

Banknote Serial Number Recognition Using Deep Learning

Xin Ma

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

2020

School of Engineering, Computer and Mathematical Sciences

Abstract

Deep learning has been broadly applied to pattern classification, object detection, image segmentation, speech recognition, and other fields in recent years. Convolutional neural networks take dominant role in the field of deep learning, which has excellent characteristics that traditional machine learning algorithms cannot reach. The problem of character recognition has also been extensively studied in recent years, whose scope is much wide, including license plate recognition, handwriting recognition, bank check, and handwriting recognition for postcodes on envelop, etc.

According to the current circulation banknote in New Zealand, this thesis applies deep learning to the character recognition of serial numbers on banknotes. The data samples used in this thesis are the images from the sixth edition of New Zealand banknote, which have been preprocessed with labelling, augmentation, scaling, and transformation, etc. The algorithms based on deep learning are proposed which have the stability for the serial number recognition in complex backgrounds.

In this thesis, a pipeline of deep neural networks is constructed for character recognition of banknote serial numbers. Since high reliability is more important than accuracy in financial applications, DenseNet is proposed as the primary classifier, the scaling transformation of SegLink is employed to locate the characters, the detection rate is up to 95.80%. A convolutional neural network with residual attention model is proposed for serial number recognition, the precision reaches up to 97.09%.

Keywords: Serial numbers of banknote, Object detection, Convolution neural networks (CNNs), Single shot multibox detector (SSD), ResNet, Inception, DenseNet, Data augmentation, SegLink, CRNN, Attention Model.

Table of Contents

Abstract.....	II
List of Figures.....	VI
List of Tables	IX
List of Algorithms.....	X
Attestation of Authorship.....	XI
Acknowledgement	XII
Chapter 1 Introduction	1
1.1 Background and Motivation	2
1.2 Research Question	4
1.3 Contribution.....	4
1.4 Objectives of This Thesis.....	5
1.5 Structure of This Thesis	5
Chapter 2 Literature Review	7
2.1 Introduction.....	8
2.2 Deep Learning.....	11
2.2 Artificial Neural Network	12
2.3 Convolutional Neural Network.....	13
2.3.1 Convolutional Neural Network Structure	14
2.3.2 Classic Convolutional Neural Network	17
2.4 Text Detection.....	20

2.4.1 SSD	21
2.4.2 SegLink.....	22
2.4.3 DenseNet.....	23
2.5 Text Recognition.....	24
2.5.1 CRNN	24
2.5.2 Attention Model.....	25
2.6 Data Augmentation	26
Chapter 3 Methodology	27
3.1 Introduction.....	28
3.2 Research Design	31
3.3 Data Preprocessing	32
3.3.1 Data Collection	32
3.3.2 Data Labelling.....	34
3.3.3 Data Augmentation.....	36
3.3.4 Training and Testing Sets	37
3.4 Architecture Design	38
3.4.1 Banknote Serial Number Area Detection	38
3.4.2 Serial Number Recognition Network.....	44
3.5 Evaluation Method.....	53
3.5.1 Loss Function for Detection.....	53
3.5.2 Loss Function for Recognition.....	55
3.5.3 Evaluation Metrics	55
3.6 Summary	57

Chapter 4 Results	59
4.1 Experimental Environment	60
4.2 Algorithm Analysis and Verification.....	60
4.2.1 Serial Number Area Detection.....	60
4.2.2 Serial Number Recognition	68
Chapter 5 Analysis and Discussion.....	72
5.1 Analysis	73
5.1.1 Analysis of Different Model for Banknote Serial Number Area Detection.....	73
5.1.2 Analysis of Different Model for Banknote Serial Number Recognition	74
5.2 Discussion.....	75
5.2.1 Discussion of Models Performance for Banknote Serial Number Area Detection..	75
5.2.2 Discussion of Models Performances for Banknote Serial Number Recognition.....	75
Chapter 6 Conclusion and Future Work	77
6.1 Conclusion	78
6.1.1 Conclusion of Detection Network	78
6.2.2 Conclusion of Recognition Network	79
6.2 Limitations	79
6.2.1 Limitations of Detection Network	79
6.2.2 Limitations of Recognition Network	80
6.3 Future Work.....	80
6.3.1 Future Work of Detection Network	80
6.3.2 Future Work of Recognition Network	81
References.....	82

List of Figures

Figure 1.1 Security features of New Zealand's series 7 banknotes.....	3
Figure 1.2 Banknote recognition workflow in an automated device.....	3
Figure 2.1 Biological neuron.....	12
Figure 2.2 Neural network structure.....	12
Figure 2.3 The CNN architecture.....	14
Figure 2.4 The Convolution operation.....	15
Figure 2.5 The graph of the sigmoid(up) and tanh(down)	15
Figure 2.6 The structure of VGG network with different depths.....	19
Figure 2.7 Recognition results with object localization.....	24
Figure 3.1 Challenges of text detection and recognition	28
Figure 3.2 Diversity of banknote texts.....	29
Figure 3.3 Complexity of background of banknote.....	29
Figure 3.4 Noise and blurring of banknote serial number	30
Figure 3.5 The pipeline of the end-to-end network for the recognition of banknote serial numbers.....	31
Figure 3.6 The research design	31
Figure 3.7 The images of banknotes with different denomination.....	33
Figure 3.8 Banknote samples for our experiments.....	34
Figure 3.9 Region labelling for serial number detection.....	35
Figure 3.10 Labelling for Serial number recognition.....	35
Figure 3.11 The samples of our data augmentation for the banknotes \$20, \$50 and \$100(NZD).....	37

Figure 3.12 K-fold cross-validation	38
Figure 3.13 A 5-layer dense block network.....	39
Figure 3.14 The network architecture for locating banknote serial numbers.....	41
Figure 3.15 Within layer links and cross-layer links.....	42
Figure 3.16 CRNN network.....	45
Figure 3.17 Bidirectional-LSTM block.....	45
Figure 3.18 CRNN with attention model.....	49
Figure 3.19 Attention feature encoder.....	49
Figure 3.20 Dense block used in modified CRNN network.....	50
Figure 3.21 Residual attention	51
Figure 3.22 Convolutional sequence modeling and CTC.....	52
Figure 3.23 Four kinds of examples.....	54
Figure 3.24 ROC curve example.....	57
Figure 4.1 Training loss with different learning rate.....	61
Figure 4.2 The convergence speeds of SGD and SGD with momentum.....	63
Figure 4.3 Performances of the network using Adam Optimizers.....	64
Figure 4.4 Performances of the network using SGD Optimizers.....	65
Figure 4.5 Performances of network using GELUS activate function.....	66
Figure 4.6 Performances of network using ReLU activate function.....	66
Figure 4.7 Training loss.....	67
Figure 4.8 Validation loss.....	67
Figure 4.9 The detection results of NZ currency.....	68
Figure 4.10 CRNN network for banknote serial number recognition	69

figure 4.11 Modified CRNN with attention model.....	70
Figure 4.12 The loss of CRNN network during training.....	71
Figure 4.13 The loss of modified CRNN network with attention block.....	71
Figure 4.14 The recognition results of 20, 50 and 100 NZD.....	71
Figure 5.1 Different detection results using different networks.....	74

List of Tables

Table 3.1 DenseNet architecture.....	40
Table 3.2 CRNN network configuration summary.....	46
Table 3.3 The architecture of the CRNN network with attention model.....	53
Table 4.1 Final results of the serial number detection.....	68
Table 4.2 Result of serial number recognition.....	69
Table 5.1 The performances of different detection models.....	73
Table 5.2 The performances of different recognition networks.....	74
Table 5.3 The training time (s) of different recognition networks.....	76

List of Algorithms

Algorithm 3.1 Combining segments	44
--	----

Attestation of Authorship

I hereby declare that this submission is my 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 acknowledgements), 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: Xin Ma

Date: 02 March 2020

Acknowledgement

This thesis 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) with 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 helps during the learning process. His patiently explanations to my questions make me finally complete my research work. In addition, I would like to thank my teachers and administrators of this school for their help in the past.

Xin Ma

Auckland, New Zealand

03 March 2020

Chapter 1

Introduction

This chapter mainly includes five parts. Firstly, the overview and necessity as well as the significance of banknote serial number recognition using deep learning in recent years will be construed. Sections 1.2 and 1.3 will list the main research questions that will be discussed in this thesis and propose meaningful contributions in the field of deep learning. Section 1.4 will explicate the important significance and the final implementation of the function. Finally, the details about this thesis and the core content of each chapter will be depicted in Section 1.5.

1.1 Background and Motivation

Although the proportion of online banking and EFTPOS payment is virtually increasing, the banknote is still one of the most widely used payment methods in people's daily life. The traditional way of liquidation, recovery, and recirculation of money has been far from meeting the developing needs of society. Therefore, the use of modern financial equipment has become the general trend of the financial system. At present, terminals such as banknote counter and banknote sorter have become indispensable tools in the banking business, they can replace bank staff to complete heavy work faster and accurate. In order to ensure the healthy and rapid development of social economy, supervision and management of banknotes are particularly important.

With the advancement of technology and usability of intelligent surveillance, the utilization is much convenient, resulting in its broad implementations in object detection such as car plate recognition, face and pedestrian detection, etc. Intelligent surveillance has become a security technique safeguarding our public and private safety, amalgamating computer science with engineering and multidiscipline, including computer vision, digital image processing, and cyber security as well as computational intelligence (Yan, 2019)

The development of an intelligent banknote identification system using intelligent surveillance, combined with the Internet of Things (IoT) technology, can implement the serial number tracking and authenticity of banknotes through the Internet between banks, this requires the assistance of banknote image processing and other technologies.

Banknote image analysis mainly includes banknote serial number recognition, banknote quality assessment, banknote feature detection, and banknote recognition (Chambers, 2014; Zhang, 2018). A banknote serial number is a specific alphanumeric identifier printed on each banknote when the banknote is produced. Each banknote has its unique identifier number, which is equivalent to an identification mark. The serial number of each banknote is not repeated. It consists of 10 characters, including 2 letters and 8 numbers indicating the name of the bank issued this banknote and serial information of each face value. They are typed and printed according to a strict rule, which is used to indicate the printed quantity and batch of banknotes. Because of the uniqueness of the banknote serial number, recognize and record the number can be used to track the origin and circulation path of the banknote, thus can be efficiently applied to detect counterfeit banknotes.

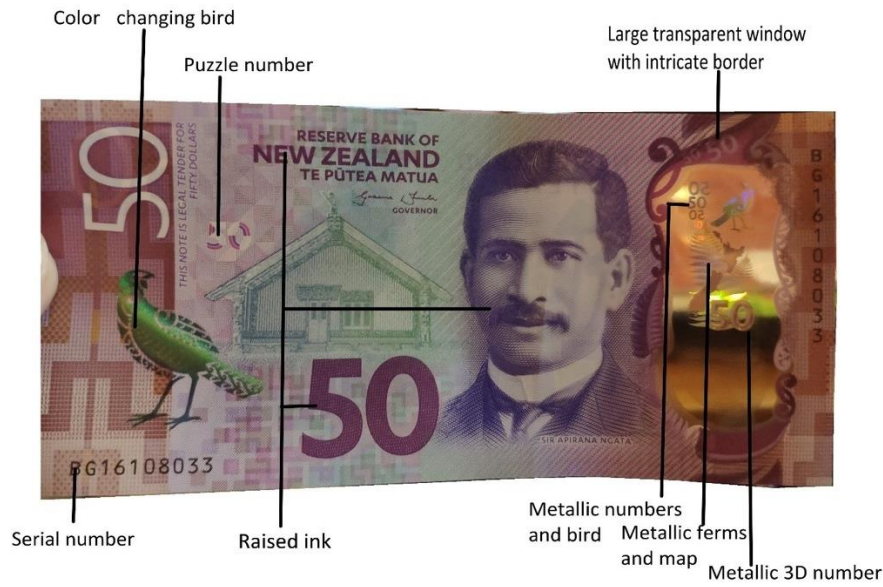


Figure 1.1 Security features of Series 7 banknotes of New Zealand

Serial number recognition technology is an essential technology for bank money counters, money detectors, sorters, ATMs, vending machines, and so on. Through the preprocessing, segmentation, and recognition of the banknote images, the purpose of identifying the serial numbers of the banknote is achieved. Due to the importance of banknotes, how to choose appropriate features, reasonable identification methods, and accurate and efficient identification of banknote serial numbers in the process of banknote recognition is an urgent problem that needs to be resolved. Therefore, the research work of fast and accurate currency identification is of great value to healthy and rapid development of bank economy.

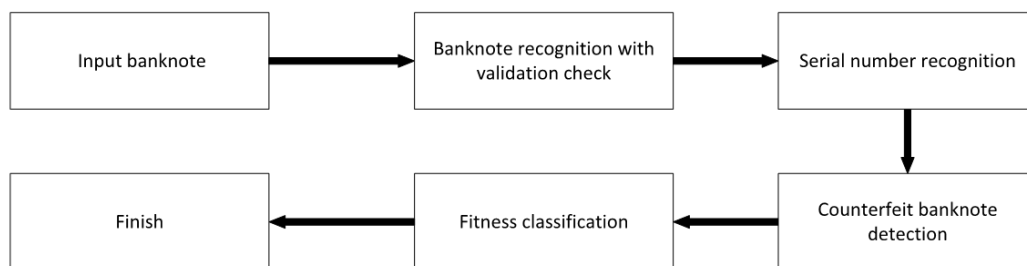


Figure 1.2 Banknote recognition workflow in an automated device

Although the current deep learning technologies represented by Faster R-CNN (Ren, He, Girshick, & Sun, 2015), You Only Look Once(YOLO) (Redmon, Divvala, Girshick, & Farhadi, 2016), and Single Shot MulutiBox Detector(SSD) (Liu et al., 2016) have achieved

excellent results in object recognition, it is still a challenge, similar to classes recognition such as digits and letters compared to much more intricate objects such as planes, vehicles, and humans (Huang et al., 2017).

1.2 Research Question

There are currently few studies on banknote serial number recognition through deep learning. Thus, the research questions in this thesis are listed as follows:

- Can deep learning techniques be used to banknote serial number detection and recognition?

More specifically, this question can be described as:

- Which algorithms can be used for the banknote serial number detection and recognition? Are there any possibilities to improve accuracy by modifying and combining existing algorithms?

1.3 Contributions

This thesis proposed a pipeline that includes banknote serial number detection and serial number recognition. Images with any resolutions, e.g., photos and images captured by cameras, will be detected and output the recognized serial number.

For the detection, in order to capture the features, DenseNet is used which turns out to be very effective. Based on the characteristics of the letter sequence, SegLink network is adopted in this thesis, which is employed to detect the segment and link the detected segments both inside a layer and cross different layers.

When it comes to the recognition, Recurrent Neural Network (RNN) is applied in the first place. Based on the problem identified during the experiments, the RNN layers are replaced by the CNN layers to obtain the context relevance of the input sequence efficiently, which is characterized by lower computational complexity and easier parallel computing. Furthermore, an attention model with residual operation is used to suppress the background noises and

improve the discriminability of CNNs. As a result, highly competitive performance and efficiency demonstrate the superiority of the proposed approach.

1.4 Objectives of This Thesis

First of all, this research aims to detect and recognize the banknote serial numbers. The process consists of two parts: In the first part, the detection of serial number region and the second part is the recognition of the numbers. Compared visual objects such as people, dogs, ships, the recognition of serial numbers on banknotes belongs to that of very small objects, which requires multiple experiments based on advanced recognition algorithms.

Secondly, the background of the serial number region on the banknotes are much complicated, there is no obvious border or outline, instead, there are a large number of patterns, vertical lines, rectangles, and shades of colors that interfere with the character recognition. In this thesis, the dataset is set up by taking pictures of banknotes, data augmentation method is offered to solve the potential problem due to insufficient samples.

Finally, to implement the real-time identification of banknote serial number, multiple neural networks, scale transformations, optimizers, and loss functions are employed in this thesis to reach a high precision of the identification from low-quality images.

1.5 Structure of This Thesis

The basic structure and content of this thesis are as follows:

In the second chapter, the previous literature will be reviewed and discussed, including recent methods and models employed to implement text detection and recognition. Therefore, Chapter 2 will review the state-of-art recognition models and data augmentation methods.

In the third chapter, the research methods will be introduced based on details from the perspective of network construction. Besides, Chapter 3 will introduce image preprocessing, dataset acquisition and processing, loss function and principles of model evaluation methods, etc.

In Chapter 4, the experimental results and datasets are summarized in tabular and graphical ways. Moreover, multiple choices of optimizers and activate functions are applied and

compared in detail. A solution to the core problem advocated in this thesis will be proposed based on our experimental results considering both speed and accuracy.

In Chapter 5, we will discuss the experimental results, tables, and figures shown in Chapter 4.

Chapter 6 is related to a summary of the entire thesis, the limitations of the research, and the plan for future work.

Chapter 2

Literature Review

With a comprehensive examination of the research question and reasonable reviews of the previous studies, the focus of this thesis is on banknote serial number region detection and recognition in digital images. In this chapter, we will review and summarize the achievements of research work in the case of object detection and text recognition over the past few years.

2.1 Introduction

A huge amount of research work has been conducted on character recognition such as handwriting recognition (HWR) and printed character recognition (PCR). The work has been applied to plenty of fields. There are numerous methods which have been developed to process the bank cheque (Xu^Y, Lam^YP, & Suen^Y, 2003), zip code recognition (Le Cun et al., 1990) and car plate serial number recognition, etc. Despite this, little work has been carried out on the recognition of banknote serial number (Zhao, Zhao, Zheng, & Zhang, 2010), which is very helpful to reduce financial crime and improve the market security by recognizing, recording, and analyzing a banknote (Wenhong, Wenjuan, Xiyan, & Zhen, 2010).

Banknote recognition is mainly split into two categories: Feature extraction, pattern classification. Since the 1990s, banknote image analysis has been investigated. Banknote recognition based on neural networks and free feature methods, probabilistic principal component analysis (Mohamad, Hussin, Shibghatullah, & Basari, 2014), competitive neural networks, learning quantization networks, etc. have turned up one after another since then. There are also multiple methods for banknote multispectral feature extraction, wavelet-based contourlet transform, and fuzzy logic for feature extraction from banknote images.

The serial number recognition in banknote image processing belongs to optical character recognition (OCR). Character is one of the most brilliant and influential creations of mankind, making it possible to reliably or effectively spread or obtain information. Letters or character strings, namely, text can be applied in a wide range of real-world applications, such as image search (Schroth, Hilsenbeck, Huitl, Schweiger, & Steinbach, 2011; Tsai et al., 2011), robots navigation (DeSouza, Kak, & intelligence, 2002; X. Liu & J. Samarabandu, 2005; X. Liu & J. K. Samarabandu, 2005; Schulz et al., 2015), instant translation (Dvorin & Havosha, 2009); (Parkinson, Jacobsen, Ferguson, & Pombo, 2016), and industrial automation (Chowdhury & Deb, 2013); (Ham, Kang, Chung, Park, & Park, 1995; Z. He, Liu, Ma, & Li, 2005). Therefore, automatic text reading from natural averments such as scene characters and symbols recognition (Zhu, Yao, & Bai, 2016) or OCR (Bissacco, Cummins, Netzer, & Neven, 2013) has become an increasingly popular and important research topic in the subject of computer vision.

There are a myriad of recognition methods of text recognition such as template matching (Ryan & Hanafiah, 2015), feature extraction method, backpropagation (BP), artificial neural

networks (Li, Cheng, Shi, & Huang, 2012), structural pattern recognition, hidden Markov models, support vector machine (SVM), deep learning, etc.

Before deep learning was widely applied, different text recognition methods have been well studied, including connected component analysis (CCA) (Epstein, Ofekx, & Wexller, 2010; Huan, Li, Yang, & Wang, 2013; Jain & Yu, 1998; Neumann & Matas, 2010; Yao, Bai, Li, Ma, & Tu, 2012; Yi & Tian, 2011; Yin, Yin, Huang, Hao, & intelligence, 2013), sliding window (SW) -based classification (Coates et al., 2011; Lee, Lee, Lee, Yuille, & Koch, 2011). CCA-based methods first abstract candidate components through multiple methods (such as clustering color or extracting exceeding region), then use manually designed rules or classifiers to filter out non-text components and use low-level functions to automatically train them. In sliding windows classification, a variable-size window slides over the input image, where each window is classified as a text region. Further classification of those classified as positive detects regions by morphological operations, conditional random field (CRF), and other graph-based methods (Dai et al., 2018).

