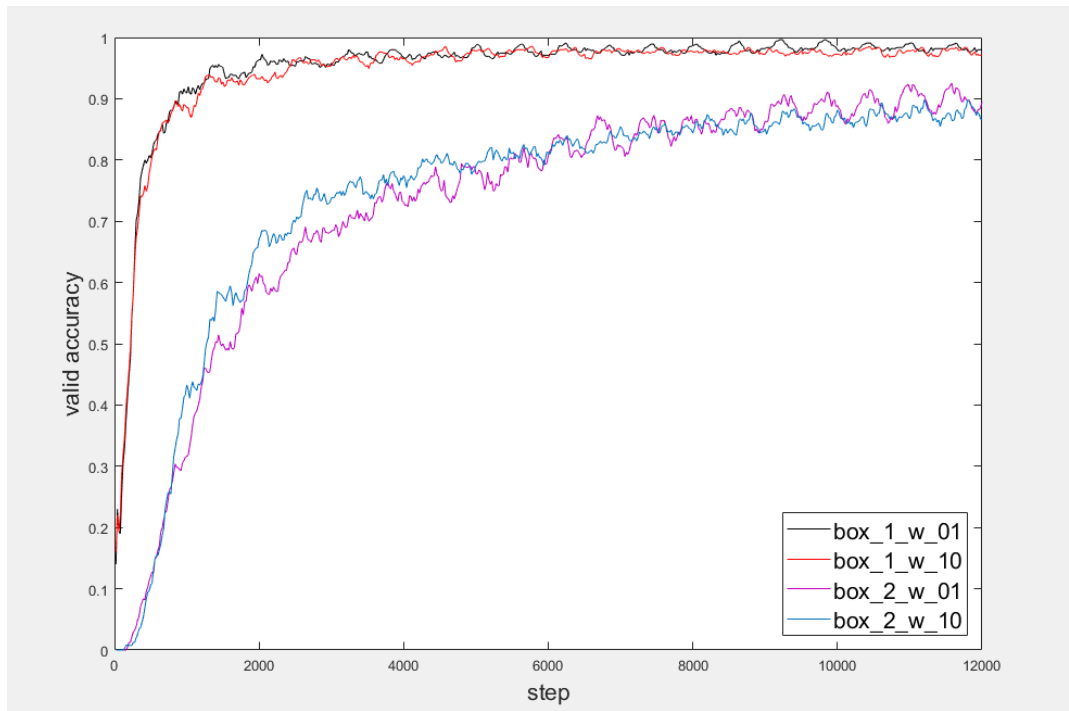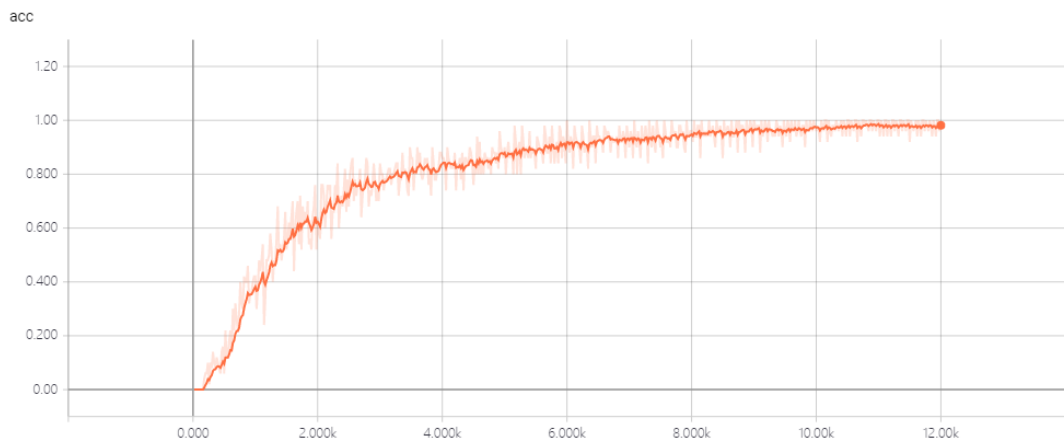
Figure 4.7 Valid accuracy

Figure 4.8 The accuracy of the model

In Figure 4.9, we clearly see that the value of the curve of this loss function is decreasing and eventually is almost zero. This indicates that there is no significant difference between the results obtained after training through the neural network and the results predicted in the verification layer of this model. This also shows that after our data has passed through a series of convolutional and MPL layers, the data has been fully trained, and the results meet our original research goal of currency recognition.
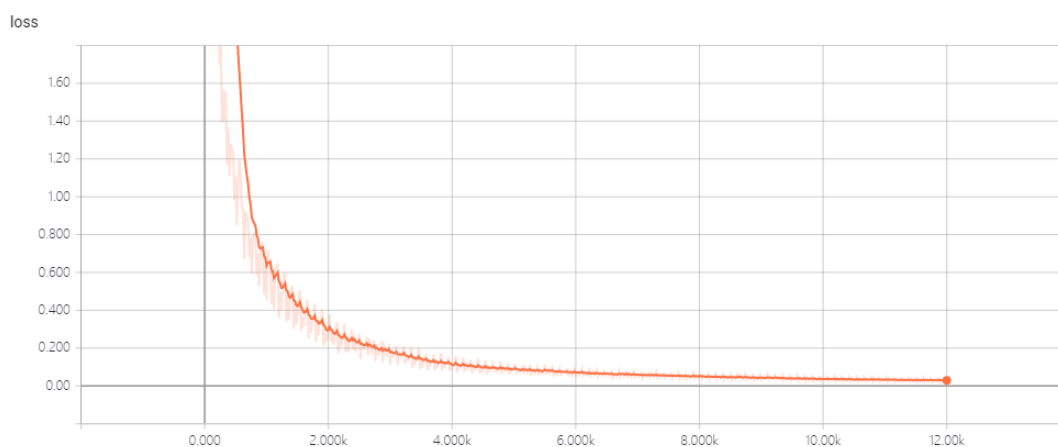
Figure 4.9 Loss function

We will use Model 1 as the final model so that we can use Model 1 to accomplish our experimental purposes. Now we will further analyze the results of the Model 1.

The data were indexed based on Model 1, using Model 1 to train 5 New Zealand dollars, 10 New Zealand dollars and 20 New Zealand dollars, the training results were specifically analyzed below. First, the overall result can be divided into three parts:

(1) The square marker box is used to mark the exact border of the currency.

(2) The enclosed rectangle of the quad is used to mark the area occupied by the currency in the image.

(3) In order to better distinguish the front and back of the currency, we chose to use different color recognition boxes to indicate the front and back of different denominations.

Specific to the currency of different denominations, there are still problems such as front and back, distance and accuracy, thus the recognition results can be analyzed according to different denominations.

In our study of currency recognition, the classification of currency is also an important part of it. Accurate classification of currency can be expressed in the final experimental results. In this thesis, we group the currency into six categories, the front and back side of the 5NZD, the front and back side of the 10NZD, and the front and back side of the 20NZD. When the currency appears in one shot, the model can automatically detect the category of the currency and display the specific class in the sort box. This part of the content is not only a high requirement for classification, but also a high requirement for currency positioning. Accurate monetary positioning can better classify and identify, there is an interactive relationship between them.

The first is to analyze the recognition results of 5 New Zealand dollar as shown in Figure 4.10. From the viewpoint of frame colors, the pink frames are used to indicate the front side and the blue recognition frame is used to mark the back side, so that the result after currency recognition could be more clearly seen. From the classification label, "5 NZD" indicates that the currency has a face value of 5 New Zealand dollar, 'F' and 'B' indicate the front and back, and '1.00' indicates that the accuracy is recognized.

Figure 4.10 The result of 5 New Zealand dollar

The second is to analyze the recognition of 10 New Zealand dollar form Figure 4.11. From the viewpoint of recognition box, the yellow recognition frame is selected on the front side and the red recognition frame is selected on the back side, so that the result after currency recognition could be more clearly seen. From the classification label, "10 NZD" indicates that the currency has a face value of 10 New Zealand dollar, 'F' and 'B' indicate the front and back, and '1.00' indicates that the accuracy is recognized.

Figure 4.11 The result of 10 New Zealand dollar

Finally, we analyze the recognition of 20 New Zealand dollar from Figure 4.12. From the viewpoint of recognition box, the front side was chosen to be marked using the blue-green color frame, and the back side has been marked by using the green recognition box, so that the result after currency recognition could be more clearly seen. From the classification label, "20 NZD" means that the currency has a denomination of 20 New Zealand dollar, 'F' and 'B' indicate the front and back, and '1.00' shows the accuracy of recognition. One of the backs has an accuracy of '0.99', which indicates the process of currency recognition. There are some differences in this demonstration, which is what our experiment hopes to achieve, because there will be room for improvement.

Figure 4.12 The result of 20 New Zealand dollar

After analyzing the results, we see that the operation of the convolutional neural network is mainly to deal with the target object in different ways. From the results, we see that all the recognition frames and detection frames are very accurate. The position and extent of the currency, regardless of the brightness of the background light and the number of background objects, will not affect the accuracy of the model for currency recognition.