For OCR, one branch adopted the feature-based methods. The segmentation-based algorithms are proposed (Shi et al., 2013) (Yao, et al., 2014)). Label embedding is employed to directly match between strings and images (Rodriguez-Serrano, et al., 2015; Gordo, 2015; Almazán, et al., 2014). Strokes (Busta, Neumann, & Matas, 2015) and character key points (Quy Phan, Shivakumara, Tian, & Lim Tan, 2013) are also detected as features for classification.

A plurality of methods have been put forward to tackle these sub-questions, which include text binarization (Lee, Kim, & Computing, 2013; Mishra, Alahari, & Jawahar, 2011; Wakahara & Kita, 2011; Zhiwei, Linlin, & Lim, 2010), text line segmentation (F. Yin, Wu, Zhang, & Liu, 2017), character segmentation (Nomura, Yamanaka, Katai, Kawakami, & Shiose, 2005; Roy, Pal, Lladós, & Delalandre, 2009; Shivakumara, Bhowmick, Su, Tan, & Pal, 2011), single character recognition (Nomura et al., 2005; Sheshadri & Divvala, 2012), and word correction (Weinman, Learned-Miller, & Hanson, 2007).

As a conclusion, methods of detecting and recognizing characters before deep learning is widely used are mostly based on extracting low-level or mid-level image features, which entail stringent and repetitious pre-processing or postprocessing. Restricted by the limitation of

representation capabilities of low-level features and the complexity of processing, these methods can barely process complex situations, e.g., blurry samples in the IIIT5K-Word.

In recent years, deep learning has achieved outstanding achievements in the field of computer vision, especially in object detection and recognition (Al-Saffar, Tao, & Talab, 2017). Compared to traditional image processing methods that seek feature vectors, deep learning applies deep neural networks to adjust parameters by itself in an end-to-end way when it is training so as to get feature maps from the input images, then identify the output (LeCun, Bengio, & Hinton, 2015). In this thesis, deep learning methods are applied, character locating, character detection, and character recognition of banknote images, which has important research value for banknote recognition. These methods in recent years are mainly categorized by the following two distinctions:

- Most methods utilize networks based on deep learning.
- Most problems are resolved from a variety of perspectives.

These solutions, piloted by the deep learning technologies, utilize the advantages that deep learning can automatically learn by itself instead of design by human and train a deep neural network associated with a huge number of low-level features. Meanwhile, computer scientists from various aspects are enhancing and improving the community into the in-depth investigation, aspiring to multiple goals, e.g., quicker and plainer pipelines (Zhou, et al., 2017), a plurality of aspect ratios, and synthetic data (Gupta, Vedaldi, & Zisserman, 2016). Many research and industrial projects have proven that the combination of deep learning has completely changed the research method for a given task, thereby expanding and deepening previous research work. In short, deep learning researches have flourished in recent years. It is divided into the following four areas:

- Character detection locates the region of characters in a natural picture.
- Character recognition interprets and transforms the content of the localized regions into linguistic symbols.
- Deep learning uses the end-to-end way that operates both characters detection and recognition in one single pipeline (shot).

- Adjuvant methods can be used to assist the main function of characters detection and recognition, e.g., data synthesis and image deblurring, etc.

2.2 Deep Learning

An important branch in machine learning is deep learning (Bharkad, 2013). The idea of deep learning is to create an artificial neural network (ANN) that simulates our human brain to implement functions such as learning, analysis and discrimination (Krogh, 2008). For example, it stimulates our human brain to process sounds, images, and texts. Deep learning algorithms are based on the ANN network (Deng, Hinton, & Kingsbury, 2013). The network is an algorithmic data model for distributed and parallel information processing by imitating the behavior of our neural systems. ANN networks are a very active branch of the field of machine intelligence. Deep learning generates the distributed feature maps of the data using convolutional operations. By creating a deep learning network, a computer can learn from data directly.

The emergence of deep learning is not long, but it develops at an exponential growth rate. In 2006, Hinton et al. published a breakthrough article in deep learning, started the new era of deep learning (Hinton, Osindero, & Teh, 2006). In 2011, Stanford Artificial Intelligence Lab and others used 16,000 computers to simulate a neural network. The network recognized cat faces by learning randomly selected videos. In 2012, Hinton et al. applied deep learning to ImageNet and achieved amazing results. In 2013, Microsoft developed a new speech recognition technology based on deep learning. In 2015, Microsoft Research Asia used residual networks (ResNet) to design neural networks, reconstructed the learning process, and redefined the information flow in deep neural networks. In 2015, the ImageNet Large Scale Visual Recognition Challenge (ILSVRC) was specially designed to evaluate algorithms for object detection and image classification at large scale. In 2016, AlphaGo, developed by Google, defeated the human Go champion in a Go game. In October of the same year, Google announced AlphaGo Zero, which was able to learn from zero without any human input, later it defeated AlphaGo.

2.2 Artificial Neural Network

When it comes to machine learning and cognitive science, Artificial Neural Network (ANN) refers to a network structure that mimics biological neural networks (Bharkad, 2013). It is called a mathematical model that simulates the biological brain system to process complex information. It is actually a complex network made up of simple elements connected to each other. It is highly nonlinear in nature, is capable of performing complex logical operations and implementing a system with nonlinear relationships. The basic operating unit of a neural network is an artificial neuron, similar to a neuron in our human brain, as shown in Figure 2.1.

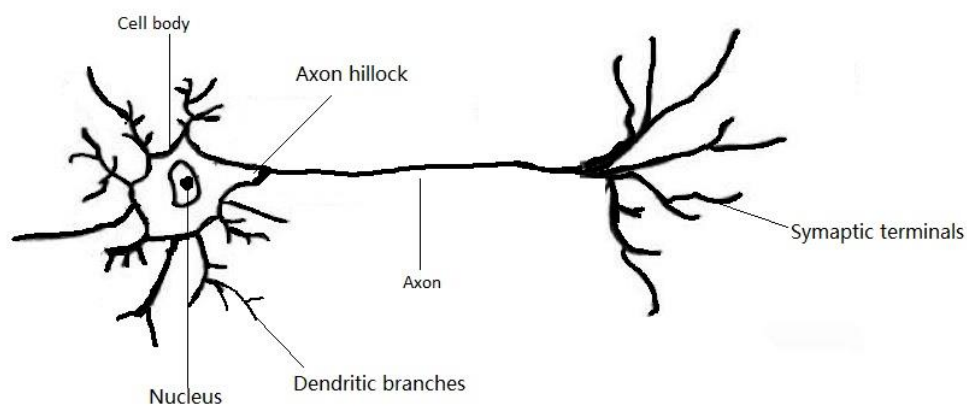


Figure 2.1 Biological neuron

A neuron cell is composed of two parts, dendrites and axons, which represent the input and output of neural signals, accordingly. A general neural network consists of three layers: an input layer, a hidden layer and an output layer.

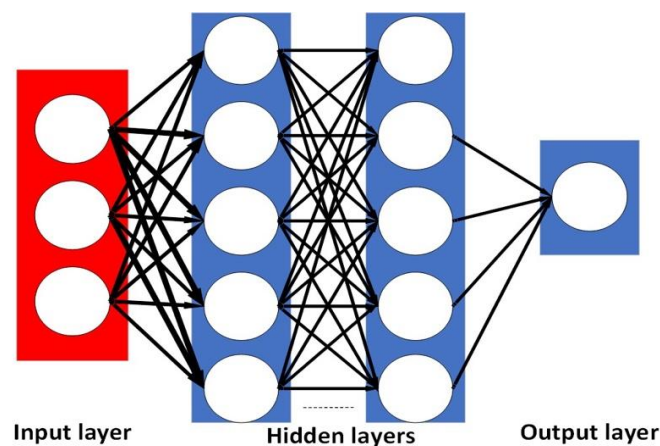


Figure 2.2 Neural network structure

Input layer: At the forefront of the entire network, the input vectors are directly accepted without any processing of the data, usually not counting the number of layers.

Hidden layers: The hidden layers can have one or more layers, deep neural networks have more layers, we call it as depth.

Output layer: The output layer is the last layer of the network which is used to output the value processed by the entire network. This value can be a classification vector, or a continuous value similar to linear regression.

2.3 Convolutional Neural Network

Among various deep network structures, convolutional neural networks (CNNs) are the most broadly employed one. Convolutional neural networks learn by itself the features of images at different aspects by using convolution and pooling operations (Shin et al., 2016), which is similar to the process of human understanding of natural images. The human cognition is understood in a layered abstraction. The first understanding is color and brightness, then the special diagnosis of local details such as edges, corners, and straight lines. The next step is the structural information such as texture and motif, then assemble the part information as an object.

Convolutional neural networks can be taken as a simple imitation of this mechanism (Gu et al., 2018). It is composed of convolutional layers. Each convolutional layer contains multiple convolution kernels. These convolution kernels scan the entire image in sequence from left to right, from top to bottom to obtain output data called feature map. The front convolution layers are devoted to acquiring the local and detailed statics of images, the receptive fields are small, and every pixel of the output image applies only a small part of the input image. The receptive fields of the subsequent convolutional layers increase layer by layer to obtain more intricate and abstractive information of the image; after multiplexing convolutional layer operations, the abstractive representation of the image at different aspects is finally acquired. CNN excels than general applications, especially in image-related tasks (Hui, Huang, Wei, Zhang, & Li, 2015).

2.3.1 The Structure of Convolutional Neural Network

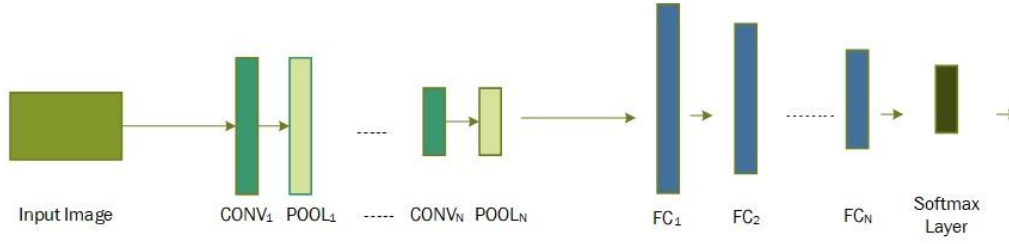


Figure 2.3 The CNN architecture

The structure of the convolutional neural network (CNN) is described in Figure 2.3. A convolutional neural network stands for a hierarchical model that inputs raw data such as audio and images (Yan, Piramuthu, Jagadeesh, Di, & Decoste, 2019). By a series of actions such as convolution, pooling, nonlinear mapping of activation functions, high-level semantics are abstracted or extracted layer by layer.

Convolutional layer: As the main module of the convolutional neural network, the convolutional layer is the core operation object of the convolutional neural network (Jia, et al., 2014). The network mainly uses the convolution operations to extract features from the images, which is also the source of the convolutional neural network. The convolution operation uses a matrix that becomes the convolution kernel from top to bottom, sliding from left to right on the image, multiplying the weight parameters in the convolution kernel with the elements at the corresponding positions that cover on the image, and finally summing to get the pixel value that needs to be output. The equation is shown as follows (Clevert, Unterthiner, & Hochreiter, 2015).

$$x_j^l = f \left(\sum_{i \in M_j} x_i^{l-1} * w_{ij}^l + b_j^l \right) \quad (2.1)$$

where $f(\cdot)$ represents the activation function of the convolution layer, which increases the nonlinearity of the algorithm and enhances the representation capability of the network. x_j^l is the j feature map of the i -th convolution layer; b is the bias parameter. w is the parameter weight in the convolution kernel; M_j is the currently selected feature map. The process is shown in Figure 2.4.

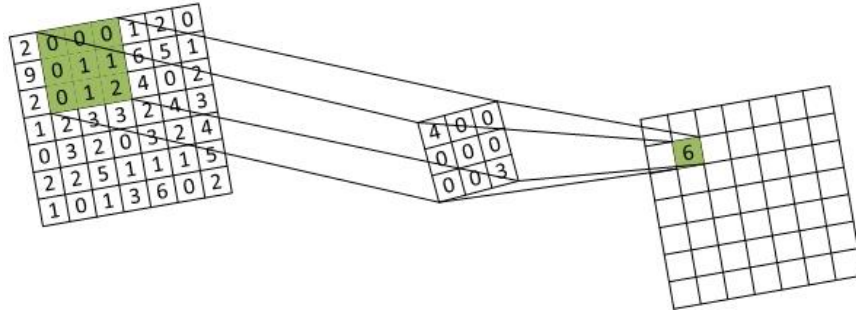


Figure 2.4 The convolution operation

The activation functions. The activation functions are indispensable parts of the network, as the introduction of it increases the nonlinear representation ability of the entire network. The activation functions include sigmoid function (Hun & Moraga, 1995), tanh function (Fan, 2000), and rectified linear activation (ReLU) (Xu, Wang, Chen, & Li, 2015).

$$f(x) = \frac{1}{(1+e^{-x})} \quad (2.1)$$

$$\tanh(x) = \frac{(e^x - e^{-x})}{(e^x + e^{-x})} \quad (2.2)$$

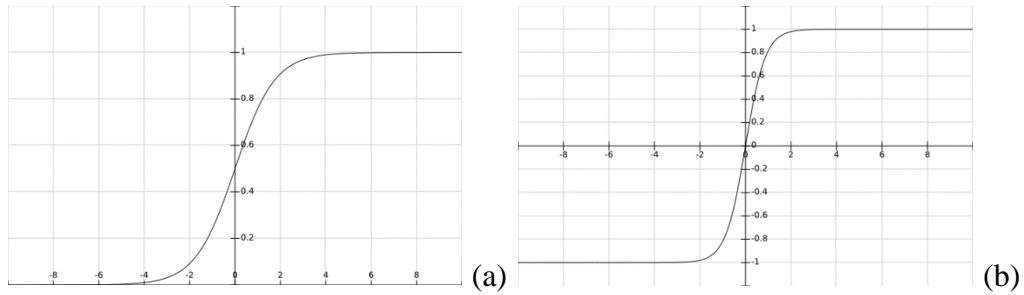


Figure 2.5 The graph of sigmoid(a) and tanh(b)

Sigmoid function is an activation function (Yin, Goudriaan, Lantinga, Vos, & Spiertz, 2003), its advantages are that the output is mapped into the interval (0,1), the function is with monotonicity and continuity, easy to derive, suitable for use as an output layer. What it lacks is that the values are greater than 5.0 or less than -5.0, thus, it will be normalized into (0, 1.0), this results in the “saturation effect” of the gradient, tend to gradient vanishing during backpropagation process.

Tanh function based on sigmoid function is called the hyperbolic tangent function. The range of this function values is fallen in (-1, 1). The mean of the outputs is 0. The convergent

speed is faster than the sigmoid function. However, as Tanh function is a sigmoid function that translates 0.5 downwards on the y-axis, which is a form of sigmoid function, the use of the Tanh function will still have gradient saturation problems.

Rectified linear units (ReLU) function is among the mainstream activation functions in current CNN (Nair & Hinton, 2010). Its expression is $f(x)=\max(0, x)$. A comparison is made between the input value and zero, and the larger value is used as the output of the ReLU function. If the input value is less than zero, the output value is zero; if the input value is greater than zero, then the output value is equal to the input value. Compared with classic activation functions, the ReLU function can speed up the training process of deep neural networks because the derivative of ReLU function is 1.0 when the input is greater than zero.

Deep neural networks do not need to spend extra time when it is being trained. When the number of layers increases, the ReLU function does not cause the gradient to vanish. This is the reason why ReLU function does not have an asymptotic upper and lower bound. Therefore, the front layer (the first hidden layer) can obtain the failings sent by the back layers to revise all the weights among layers. On the contrary, the value of a typical activation function like sigmoid is defined between 0 and 1.0, the errors are tiny for the first hidden layer which will result in a deficiently neural network after training.

Gaussian error linear units (GELUs) function is an activation function for the high-performance neural network (Hendrycks & Gimpel, 2016). The function is expressed as

$$xP(X \leq x) = x\Phi(x) \quad (2.3)$$

where $\Phi(x)$ refers to the cumulative distribution of the Gaussian normal distribution of x , the full form is

$$xP(X \leq x) = x \int_{-\infty}^x \frac{e^{-\frac{(X-\mu)^2}{2\sigma^2}}}{\sqrt{2\pi}\sigma} dX \quad (2.4)$$

GELUs function introduced the idea of random regularity during the activation process. This is a probability description of neuron input, which is intuitive in line with natural understanding. The outcome of our experiment has shown that the manifestation of this function is more superior than ReLU.

Pooling layer. Pooling is the average or maximum value of the convolution results (Scherer, Müller, & Behnke, 2010). It is used to progressively decrease the dimensional size of the representation so as to decrease the number of parameters, reduce the calculated cost of the network. Two methods are widely used for pooling operation: average pooling and max pooling.

Pooling and convolution operations are different. The pooling operation does not include the network parameters that the algorithm needs to learn. The only thing needs to be done is to specify the type of pooling operation, the size of the filter scan window, the stride value, and whether padding is needed. The pooling operation is shown in the figure, the kernel size is 2×2 , and the step size is 2.

Both average pooling and max pooling can complete the downsampling operation. The former is a linear function, while the latter is a nonlinear function. In general, max pooling is better.

Fully connected layer. In a convolutional neural network, the fully connected layers are used as a classifier (Andrearczyk & Whelan, 2016). The convolutional layer, pooling layer, and activation function layer are used to transform the primitive data to the feature space of hidden layer, while the fully connected layer is to map the learned features to the sample label space. The equation for the output of the fully connected layer is shown as eq. (2.5). $f(\bullet)$ stands for the activation function of the fully connected layer, with ‘e’ presents the e -th neural unit of the layer

$$y_e = f(x_e) = f\left(\sum_{i \in M_j} x_i^j * w_j + b_j\right). \quad (2.5)$$

2.3.2 Convolutional Neural Network

2.3.2.1. AlexNet

AlexNet is thought as the originator of deep convolutional neural networks (Alom, et al., 2018). Compared with the previous convolutional neural network, it has deeper layers and larger parameter scales.

AlexNet consists of five kinds of layers, following a max pooling layer for downsampling, and three fully connected layers. The last layer is the softmax output layer with a total of 1000

nodes, corresponding to 1000 image classifications of the ImageNet dataset. Part of the convolutional layer is divided into two groups for independent calculations, which is conducive to GPU parallelism and reduces the number of calculations.

ReLU function is the activation function for AlexNet and the dropout technique is applied. AlexNet regulates the output of each hidden neuron to 0 and the probability is 0.5, in this way, ensuring that the suppressed neurons do not take part in forward and backward propagation. Dropout mitigates overfitting through regularization.

2.3.2.2.VGG

VGG was developed based on AlexNet and put forward by the visual geometry group Oxford University in 2014 (Simonyon & Zisserman, 2014). VGG applies the same size convolution kernel (3×3), the convolutional stride is set to 1, the size of the pooling layer is 2×2 ; then three fully connected layers are added and the two of these layers use ReLU as activation function (Senguta, Ye, Wang, Liu, & Roy, 2019). The network structure of VGG with different depths is listed in Figure 2-6.