4.4 Limitations of the experiment

For currency recognition, using CNN as the feature extractor, SSD as the basic model of the overall framework, deep neural networks can be implemented for detection and recognition of currency; but there are still some limitations. These limitations will have a slight effect on the results of our currency recognition, we need to improve them in future research.

(1)   Since the currency chosen by our experiment is limited, we only select three denominations. We also need to consider more rich currencies with different denominations.

(2)   Currency recognition should not be limited to New Zealand dollars. The identification methods that currency recognition needs in different countries are also what we need to study.

(3)   At present, currency recognition we are studying is the information obtained under

the static lens. In future, we need to quickly capture the money under the moving lens, and simultaneously detect and recognition it.

(4)   In our research project, we mainly use CNN for training. In order to improve the robustness of the research, we choose to use more different methods to conduct experiments in future, and thus obtain a more stable research model.

4.5 Summary

In this chapter, we will detail the entire process of data collection and data tagging. In order to better select the optimal model, we use two different quadrilateral coding methods, which took weights of 1 and 10 respectively, so that four models are obtained, and four models are introduced to compare the four aspects. The model that best fits our requirements. At the same time, the results of currency recognition of the three denominations of the currency after training are also analyzed. In next chapter, we will continue to analyze the results and compare them to the other two model methods.

Chapter 5

Analysis and Discussions

In this chapter, experimental results will be clearly analyzed and discussed. This chapter will compare all methods used for currency recognition, discuss the best method of recognition, and illustrate its importance by analyzing the results.

5.1 Introduction

After completing the neural network training for currency recognition, we got satisfactory results. In addition to these results, we also need to analyze the model used. CNN is used to extract feature maps; after the convolution operation, the output is sent to complete the classification, this step can well locate the currency in the map and help the currency to be classified in the display.

In addition to analyze the model and experimental results, we also need the ratio of currency and image area. The currency itself is displayed in a complete image, then the other interference parts will definitely affect the completion of subsequent recognition work, so we confirm that the relationship between the currency and the whole image will affect the recognition effect.

We have chosen two other models for currency recognition. After completing the training of the dataset, we analyze the algorithms and results of these two models. By analyzing their structures, we find the difference between these two models. Comparative analysis shows that the accuracy of currency recognition is the highest one.

5.2 Analysis of CNN Model

From Chapter 4, we already see that our data has been fully trained under the CNN model. Using the CNN model will reduce the occurrence of overfitting. In the loss function, we see that overfitting has been successfully avoided. In the final comparison of the four models, we chose to use the first model with a weight of 1.0.

From the final results, we see that the rectangular boxes, bounding boxes, and classification labels meet our initial requirements, the accuracy also matches our needs. Before we reach the final conclusion, we consider three factors that will affect the correct rate of the model:

(1) *The area ratio of the currency*: the area of the enclosed rectangles where the currency occupies the proportion of the entire screen. The larger the ratio, the clearer the currency. Conversely, the smaller ratio, the more blurred currency.

(2) *Long-side ratio*: set the length of the two long sides of the currency in the image to w_1 and w_2, the long-side ratio is $L_w = \max(w_1/w_2, w_2/w_1)$. The closer this value is, the more the currency tends to the image plane; otherwise, higher degree of tilt of the currency reflects the degree of tilt of the currency in the longitudinal direction.

(3) *Short-side ratio*: set the length of two short sides of the currency in the image as h_1 and h_2, then the short side ratio is $L_h = \max(h_1/h_2, h_2/h_1)$. This value is similar to the long side ratio and reflects the lateral tilt of the currency.

Because the correct rate only describes the sample ratio of *IoU*>0.5, in order to accurately reflect the accuracy of the model, we will also compare the relationship between these factors and the average *IoU*. The closer the *IoU* is, the more accurate the positioning will be.
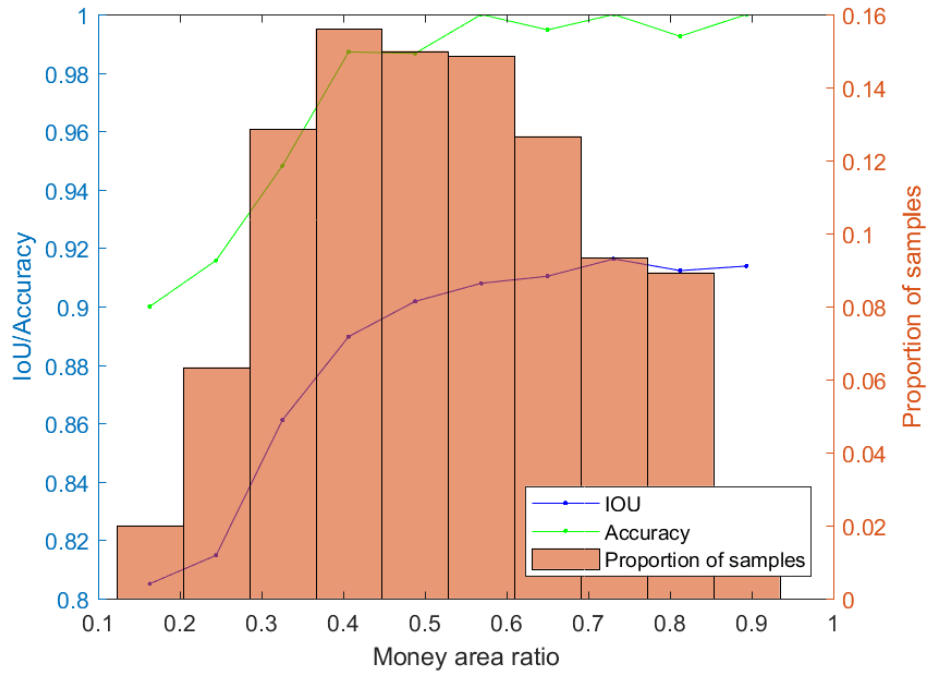
Figure 5.1 Money area ratio

As shown in Figure 5.1, the area ratio of the samples we generated is basically normal distribution. As the area ratio increases, the correct rate and IoU also increase. The larger the sample area is, the clearer the currency image will be; the model is much easier to identify and locate. This shows that when the currency is identified, the clarity of the sample, the position information and the degree of tilt will directly affect the final recognition accuracy.
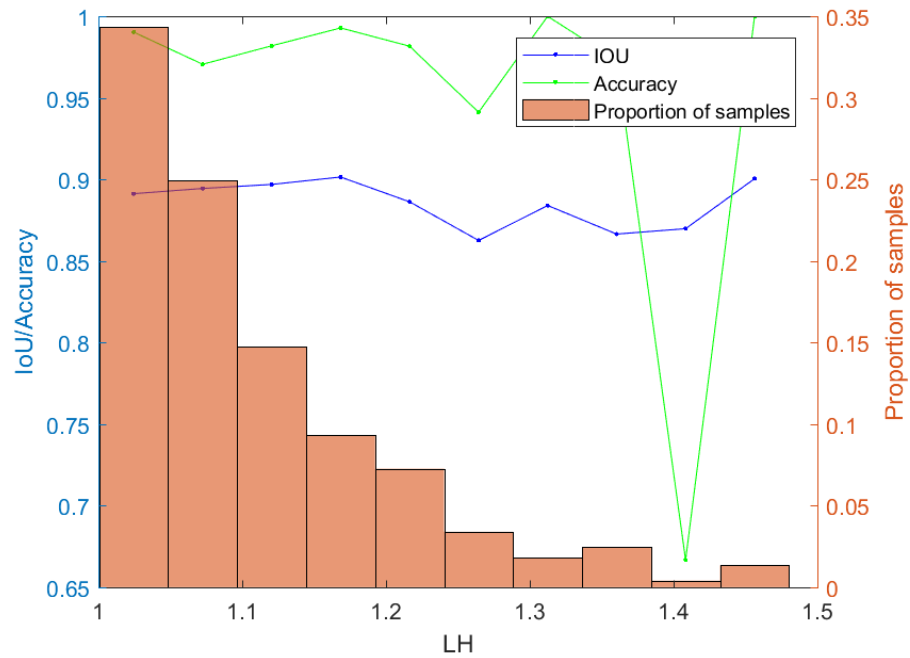

Figure 5.2 Line high

In Figure 5.2, the relationships among the ratio, IoU and the correct rates are similar to the long side ratio. The difference is that the change in the short side ratio has less effect on the IoU and the correct rate. This is because the short side ratio reflects the degree of inclination in the long side direction; under the same ratio, the short side ratio causes less loss of currency information.
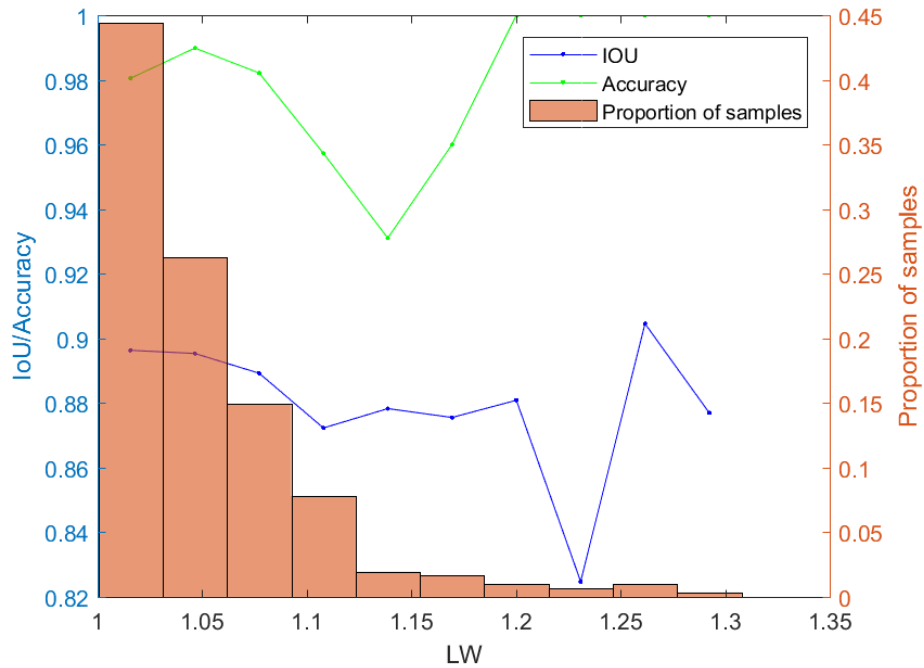

Figure 5.3 Line wide

From Figure 5.3, the closer the long side ratio is, the more samples there are. The correct rate and IoU increase first and then fluctuate with the increase of the long-side ratio. Subsequent fluctuations are caused by insufficient sample size. Therefore, on the whole, the larger the long side ratio, the higher the model correct rate and IoU. That means, the more parallel the banknote is, the higher the recognition accuracy.

Therefore, in general, the clearer the banknote is, the higher the accuracy of the model recognition.

5.3 Analysis of Another Models

We have learned that in the process of research, we mainly used the CNN model based on the SSD framework. In order to ensure the rightness of our experiments and to find the most suitable method for currency recognition, we also select it. The other models are tested. The next section will introduce the experimental results of the remaining models; at the end of the resultant comparison, we select the most suitable model.

5.3.1 Analysis of MobileNet Model

MobileNetV1 is a model that reduces the network size and speed up the reasoning. It is a model based on a streamlined structure that uses a deep separable convolution to construct a light weight deep neural network. We also tried to use this method for currency recognition. We know that we divided the data into four different models based on the encoding of the quadrilateral; in MobileNet, we choose to use these four models for training. It decomposes the standard convolution into deep convolutions, only uses a single filter for filtering and a point-by-point convolution to apply a 1×1 convolution operation so as to combine the output of all deep convolutions.

Table 5.1 shows that we train the four models using MobileNet, which are organized in three aspects: accuracy, training and verification results.

Table 5.1 Results of the different models after training using MobileNet

model	mean	Train	vaild
box_1_w_01	0.9543	0.944	0.9410
box_1_w_02	0.947	0.9431	0.9262
box_2_w_01	0.8654	0.8723	0.8531
box_2_w_02	0.8885	0.8343	0.821

From Table 5.1, we see that for Model 1 and Model 2, the accuracy is still promising, because MobileNet will decompose large models into small models and its accuracy is very high, but the decomposition model takes a certain amount of time, which will lead us to spend more time in the training process and make the research process more complicated, this method is not suitable for our research.

5.3.2 Analysis of Faster R-CNN Model

We also chose to use Faster R-CNN (Zhang, Lin, Liang, & He, 2016) as another algorithm. For Faster R-CNN, it sees from Figure 5.4 that it has a more RPN step in order to better detect and recognition of the object. For our currency recognition, it splits the identified currency into multiple 3×3 recognition boxes and enters the proposals layer; it is predicted that the position where the target may appear in the figure is marked for better RoI; this step is performed simultaneously with the formation of feature maps and enters RoI pooling, adjusts the recognition frame to a fixed size, extracts the features of the RoI, and finally enters the classification and positioning part.
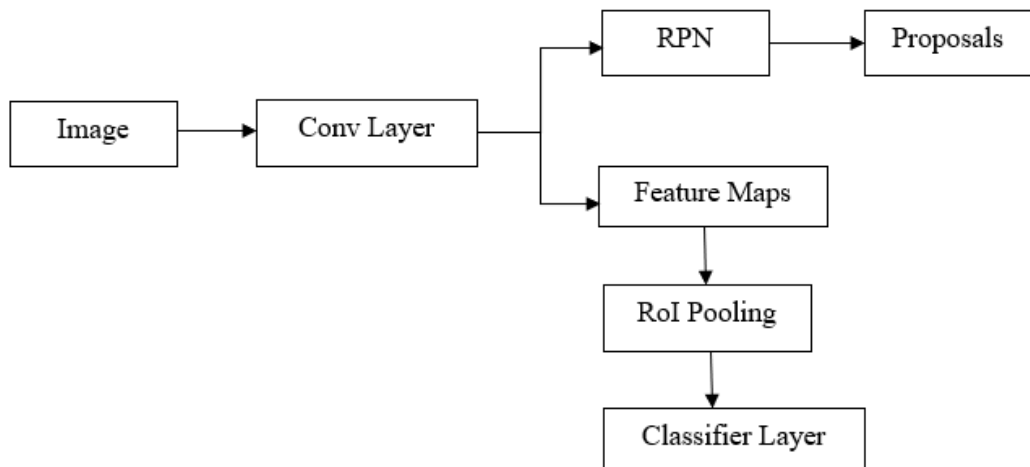
Figure 5.4 Flowchart of Faster R-CNN

From Table 5.2, we clearly see the results of using the Faster R-CNN to train the four models. Model 1 and Model 2 are still satisfactory in training, because the two models are more adequately trained and fully trained during the training process to avoid overfitting.

The training results of Model 1 and Model 2 are similar, but from Model 3 and Model 4, we clearly see that the accuracy of training has decreased. This indicates that there are insufficient training problems in these two models, which makes it overfitting. This will be reflected in the final accuracy.

Table 5.2 Results of the different models after training using Faster R-CNN

model	mean	train	vaild
box_1_w_01	0.9587	0.9542	0.9497
box_1_w_02	0.9534	0.9445	0.9437
box_2_w_01	0.9338	0.8827	0.8683
box_2_w_02	0.8935	0.8963	0.8773

5.3.3 Analysis of Three Different Models

We know that when the recognition task is completed, the most important algorithm is to see how to mark the target. The setting of the bounding box is different, which can directly affect the final recognition effect. The deep learning algorithm is to perform feature extraction on the target after multilayer convolution operation. This is also related to the setting of the number of convolution layers, when setting the number of convolution

layers, we also need to configure the network according to the hardware, environment needs, and the data size. Because different algorithms have different settings for the bounding box, the final result will be different. The accuracy is based on the area ratio of the overlapping between the bounding box and the original image. We can make appropriate adjustment to the bounding box based on the final accuracy and make the appropriate adjustments to more accurately detect the position of the target object.

For target recognition and detection tasks, where the bounding box is a critical step in performing a series of tasks, the bounding box setting is the fundamental operation before the convolution. The bounding box marks the detection target in the target object, and the RPN will give the target object's offer in the input image, but this proposal is not completely correct, there will be some overlapping between the original ground truth box and the original ground truth box. In order to improve the accuracy of positioning, we will adjust the RPN to be closer to the ground truth box.

In order to better compare these three methods, we chose to use the same box setting method, in which the method of setting box_1 is the same, but the weights are $w=1$ and $w=10$, respectively. The setting method of box_2 is the same, and the weights are also $w=1$ and $w=10$, respectively so that four models are obtained. Using the same model, we can clearly compare the different methods and visually see the difference in results between the different methods.

(1) Mobilenet

MobileNet is an operation based on convolutional neural networks for the detachable convolution of priests. The convolutional neural network we usually use is a standard convolution, MobileNet (Zhu, & Gupta, 2017) integrates the standard volume into depthwise convolution and pointwise convolution. This is done to reduce the number of parameters. When we are conducting currency recognition, we classify the data through these two convolution methods to simplify the calculation process.

As shown in Figure 5.5, it is related to MobileNet's depthwise separable convolution decomposition. We need to enter a feature map **F** at the beginning and set its size to (D_F, D_F, M). In order to distinguish it from the standard convolution, the known standard convolution is K, and the size is (D_K, D_K, M, N) as shown in Figure 5.5(a), then the final output feature map value is G and the size is (D_G, D_G, N), the well-known standard convolution is shown as Eq. (5.1).

$$G_{k,l,n} = \sum_{i,j,m} K_{i,j,m,n} \cdot F_{k+i-1,l+j-1,m}$$

<div align="right">(5.1)</div>

Figure 5.5(b) shows that the standard convolution is split into depthwise convolution, the deep convolution is responsible for filtering, and the size of the currency is reduced to $(D_k, D_k, 1, M)$, then the output size after depthwise convolution is (D_G, D_G, M). Figure 5.5(c) shows another part of the standard convolution split, which is a pointwise convolution and responsible for the conversion to different channels in the currency recognition process with dimensions $(1, 1, M, N)$. Then, the output size after pointwise convolution is (D_G, D_G, N).