A	B	C	D	E
11 weight layers	13 weight layers	16 weight layers	16 weight layers	19 weight layers
input image				
conv(3,64)	conv(3,64)	conv(3,64)	conv(3,64)	conv(3,64)
	conv(3,64)	conv(3,64)	conv(3,64)	conv(3,64)
Maxpooling				
conv(3,128)	conv(3,128)	conv(3,128)	conv(3,128)	conv(3,128)
	conv(3,128)	conv(3,128)	conv(3,128)	conv(3,128)
Maxpooling				
conv(3,256)	conv(3,256)	conv(3,256)	conv(3,256)	conv(3,256)
conv(3,256)	conv(3,256)	conv(3,256)	conv(3,256)	conv(3,256)
		conv(3,256)	conv(3,256)	conv(3,256)
				conv(3,256)
Maxpooling				
conv(3,512)	conv(3,512)	conv(3,512)	conv(3,512)	conv(3,512)
conv(3,512)	conv(3,512)	conv(3,512)	conv(3,512)	conv(3,512)
		conv(3,512)	conv(3,512)	conv(3,512)
				conv(3,512)
Maxpooling				
conv(3,512)	conv(3,512)	conv(3,512)	conv(3,512)	conv(3,512)
conv(3,512)	conv(3,512)	conv(3,512)	conv(3,512)	conv(3,512)
		conv(3,512)	conv(3,512)	conv(3,512)
				conv(3,512)
Maxpooling				
FC(4096)				
FC(4096)				
FC(1000)				
Softmax				

Figure 2.6 The structure of VGG network with different depths

2.3.2.3. GoogLeNet

Google network first appeared in the ILSVRC2014 competition and won the championship with a great advantage. GoogLeNet introduces a new module Inception, thus GoogLeNet is also called Inception v1. GoogLeNet uses Inception instead of manual to select the convolution type and then stacks inception blocks to form an Inception network (Szegdy, et al., 2015). The network removes the fully connected layer and uses global mean pooling, which effectively decreases the number of parameters. The Inception module firstly uses a 1×1 convolution kernel to perform convolution operations for data dimensionality reduction. Then it utilizes convolution kernels of size 1×1 , 3×3 , and 5×5 , respectively to perform convolution operations and generate feature maps with various sizes. Convolution kernels of different sizes have different receptive fields when conducting convolution operations on images. Therefore, by

using convolution kernels with different sizes for convolutions, the ability to model different scale features can be improved significantly.

2.3.2.4. ResNet

ResNet won the championship in the 2015 ILSVRC competition in two projects: Object detection and object recognition which do not rely on external data. It trains the network having 152 layers through the residual module and solves the gradient degradation problem. After using the residuals, as the network grows deeper, the error rate decreases further. The main novelty of ResNet is residuals. The residual network can be thought as a special example in an expressway. The carrying gates and transform gates in the expressway network are all identity maps. The residual module consists of two branches, the left one is the residual function, and the right one is the input identity map (Szegedy, Ioffe, Vanhoucke, & Alemi, 2017). After these two parts are simply combined (addition of relevant elements), they are then subjected to a nonlinear ReLU activation function to generate the whole residual learning block. A network structure by stacking multiple residual modules together construct a "residual network".

2.4 Text Detection

Since 2006, under the leadership of Hinton, a multitude of researches have been done on deep convolutional networks. In 2012, Hilton and his team won the championship in the ImageNet image recognition competition. From then on, neural networks have been receiving widespread attention. Object detection means to locate the region of interest from an image or video, and output the position and class of the detected visual object at the same time. It is among the core topics when it comes to the computer vision field. Object detection has been investigated for nearly two decades. With the rapid growth of deep learning technology, visual object detection algorithms have also shifted from traditional machine algorithms based on small datasets to deep neural network based on big datasets (Kim, 2014).

There are many uncertain impacts on object detection. For example, the number of objects in one single image is uncertain, the objects have various shapes, visual appearances, or motions. In addition, when the object is captured, there will have interferences from multiple aspects such as light and occlusion, which will make the detection difficulty. The current deep learning methods in the field of object detection are mainly divided into two parts: One-stage detection algorithms and two-stage detection algorithms. One-stage algorithms do not have

region proposals, but straightly produce the class probability and position coordinates of the object. Representative algorithms include YOLO, SSD, and CornerNet. Two-stage algorithms split the detection procedure into two stages: The first stage is to generate candidate regions. This step can be set manually or learned automatically by a neural network. The following step is to categorize the generated candidate regions and regress the positions of the candidate regions to finally obtain an accurate region. Two-stage algorithms include R-CNN, SPP-Net, Fast R-CNN, and Faster R-CNN.

The main indicators for measuring the object detection algorithms are the accuracy and speed of detection. Accuracy refers to the pattern classification accuracy. In general, when the accuracy takes priority, two-stage algorithms are employed; when the executing speed is considered, one-stage algorithms are adopted. However, with the fast growth of deep learning technology, both kinds of algorithms are continuously improved, and both can achieve excellent results in terms of accuracy and speed.

The one-stage object detection algorithms can straightly produce the class probability and position coordinates of the object in one step. In comparison with two-stage algorithms, there are not region proposals, and the overall process is simpler. Image-decoding generates corresponding detection frame; the training process encodes the ground truth into the format corresponding to the CNN outputs so as to get the loss.

The two-stage object detection algorithms can be regarded as operating one-stage detection twice. The overall process is input images first and generates ROIs; classify ROIs into specific categories and more accurately locate (regress); decode and generate corresponding detection frames; during training, the ground truth is encoded into CNN output format corresponding to calculate the corresponding loss (Girshick, 2015).

2.4.1 SSD

An object detection algorithm based on Single Shot MulutiBox Detector (SSD) was proposed (Wei Liu et al. ,2016). In comparison with the classic Faster R-CNN and YOLO networks, this network has better speed and accuracy. SSD500 achieved 75.1% accuracy and 23fps speed based on Pascal VOC2007 dataset. The SSD network abandons the step of extracting candidate regions and extracts candidate networks on feature maps with various scales. SSD network mainly includes two parts: A pre-trained network and multiscale convolutional neural network.

A multiscale convolutional neural network is mainly used to extract the multiscale features needed for object classification and position regression.

The advantages of the SSD algorithm include: Its running speed is comparable to YOLO, and its detection accuracy is comparable to Faster R-CNN.

The disadvantages of SSD mainly include: The priority box is needed to be set manually. The fundamental dimension and proportion of the priority box in the network cannot be directly acquired during the learning process but need to be fixed by human. Each layer of features in the network use the same priority boxes with the same dimension and proportion, resulting in the training process being very dependent on personal experience. In spite of the fact that the idea of pyramid hierarchy is employed, the detection performance of small objects is average.

2.4.2 SegLink

For most text detection algorithms, it is difficult to detect long texts, or it is more difficult to detect the entire line of text at one time. In response to this issue, SegLink put forward a new idea based on SSD but the position of the text area is not returned through the rectangular box.

The approach of the SegLink algorithm is: Cut each word into small directional segments that are easier to be detected, and then link adjacent small text blocks into words using adjacent connections. That is, the network outputs two types of information:

One is segmented, which may be a character or several characters, etc. It is not a box of the entire text line, but a section of the text line. This information is angled.

The other is the link information between different segments, and this link is also automatically learned in the network. The network determines which segments are parts of the line of text.

After predicting segments and links, it is needed to combine the predicted segments or call them connected.

The SegLink algorithm follows the steps: Filter the segments and links predicted by the network through α and β (these two values are found by grid search). Treat each segment as a node, the link is regarded as an edge, a graph model is established, and DFS (depth-first search)

is used to locate linked components. Each connected component consists of a series of segments. Finally, output the text box after connecting the segments.

The SegLink backbone network follows the SSD network, but the last pooling layer is modified to a convolutional layer. Specifically, the network uses VGG16 as the base net, and changes the last two fully connected layers of VGG16 to convolutional layers. To extract deeper features, additional convolutional layers were added. The last pooling layer of SSD was changed to a convolutional layer. SegLink extracts feature maps of different layers and generate the final output by using a 3×3 convolution kernel to these layers. Finally, the box information and link information of the segment are fused to obtain the final text line (B. Shi, Bai, & Belongie, 2017).

2.4.3 DenseNet

As the Best Paper of CVPR 2017, DenseNet has been proposed from the idea of deepening the network layer (ResNet) and broaden the network structure (Inception) to promote network performance. From the aspect of features, through feature reuse and bypass settings, DenseNet not only greatly reduces the number of network parameters, but also resolve the gradient vanishing problem (G. Huan, Li, Van der Maten, & Weinberger, 2018). Combined the assumptions of information flow and feature reuse together, DenseNet deserves to be the best paper of the 2017 computer vision summit.

Convolutional neural networks have slept for nearly 20 years, and now they have become one of the most significant networks in deep learning fields. From the beginning of LeNet, which has only a five-layer structure to VGG, which has a 19-layer structure, to the first 100-layer network ResNet, deepening network layers has become one of the main directions of CNN development. With the continuous expansion of the amount of CNN network layers, the problems of gradient vanishing and model degradation have appeared (Zhang, Tian, Kong, Zhong, & Fu, 2018).

DenseNet, as another type of CNN with a deeper structure, has the following unique advantages:

- Compared with ResNet, it has a smaller number of parameters.
- Bypass enhances feature reuse.

- DenseNet is easy to train and has an excellent performance.
- Mitigated the problems of gradient vanishing and model degradation.

2.5 Text Recognition

Text recognition is one of the main research directions in computer vision (CV). For text recognition, generally, the text is firstly located in the image by using text detection (Shi, Wang, Lyu, Yao, & Bai, 2016). Based on the features of the region that are extracted, special character recognition is performed. However, with the continuous development of this field, many new end-to-end OCR technologies have appeared. The text located at the area of the image, and the accuracy of detection directly affects recognition results.



Figure 2.7 The results with object locating

The objective of detecting an object is to locate the ROI in a given image, the positioning accuracy directly affects subsequent recognition results. As listed in Figure 2.7, the red rectangle stands for the ground truth (GT), and the green rectangle represents the detection box. When the GT and the detection box have the same IoU, the recognition results will be perfect. Therefore, object locating has a significant impact on the following object recognition. For recognizing the detected objects, Convolutional Recurrent Neural Network (CRNN) or Seq2Seq algorithms are generally used (Sheng, Chen, & Xu, 2019).

2.5.1 CRNN

In the field of computer vision, a large number of image-based sequence recognition research topics have been carried out for a long time (Lee & Osindero, 2016). CRNN targeted to resolve

the issue of scene text recognition, which is one of the most significant and inspiring problems in image-based sequence recognition. CRNN proposed a new neural network that synthesizes feature extraction, sequence modelling and transcription into a unitary framework. Compared with other text recognition network, the proposed architecture has four different characteristics.

Compared to most existing algorithms which require separate training, CRNN model is an end-to-end network. It naturally deals with sequences of arbitrary length and does not involve segmenting character or normalizing horizontal scale. It is not limited to any predefined vocabulary but still has achieved outstanding performance in dictionary-free and dictionary-based text recognition. CRNN produces an efficacious and much smaller model, which is more practical for real-world applications. The experiments based on standard datasets including IIIT-5K(Mishra, Alahari, & Jawahar, 2012), Street View Text (K. Wang, Babenko, & Belongie, 2011), and ICDAR datasets (Karatzas et al., 2015) prove that the proposed algorithm has advantages over existing technologies.

2.5.2 Attention Model

Attention model has been widely used in many different fields of deep learning in recent years including image processing, speech recognition, and natural language processing. Therefore, understanding the workflow of the attention model is very necessary for the developing of deep learning applications. The attention model in deep learning is similar to the selective visual attention of human beings, in essence, the core goal is to select the information that is much critical to the target from current information (Vasweni, et al., 2017).

In short, the attention model is a method of extracting specific vectors from vector expression sets for weighted combination according to the rules. In a simple case, as long as we perform weighted summing from partial vectors, then attention is applied.

Currently, attention has been widely applied, including soft attention(Kumar, Sangwan, Arora, Nayyar, & Abdel-Basset, 2019), global attention, dynamic attention, self-attention, key-value attention, multi-head attention (Tao et al., 2018), and 2D attention (Elbayad, Besacier, & Verbeek, 2018).

2.6 Data Augmentation

Although a two-layer network can theoretically fit all distributions, it is not easy to be trained. In practice, the depth and breadth of the neural network thus are usually increased, so that the learning ability of the neural network is enhanced and fit to the distribution of training data. In convolutional neural networks, the relevant experiments have shown that depth is more important than breadth (Montufar, Pascanu, Cho, & Bengio, 2014).

However, as neural networks have been deepened, the number of parameters need to be calculated will be increased, which will more easily lead to overfitting (Hawkins & sciences, 2004). When the dataset is small, too many parameters will fit all the dataset, rather than the commonality between the data. Overfitting means that the neural network can highly fit the distribution of the data that trained, but it has a low accuracy rate for the test data and lacks generalization capabilities (Cawley & Talbot, 2010).

Therefore, in this case, in order to prevent the network from overfitting, the methods such as data augmentation, regular terms, and dropout are often used which are turned out to be very effective. Common data augmentation methods include random rotation, random cropping, color dithering, Gaussian noise, horizontal flip, vertical flip, random erasing, etc. (Zhong, Zheng, Kang, Li, & Yang, 2017).

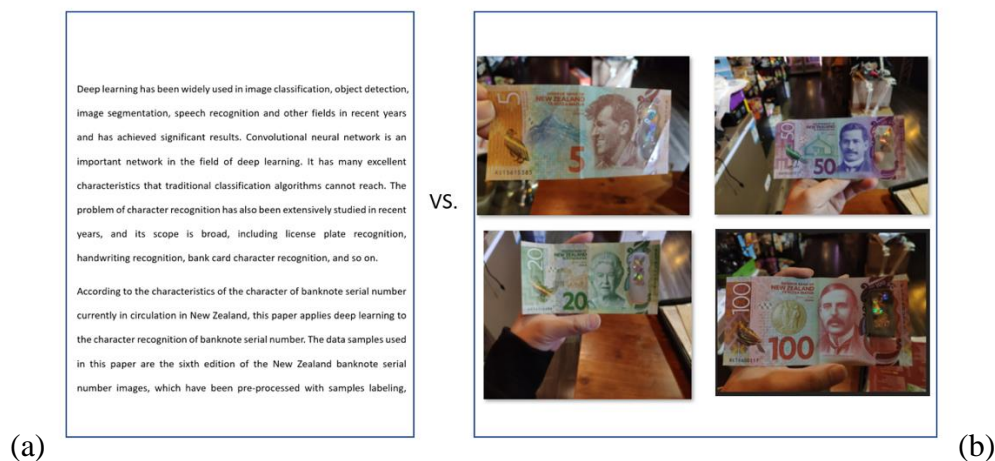
Chapter 3

Methodology

This chapter mainly describes the research methods of banknote serial number recognition by describing the details of banknote serial number region detection and text recognition. Moreover, this chapter also details the evaluation methods applied in this thesis.

3.1 Introduction

There are many challenges in banknote serial number detection and recognition. First of all, it is totally different from the traditional Optical Character Recognition (OCR) (Darshan, Gopalkrishna, & Hanumantharaju, 2015), because the text in natural scenes has been distorted a lot, as shown in Figure 3.1, Figure 3.1(a) is a typical scanned document, while Figure 3.1(b) is a collection of natural scene images.



- Clean background vs. cluttered background
- Regular font vs. various fonts
- Plain layout vs. complex layouts
- Monotone color vs. different colors

Figure 3.1 Challenges of text detection and recognition

Through comparisons, we find that:

- The background in Figure 3.1(a) is very clean, while the contrary part in Figure 3.1(b) is very messy;
- The font in Figure 3.1(a) is very regular, while in Figure 3.1(b), it is quite changing;
- The layout in Figure 3.1(a) is flatter and more unified, while that in Figure (b) is diverse, complex, and lacking specifications;
- The colors in Figure 3.1(a) are monotonous, while the show in Figure 3.1(b) are diverse.

In general, there are three major challenges in text detection and recognition:

- Diversity of scene text, such as text colour, size, orientation, language, font, etc.



Figure 3.2 Diversity of banknote texts

- Variances of the image background, e.g., signals, indicators, fences, roofs, windows, bricks, flowers, etc., which are everywhere in daily life, have the similarities with the texts, which has greatly disrupted in the text detection and recognition process. Elements like bricks, circles, and dots are virtually indistinguishable from the true text.



Figure 3.3 Complexity of background of banknote

- The third challenge comes from the imaging process of the image itself. For example, taking pictures containing text has the problems such as noises, blurry, nonuniform lighting (high reflection, shadows), low resolution, and partial occlusion are also very big challenges to the detection and recognition algorithms.



Figure 3.4 Noises and blurring of banknote serial number

Because of the multiple challenges, in this thesis, we address them from a variety of aspects. It mainly includes:

- Drawing inspiration from semantic segmentation and object detection methods.
- Simplifying the pipeline.
- Processing oblique texts with an angle.
- Using the attention model.

The identification of the banknote serial number in this research is mainly divided into two parts: the region detection for the serial number and the serial number recognition.

The first step is to collect and preprocess the samples. We take photos of banknotes and make sure that the banknotes are flat enough. Then, we label the samples, which include the label of each serial number. Finally, the dataset is augmented through data preprocessing methods as the enhancement method for the robustness of the model.

Thus, the network is built. The basic network for the detection uses DenseNet as the backbone and the SegLink is adopted. Thus, we extract features at multiple scales in different layers of the network. During training, the focal loss is used as the loss function which improves the accuracy. At last, the detected text region is sent to the deep neural network for text recognition and output the result.

Finally, the performances of different networks are compared. For the detection, the algorithms mainly include SegLink, SegLink + Inception V4+ ResNet, SegLink + DenseNet. for recognition part, CNN + RNN + CTC Loss network, CNN + RNN + Attention Model are used.

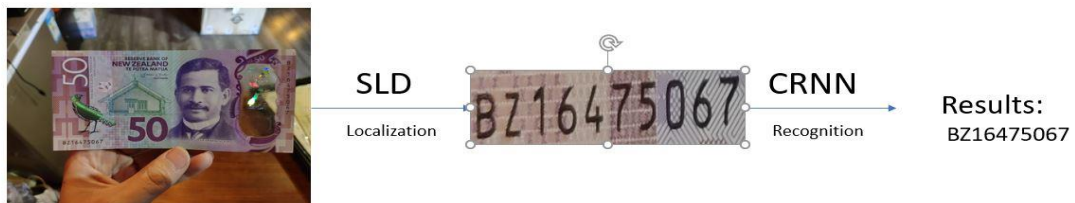


Figure 3.5 The pipeline of the end-to-end network for the recognition of banknote serial numbers

3.2 Research Design

The goal of this experiment is to identify currency serial numbers. To achieve this goal, the experimental process design is shown in the figure below.

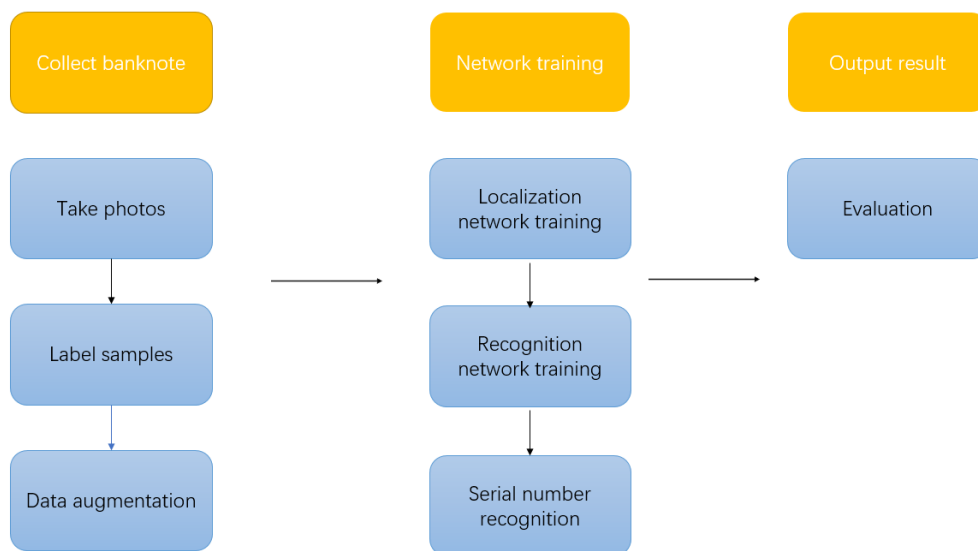


Figure 3.6 The research design

3.3 Data Pre-processing

3.3.1 Data Collection

There are 1,000 samples collected for this experiment, including the NZ banknotes issued in 2016 with denominations \$20, \$50, and \$100. Each banknote contains a string of 2 letters and 8 digits. The serial numbers of different denominations have been positioned with different backgrounds. The images of each denomination are shown in Figure 3.7.