The formula for the known depthwise convolution is shown as Eq. (5.2).

$$\hat{G}_{k,l,n} = \sum_{i,j} \hat{K}_{i,j,m} \cdot F_{k+i-1,l+j-1,m}$$

<div align="right">(5.2)</div>

where $\hat{K}$ is the depthwise convolution and the convolution kernel is $(D_K, D_K, 1, M)$. When the currency recognition proceeds to the convolution kernel, the application should be on the m_{th} channel in F and the depthwise convolution will be output on the m_{th} channel on $\hat{G}$.

Through the analysis of the depthwise convolution diagram of MobileNet and the calculation formula of depthwise convolution, we see that the calculation cost of MobileNet is less than the standard convolution and the required parameters are less than the standard convolution.
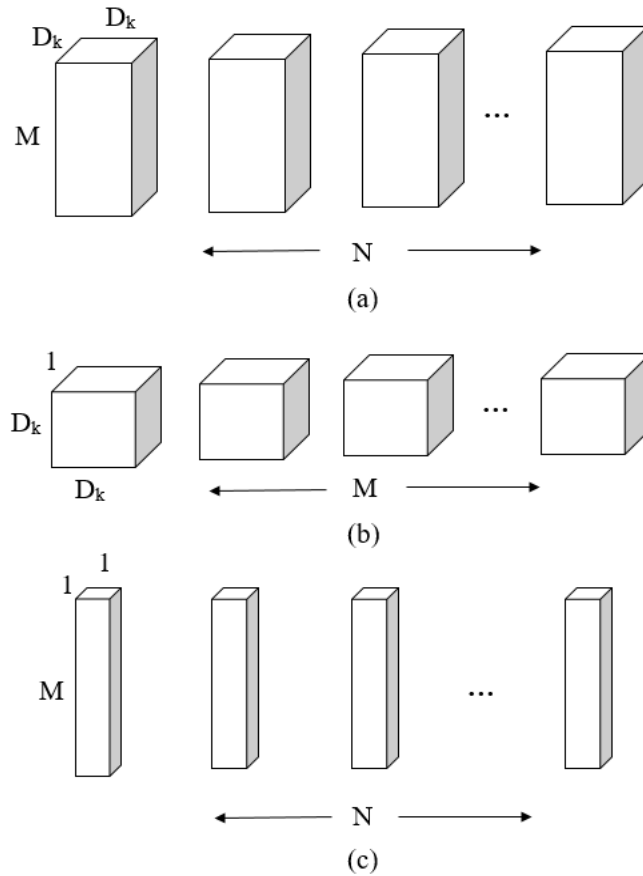
Figure 5.5 Depthwise Separable Convolution Decomposition Diagram

The basic calculations for depthwise convolution and pointwise convolution are shown in Eq. (5.3).

$$D_K \cdot D_K \cdot M \cdot D_F \cdot D_F + M \cdot N \cdot D_F \cdot D_F$$

(5.3)

After completed the deep separable convolution operation, the amount of calculation costs is reduced to Eq. (5.4).

$$\frac{D_K \cdot D_K \cdot M \cdot D_F \cdot D_F + M \cdot N \cdot D_F \cdot D_F}{D_K \cdot D_K \cdot M \cdot N \cdot D_F \cdot D_F} = \frac{1}{N} + \frac{1}{D_K^2}$$

(5.4)

For MobileNet, the overall structure is to add a batchnorm and a ReLU nonlinear activation function (Cybenko,1989) after all layers. For the last layer of the structure, no nonlinear activation function is required (Chen, & Chen, 1995), the SoftMax layer performs target classification directly.

(2) Faster R-CNN

Faster R-CNN is usually suitable for large datasets. It will first select a model as a feature extractor, and then move on to the next layer. The Region Propose Network (RPN) is used to process the image extracted by the feature. The main role is to find the bounding box of possible objects. The biggest improvement of Faster R-CNN is to abandon the

traditional method of generating detection frames, and instead use RPN as a method to detect bounding boxes.

When using Faster R-CNN for currency recognition, there are mainly four parts of the operation. The first part is the convolution layer. This part is as same as CNN's target detection method. The convolution layer is used to extract the features in the currency image to form feature maps, and the feature maps will be shared with the RPN layer and the fully connected layer.

The second part is the Region Proposal Networks (RPN) layer (Cai, Fan, Feris, & Vasconcelos, 2016), which is used to propose regional proposals for currency, distinguish the boundary box of the target currency and background, and then use the bounding box regression to correct the previously proposed currency anchor to obtain accurate proposals. The third part is the RoI Pooling layer. This part mainly integrates the currency feature maps and proposals obtained by using the first two layers. This result is input to the fully connected layer and used to determine the target classification. The last layer is the classification layer. The results of the previous RoI Pooling are collated (Dai, He, & Sun, 2016), and then the specific category of the currency is calculated, the detected currency recognition box is corrected again using bounding box regression to obtain the prepared currency position.

In Faster R-CNN, the most important part is the addition of the RPN layer. The advantage of using RPN is that it can budget the bounding box of the background and target. At the time of currency identification, because the currency is in a disturbing environment, when using RPN for feature extraction processing, the background matter is separated to calculate the foreground Softmax probability. The specific RPN running process is shown in Figure 5.6.
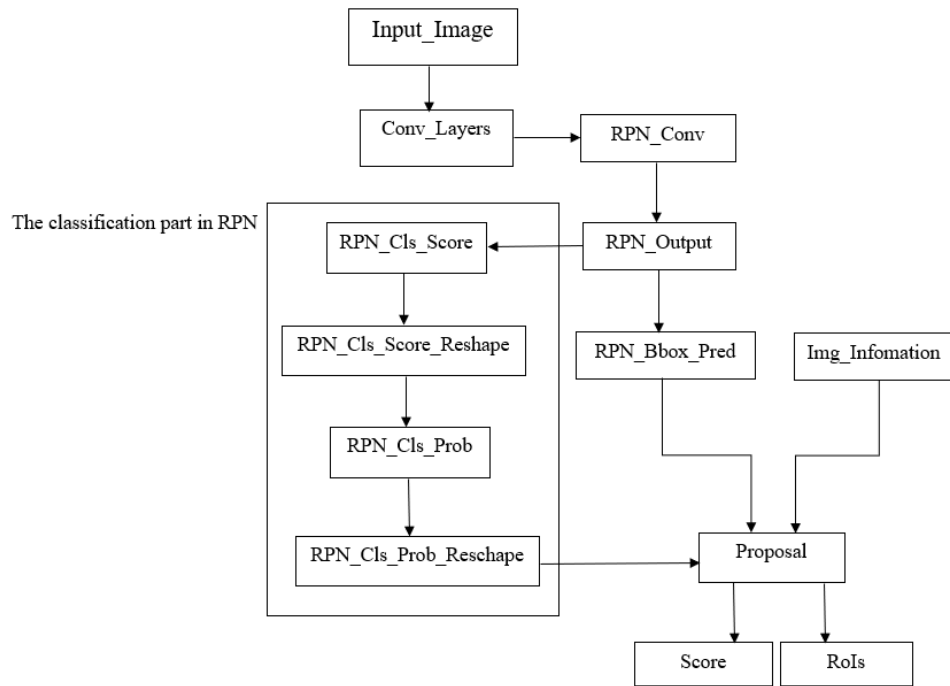
Figure 5.6 The structure of RPN layer

After the analysis of these three algorithms, the main content of this part is to compare which model is better and more accurate in the issue of currency recognition between three different models. This is why we choose to use the same four quadrangle encoding methods to apply to the three models, so that we see the difference more intuitively. We integrated the accuracy of the three models together and formed Figure 5.7 based on the three datasets. From the statistics, the accuracy of CNN in mean, training and validation is the most consistent with our requirements. On the whole, the accuracy of these three models is satisfactory. When *box_2* takes *w*=10, the accuracy of these three models is significantly decreased, which indicates that there is excessive fitting in this dataset, and such results could not meet our requirements.

From the partial analysis of the mean, for box_1, whether it is 1 or 10 is the accuracy of CNN, but the difference between the three models is not large; when the weight equals to 1, the accuracy of the models is outside 95%. For *box_2*, there is a clear gap between the three models. When the weight value is 1, the accuracy of CNN and Faster R-CNN is similar, but the accuracy of MobileNet is lower than 90%; when the weight is heavy at 10, the accuracy using CNN remained above 90%, but MobileNet and Faster R-CNN were less than 90% accurate.

From the partial analysis of train, for *box_1*, when the weight is 1, only MobileNet drops below 95%, which shows that though the training effect using MobileNet model is

74

very high, compared with the other two models, There is still a problem of insufficient training; when the weight is 10, for CNN, the training effect is better than when the weight is 1, but for MobileNet and Faster R-CNN, the training effect is similar, but both are below 95%. For *box_2*, when the weight is 1, CNN training results reach more than 90%, while MobileNet and Faster R-CNN are both below 90%, which indicates that the training of these two models is not enough and may exist.

When the weight is 10, CNN training results reach more than 95%, this indicates that CNN is fully trained. In the process of training, there is no overfitting, and MobileNet is lower than 85%, that indicates training MobileNet on this model is not enough, and there is overfitting, relative R-CNN training is more adequate than MobileNet.

From the partial analysis of valid, for *box_1*, when the weight is 1.0, the accuracy of verification set of CNN and that of Faster R-CNN are similar, this indicates that the two models are fully trained in the verification set. The accuracy rate is very high; when the weight is 10, the accuracy of the three models decreases, but the gap between CNN and Faster R-CNN keeps small.

For MobileNet, there are problems with insufficient training. For *box_2*, as a whole, the three models are not enough for training regardless of the weight of 10; when the weight is 1.0, the accuracy of CNN is still the highest one among the three models; when the weight is 10, Faster R-CNN is more than 85% accurate, while those of CNN and MobileNet are less than 85%.
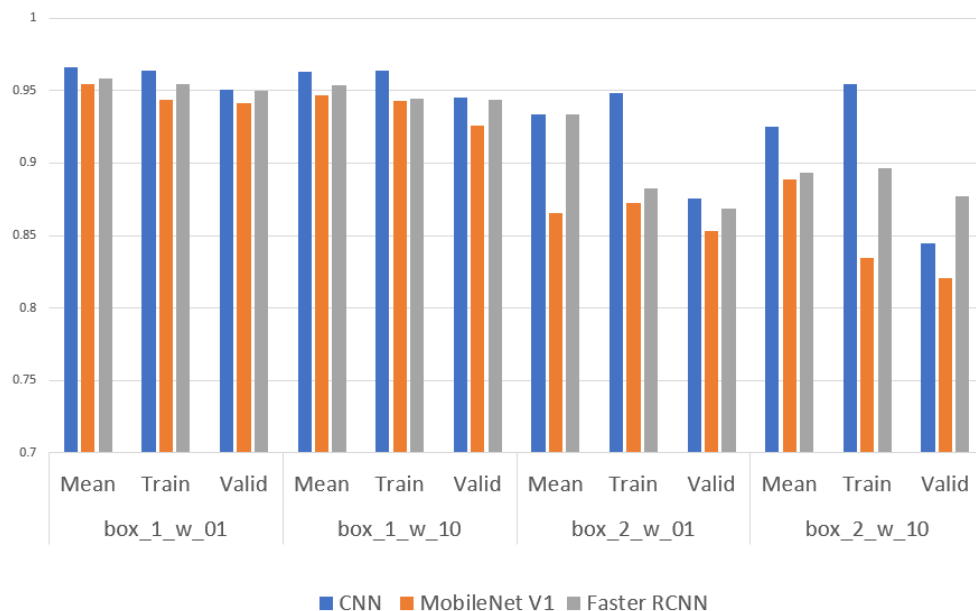

Figure 5.7 Statistical comparison of accuracy of the three models

After the analysis of these three models, Table 5.3 is the specific result of the three models for currency recognition, the specific accuracy of these three models under the four rectangular models can be visually seen through the table.

For the three models, the weight of box_1 is the highest when the weight is 1.0, the CNN reaches 96.9%, the accuracy of MobileNet is 95.4%, and the accuracy of Faster R-CNN is 95.8%. First of all, it is explained that when the weight is 1.0, the model of *box_1* is the best one of the four models which is the one with the highest accuracy for these three algorithm models. At the same time, we see that the accuracy of CNN under four models is relatively stable and the accuracy is satisfactory. The difference between the recognition accuracy of MobileNet is relatively low. The accuracy of *box_2* is lower, and the accuracy of Faster R-CNN. It is also relatively stable, except that when the weight is 10, the accuracy of *box_2* is lower, and the accuracy of the other three models is more than 90%.

For the training, overall accuracy of *box_1* is stable in the three models, and the accuracy is also high, this indicates that the data of *box_1* is fully trained regardless of the weight of 1.0 or 10. When the weight is 1.0, the training accuracy of these three boxes more than 95% for CNN and Faster R-CNN, and the accuracy rate of MobileNet is also 94.4%. But when we started training *box_2*, we see that both the Faster R-CNN and MobileNet dropped below 90% accuracy of training set, which shows that the box_2 model is not adequately for the two algorithm models.

For validation, the higher the accuracy, the more adequate the training dataset, and the lower the overfitting during training. It is clear that the three models are the most adequately trained for CNN. For *box_1*, CNN avoids overfitting during training, so the accuracy of valid is higher; but for MobileNet and Faster R-CNN, the accuracy of *box_1* is higher than that of *box_2*, which indicates that during the training process of *box_1*, the dataset is trained as much as possible, and there is overfitting in the training process, but it is not serious. However, for *box_2*, regardless of whether the weight value is 1 or 10, there is a problem of insufficient training, and there is also an overfitting in the training process, which affects the final accuracy.

Table 5.3 Comparison of the results of the three methods

Models	Types	CNN	MobileNet	Faster RCNN
box_1_w_01	Mean	0.9660	0.9543	0.9587
	Train	0.9638	0.9440	0.9542
	Valid	0.9507	0.941	0.9497
box_1_w_10	Mean	0.9631	0.947	0.9534
	Train	0.9634	0.9431	0.9445
	Valid	0.9453	0.9262	0.9437
box_2_w_01	Mean	0.9335	0.8654	0.9338
	Train	0.9480	0.8723	0.8827
	Valid	0.8753	0.8531	0.8683
box_2_w_10	Mean	0.9248	0.8885	0.8935
	Train	0.9548	0.8343	0.8963
	Valid	0.8447	0.821	0.8773

It is worth mentioning that the biggest difference between the Faster R-CNN and the SSD model is that SSD is only used, and then the MLP is used to classify and locate the object. At the beginning, Faster R-CNN is shown in Figure 5.8, it will perform weak classification and rough positioning of the object in the first stage. The highest feature of this stage is to cut the result according to the recommendation of RPN and input it into a new CNN for new features.
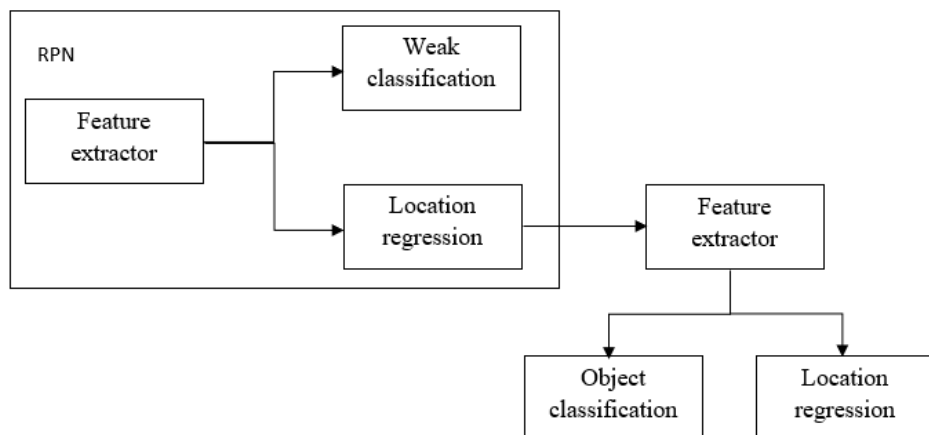

Figure 5.8 Faster R-CNN model

5.4 Summary

The entire fifth chapter is splitted into three parts. The first part analyzes the three parts that may affect the accuracy when using the CNN model: the ratio of the currency to the whole picture, the long side ratio to that the short side. In comparison, the statistical diagrams were analyzed for these three points, and the conclusion was that the higher the accuracy, the more parallel the currency is in the whole picture, the larger the area ratio. The second part is to use the different algorithms of MobileNet and Faster R-CNN for currency recognition and draw conclusions. The third part is to analyze and compare three different algorithms. It is found that when performing currency recognition, the accuracy of CNN is still higher and more stable. Faster R-CNN could also meet the demand at the certain time, while MobileNet needs further research.

Chapter 6

Conclusion and

Future Work

Throughout the description in this thesis, we have investigated the methods for currency recognition. The optimal method was compared with others, we met the research purpose and answered the research questions which were asked at the beginning of this thesis. In this chapter, we will summarize the contents of the full text;in the end, we will introduce the work that we will continue to study in future.

6.1 Conclusion

The main purpose of this thesis is to carry out currency recognition. In fact, it includes the denomination of currency. We trained the SSD framework, tested four different models and finally selected the best one. These models are based on empirical methods. We have also received satisfactory results. After completing currency recognition, we also summarized the main contributions of this thesis.

We chose to create a 6-layer CNN model. We select to use quadrilateral $box_1 = [x_1, y_1, x_2, y_2, x_3, y_3, x_4, y_4]$ and set the initial weights to 1.0 for our currency recognition training. Finally, the trained model could reach 96.6% accuracy, which shows that our dataset has been fully trained. From the loss function, we see that our model does not have overfitting during the training process.