Years Issued	Denominations	Samples	Quantity
2016	20		374
2016	50		338
2016	100		288

Figure 3.7 The images of banknotes with different denominations

The banknotes were taken by using our mobile camera for data collection. In order to get high-quality images, the entire banknote was photographed from a top-down or parallel angle. All banknotes are not taken out by banks or ATMs but collected in a casual way to avoid the consecutive serial numbers, which ensures the randomness of the letters and numbers, and reduces the hints. The image data is shown in Figure 3.8.

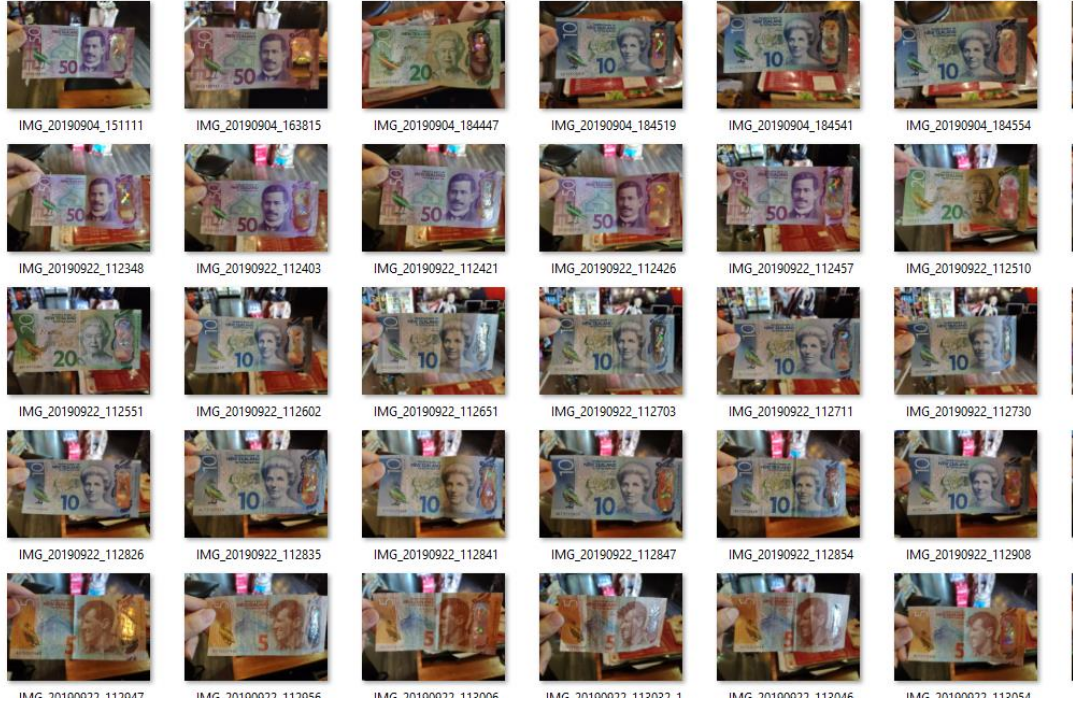


Figure 3.8 Banknote samples for our experiments

3.3.2 Data Labelling

Serial number detection and recognition both followed the supervised learning (Caruana & Niculescu-Mizil, 2006), the algorithm needs to know exactly what is in the graph and where these objects are. This study applied the open-source image annotation tool LabelImg to generate an XML file. The first step is to use LabelImg to label the serial number area of the banknote.

Then, we label the numbers and letters in the region of serial numbers. The serial number contains a total of 26 letters and 10 numbers. The annotation effect is shown in the following Fig. 3.10.

The generated .xml code is detailed. Among them, under the `<size>` tag, `<width> 4032 </width>` indicates that the width of the picture is 4032, `<height> 3024</ height>` indicates that the height is 3024, and `<depth> 3 </ depth>` indicates that the image is colored, and the number of channel is 3. The labeled detection information is indicated under the `<object>` tag which mainly includes: `<name> w </ name>`, which indicates the class of the object, `<difficult> 0 </ difficult>` whether the object is a different sample, and `<bndbox >` indicates the location information of the object. `<xmin> 1265 </ xmin>` means the smallest x-axis coordinate is 1265,

<ymin> 2216 </ ymin> means the smallest y-axis coordinate is 2216, and <xmax> 1304 </ xmax> means the largest x-axis coordinate is 1304 , <Ymax> 2273 </ ymax> means that the maximum y-axis coordinate is 2273. According to the above four coordinates, the position of the object can be determined.



Figure 3.9 Region labelling for serial number detection

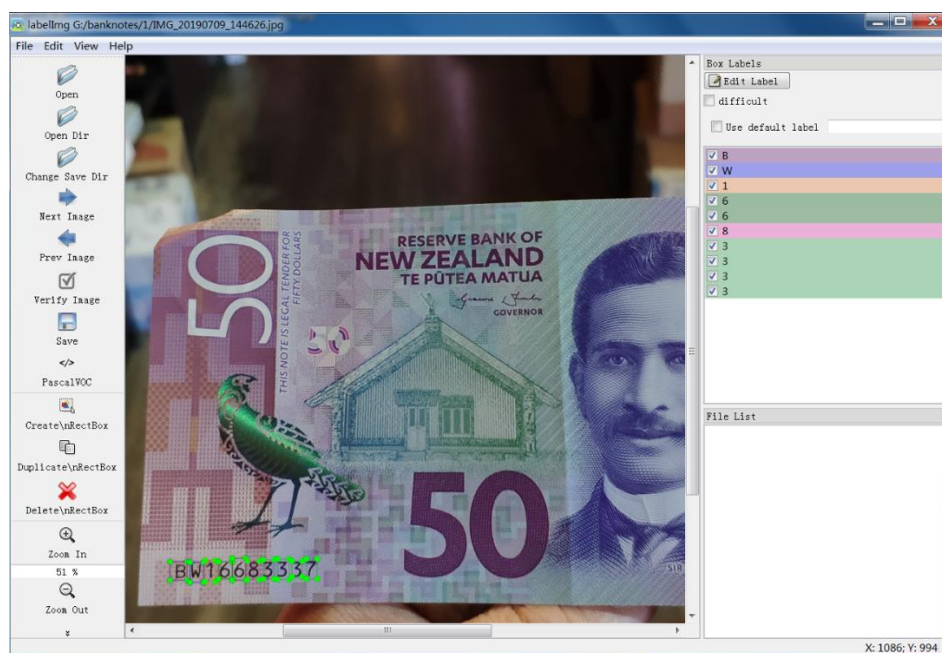


Figure 3.10 Labelling for serial number recognition

3.3.3 Data Augmentation

In order to increase the number of training data, data augmentation method is employed to augment the dataset. The main approaches are rotation, translation, color jittering, and adding Gaussian noises. Our data enhancement is mainly achieved through the ImageDataGenerator function of Keras.

Rotations. The images are randomly rotated by an angle and the orientation of the image are changed.

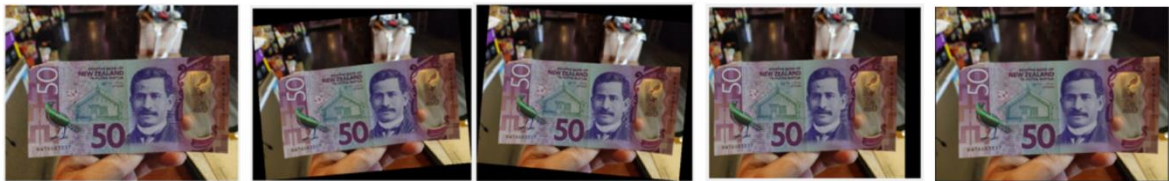
Translations. We translate the image, either in a random or specified way according to the specified step horizontally or vertically and locate the image region.

Gaussian noises. Overfitting usually takes place when neural networks learn potentially useless high-frequency features (a large number of patterns). Gaussian noises with zero means have data points in basically distributed in all frequencies, which can effectively distort high-frequency characteristics. It also means that lower frequency components (usually expected data) will also be distorted, but the neural networks can avoid or surpass it. Adding the right number of noises can strengthen the learning ability of networks and reduce overfitting.

In this research, the number of samples is increased to 5,000 by rotating every 5 degrees clockwise of images and every 5 degrees counterclockwise, randomly shifting 300 pixels and adding Gaussian noises.



(a) Our data augmentation on \$100 (NZD)



(b) Our data augmentation on \$50 (NZD)



(c) Our data augmentation on \$20 (NZD)

Figure 3.11 The samples of our data augmentation for the banknotes \$20, \$50 and \$100 (NZD)

3.3.4 Training and Test Datasets

Deep neural networks demand a large number of samples as the training dataset and testing dataset. The training dataset is used for model learning features, and the testing dataset is employed to assess the generalization ability of the model (Storkey, 2009). In the case of limited or insufficient data, we need to consider choosing multiple methods to generate the datasets. The evaluation method is the cross validation (Krogh & Vedelsby, 1995), which mainly includes holdout cross-validation, leave out cross-validation, and k -fold cross-validation.

The hold-out cross-validation refers to statically divide the dataset into a training set, a validation set, and a test set according to a fixed ratio (Yadav & Shukla, 2016).

The leave-out cross-validation indicates that each test set has only one sample, and m times training and prediction are performed (Yadav & Shukla, 2016). This method uses only one sample instead of the overall dataset, so it is closest to the original sample distribution. But the training complexity increases because the number of models is as same as the number of original data samples. It is used when data is scarce.

Because the static leave-out cross-validation method is more sensitive to the data division, different division methods may lead to different models. In order to weaken this effect aroused by the static stroke method, the K-fold cross-validation approach adopts a dynamic verification method. The specific steps are:

- Divide the data set into a training set and a test set, and set the test set aside.
- Divide the training set into k shares.

- Use one of the k copies at one time as the validation dataset and the rest as the training set.
- After k times training, we get k different models.
- Evaluate the performance of k models and select the best hyperparameters.
- Use the optimal hyperparameters, and retrain the model using all k datasets as the training set to get the final model. In our experiments, K-fold cross-validation is applied with $k=10$.

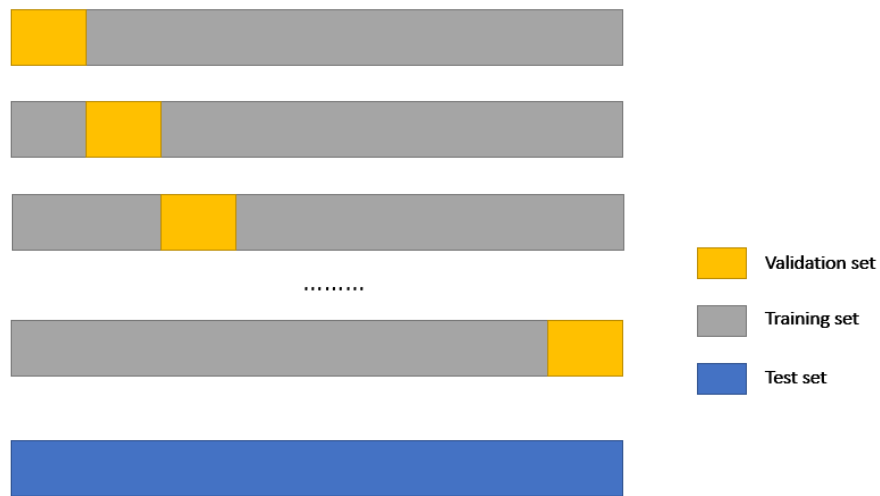


Figure 3.12 K-fold cross-validation method

3.4 Architecture Design

3.4.1 Detecting the Region with Banknote Serial Numbers

In the serial number detection, we adopt DenseNet to construct the backbone network and use the multiscale fusion strategy of SegLink in the later layers of the network. ResNet and Inception v4 have been set up as a backbone network for the purpose of comparisons. DenseNet uses a feedforward method to connect each layer with all other layers, and there are $L(L+1)/2$ direct connections in the L layer network. DenseNet has several advantages, it can reduce the gradients vanishing, enhance backpropagation, improve the reuse of feature maps, and substantially cut down the number of parameters.

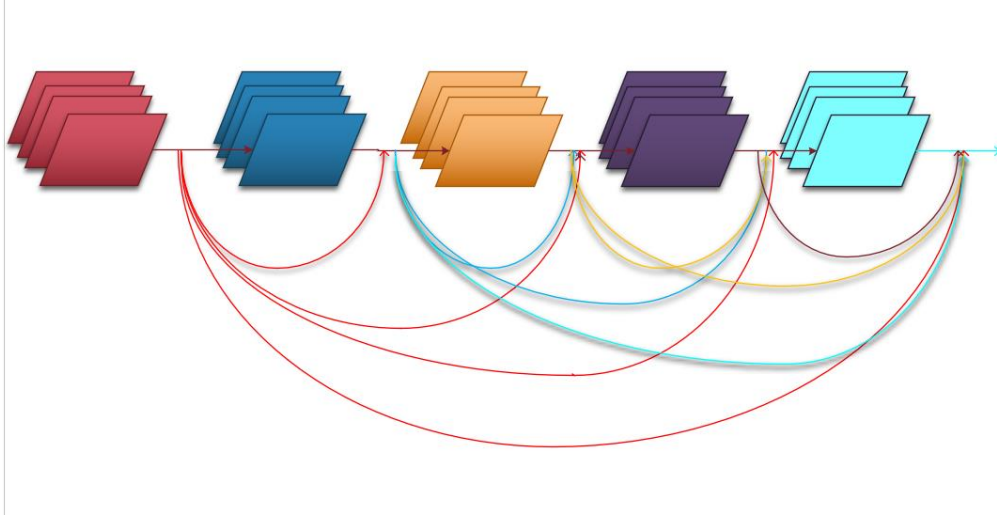


Figure 3.13 A 5-layer DenseNet block

Layer 1 acquires the feature maps of all previous layers, as the input, $x_1 = H_1(x_0, x_1, \dots, x_{l-1})$. $(x_0, x_1, \dots, x_{l-1})$ represents the connection (merging) of feature maps generated from the 0-th layer to the 1-st layer. A large number of connections does not increase hyperparameters.

$H_l(\bullet)$ consists of three parts: batch normalization (BN) layer, ReLU layer, and a 3×3 convolutional layer. The layers between blocks are defined as transition layers, which include a BN layer, a 1×1 convolutional layer, and a 2×2 average pooling layer.

If the function $H_l(\bullet)$ generates k feature maps, then one layer has $(k_0 + k_1 + \dots + k_{l-1})$ feature maps as input, where k_0 is the number of channels of the input layer. The hyperparameter k is the growth rate of the network. Each layer put in its k feature maps to the global state of the network. The growth rate shows the amount of new information of each layer contributes to the global state of the network.

In this research project, the network architecture adopted for positioning banknote serial numbers is shown in Table 3.1.

Table 3.1 DenseNet architecture

Layers	Output Size	Parameters
Convolution	112×112	7×7 conv, stride =2
Pooling	56×56	3×3 max pool, stride=2
Dense Block 1	56×56	$\left\{ \begin{array}{l} 1 \times 1 \text{ conv} \\ 3 \times 3 \text{ conv} \end{array} \right\} \times 6$
Transition Layer 1	56×56	1×1 conv
	28×28	2×2 average pool, stride=2
Dense Block 2	28×28	$\left\{ \begin{array}{l} 1 \times 1 \text{ conv} \\ 3 \times 3 \text{ conv} \end{array} \right\} \times 12$
Transition Layer 2	28×28	1×1 conv, stride=2
	14×14	2×2 average pool, stride=2
Dense Block 3	14×14	$\left\{ \begin{array}{l} 1 \times 1 \text{ conv} \\ 3 \times 3 \text{ conv} \end{array} \right\} \times 24$
Transition Layer 3	14×14	1×1 conv, stride=2
	7×7	2×2 average pool, stride=2
Dense Block 3	7×7	$\left\{ \begin{array}{l} 1 \times 1 \text{ conv} \\ 3 \times 3 \text{ conv} \end{array} \right\} \times 16$

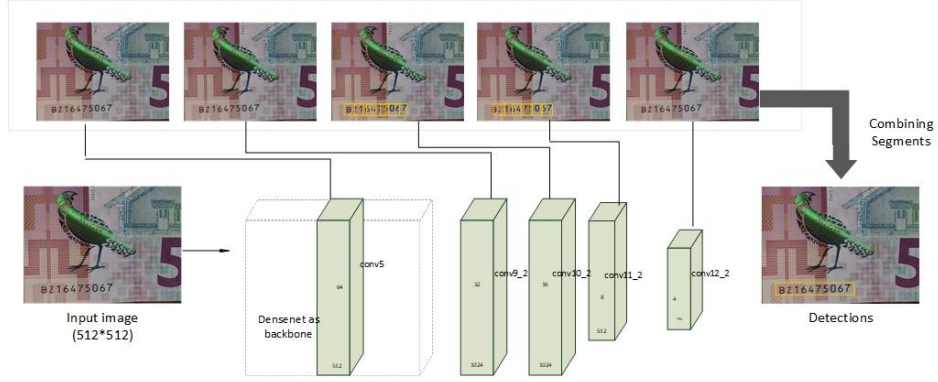


Figure 3.14 The network architecture for locating banknote serial numbers

This network includes two parts to achieve the detection: segmentation and link detection. The expression is

$$S = (x_s, y_s, w_s, h_s, \theta_s). \quad (3.1)$$

Compared to SSD network, the segment contains extra angle information. The SSD network uses default boxes with different ratios of 1, 2, 3, 1/2, and 1/3, and SegLink algorithm uses only one default box with an aspect *ratio* = 1 in each position of each feature map that is the reason why this algorithm has a fast detection speed. Regarding the scale size of the default box, the SSD network sets it manually, while SegLink algorithm sets it according to the receptive field of the current layer.

$$\alpha_1 = \gamma \frac{w_l}{w_l} \quad (3.2)$$

where $\gamma = 1.5$, w_l stands for the width of the input image, w_l stands for the width of the current feature map. The calculation of the segment is as:

$$x_s = \alpha_1 \Delta x_s + x_a \quad (3.3)$$

$$y_s = \alpha_1 \Delta y_s + y_a \quad (3.4)$$

$$w_s = \alpha_l \exp(\Delta w_s) \quad (3.5)$$

$$h_s = \alpha_l \exp(\Delta h_s) \quad (3.6)$$

$$\theta_s = (\Delta \theta_s) \quad (3.7)$$

Link detection is mainly used to connect the segments, it is divided into within-layer link detection and cross-layer link detection.