The final research results are satisfactory, including determining the currency range in the classification label, currency denomination, currency front and back; currency recognition accuracy could be expressed, the accuracy is very high.

After the analysis, we found that when the currency is in a clear state on the entire screen and the angles are parallel, our recognition is fast, and the precision is high. When the currency moves at a very obvious angle or appears on the screen far away from the camera, the accuracy of currency recognition will decrease slightly, but because the dataset is fully trained, the experiments of currency recognition can still be conducted well.

In the final part of this thesis, we chose to use the MobileNet model and Faster R-CNN for currency recognition. The purpose of this is mainly to study the accuracy of currency recognition in other models, analyze and compare the three models. Experiments show that the proposed method of using the SSD model is the most accurate one for currency recognition, and its practicality is also the best.

6.2 Future Work

Our future work includes,

(1) In future, we will select the currency of several different countries for recognition and identify the country name and other information through the recognition results.

(2) For currency recognition, we can also recognize the serial number of currency, patterns on the currency surface and complete the currency recognition.

(3) We can also consider that currency recognition with distance and background problems. In the context of different distances or more interfering objects, we identify currency denominations using deep learning.

(4) The model we use can also be extended to ResNet-101, the Inception_V2 model and other deep learning models.

References

Abdel-Hamid, O., Mohamed, A. R., Jiang, H., & Penn, G. (2012). Applying convolutional neural networks concepts to hybrid NN-HMM model for speech recognition. In IEEE International Conference On Acoustics, Speech and Signal Processing (ICASSP) (pp. 4277-4280). IEEE.

Angin, O., Campbell, A. T., Kounavis, M. E., & Liao, R. F. (1998). The Mobiware toolkit:Programmable support for adaptive mobile networking. IEEE Personal Communications, 5(4), 32-43.

Arbelaez, P., Maire, M., Fowlkes, C., & Malik, J. (2011). Contour detection and hierarchical image segmentation. IEEE transactions on pattern analysis and machine intelligence, 33(5), 898-916.

Barequet, G., & Har-Peled, S. (2001). Efficiently approximating the minimum-volume bounding box of a point set in three dimensions. J. Algorithms, 38(1), 91-109.

Basu, A. P., & Ebrahimi, N. (1991). Bayesian approach to life testing and reliability estimation using asymmetric loss function. Journal of statistical planning and inference, 29(1-2), 21-31.

Bengio, Y. (2009). Learning deep architectures for AI. Foundations and trends in Machine Learning, 2(1), 1-127.

Bharkad, A. A. S. S. (2013). Survey of currency recognition system using image processing. International Journal of Computational Engineering Research, 3(7).

Blake, A., Rother, C., Brown, M., Perez, P., & Torr, P. (2004, May). Interactive imagesegmentation using an adaptive GMMRF model. In European conference on computer vision (pp. 428-441). Springer, Berlin, Heidelberg.

Bobrowski, L. (1978). Learning processes in multilayer threshold nets. Biological Cybernetics, 31(1), 1-6.

Boykov, Y. Y., & Jolly, M. P. (2001). Interactive graph cuts for optimal boundary & region segmentation of objects in ND images. In Computer Vision, 2001. ICCV 2001.

Proceedings. Eighth IEEE International Conference on (Vol. 1, pp. 105-112). IEEE.

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, Cham.

Caruana, R., Lawrence, S., & Giles, C. L. (2001). Overfitting in neural nets: Backpropagation, conjugate gradient, and early stopping. In Advances in neural information processing systems (pp. 402-408).

Chambers, J., Yan, W., Garhwal, A., Kankanhalli, M. (2015) Currency security and forensics: a survey. Multimedia Tools Appl. 74(11): 4013-4043.

Chen, L., & Zhang, C. S. (2006). AFEM@ matlab: a Matlab package of adaptive finite element methods. Technique Report, Department of Mathematics, University of Maryland at College Park.

Chen, T., & Chen, H. (1995). Universal approximation to nonlinear operators by neural networks with arbitrary activation functions and its application to dynamical systems. IEEE Transactions on Neural Networks, 6(4), 911-917.

Cireşan, D. C., Giusti, A., Gambardella, L. M., & Schmidhuber, J. (2013). Mitosis detection in breast cancer histology images with deep neural networks. In International Conference on Medical Image Computing and Computer-assisted Intervention (pp. 411-418). Springer, Berlin, Heidelberg.

Ciresan, D. C., Meier, U., Masci, J., Maria Gambardella, L., & Schmidhuber, J. (2011). Flexible, high performance convolutional neural networks for image classification. In IJCAI (Vol. 22, No. 1, p. 1237).

CireşAn, D., Meier, U., Masci, J., & Schmidhuber, J. (2012). Multi-column deep neural network for traffic sign classification. Neural networks, 32, 333-338.

Cohn, J. F., Zlochower, A. J., Lien, J. J., & Kanade, T. (1998). Feature-point tracking by optical flow discriminates subtle differences in facial expression. In FG (p. 396). IEEE.

Cucchiara, R., Grana, C., Piccardi, M., Prati, A., & Sirotti, S. (2001). Improving shadow

suppression in moving object detection with HSV color information. In Intelligent Transportation Systems, (pp. 334-339). IEEE.

Cybenko, G. (1989). Approximation by superpositions of a sigmoidal function. Mathematics of control, signals and systems, 2(4), 303-314.

Dahl, G. E., Yu, D., Deng, L., & Acero, A. (2012). Context-dependent pre-trained deep neural networks for large-vocabulary speech recognition. IEEE Transactions on audio, speech, and language processing, 20(1), 30-42.

Dai, J., He, K., & Sun, J. (2016). Instance-aware semantic segmentation via multi-task network cascades. In IEEE Conference on Computer Vision and Pattern Recognition (pp. 3150-3158).

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).

Debnath, K. K., Ahmed, S. U., Shahjahan, M., & Murase, K. (2010). A paper currency recognition system using negatively correlated neural network ensemble. Journal of Multimedia, 5(6), 560.

Deng, L., Li, J., Huang, J. T., Yao, K., Yu, D., Seide, F., ... & Gong, Y. (2013). Recent advances in deep learning for speech research at Microsoft. In ICASSP (Vol. 26, p. 64).

Deng, L., & Yu, D. (2014). Deep learning: methods and applications. Foundations and Trends in Signal Processing, 7(3–4), 197-387.

Dollár, P., & Zitnick, C. L. (2013). Structured forests for fast edge detection. In Proceedings of the IEEE International Conference on Computer Vision (pp. 1841-1848).

Du, Y., Fu, Y., & Wang, L. (2016). Representation learning of temporal dynamics for skeleton-based action recognition. IEEE Transactions on Image Processing, 25(7), 3010-3022

Elhofi, A. H., & Helaly, H. A. (2015). Comparison between digital and manual marking for toric intraocular lenses: a randomized trial. Medicine, 94(38).

Epshtein, B., & Ullman, S. (2005). Feature hierarchies for object classification. In null (pp. 220-227). IEEE.

Deng Farfade, S. S., Saberian, M. J., & Li, L. J. (2015). Multi-view face detection using deep convolutional neural networks. In International Conference on Multimedia Retrieval (pp. 643-650). ACM.

Fawzi, A., Samulowitz, H., Turaga, D., & Frossard, P. (2016). Adaptive data augmentation for image classification. In IEEE International Conference On Image Processing (ICIP) (No. EPFL-CONF-218496, pp. 3688-3692). IEEE.

Feng, L. J., Bo, L. S., & Long, T. X. (2003). An Algorithm of Real-Time Paper Currency Recognition [J]. Journal of Computer Research and Development, 7, 1057-1061.

Frosini, A., Gori, M., & Priami, P. (1996). A neural network-based model for paper currency recognition and verification. IEEE Transactions on neural networks, 7(6), 1482-1490.

Garcia, C., & Delakis, M. (2004). Convolutional face finder: A neural architecture for fast and robust face detection. IEEE Transactions on pattern analysis and machine intelligence, 26(11), 1408-1423.

Gardner, M. W., & Dorling, S. R. (1998). Artificial neural networks (the multilayer perceptron) - a review of applications in the atmospheric sciences. Atmospheric environment, 32(14-15), 2627-2636.

Garg, R., BG, V. K., Carneiro, G., & Reid, I. (2016). Unsupervised CNN for single view depth estimation: Geometry to the rescue. In European Conference on Computer Vision (pp. 740-756). Springer, Cham.

Gehler, P., & Nowozin, S. (2009). On feature combination for multiclass object classification. In International Conference on Computer Vision (pp. 221-228). IEEE.

Girshick, R., Donahue, J., Darrell, T., & Malik, J. (2016). Region-based convolutional networks for accurate object detection and segmentation. IEEE Transactions on pattern analysis and machine intelligence, 38(1), 142-158.

Girshick, R. (2015). Fast R-CNN. In IEEE international conference on computer

vision (pp. 1440-1448).