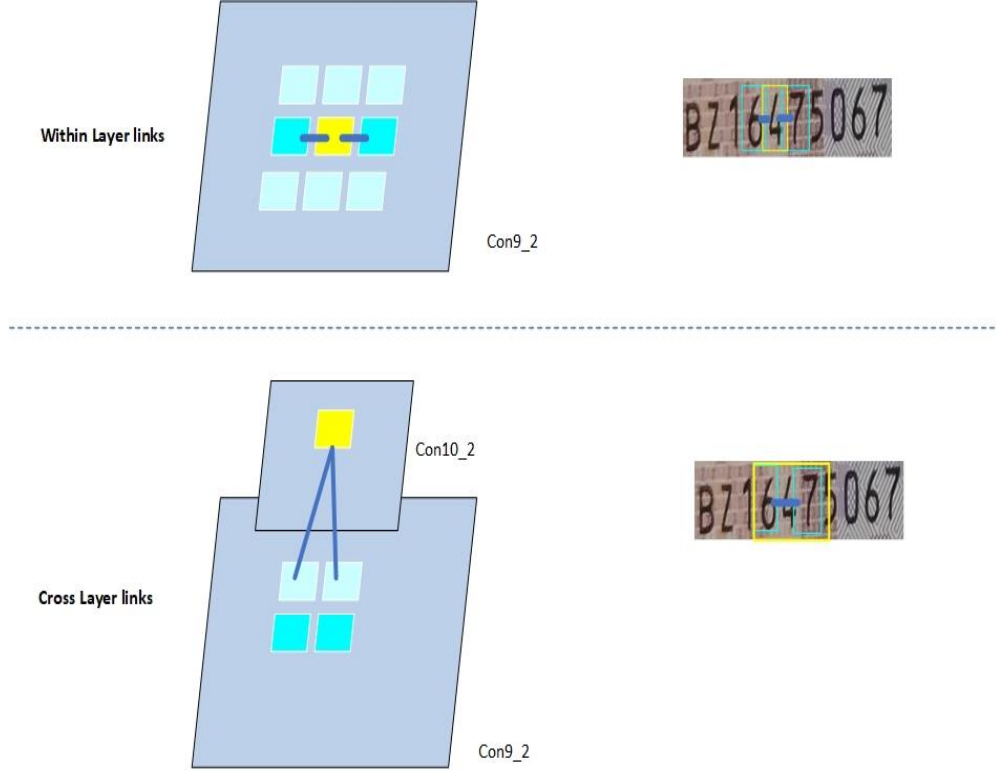


Figure 3.15 Within layer links and cross-layer links

In the same feature map, only one segment is predicted for each position in the feature map. To calculate the link within the layer, we need to consider the eight neighbourhoods of the current segment, that is used to determine each connectivity of the segment with the eight segments surrounding it. Each link has two scores, one for a positive score and one for a negative score. A positive score indicates whether the two belong to the same word; a negative score indicates whether the two belong to different words and should be disconnected. Therefore, the link of each segment should be a 16-dimensional vector ($8 \times 2 = 16$). The specific is shown in equation (3.8):

$$\mathcal{N}_{s^{(x,y,l)}}^w = \{s^{(x',y',l)}\}_{x-1 \leq x' \leq x+1, y-1 \leq y' \leq y+1} \setminus s^{(x,y,l)} \quad (3.8)$$

Since the segments may be detected through multiple feature maps, in order to solve the redundancy issue of this repeated detection, the algorithm uses a method based on the cross-layer link. It is mainly used to detect two consecutive layers of links (the previous layer is the neighbour of the next layer, but the next layer is not the neighbour of the previous layer), so only the cross-links of full or parts of the layers 8, 9, 10, 11, 11, and 12 are detected.

To combine the segments with links, we first manually filter the segments and links predicted by the network through α and β (these two values are found by grid search). Second, consider each segment as a node and the link as an edge to create a graph model, use a depth-first search (DFS) to find connected components. Each connected component includes a series of segments (represented by B). The algorithm that outputs the word box is described as algorithm 3.1.

Algorithm 3.1 Combining segments

Step 1: Input $\beta = (\{s^i\})_{i=1}^{|\beta|}$ is a set of segments connected by links, where

$$s^i = (x_s^i, y_s^i, w_s^i, h_s^i, \theta_s^i) \quad (3.9)$$

Step 2: Find the average angle

$$\theta_b := \frac{1}{|\beta|} \sum_{\beta} \binom{n}{k} \theta_s^i \quad (3.10)$$

Step 3: For a straight line $(\tan \theta_b)x + b$, find the b that minimizes the sum of distances to all segment centres (x_s^i, y_s^i) .

Step 4: Find the perpendicular projections of all segment centres onto the straight line.

Step 5: From the projected points, find the two with the longest distance. Denote them by (x_p, y_p) and (x_q, y_q)

$$x_b := \frac{1}{2}(x_p + x_q)$$

$$y_b := \frac{1}{2}(y_p + y_q)$$

$$y_b := \sqrt{(x_p - x_q)^2 + (y_p - y_q)^2} + \frac{1}{2}(w_p + w_q)$$

$$h_b := \frac{1}{|\beta|} \sum_{\beta} h_s^i$$

$$b := (x_b, y_b, w_b, h_b, \theta_b) \quad (3.11)$$

Step 6: Output **b** is the combined bounding box

3.4.2 The Network for Serial Number Recognition

When the serial number region is detected by the locating network, the data of this region will be sent to the recognition network for character and number identification. In this experiment, two networks are adopted and compared to implement the recognition which is CNN + RNN network and CNN + ResNet + Attention model.

CRNN is a convolutional recurrent neural network applied to handle image-based sequence recognition problems, specifically for scene text recognition. The CRNN is mainly applied for end-to-end recognition of indefinite-length text sequences. The network does not cut a single text but turns text recognition into a sequence-dependent sequence learning question, which is based on image sequence recognition. The network uses CNN to extract the features of the input image, then takes use of RNN to predict the sequence, and finally obtains the final result through a CTC translation layer (Graves, Fernández, Gomez, & Schmidhuber, 2006). The entire CRNN network is split into three parts, they are:

Convolutional layers. Use deep CNN to extract features from the input image to get feature maps.

Recurrent layers. These layers use bidirectional RNN (BLSTM) to predict the feature sequence, learn each feature vector in the sequence, and output the predicted label (true value) distributions (Fernández, Graves, & Schmidhuber, 2008).

Transcription layer. A series of label distributions acquired from the loop layer is transformed into the final label sequence by using the CTC loss.

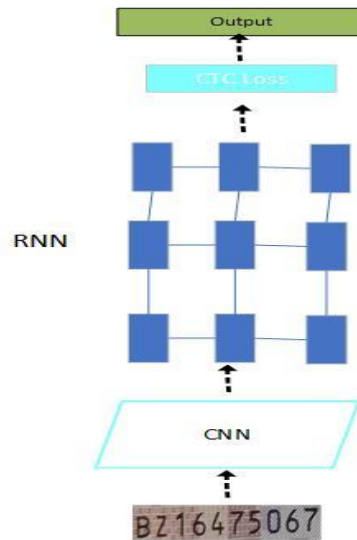


Figure 3.16 CRNN network

The CRNN network inputs a $256 \times 32 \times 1$ normalized image, extracts a feature map based on a 7-layer CNN (VGG16 as the backbone network) and divides the feature map into columns (Map-to-Sequence). Each column has 512-dimensional features. Bidirectional LSTM with 256 units in each layer is classified. During the training process, the CTC loss function is employed to align the character position with the class mark.

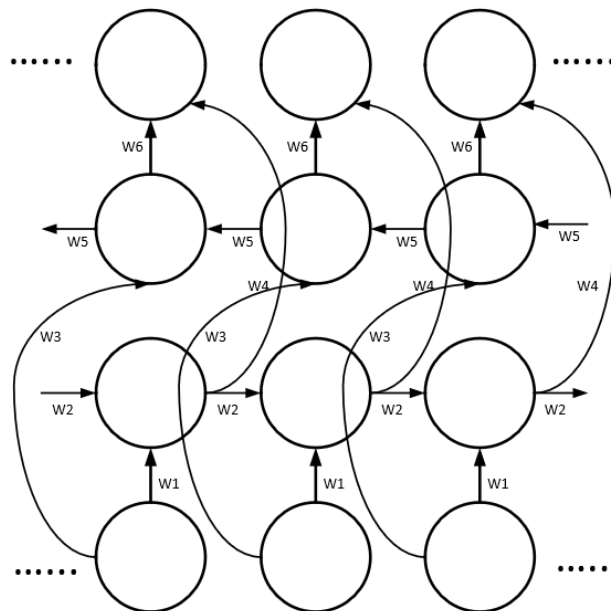


Figure 3.17 Bidirectional-LSTM block

CRNN borrows the method from LSTM + CTC in speech recognition. The difference is that the features input into LSTM is feature maps extracted from the CNN network. It combines the potential of CNN for image feature engineering with the potential of LSTM for serialized

recognition, which not only extracts robust features but also avoids the difficult single character segmentation as well as single character recognition in traditional algorithms through sequence recognition. Serialized recognition also embeds timing dependencies. During the training phase, CRNN uniformly scales the training image to 256×32 (width \times height). During the test, in order to reduce the wrong recognition rate caused by character stretching, CRNN maintains the input image size ratio, but the image height is uniformly 32 pixels. The size of the convolutional feature map dynamically determines the LSTM timing length.

LSTM has 256 hidden nodes. After LSTM, it becomes a vector of length $T \times 36$; after softmax processing, each element of the column vector represents the corresponding character prediction probability. Finally, the prediction result is combined with a complete recognition result.

Table 3.2 CRNN network configurations

Type	Configurations ('k' stands for kernel size, 's' is stride and 'p' is padding size)
Input	256×32×1 greyscale image
Convolution	Maps:64, k:3×3, s=1, p=1
Max pooling	Window: 2×2, s=2
Convolution	Maps:128, k:3×3, s=1, p=1
Max pooling	Window: 2×2, s=2
Convolution	Maps:256, k:3×3, s=1, p=1
Convolution	Maps:256, k:3×3, s=1, p=1

Max pooling	Window: 1×2 , $s=2$
Convolution	Maps:512, $k:3 \times 3$, $s=1$, $p=1$
Batch normalization	-
Convolution	Maps:512, $k:3 \times 3$, $s=1$, $p=1$
Batch normalization	-
Max pooling	Window: 1×2 , $s=2$
Convolution	Maps:512, $k:2 \times 2$, $s=1$, $p=1$
Map-to-sequence	-
Bidirectional-LSTM	Hidden units:256
Bidirectional-LSTM	Hidden units:256
Transcription	-

CRNN with Attention Model

After using the CRNN network for character recognition, our experiment replaced the RNN part with CNN and added the residual attention model to the feature extraction network. This is the reason why each step of RNN depends on the previous steps and the calculation cannot be parallelized, in spite that RNN processes sequence signals very effectively to obtain the long-term dependencies. Therefore, the calculation of the RNN model relies on the length of the input sequence, and often need a long time to calculate. Training RNN network is much

more difficult than to train CNN network, and the issue of vanishing gradients or exploding gradients often occurs. In contrast, CNNs can be highly parallelized and have low computational complexity.

At present, CNN has been applied to process sequence of machine translation and construct language models. Therefore, in this experiment, CNN and CTC were used in combination without using any recurrent unit, and no circulation unit was needed. First, a sequence-to-feature map is used to convert the sequence into a 2D feature map as an input to the CNN. Then, the stacked CNN is employed to extract feature representations of different levels of contexts to obtain long-term dependencies. The length of the dependency can be restrained by using a number of convolutions. This structure is a fully convolutional structure, which is very easy to be parallelized, and there is no special requirement on the length of the sequence.

In order to strengthen the expressive ability of text and suppress noise, a residual attention model is used in small Dense networks to obtain more separated attention features. This is the reason why the attention model takes a very important part in the process of feature learning. It can focus on the salient region and improve the expression ability of relevant parts. For the serial number recognition of banknotes, there are many noises, including shadows, irrelevant symbols, and background textures, etc. Using the attention model can effectively suppress these interferences in the background. The transformed network includes an attention block, a convolutional sequence model, and CTC.

Attention Feature Encoder

To enhance the discriminability of CNN features, a residual attention encoder network with DenseNet block is designed to obtain the attention aware representation. Dense connectivity mechanism can be used to enhance the information movement among different layers; meanwhile, by using an attention model with residuals (Wang, et al., 2017), the disturbance of background noise can be efficaciously restrained. Moreover, the feature maps of the input image can be turned into a sequential expression. Specifically, the same columns of feature maps can be taken out and concatenated into a vector from left to right, which is correlated to a local rectangle area of the input image.

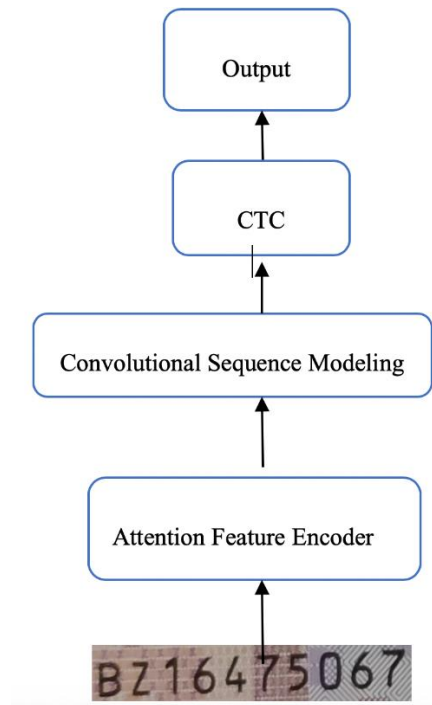


Figure 3.18 CRNN with attention model

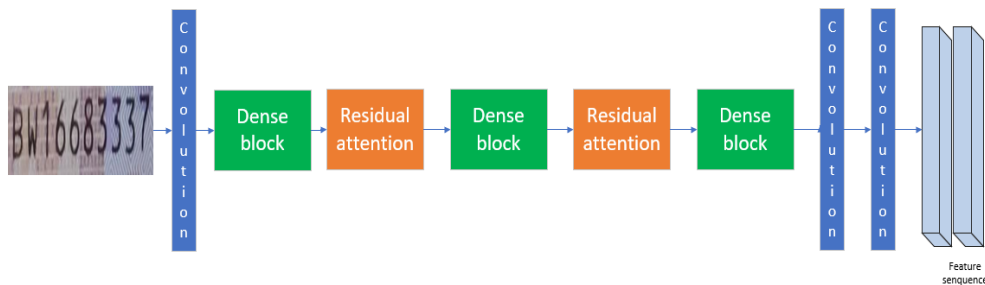


Figure 3.19 Attention feature encoder

Dense Connectivity

Making full use of the power of DenseNet (Huang, Liu, Van Der Maaten, & Weinberger, 2017), the dense connectivity is used to enhance the flow of information and gradient propagation in the encoder network. Direct connections exist between all layers in the dense block. Consequently, each layer can obtain the information from all previous layers and send its message to all following layers. Additionally, instead of performing the gradient backpropagation layer by layer, each layer can be supervised in-depth, which simplifies the learning process. Feature maps generated by previous layers are concatenated and used as the input of the subsequent layers.

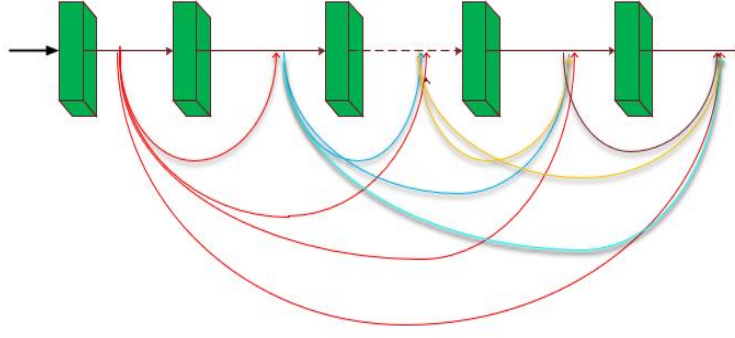


Figure 3.20 Dense block used in modified CRNN network

Attention Model with residual

Attention model takes an important leading part in the process of feature learning, its goal is to selectively focus on salient areas of the object and improve the representation of pertinent parts. For text in natural images, there often exists noises, including shadow, unrelated icons, and background texture. The scene text with various appearance is often confused by different kinds of influences. Therefore, a residual attention mechanism (Wang, et al., 2017) is used to improve the representation of font text and restrain background noise.

The residual attention model acts as the transition between dense blocks. It consists of two parts. Specifically, the feature part fulfils the feedforward process and the attention part produces the soft attention weights. The attention part is used with bottom-up top-down structure so that the high-level semantic information can be obtained to conduct the discriminative feature selection. To enlarge the receptive field effectively and gather global information, max pooling layer and a convolutional layer which is stacked several times is used. Then asymmetrical structure with bilinear interpolation for up-sampling is used to restore the resolution. The attention maps as soft weights are put into corresponding feature maps in each position afterwards. Since the value of attention weight ranges from zero to one, the element-wise product between the feature map and the attention map may cause serious degradation of useful information. Therefore, residual attention learning is used to solve this problem. Just like Resnet (He, Zhang, Ren, & Sun, 2016) does, the output of the residual attention model is

$$T = (1+A) \times F(1) \quad (3.12)$$

where F and A represent the output of the feature part and the attention part, accordingly.

As a result, the background noises can be constrained effectively and improving the discriminability of original features at the same time. Moreover, different attention modules produce the attention maps adjusted to the relevant features. The low-level attention module mainly focuses on the details including border, color and text, and the high-level attention ones acquire more semantic information. By using this mechanism, the noise is effectively suppressed, so the feature encoder can get a more discriminative indication.

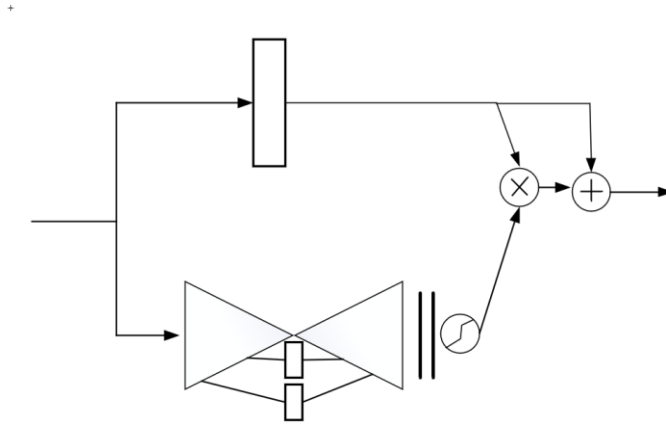


Figure 3.21 Residual attention model

Convolutional Sequence Modeling

As the main method of sequence-to-sequence learning, RNN has a wide range of applications in the field of computer vision, as well as speech recognition (Hannun, et al., 2014), language modelling (Jozefowicz, Vinyals, Schuster, Shazeer, & Wu, 2016), and machine translation (Luong, Sutskever, Le, Vinyals, & Zaremba, 2014). Nevertheless, based on the calculation of the previous steps, the recurrent connection is unable to perform parallel calculations. Moreover, RNN is difficult to be trained due to the problem of gradient vanishing and exploding. So, in this thesis, CNN is employed to obtain the sequential dependencies with bidirectional to recognize scene text, which performs faster than RNN network. Considering the feature sequence generated by encoder, which is indicated as $f = (f_1, f_2, \dots, f_w)$. In order to obtain the contextual information $c = (c_1, c_2, \dots, c_w)$, the RNN produces the contextual representation through the recurrent connection $c_i = R(c_{i-1}, f_i)$, which is a chain structure and is unable to perform parallel computation.

The proposed method generates sequential dependencies by using totally convolutional operation. First of all, we use elements of the feature sequence to produce a 2D map together, where each column is related to a local area of the original text image from left to right. Then the input is convolved by a filter with width k and generate the contextual information on k elements of the input sequence. The convolutional layers produce the hierarchical representation to expand the size of the receptive field effectively. As a result, the range of spatial dependencies to be modelled can be easily controlled through the number of convolutional layers.

If there are enough layers, the high-level features are able to acquire the contextual information need. Moreover, the convolution operation is not determined by the state of the previous step and is independent of the length of the input sequence. As a result, the calculation over the entire sequence can be processed at the same time, which could effectively boost the process of sequence modelling. Furthermore, the convolutional network costs less memory space and running time due to fewer parameters and lower computation complexity.

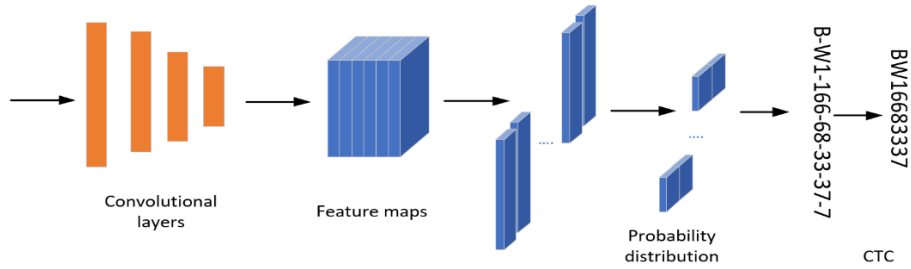


Figure 3.22 Convolutional sequence modelling and CTC

In the process of sequence modelling, the length of the sequence can stay unchanging by operating convolution with zero paddings. Afterwards, in order to get the sequential representation to serve as the input of CTC, we restore the output feature maps to a sequence again by using the same map-to-sequence operation in the feature encoder. Assuming the feature maps generated by using CNN have the dimension of $C \times H \times W$, where C , H , and W represent the channels, height and width accordingly. Specifically, we crop each channel of feature maps by using column and then concatenate the same columns of all channels into a vector, which has a dimension of $C \times H$. Therefore, we can acquire a sequence with W vectors, which is the contextual information $c = (c_1, c_2, \dots, c_w)$.

Finally, for the obtained sequence, we can get the probability distribution over the label space for the per frame in the sequence through a linear layer

$$y_t = \text{softmax}(\mathbf{W}c_t + \mathbf{b}), t = 1, 2, \dots \quad (3.13)$$

where \mathbf{W} and \mathbf{b} denote the weight matrix and bias separately.

Table 3.3 The architecture of the CRNN network with attention model

Module	Layer	Configurations
Encoder	Convolution	3×3 , 36, stride 1×1
	Dense Block	$[3 \times 3, \text{stride } 1 \times 1] \times 4$
	Attention Module	Attention 1
	Average Pooling	2×2 , stride 2×2
	Dense Block	$[3 \times 3, \text{stride } 1 \times 1] \times 4$
	Attention Module	Attention 1
	Average Pooling	2×2 , stride 2×2
	Dense Block	$[3 \times 3, \text{stride } 1 \times 1] \times 4$
	Convolution	3×3 , 512, stride 1×1
	Average Pooling	2×2 , stride 2×1
CNN	Convolution	3×3 , 512, stride 1×1
	Convolution	3×3 , 1, stride 2×1
	Convolution	3×3 , 1, stride 2×1
	Convolution	3×3 , 1, stride 2×1
CTC	CTC	-

3.5 Evaluation Methods

3.5.1 Loss Function for Detection

One-stage object detection algorithm, represented by YOLO (Redmon, et al., 2016) and SSD, abandons the process of extracting the proposal and completes recognition/regression using only one stage (Tian, Shen, Chen, & He, 2019). Although it is faster, the accuracy is not as good as the two-stage object represented by Fast RCNN. Because the low one-stage accuracy is determined by class imbalance. The bbox for calculating loss can be divided into two categories: positive and negative (Liu & Jin, 2017). When the intersection over union (IoU) between the bbox (obtained by the anchor plus the offset) and the ground truth is greater than the upper threshold (usually 0.5), the bbox is considered to be a positive example. If the IoU is less than the lower threshold, the bbox is considered to be negative (Kong, Yao, Chen, & Sun, 2016). In an input image, the proportion of the object is generally much smaller than the

proportion of the background, so the two examples are mainly negative, which raises two problems:

- Too many negative examples cause the loss to be too large so that the positive losses are submerged, which is not conducive to the convergence of the object.
- Most negative examples are not on the transition area between foreground and background, and the classification is very clear. The corresponding score of background class will be enlarged during training. From another perspective, the loss of a single example is small. The gradient is small when calculating the backwards. A small gradient causes the easy negative example to have a limited effect on the parameter convergence. We need an example with a large loss that has a greater effect on the parameter convergence, i.e., a hard positive/negative example.



Figure 3.23 Four kinds of examples

In this experiment, the focal loss is used to adjust the calculation of loss so that one-stage algorithms can achieve the same accuracy as Fast R-CNN does. The focal loss function is

$$FL(p_t) = -\alpha_t(1 - p_t)^\gamma \log(p_t) \quad (3.14)$$

where P_t is the classification probability of different categories, γ is a value greater than 0, α_t is a decimal between [0, 1]. γ and α_t are fixed values, and they do not participate in training.

For both foreground and background are shown in eq. (3.9), the larger the p_t is, the smaller the weight $(1 - p_t)^\gamma$ will be. In other words, an easy example can be suppressed by adjusting weight; where α_t is used to adjust the ratio of positive and negative. When α_t is used for the foreground class, $1 - \alpha_t$ is used for the corresponding background category. The optimal values

of γ and α_t are affected by each other, thus, we need to adjust the two values at the same time when evaluating the accuracy.

3.5.2 Loss Function for Recognition

CTC is a loss calculation method (Shi, Bai, Yao, & intelligence, 2016). CTC is used instead of the softmax loss (Liu, Wen, Yu, & Yang, 2016), and the training samples do not need to be aligned. CTC features:

- Introduce blank characters to solve the problem of no characters in some positions.
- Fast calculation of gradients through recursion.

The training process of CTC essentially adjusts the LSTM parameter \mathbf{w} through the gradient $\frac{\partial p(l|x)}{\partial \mathbf{w}}$, so that $p(l|x)$ is maximized when the input sample is $\pi \in B^{-1}(l)$. CTC uses the forward-backwards algorithm to calculate $p(l|x)$.

3.5.3 Evaluation Metrics

The prediction results are divided into four categories:

- *TP* is True Positive, judged as a positive sample, in fact also a test sample.
- *TN* is True Negative, judged as a negative sample, in fact a negative sample.
- *FP* is False Positive, judged as a positive sample, is actually a negative sample.
- *FN* is False Negative, judged as a negative sample, but in fact a positive sample

Truth \ Prediction	Positive	Negative
Positive	TP	FN
Negative	FP	TN

In this experiment, accuracy, error rate, precision, recall, F1 Score, receiver operation characteristic (ROC) curve are used as a performance evaluation metric. Accuracy is the rate that the number of all correctly classified samples divided by total samples

$$\text{Accuracy (\%)} = \frac{(TP+TN)}{\text{All values}} \times 100. \quad (3.15)$$

The error rate is the rate that the number of all misclassified samples divided by the total number of samples.

$$\text{error}(f; D) = \frac{1}{N} \sum_{i=0}^N I(f(x_i) \neq y_i) \quad (3.16)$$

$$\text{Accuracy} + \text{Error rate} = 1 \quad (3.17)$$

Accuracy or error rate is the most basic evaluation index for classification. But the number of samples in each class is not the same. For example, out of a total of 100 test samples, there are 98 positive samples and only 2 negative samples. Then, we only treat all samples as positive, so the accuracy can reach 98%.

Even if the number of samples in each category is relatively balanced, if a specified class needs to be more concerned, then the indicators should be selected to evaluate the quality of the model. Therefore, indicators such as recall and precision are used. Precision (%) is the proportion of correct predictions in all samples with positive predictions.

$$\text{Precision (\%)} = \frac{TP}{TP+FP} \times 100. \quad (3.18)$$

Recall is the correct proportion predicted in all actually positive samples.

$$\text{Recall (\%)} = \frac{TP}{TP+FN} \times 100. \quad (3.19)$$

Accuracy and recall are a pair of contradictory measures. In general, when the recall rate is high, the precision rate is usually low; when the recall rate is high, the precision rate is often low.

F1 Score is also known as the F Score, which is defined as the harmonic average of precision and recall. F1 Score indicator combines the results of precision and recall. The value

of F1 Score ranges from 0 to 1.0, where 1.0 stands for the best model output and 0 represents the worst model output

$$F1 = \frac{2 \times Precision \times Recall}{Precision + Recall}. \quad (3.20)$$

ROC curve is a curve drawn based on “True Positive Rate” (TPR) as the y-axis and “False Positive Rate” (FPR) as the x-axis. The closer the ROC curve is to the upper left corner (true positive rate approximates to 1.0, the false positive rate will be 0), the better the model performance is. If the ROC curve of one model is completely covered by the curve of another model, it asserts that the performance of the latter is better than the former; if there is a crossover, the performance can be judged by comparing the area under the ROC curves.

$$TPR = TP/(TP + FN) \quad (3.21)$$

$$FPR = FP/(TN + FP) \quad (3.22)$$

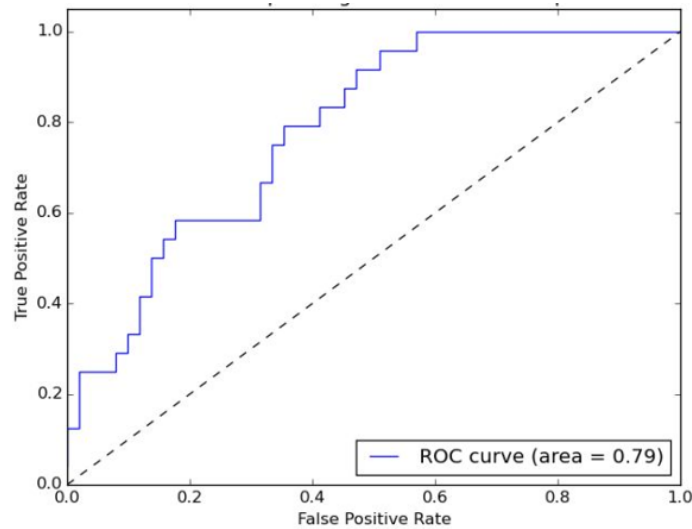


Figure 3.24 ROC curve

3.6 Summary

In this chapter, the methodology of the research is discussed in detail. To complete the experiment, the first step is to prepare the data, 1000 samples of different demission are collected, and labelled. The labelling includes two parts: serial number region labelling and serial number characters labelling. The second is to enlarge the dataset by using data

augmentation which will strengthen the robustness of the model. The recognition of banknote serial number is divided into two steps: serial number region detection and serial number recognition. The algorithms involved in this chapter are SegLink, DenseNet, CRNN, ResNet and Attention model. The modified networks and pipeline are proposed to take the advantages of networks above. Serial number region detection is implemented by using DenseNet with SegLink, the serial number recognition is realized by using CRNN with a residual attention mechanism.

Through the literature review, we find that different activation functions and optimizers also have effects on the training speed and accuracy. In the next chapter, the experimental results will be presented using different networks, activate functions, and optimizers. The comparisons will be conducted among the results of different networks.

Chapter 4

Results

The main content of this chapter is to introduce the schema of the whole methods and the implementation of banknote serial number recognition. The experimental environment will be built in this chapter. In addition, this chapter will clarify the results of banknote serial number region detection and text recognition.

4.1 Experimental Environment

Based on the design of the algorithm in Chapter 3, this chapter sets up the actual environment for implementation, performs experiments, and obtains the results. The experimental platform is divided into a software platform and a hardware platform. The specific configuration is as follows:

- Operating system: Windows 10 professional
- Hardware configuration: CPU: INTEL I7 6700K; GPU: NVIDIA GeForce GTX1060; Memory: 16GB
- Development language: Python 3.6
- Environment: CUDA, cuDNN, TensorFlow, OpenCV, Keras.

4.2 Algorithm Analysis and Verification

4.2.1 Serial Number Region Detection

The banknote dataset used in this experiment contains 1,000 original samples, which are expanded to 5,000 after data augmentation. The K-fold cross-validation method was used in the experiment.

The part of region positioning for serial numbers uses the SegLink with DenseNet as the backbone network, the loss function is the focal function, respectively. The backbone of the network was initialized with weights from a pre-trained SegLink model which is trained on SynthText dataset.

As a neural network, the input and output layers need to be designed for the attributes of different output data. After continuous optimizations in our experiments, it was determined that the output image size was uniform at 512×512 pixels. The output of the network is divided into two categories: Serial number and background. This thesis defines the conv5_2, conv9_2, conv10_2, conv11_2, conv12-2 layers as multi-scale feature extraction layers.

Learning Rate

The learning rate, as an important hyperparameter in supervised learning and deep learning, determines whether the objective function converges to the minimum value and when it converges to the value. The learning rate refers to the hyperparameters when the weights are updated during the gradient descent, as α is defined as

$$\theta = \theta - \alpha \frac{\partial}{\partial \theta} J(\theta) \quad (4.1)$$

The lower the learning rate, the slower the change rate of the loss function, and the more likely it is overfitting. Although using a low learning rate can ensure that the training process does not miss any local minima, it also means that it will take more time to converge, especially when trapped in the local best. If the learning rate is too high, the gradient explosion is easy to occur, and the amplitude of loss vibration is large, which makes it difficult for the model to converge. A reasonable learning rate allows the model to converge to the smallest point instead of the local best point or saddle point. Therefore, it is very important to choose the right learning rate.

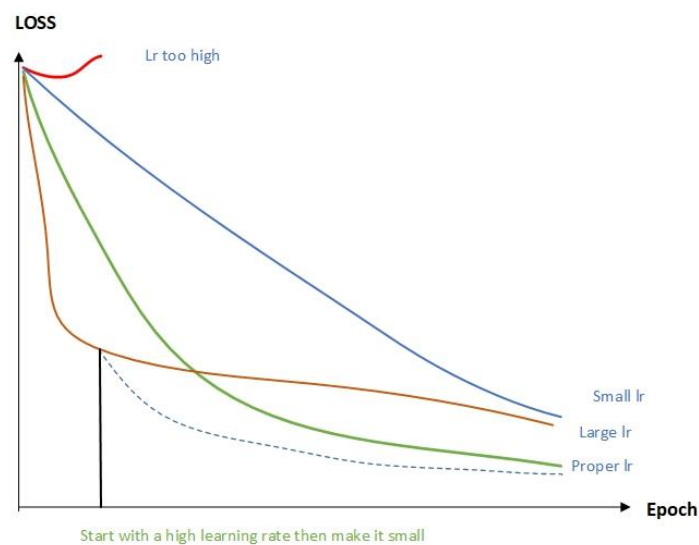


Figure 4.1 Training loss with different learning rate

The red curve rises from the beginning, indicating that the initial learning rate is too large and cause oscillations, thus, we should reduce the learning rate and train from scratch.

The yellow curve hardly changes after a period of rapid decline, this is because the learning rate is too large, which leads to convergence. The learning rate should reduce and retrain the later epochs.

The blue curve keeps falling slowly, which means that the learning rate is too small, and the convergence rate is slow. The learning rate should increase and start training from scratch.

Therefore, this experiment sets the learning rate to a dynamic value. In the first few rounds of epochs, large learning rates are used so that the training speed is fast.

As the epoch gradually increases, the learning rate also decreases. The attenuation mechanisms are: Step decay, exponential decay, and $1/t$ decay. Exponential decay is proved to be more suitable for this experiment, it is calculated as follows

$$decayed_lr = lr0 \times (decayrate^{\frac{globalsteps}{decaysteps}}) \quad (4.2)$$

where $decayed_lr$ is the decayed learning rate, which is the actual learning rate used in current training, $lr0$ is the initial learning rate. In this experiment, it is set as $lr0 = 0.001$. $decayrate$ is the decay rate, which is the ratio of each decay. $Globalsteps$ is the current training steps, $decaysteps$ is the number of decay steps, that is, how many steps to complete the decay.

Optimizers

The choice of optimization algorithm has a great impact on the accuracy of the model. Even when the dataset and model architecture is exactly the same, using different optimizations may lead to very different training results.

Gradient descent is one of the most widely used optimization algorithms in neural networks. Gradient descent means that given the model parameters $\theta \in R^d$ and objective function $J(\theta)$ to be optimized, the algorithm updates the value of θ in the opposite direction of the gradient of $\nabla J(\theta)$ to the minimum, the value of $J(\theta)$, learning rate η determines the update step size at each moment. The calculation process of gradient descent is as follows. This experiment mainly uses two gradient descent algorithms, SGD with Momentum and Adam.

Step 1. Calculate the gradient of the objective function concerning parameters:

$$g_t = \nabla J(\theta) \quad (4.3)$$

Step 2. Calculate momentum based on a historical gradient:

$$m_t = \phi(g_1, g_2, \dots, g_t) \quad (4.4)$$

$$v_t = \Psi(g_1, g_2, \dots, g_t) \quad (4.5)$$

Step 3. Update the parameters

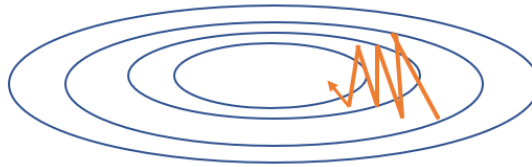
$$\theta_{t+1} = \theta_t - \frac{1}{\sqrt{v_t + \varepsilon}} m_t \quad (4.6)$$

where $\varepsilon = 1e - 8$.

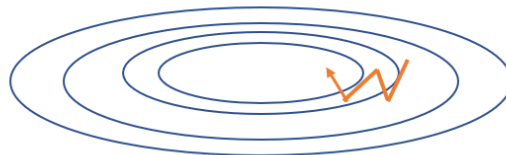
SGD with Momentum

SGD is prone to shocking when encountering gully. Therefore, momentum is introduced to accelerate the decline of SGD in the correct direction and suppress the shock. SGD with momentum is above the original step size, adding γm_{t-1} related to the previous step size. This makes the parameter update direction not only determined by the current gradient but also related to the previously accumulated descent direction. The dimension in which the gradient direction does not change much can accelerate the update and the dimension in which the gradient direction changes greatly decreases the update range. This has the effect of accelerating convergence and reducing shock.

$$M_t = \gamma m_{t-1} + \eta g_t \quad (4.7)$$



SGD



SGD with momentum

Figure 4.2 The convergence speeds of SGD and SGD with momentum

In Figure 4.2, the introduction of momentum effectively accelerates the convergence process of gradient descent. In this experiment, the momentum is set to 0.9.

Adam

Adam is a combination of *RMSprop* and momentum. The update process is

$$m_t = \eta[\beta_1 m_{t-1} + (1 - \beta_1)g_t] \quad (4.8)$$

$$v_t = \beta_2 v_{t-1} + (1 - \beta_2) \text{diag}(g_t^2) \quad (4.9)$$

$$\hat{m}_t = \frac{m_t}{1 - \beta_1^t} \quad (4.10)$$

$$\hat{v}_t = \frac{v_t}{1 - \beta_2^t} \quad (4.11)$$

$$\theta_{t+1} = \theta_t - \frac{1}{\sqrt{\hat{v}_t} + \epsilon} \hat{m}_t \quad (4.12)$$

where $m_0 = 0$, $v_0 = 0$.

In this experiment, both optimizers are compared. The training loss and validation loss are shown in Figure 4.3 and Figure 4.4.

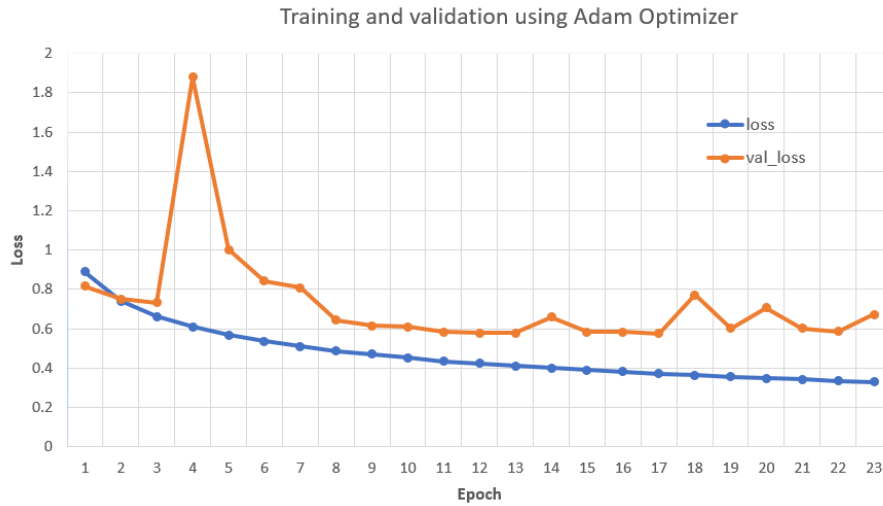


Figure 4.3 Performance of the network using Adam Optimizer

Activation Functions

The role of the activation function is to add nonlinearity to the network. The network needs to calculate the activation value of the next layer based on the activation function, weight, and bias of the previous layer. Before sending the value to the next layer, the activation function is used to scale.