Gottschalk, S., Lin, M. C., & Manocha, D. (1996). OBBTree: A hierarchical structure for rapid interference detection. In conference on Computer graphics and interactive techniques (pp. 171-180). ACM.

Granger, E., Kiran, M., & Blais-Morin, L. A. (2017). A comparison of CNN-based face and head detectors for real-time video surveillance applications. In International Conference on Image Processing Theory, Tools and Applications (IPTA) (pp. 1-7). IEEE.

Greig, D. M., Porteous, B. T., & Seheult, A. H. (1989). Exact maximum a posteriori estimation for binary images. Journal of the Royal Statistical Society. Series B (Methodological), 271-279.

Gunaratna, D. A. K. S., Kodikara, N. D., & Premaratne, H. L. (2008). ANN based currency recognition system using compressed gray scale and application for Sri Lankan currency notes-SLCRec. Proceedings of World academy of science, engineering and technology, 35,235-240.

Guo, H., & Gelfand, S. B. (1992). Classification trees with neural network feature extraction. IEEE Transactions on Neural Networks, 3(6), 923-933.

Hassanpour, H., & Farahabadi, P. M. (2009). Using Hidden Markov Models for paper currency recognition. Expert Systems with Applications, 36(6), 10105-10111.

Hasanuzzaman, F. M., Yang, X., & Tian, Y. (2011). Robust and effective component-based banknote recognition by SURF features. In Wireless and Optical Communications Conference (WOCC), (pp. 1-6). IEEE.

Hawkins, D. M. (2004). The problem of overfitting. Journal of chemical information and computer sciences, 44(1), 1-12.

Hochstadt, A. M. (1975). U.S. Patent No. 3,869,600. Washington, DC: U.S. Patent and Trademark Office.

Hoo-Chang, S., Roth, H. R., Gao, M., Lu, L., Xu, Z., Nogues, I., ... & Summers, R. M. (2016). Deep convolutional neural networks for computer-aided detection: CNN architectures, dataset characteristics and transfer learning. IEEE Transactions on

medical imaging, 35(5), 1285.

Howard, A. G., Zhu, M., Chen, B., Kalenichenko, D., Wang, W., Weyand, T., ... & Adam, H. (2017). Mobilenets: Efficient convolutional neural networks for mobile vision applications. arXiv preprint arXiv:1704.04861.

Huang, J., Rathod, V., Sun, C., Zhu, M., Korattikara, A., Fathi, A., ... & Murphy, K. (2017). Speed/accuracy trade-offs for modern convolutional object detectors. In IEEE CVPR (Vol. 4).

Huang, K., & Yan, H. (1997). Off-line signature verification based on geometric feature extraction and neural network classification. Pattern Recognition, 30(1), 9-17.

Huang, W., Qiao, Y., & Tang, X. (2014). Robust scene text detection with convolution neural network induced MSER trees. In European Conference on Computer Vision (pp. 497-511). Springer, Cham.

Janocha, K., & Czarnecki, W. M. (2017). On loss functions for deep neural networks in classification. arXiv preprint arXiv:1702.05659.

Javed, O., & Shah, M. (2002). Tracking and object classification for automated surveillance. In European Conference on Computer Vision (pp. 343-357). Springer, Berlin, Heidelberg.

Karpathy, A., Toderici, G., Shetty, S., Leung, T., Sukthankar, R., & Fei-Fei, L. (2014). Large-scale video classification with convolutional neural networks. In IEEE conference on Computer Vision and Pattern Recognition (pp. 1725-1732).

Kim, Y. (2014). Convolutional neural networks for sentence classification. arXiv preprint arXiv:1408.5882.

Ko, Y. H., Kim, K. J., & Jun, C. H. (2005). A new loss function-based method for multiresponse optimization. Journal of Quality Technology, 37(1), 50-59.

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).

Kuo, C. C. J. (2016). Understanding convolutional neural networks with a mathematical

model. Journal of Visual Communication and Image Representation, 41, 406-413.

Lau, R., Weir, J., Yan, W. (2009): Digital Image Splicing Using Edges. PCM 2009: 697-707.

Lauer, F., Suen, C. Y., & Bloch, G. (2007). A trainable feature extractor for handwritten digit recognition. Pattern Recognition, 40(6), 1816-1824.

Lawrence, S., Giles, C. L., Tsoi, A. C., & Back, A. D. (1997). Face recognition: A convolutional neural-network approach. IEEE Transactions on neural networks, 8(1), 98-113.

Le, Q. V., Ngiam, J., Coates, A., Lahiri, A., Prochnow, B., & Ng, A. Y. (2011). On optimization methods for deep learning. In Proceedings of the 28th International Conference on International Conference on Machine Learning (pp. 265-272). Omnipress.

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

Lempitsky, V. S., Kohli, P., Rother, C., & Sharp, T. (2009). Image segmentation with a bounding box prior. In ICCV (pp. 277-284).

Li, B., Chen, H., Chen, Y., Dai, Y., & He, M. (2017). Skeleton boxes: Solving skeleton based action detection with a single deep convolutional neural network. arXiv preprint arXiv:1704.05643.

Li, F., & Yang, Y. (2003). A loss function analysis for classification methods in text categorization. In Proceedings of the 20th International Conference on Machine Learning (ICML-03) (pp. 472-479).

Liao, M., Shi, B., Bai, X., Wang, X., & Liu, W. (2017). TextBoxes: A Fast Text Detector with a Single Deep Neural Network. In AAAI (pp. 4161-4167).

Lin, T. Y., Dollár, P., Girshick, R. B., He, K., Hariharan, B., & Belongie, S. J. (2017). Feature Pyramid Networks for Object Detection. In CVPR (Vol. 1, No. 2, p. 4).

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, Cham.

Liu, Z., Yan, W., Yang, B. (2018) Image denoising based on a CNN model, ICCAR 1 (1).

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

Mao, J., & Jain, A. K. (1995). Artificial neural networks for feature extraction and multivariate data projection. IEEE Transactions on neural networks, 6(2), 296-317.

Masci, J., Meier, U., Cireşan, D., & Schmidhuber, J. (2011). Stacked convolutional auto-encoders for hierarchical feature extraction. In International Conference on Artificial Neural Networks (pp. 52-59). Springer, Berlin, Heidelberg.

Matan, O., Baird, H. S., Bromley, J., Burges, C. J. C., Denker, J. S., Jackel, L. D., ... & Thompson, T. J. (1992). Reading handwritten digits: A zip code recognition system. Computer, 25(7), 59-63.

Meng, X. L., & Van Dyk, D. A. (1999). Seeking efficient data augmentation schemes via conditional and marginal augmentation. Biometrika, 86(2), 301-320.

Merriënboer, B., Bahdanau, D., Dumoulin, V., Serdyuk, D., Warde-Farley, D., Chorowski, J., Bengio, Y. (2015). Blocks and fuel: Frameworks for deep learning. arXiv preprint arXiv:1506.00619.

Murtagh, F. (1991). Multilayer perceptrons for classification and regression. Neurocomputing, 2(5-6), 183-197.

Najafabadi, M. M., Villanustre, F., Khoshgoftaar, T. M., Seliya, N., Wald, R., & Muharemagic, E. (2015). Deep learning applications and challenges in big data analytics. Journal of Big Data, 2(1), 1.

Ng, A. Y., & Jordan, M. I. (2002). On discriminative vs. generative classifiers: A comparison of logistic regression and Naive Bayes. In Advances in neural information processing systems (pp. 841-848).

Ngiam, J., Khosla, A., Kim, M., Nam, J., Lee, H., & Ng, A. Y. (2011). Multimodal deep learning. In Proceedings of the 28th international conference on machine learning (ICML-11) (pp. 689-696).

Peduzzi, P., Concato, J., Kemper, E., Holford, T. R., & Feinstein, A. R. (1996). A simulation study of the number of events per variable in logistic regression analysis. Journal of clinical epidemiology, 49(12), 1373-1379.

Perez, L., & Wang, J. (2017). The effectiveness of data augmentation in image classification using deep learning. arXiv preprint arXiv:1712.04621.

Pham, T. D., Nguyen, D. T., Kim, W., Park, S. H., & Park, K. R. (2018). Deep Learning-Based Banknote Fitness Classification Using the Reflection Images by a Visible-Light One-Dimensional Line Image Sensor. Sensors, 18(2), 472.

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).

Ren, Y. (2017). Banknote Recognition in Real Time Using ANN (Doctoral dissertation, Auckland University of Technology).

Y Ren, M Nguyen, W Yan. (2018) Real-Time Recognition of Series Seven New Zealand Banknotes, IJDCF 10 (3), 50-66

Reponen, E., Huuskonen, P., & Mihalic, K. (2008). Primary and secondary context in mobile video communication. Personal and Ubiquitous Computing, 12(4), 281-288.

Ronneberger, O., Fischer, P., & Brox, T. (2015). U-net: Convolutional networks for biomedical image segmentation. In International Conference on Medical image computing and computer-assisted intervention (pp. 234-241). Springer, Cham.