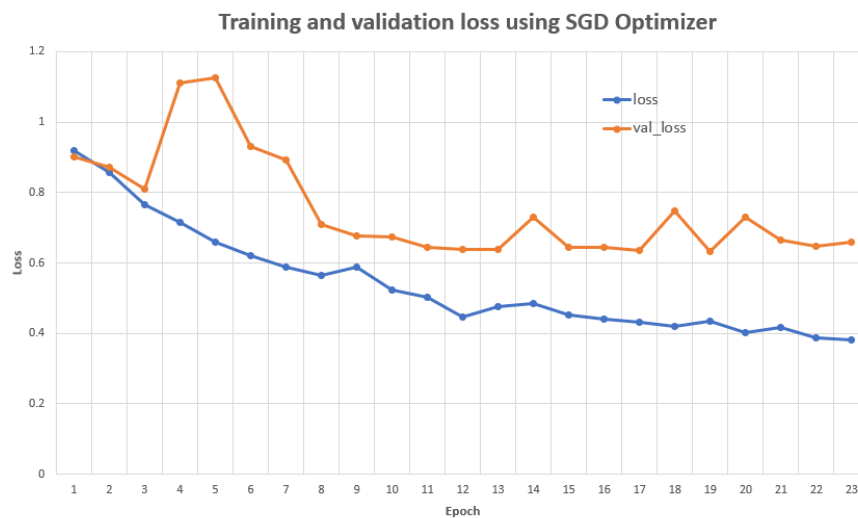


Figure 4.4 Performance of the network using SGD Optimizer

This experiment uses ReLU and GELUs as activation functions respectively. Because there is no GELUs in Keras, we manually add the code as

```
def gelu(input_tensor):  
    cdf = 0.5 * (1.0 + tf.erf(input_tensor / tf.sqrt(2.0)))  
    return input_tensor*cdf
```

The convergence result of the detection network of serial number using the above two activation functions is shown as Fig. 4.5 and Fig. 4.6.

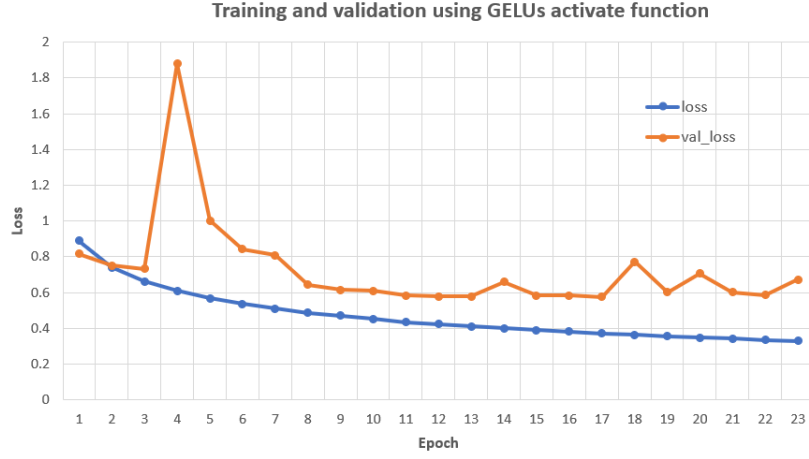


Figure 4.5 Performances of the network using GELUS activate function

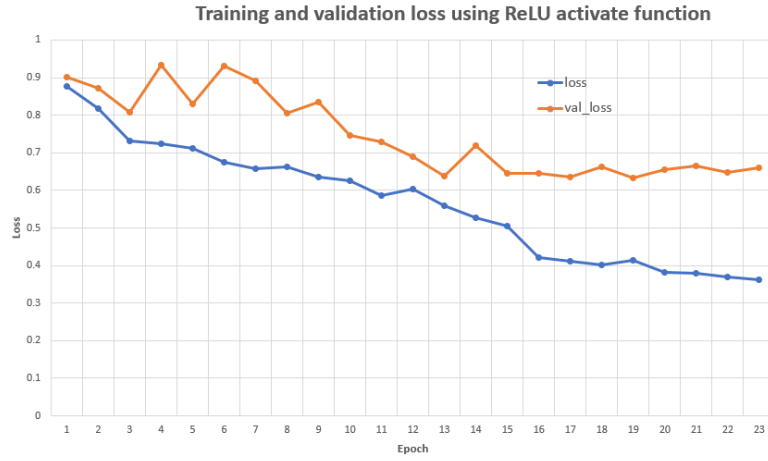


Figure 4.6 Performances of the network using ReLU activate function

We see from the comparative experiments that using GELUs as the activate function and Adam as optimizer performs best in terms of the convergence speed and loss function.

The focusing parameter γ of the focal loss was set to 2.0 for both segments and links. The weighting of the different loss terms has been adjusted to the better scale of the focal loss ($\lambda_{\text{segments}} = 100.0$, $\lambda_{\text{offsets}} = 1.0$, and $\lambda_{\text{links}} = 1.0$). The final results of the detection are shown in Table 4.1.

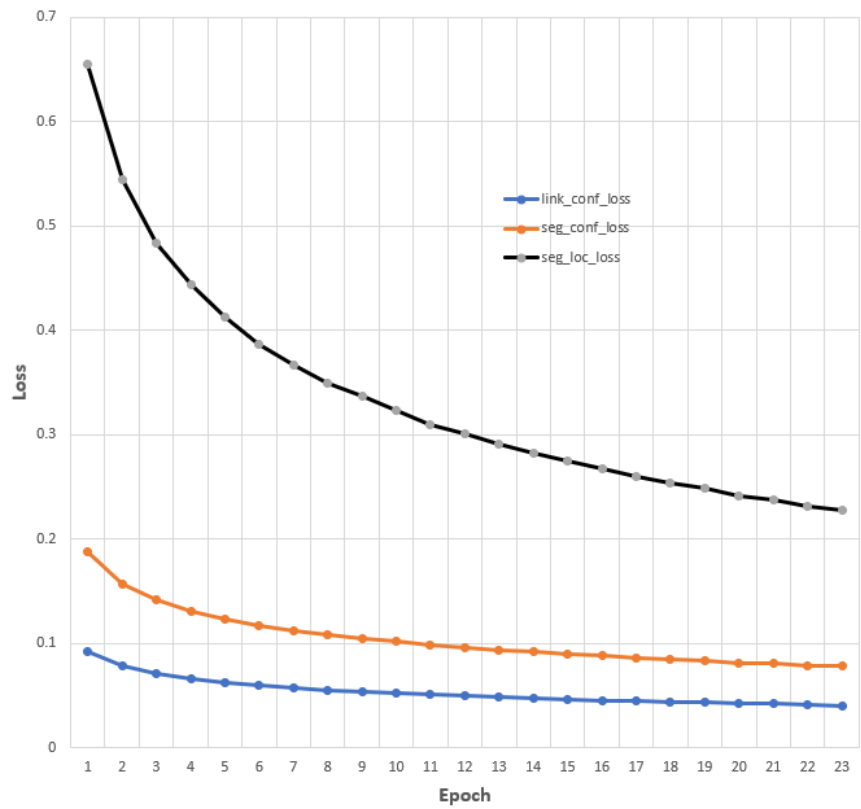


Figure 4.7 Training loss

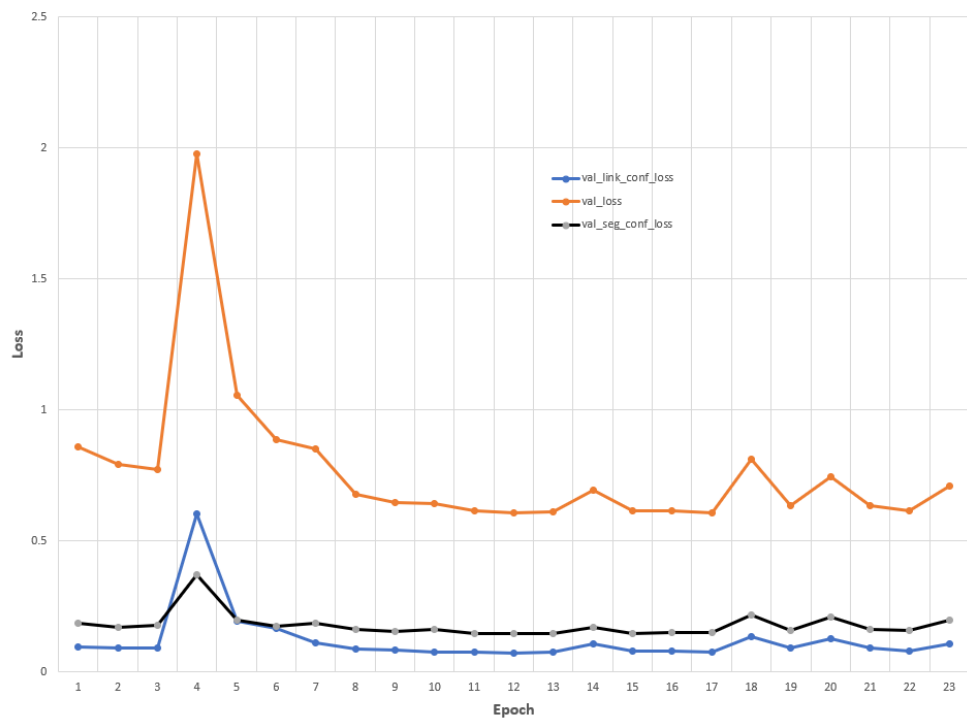


Figure 4.8 Validation loss

Table 4.1 Final results of the serial number detection

Model	Precision	Recall	F-measure
SegLink + DenseNet	0.95807	0.95208	0.95507

The detection results samples are shown in Fig. (4.9).



(a) The detection result of 20 NZD (b) The detection result of 50NZD



(c) The detection result of 100 NZD

Fig. 4.9 The detection results of NZ currency

4.2.2 Serial Number Recognition

When the region recognition network completes detection, the detected area is sent to the text recognition network for recognition. The following two networks were used in this experiment:

- Convolutional Recruitment Neural Network (CRNN).

- Modify CRNN with attention module.

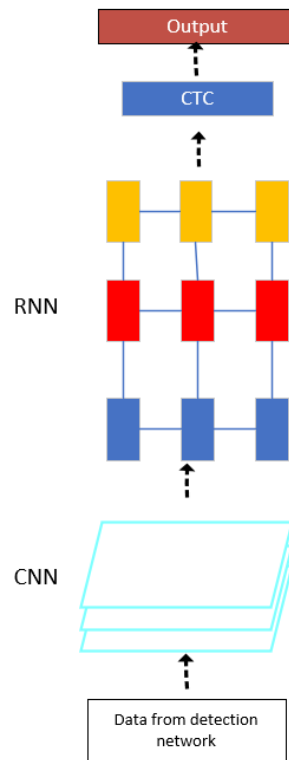


Figure 4.10 CRNN network for banknote serial number recognition

In Figure 4.9, the detection network is mainly composed of three parts: CNN block, RNN block, and CTC.

First, CRNN network is trained as the loss during the training is shown in Figure 4.11. Then the RNN part is replaced with CNN (DenseNet block + ResNet + Attention Block) and trained on the same dataset again to compare the training speed and accuracy. The modify network constructor is shown in Figure 4.12. All the parameters details are listed in Chapter 3.

Table 4.2 Result of Serial Number Recognition

Model	Precision	Recall	F-measure
CRNN	0.9649	0.9669	0.9659
Modified CRNN with attention module	0.9709	0.9609	0.9659

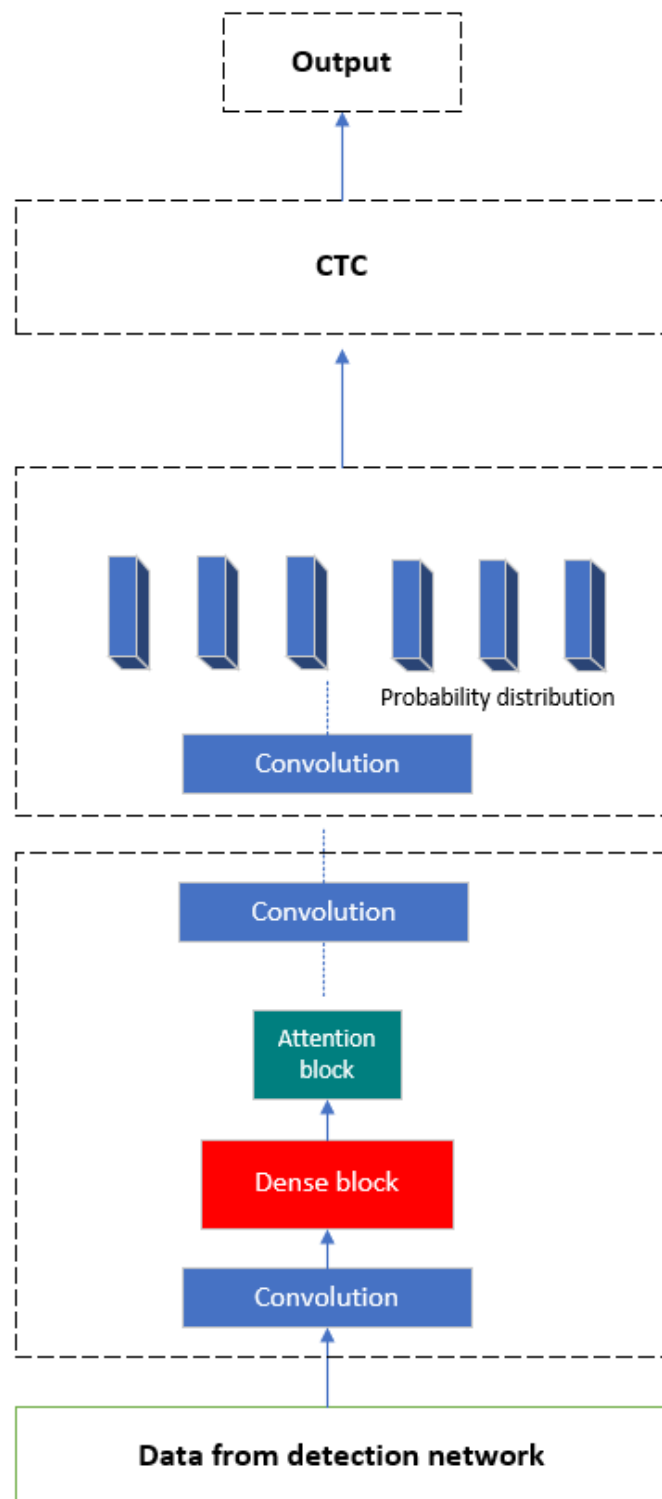


Figure 4.11 Modified CRNN with attention model

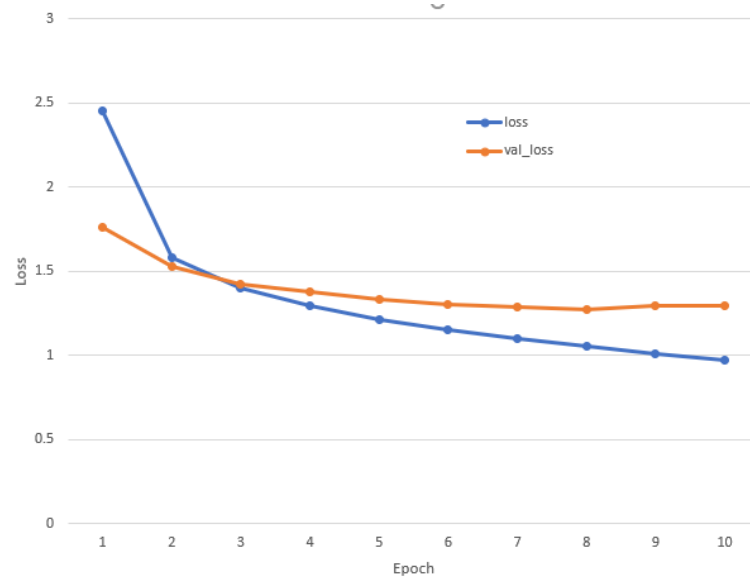


Figure 4.12 The loss of CRNN network during training

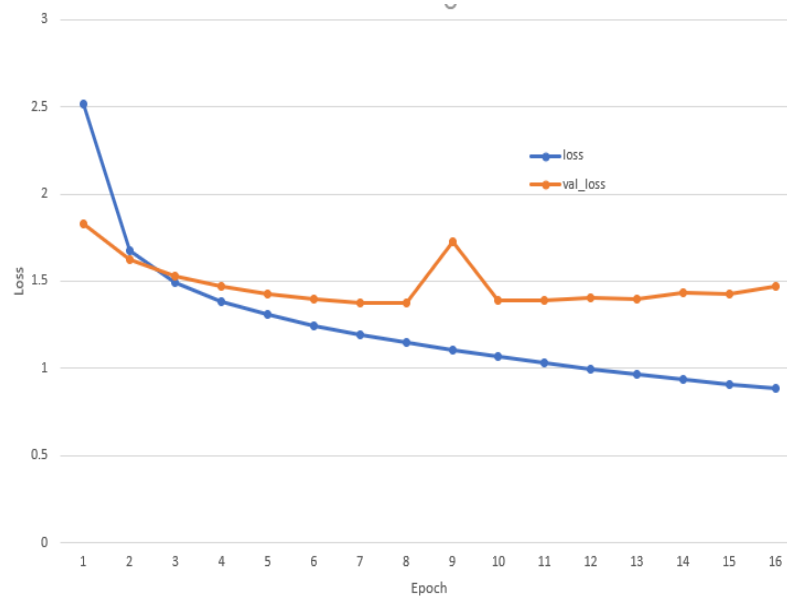


Figure 4.13 The loss of modified CRNN network with attention block



(a) The result of \$20 NZD (b) The result of \$50 NZD (c) The result of \$100NZD

Figure 4.14 The recognition results of \$20, \$50, and \$100 NZD

Chapter 5

Analysis and Discussion

In this chapter, dialectical comparisons of the performances between designed models and other models are made. Moreover, this project also illustrates the effects of different optimizers and the activate functions involved in the model.

5.1 Analysis

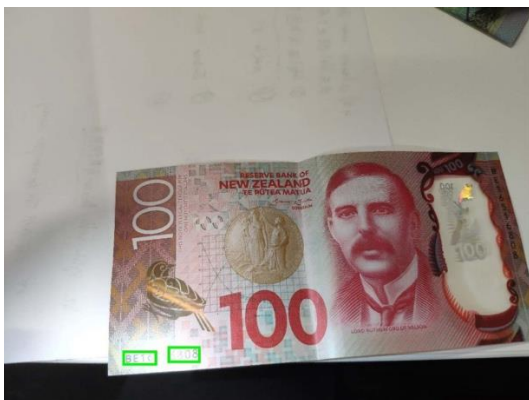
In this chapter, the performances of different networks which are used for banknote serial number region detection and serial number recognition are analyzed in detail.

5.1.1 Analysis of Different Model for Banknote Serial Number Area Detection

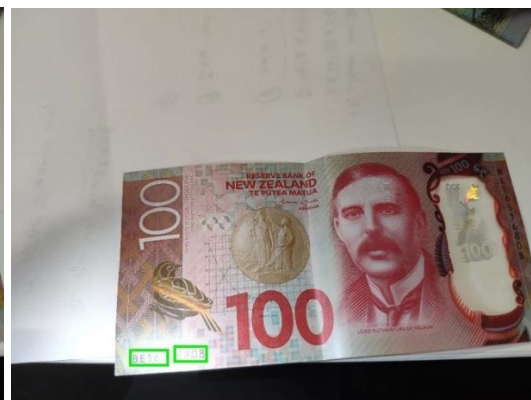
In order to compare with the designed model, a SegLink model with Inception v4 + ResNet152 as the backbone was added to this experiment.

Table 5.1 The performances of different detection models

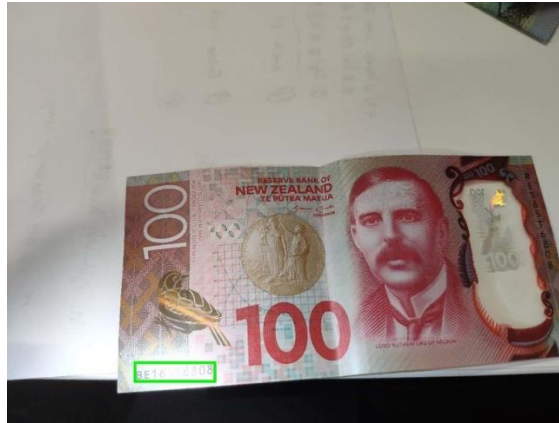
Model	Precision	Recall	F-measure
SegLink	0.88764	0.87778	0.88268
SegLink + Inception	0.92391	0.91398	0.91892
SegLink + DenseNet	0.95807	0.95208	0.95507



(a) Wrong detection using SegLink network



(b) Wrong detection using SegLink with Inception



(c) Right detection using SegLink with DenseNet

Figure 5.1 Different detection results using different networks

We see from Fig. 5.1, because DenseNet strengthens the transmission of features through DenseNet, it has certain advantages in extracting small object features; but due to the increase the number of calculations, the recognition speed has decreased compared to Inception + ResNet.