Rriedman, J., Hastie, T., & Tibshirani, R. (2000). Additive logistic regression: A statistical view of boosting. The Annals of Statistics, 38:2, 337–374.)

Sahiner, B., Chan, H. P., Petrick, N., Wei, D., Helvie, M. A., Adler, D. D., & Goodsitt, M. M. (1996). Classification of mass and normal breast tissue: a convolution neural network classifier with spatial domain and texture images. IEEE Transactions on Medical Imaging, 15(5), 598-610.

Sainath, T. N., Mohamed, A. R., Kingsbury, B., & Ramabhadran, B. (2013). Deep convolutional neural networks for LVCSR. In IEEE international conference

on Acoustics, speech and signal processing (ICASSP) (pp. 8614-8618).

Salamon, J., & Bello, J. P. (2017). Deep convolutional neural networks and data augmentation for environmental sound classification. IEEE Signal Processing Letters, 24(3), 279-283.

Sarle, W. S. (1996). Stopped training and other remedies for overfitting. Computing science and statistics, 352-360.

Schmidhuber, J. (2015). Deep learning in neural networks: An overview. Neural networks, 61, 85-117.

Seide, F., Li, G., Chen, X., & Yu, D. (2011). Feature engineering in context-dependent deep neural networks for conversational speech transcription. In IEEE Workshop on Automatic Speech Recognition and Understanding (ASRU) (pp. 24-29).

Seide, F., Li, G., & Yu, D. (2011). Conversational speech transcription using context-dependent deep neural networks. In Twelfth annual conference of the international speech communication association.

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.

Severyn, A., & Moschitti, A. (2015). Twitter sentiment analysis with deep convolutional neural networks. In International ACM SIGIR Conference on Research and Development in Information Retrieval (pp. 959-962). ACM.

D Shen, X Chen, M Nguyen, W Yan (2018) Flame detection using deep learning. ICCAR'18, 1 (1).

Shen, W., Wang, X., Wang, Y., Bai, X., & Zhang, Z. (2015). Deepcontour: A deep convolutional feature learned by positive-sharing loss for contour detection. In IEEE Conference on Computer Vision and Pattern Recognition (pp. 3982-3991).

Simonyan, K., & Zisserman, A. (2014). Very deep convolutional networks for large-scale image recognition. arXiv preprint arXiv:1409.1556.

Song, Z., Chen, Q., Huang, Z., Hua, Y., & Yan, S. (2011). Contextualizing object detection

and classification. In IEEE Conference on Computer Vision and Pattern Recognition (CVPR) (pp. 1585-1592). IEEE.

Springenberg, J. T., Dosovitskiy, A., Brox, T., & Riedmiller, M. (2014). Striving for simplicity: the all convolutional net. arXiv preprint arXiv:1412.6806.

Srivastava, N., Hinton, G., Krizhevsky, A., Sutskever, I., & Salakhutdinov, R. (2014). Dropout: a simple way to prevent neural networks from overfitting. The Journal of Machine Learning Research, 15(1), 1929-1958.

Sun, X., Wu, P., & Hoi, S. C. (2018). Face detection using deep learning: An improved faster RCNN approach. Neurocomputing, 299, 42-50.

Sun, Y., Liang, D., Wang, X., & Tang, X. (2015). Deepid3: Face recognition with very deep neural networks. arXiv preprint arXiv:1502.00873.

Sun, Y., Wang, X., & Tang, X. (2013). Deep convolutional network cascade for facial point detection. In IEEE conference on computer vision and pattern recognition (pp. 3476-3483).

Sutskever, I., Martens, J., Dahl, G., & Hinton, G. (2013). On the importance of initialization and momentum in deep learning. In International conference on machine learning (pp. 1139-1147).

Szegedy, C., Ioffe, S., Vanhoucke, V., & Alemi, A. A. (2017). Inception-v4, inception-resnet and the impact of residual connections on learning. In AAAI (Vol. 4, p. 12).

Szegedy, C., Vanhoucke, V., Ioffe, S., Shlens, J., & Wojna, Z. (2016). Rethinking the inception architecture for computer vision. In Proceedings of the IEEE conference on computer vision and pattern recognition (pp. 2818-2826).

Takeda, F., & Omatu, S. (1995). A neuro-paper currency recognition method using optimized masks by genetic algorithm. In IEEE International Conference on Systems, Man and Cybernetics (Vol. 5, pp. 4367-4371).

Tian, Z., Huang, W., He, T., He, P., & Qiao, Y. (2016). Detecting text in natural image with connectionist text proposal network. In European conference on computer vision (pp. 56-72). Springer.

Trier, O. D., Jain, A. K., & Taxt, T. (1996). Feature extraction methods for character recognition-a survey. Pattern recognition, 29(4), 641-662

Van Merriënboer, B., Bahdanau, D., Dumoulin, V., Serdyuk, D., Warde-Farley, D., Chorowski,

Vedaldi, A., & Lenc, K. (2015). Matconvnet: Convolutional neural networks for matlab. In ACM International Conference on Multimedia (pp. 689-692). ACM.

Waibel, A. (1989). Modular construction of time-delay neural networks for speech recognition. Neural computation, 1(1), 39-46.

Wang, G., Liu, F., Yan, W. (2014) Braille for Visual Cryptography. IEEE ISM 2014: 275-276.

Wang, G., Wu, X., Yan, W. (2017) The State-of-the-Art Technology of Currency Identification: A Comparative Study. IJDCF 9(3): 58-72

Wang, Y., Xia, L., Tang, T., Li, B., Yao, S., Cheng, M., & Yang, H. (2016). Low power convolutional neural networks on a chip. In IEEE International Symposium on Circuits and Systems (ISCAS) (pp. 129-132).

Wen, Y., Zhang, K., Li, Z., & Qiao, Y. (2016). A discriminative feature learning approach for deep face recognition. In European Conference on Computer Vision (pp. 499-515). Springer, Cham.

Wei, Y., Xia, W., Huang, J., Ni, B., Dong, J., Zhao, Y., & Yan, S. (2014). CNN: single-label to multi-label. arXiv preprint arXiv:1406.5726.

Weir, J., Yan, W., Kankanhalli, M. (2012) Image hatching for visual cryptography. ACM TOMCCAP 8(S2): 32:1-32:15.

Witschorik, C. A. (2000). U.S. Patent No. 6,131,718. Washington, DC: U.S. Patent and Trademark Office.

Wu, J. W., Chen, M. C., Fan, Z., & Phong, K. A. (2001). U.S. Patent No. 6,317,524. Washington, DC: U.S. Patent and Trademark Office.

Wu, R., Yan, S., Shan, Y., Dang, Q., & Sun, G. (2015). Deep image: Scaling up image recognition. arXiv preprint arXiv:1501.02876.

Xing, C., Ma, L., & Yang, X. (2016). Stacked denoise autoencoder based feature extraction and   classification for hyperspectral images. Journal of Sensors, 2016.

Xu, L., Ren, J. S., Liu, C., & Jia, J. (2014). Deep convolutional neural network for image deconvolution. In Advances in Neural Information Processing Systems (pp. 1790-1798).

Yadav, B. P., Patil, C. S., Karhe, R. R., & Patil, P. H. (2014). An automatic recognition of fake Indian paper currency note using MATLAB. Int. J. Eng. Sci. Innov. Technol, 3, 560-566.

Yan, W. (2017) Introduction to intelligent surveillance, Springer.

Yan, W., Chambers, J. (2013) An empirical approach for digital currency forensics. IEEE ISCAS 2013: 2988-2991.

Yan, W., Chambers, J., Garhwal, A. (2015) An empirical approach for currency identification. Multimedia Tools Appl. 74(13): 4723-4733.

Yeh, C. Y., Su, W. P., & Lee, S. J. (2011). Employing multiple-kernel support vector machines   for counterfeit banknote recognition. Applied Soft Computing, 11(1), 1439-1447.

Yu, D., & Deng, L. (2011). Deep learning and its applications to signal and information processing. IEEE Signal Processing Magazine, 28(1), 145-154.

Zanaty, E. A. (2012). Support Vector Machines (SVMs) versus multilayer perception (MLP) in data classification. Egyptian Informatics Journal, 13(3), 177-183.

Zhang, E. H., Jiang, B., Duan, J. H., & Bian, Z. Z. (2003). Research on paper currency recognition by neural networks. In International Conference on Machine Learning and Cybernetics (Vol. 4, pp. 2193-2197).

Zhang, L., Lin, L., Liang, X., & He, K. (2016). Is faster R-CNN doing well for pedestrian detection? In European Conference on Computer Vision (pp. 443-457). Springer.

Zhao, R., Ouyang, W., Li, H., & Wang, X. (2015). Saliency detection by multi-context deep learning. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (pp. 1265-1274).

Zheng, K., Yan, W., Nand, P. (2018) Video Dynamics Detection Using Deep Neural Networks. IEEE Transactions on Emerging Topics in Computational Intelligence, 2(3): 224-234.

Zhu, M., & Gupta, S. (2017). To prune, or not to prune: exploring the efficacy of pruning for model compression. arXiv preprint arXiv:1710.01878.