5.1.2 Analysis of Different Model for Banknote Serial Number Recognition

In the serial number recognition, two models, CRNN with CTC and modified CRNN with DenseNet and residual attention block, were used for training.

Table 5.2 The performances of different recognition networks

Model	Precision	Recall	F-measure	Run time
CRNN	0.9649	0.9669	0.9659	119ms
Modified CRNN with attention module	0.9709	0.9609	0.9659	5ms

5.2 Discussions

5.2.1 Discussion of Models Performance for Banknote Serial Number Area Detection

We see from the results that the sequence number region detection with deep learning has achieved good results. In particular, it can identify banknotes with a tilt angle, which greatly improves the usability of the model in daily life. It can quickly identify banknotes from real-time monitoring and overcomes the knowledge required by traditional SERA (dice) (fixed position and relative position), or pigment swelling (a large number of thick lines and dots in the background of the banknote, causing more interference), and other defects. The character is detected by using the segment of serial number, and the serial number region is effectively achieved through the link detection with high accuracy.

This study also used the traditional SSD method, but during the training process, it was found that the model could not converge for three reasons: the dataset is not big enough, the number of layers in the network was not deep enough, and convolutional networks usually rely on contours to learn. Compared to other objects such as license plates, the serial number of the banknote does not have obvious edges and contours, which causes the model to fail, we need to learn enough features to achieve recognition. With the introduction of DenseNet, the depth of the network has been enhanced. With a limited dataset, high accuracy is achieved. It shows that deepening the network can effectively improve the accuracy of the network in small object recognition.

5.2.2 Discussion of Models performances for Banknote Serial Number Recognition

Traditional methods often need to cut a series of text into a single text using projection method (Wang et al., 2011), and send it to CNN for classification (Wang, Wu, Coates, & Ng, 2012). However, the background of the banknote serial number region is very complicated, with a large number of thick lines and dots. If the projection method is used, it is very easy to cause misidentification. CRNN abandons the traditional method of "cut word then classify" and converts text recognition to sequence learning problems.

In this experiment, the residual attention model and DenseNet are added to the feature extraction network of CRNN. The distinct difference from the original CRNN network is how to convert the feature maps of the banknote serial number area into sequence feature

information. Since the RNN network was replaced by CNN, the recognition time of the network has been reduced over 10 times. Finally, CTC loss was used in the alignment, and good results were gained.

Through our experiments, it can be found that the CRNN network using LSTM performs very good in the recognition of text under more complex backgrounds by reaching the precision of 96.49%.

Table 5.3 The training time (s) of different recognition networks

Epoch \ Model	1	2	3	4	5	6	7	8	9	10
CRNN	19044	37650	57310	77010	96377	112421	137946	154334	177601	204791
Modify CRNN with attention	6943	15137	21348	30781	38684	46359	55796	71956	80317	87167

We see that the modified CRNN with residual attention network borrows the LSTM + CTC method in speech recognition. The difference is the input features. The acoustic features (MFCC, etc.) in the speech field are replaced by the image feature vectors extracted by the CNN network. The potential of CNN for image feature engineering is combined with the potential of LSTM for serialized recognition. It not only extracts robust features but also avoids the difficulties of single character segmentation and single character recognition in traditional algorithms through sequence recognition. Simultaneously, serialization recognition also embeds timing dependencies (implicit use of corpora). The use of residual attention enables the whole network easier and faster to train, the speed for recognition is much faster than the ANN does, when RNN is replaced by CNN, the running time of this network is reduced from 119ms to 20ms, and the accuracy has not much decreased.

Chapter 6

Conclusion and Future Work

In this project, an in-depth explanation of performances of different networks is discussed. We elaborated on the research results and the innovation of research methods. In this chapter, we will present this argument at the scholar level. Additionally, we also integrate and organize the conclusions into the context, meanwhile point out the future work at the end of this thesis.

6.1 Conclusion

6.1.1 Conclusion of Detection Network

The purpose of this experiment is to quickly detect the serial number region of a banknote through deep learning. This experiment overcomes the previous drawbacks that require pre-knowledge, that means, it does not need to specify a specific range, nor a specified position for identifying the banknote serial numbers.

This experiment demonstrates the pipeline of image collection, data labelling, image augmentation, region locating and positioning, network training, and export the outputs for banknote serial numbers. In order to avoid any influence of the currency samples, all the images in this thesis were taken directly and randomly from a bank. The banknotes for this experiment from the bank are likely to have the continuous serial numbers and the banknotes were quite new which may not affect the training of the network. The data augmentation was used to expand the dataset and achieve ideal results.

Compared to the recognition of license plate numbers, the serial number region of banknotes only occupies 0.97% of the entire banknote area, the area of a single character is only 0.12% of the banknote area. In a complex background, the banknotes are only taken into account as a whole photo. Therefore, the serial number locating and serial number recognition belongs to the small object recognition. Hence, the YOLO model does not achieve good results; on the contrary, the SSD models perform well.

In order to extract enough features based on the limited dataset for character detection and recognition, the network needs to be further deepened and widened. Therefore, in this experiment, DenseNet as the primary model was adopted. By using SegLink-based feature extraction strategy, the good results were obtained. At the same time, a deep learning model based on ResNet152 and Inception v4 were employed as the comparative experiment. The results show that DenseNet performs better than ResNet 152 + Inception v4 based on the dataset of this experiment for feature extracting with the precision 95.807%.

6.2.2 Conclusion of Recognition Network

After the region detection, the region having serial number data is sent to the network for identification. In our experiment, first of all, CRNN network was employed and achieved a 96.48% precision. In spite that RNN is a very effective way to process series number because it can obtain long-term dependencies during the calculating, each step depends on the previous steps. Therefore, the RNN model is very dependent on the length of the input sequence, and the calculation time is very long. At the same time, RNN training is not easy which often encounters the problem of gradient vanishing and exploding.

In order to solve the above problems and improve the time-consuming problem, we used the modified CRNN with residual attention network. By replacing RNN and LSTM with CNN model, the training and recognition speed is effectively improved. At the same time, the attention model and ResNet are introduced, which assist the modified CRNN network and achieve the accuracy of 97.09% without significantly increasing the computational cost. It shows that the attention mechanism has played a pivot role in removing unnecessary interference.

6.2 Limitations

The proposed algorithms have been implemented successfully in this thesis for the recognition of the banknote serial number. However, there are still some limitations that should be improved in the near future.

6.2.1 Limitations of Detection Network

We see from the experiments that the detection network has the following shortcomings:

- (1) A large number of banknote images are needed as the training data. Collection and labelling work need a huge number of human labour supports.
- (2) In order to improve the accuracy, the primary network is very complicated, and the detection speed is not very fast. In particular, taken DenseNet as the primary network, training under Keras is slow, DenseNet does not take the benefits from reducing the number of parameters.

(3) Because the input size of the input layer is set to 512×512 and the resolution is small, the banknote only occupies a small part of the entire image, or it is too blurry, the network fails to detect the serial numbers.

(4) We did not develop the mobile-based neural network, real-time recognition of serial numbers cannot be implemented in this thesis at this moment.

6.2.2 Limitations of Recognition Network

We see from the experiments that the recognition network has the following drawbacks:

(1) A large number of banknote images are required. Collection and labelling work requires huge human labour and overheads.

(2) Limited to hardware resources, transfer learning was not added to the comparison experiments.

6.3 Future Work

6.3.1 Future Work of Detection Network

Our future work of currency detection is listed as follows:

(1) Because the input of this model uses the image resolution 512×512 , the serial number region is with a very small character accordingly, which leads to the problems during the serial number detection. In the next step, the input must be adjusted to 800×600 , 1024×768 , and other resolutions; meanwhile, the aspirations, such as 4: 3 or 16: 9, should be added as the input.

(2) We will prune the deep learning model to reduce the size of the model effectively, thus the training time will be reduced correspondingly.

(3) In the detection network, the features extracted from each layer are different. We can further investigate the corresponding features from the corresponding layer to implement the identification of the currency denomination.

(4) We will use MobileNet v3 as the basic network, make full use of the depthwise separable convolution, inverted residuals, linear bottlenecks, and network architecture search (NAS) to implement real-time recognition.

6.3.2 Future Work of Recognition Network

Our future work of currency recognition is listed as follows:

- (1) We will use the transfer learning to construct a new recognition network.
- (2) We will enhance our recognition by using large-angle rotated letters and deformed characters.
- (3) We will combine blockchain technology to achieve management and monitoring of banknote circulation based on the serial number.

References

- Al-Saffar, A. A. M., Tao, H., & Talab, M. A. (2017). Review of deep convolution neural network in image classification. In the International Conference on Radar, Antenna, Microwave, Electronics, and Telecommunications (ICRAMET) (pp. 26-31). IEEE
- Almazán, J., Gordo, A., Fornés, A., Valveny, E. (2014). Word spotting and recognition with embedded attributes. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 36(12), 2552-2566.
- Andrearczyk, V., & Whelan, P. (2016). Using filter banks in convolutional neural networks for texture classification. *Pattern Recognition Letters* 84(1), 63-69.
- Bharkad, A. (2013). Survey of currency recognition system using image processing. *International Journal of Computational Engineering Research*, 3(7).
- Bissacco, A., Cummins, M., Netzer, Y., & Neven, H. (2013). PhotoOCR: Reading text in uncontrolled conditions. In the IEEE International Conference on Computer Vision. (pp. 785-792)
- Busta, M., Neumann, L., & Matas, J. (2015). Fastext: Efficient unconstrained scene text detector. In the IEEE International Conference on Computer Vision. (pp. 1206-1214)
- Caruana, R., & Niculescu-Mizil, A. (2006). An empirical comparison of supervised learning algorithms. In the International Conference on Machine Learning. (pp. 161-168)
- Cawley, G. C., & Talbot, N. (2010). On over-fitting in model selection and subsequent selection bias in performance evaluation. *Journal of Machine Learning Research*, 11(Jul), 2079-2107.
- Chambers, J. Yan, W., Garhwal, A., Kankanhalli, M. (2014) Currency security and forensics: A survey *Multimedia Tools and Applications*, 74(11), 4013-4043.
- Yan, W., Chambers, J., Garhwal, A. (2015) An empirical approach for currency identification *Multimedia Tools and Applications*, 74, 4723–4733

- Chowdhury, M. A., & Deb, K. (2013). Extracting and segmenting container name from container images. *International Journal of Computer Applications*, 74(19): 18-22.
- Coates, A., Carpenter, B., Case, C., Satheesh, S., Suresh, B., Wang, T., Ng, A. (2011). Text detection and character recognition in scene images with unsupervised feature learning. In the *IEEE International Conference on Document Analysis and Recognition*. (pp. 440-445). IEEE.
- Dai, Y., Huang, Z., Gao, Y., Xu, Y., Chen, K., Guo, J., & Qiu, W. (2018). Fused text segmentation networks for multi-oriented scene text detection. In the *International Conference on Pattern Recognition (ICPR)*. (pp. 3604-3609). IEEE.
- Darshan, H., Gopalkrishna, M., & Hanumantharaju, M. (2015). Text detection and recognition using camera based images. In the *International Conference on Frontiers of Intelligent Computing: Theory and Applications (FICTA) 2014* (pp. 573-579). Springer, Cham.
- Deng, L., Hinton, G., & Kingsbury, B. (2013). New types of deep neural network learning for speech recognition and related applications: An overview. In the *IEEE International Conference on Acoustics, Speech and Signal Processing* (pp. 8599-8603). IEEE.
- DeSouza, G. (2002). Vision for mobile robot navigation: A survey. *IEEE PAMI*, 24(2), 237-267.
- Dvorin, Y., & Havosha, U. (2009). Method and device for instant translation: Google Patents.
- Elbayad, M., Besacier, L., & Verbeek, J. (2018). Pervasive Attention: 2D Convolutional Neural Networks for Sequence-to-Sequence Prediction. arXiv:1808.03867
- Epshtein, B., Ofek, E., & Wexler, Y. (2010). Detecting text in natural scenes with stroke width transform. In the *IEEE Conference on Computer Vision and Pattern Recognition* (pp. 2963-2970). IEEE
- Fan, E. (2000). Extended tanh-function method and its applications to nonlinear equations. *Physics Letters A*, 277(4-5), 212-218.
- Fernández, S., Graves, A., & Schmidhuber, J. (2008). Phoneme recognition in TIMIT with BLSTM-CTC. Technical Report No. IDSIA-04-08.

- Gehring, J., Auli, M., Grangier, D., Yarats, D., & Dauphin, Y. N. (2017). Convolutional sequence to sequence learning. In Proceedings of the 34th International Conference on Machine Learning-Volume 70 (pp. 1243-1252). JMLR. org.
- Girshick, R. (2015). Fast R-CNN. In Proceedings of the IEEE International Conference on Computer Vision (pp. 1440-1448).
- Gordo, A. (2015). Supervised mid-level features for word image representation. In the IEEE Conference on Computer Vision and Pattern Recognition (pp.2956-2964).
- Graves, A., Fernández, S., Gomez, F., & Schmidhuber, J. (2006). Connectionist temporal classification: labelling unsegmented sequence data with recurrent neural networks. In the International Conference on Machine Learning (pp. 369-376).
- Gu, J., Wang, Z., Kuen, J., Ma, L., Shahroudy, A., Shuai, B., Cai, J. (2018). Recent advances in convolutional neural networks. *Pattern Recognition*, 77, 354-377.
- Gupta, A., Vedaldi, A., & Zisserman, A. (2016). Synthetic data for text localisation in natural images. In IEEE Conference on Computer Vision and Pattern Recognition (pp. 2315-2324).
- Ham, Y., Kang, M. , Chung, H. , Park, R., & Park, G. (1995). Recognition of raised characters for automatic classification of rubber tires. *SPIE Optical Engineering* 34(1), 102-110.
- Han, J., & Moraga, C. (1995). The influence of the sigmoid function parameters on the speed of backpropagation learning. In the International Workshop on Artificial Neural Networks (pp. 195-201). Springer, Berlin, Heidelberg.
- Hawkins, D. (2004). The problem of overfitting. *Journal of Chemical Information and Computer Sciences*, 44(1), 1-12.
- He, K., Zhang, X., Ren, S., & Sun, J. (2016). Deep residual learning for image recognition. In the IEEE Conference on Computer Vision and Pattern Recognition, pp. 770-778.2016.
- He, Z., Liu, J., Ma, H., & Li, P. (2005). A new automatic extraction method of container identity codes. *IEEE Transactions on Intelligent Transportation Systems*, 6(1), 72-78.
- Hendrycks, D., & Gimpel, K. (2016). Gaussian error linear units (gelus). *arXiv:1606.08415*.

- Hinton, G., Osindero, S., & Teh, Y. (2006). A fast learning algorithm for deep belief nets. *Neural computation*, 18(7), 1527-1554.
- Hu, W., Huang, Y., Wei, L., Zhang, F., & Li, H. (2015). Deep convolutional neural networks for hyperspectral image classification. *Journal of Sensors*, 2015. (pp. 34-39).
- Huang, G., Liu, S., Van der Maaten, L., & Weinberger, K. (2018). CondenseNet: An efficient densenet using learned group convolutions. In the *IEEE Conference on Computer Vision and Pattern Recognition* (pp. 2752-2761).
- Huang, G., Liu, Z., Van Der Maaten, L., & Weinberger, K. (2017). Densely connected convolutional networks. In the *IEEE Conference on Computer Vision and Pattern Recognition* (pp. 4700-4708).
- Huang, J., Rathod, V., Sun, C., Zhu, M., Korattikara, A., Fathi, A., Guadarrama, S. (2017). Speed/accuracy trade-offs for modern convolutional object detectors. In the *IEEE Conference on Computer Vision and Pattern Recognition* (pp. 7310-7311).
- Huang, W., Lin, Z., Yang, J., & Wang, J. (2013). Text localization in natural images using stroke feature transform and text covariance descriptors. In the *IEEE International Conference on Computer Vision* (pp. 1241-1248).
- Jain, A. K., & Yu, B. (1998). Automatic text location in images and video frames. *Pattern Recognition*, 31(12), 2055-2076.
- Ji, H., Liu, Z., Yan, W., Klette, R. (2019) Early diagnosis of Alzheimer's disease based on selective kernel network with spatial attention. *ACPR 2* (1), 503-515
- Ji, H., Yan, W., Klette, R. (2019) Early diagnosis of Alzheimer's disease using deep learning. *ICCCV*, pp.87-91.
- Jia, Y., Shelhamer, E., Donahue, J., Karayev, S., Long, J., Girshick, R., Darrell, T. (2014). Caffe: Convolutional architecture for fast feature embedding. In the *ACM International Conference on Multimedia* (pp. 675-678).
- Karatzas, D., Gomez-Bigorda, L., Nicolaou, A., Ghosh, S., Bagdanov, A., Iwamura, M., Lu, S. (2015). Competition on robust reading. In the *International Conference on Document Analysis and Recognition (ICDAR)* (pp. 1156-1160). *IEEE*

- Kim, Y. J. (2014). Convolutional neural networks for sentence classification. In the International Conference on Empirical Methods in Natural Language Processing (EMNLP)
- Kong, T., Yao, A., Chen, Y., & Sun, F. (2016). HyperNet: Towards accurate region proposal generation and joint object detection. In the IEEE Conference on Computer Vision and Pattern Recognition (pp. 845-853).
- Krogh, A. (2008). What are artificial neural networks? *Nature biotechnology*, 26(2), 195.
- Krogh, A., & Vedelsby, J. (1995). Neural network ensembles, cross validation, and active learning. In the Advances in neural information processing systems (pp. 231-238).
- Kumar, A., Sangwan, S. R., Arora, A., Nayyar, A., & Abdel-Basset, M. (2019). Sarcasm detection using soft attention-based bidirectional long short-term memory model with convolution network. *IEEE Access* 7, 23319-23328.
- Le Cun, Y., Matan, O., Boser, B., Denker, J. S., Henderson, D., Howard, R. E., Baird, H. S. (1990). Handwritten zip code recognition with multilayer networks. In the International Conference on Pattern Recognition (pp. 35-40).
- LeCun, Y., Bengio, Y., & Hinton, G. (2015). Deep learning. *Nature* 521(7553), 436-444.
- Lee, C.-Y., & Osindero, S. (2016). Recursive recurrent nets with attention modeling for OCR in the wild. In the IEEE Conference on Computer Vision and Pattern Recognition(pp.2231-2239).
- Lee, J., Lee, P., Lee, S., Yuille, A., & Koch, C. (2011). Adaboost for text detection in natural scene. In the International Conference on Document Analysis and Recognition(pp. 429-434).
- Lee, S., Kim, J. H. (2013). Integrating multiple character proposals for robust scene text extraction. *Journal of Image and Vision Computing* 31(11), 823-840.
- Li, J., Cheng, J., Shi, J., & Huang, F. (2012). Brief introduction of back propagation (BP) neural network algorithm and its improvement. In the *Advances in computer science and information engineering*, pp. 553-558.

- Li, P., Nguyen, M., Yan, W. (2018) Rotation correction for license plate recognition. ICCAR.
- Liu, C. (2007). Normalization-cooperated gradient feature extraction for handwritten character recognition. *IEEE Transactions on Pattern Analysis and Machine Intelligence* 29(8): 1465-1469.
- Liu, W., Anguelov, D., Erhan, D., Szegedy, C., Reed, S., Fu, C., & Berg, A. (2016). SSD: Single shot multibox detector *Springer*. In European Conference on Computer Vision (pp. 21-37).
- Liu, W., Wen, Y., Yu, Z., & Yang, M. (2016). Large-margin softmax loss for convolutional neural networks. In the ICML (Vol. 2, No. 3, p. 7).
- Liu, X., & Samarabandu, J. (2005). An edge-based text region extraction algorithm for indoor mobile robot navigation. In the IEEE International Conference Mechatronics and Automation, 2005 (Vol. 2, pp. 701-706). IEEE.
- Liu, X., & Samarabandu, J. (2005). A simple and fast text localization algorithm for indoor mobile robot navigation. *SPIE Image Processing: Algorithms and Systems IV* (Vol. 5672, pp. 139-150). International Society for Optics and Photonics.
- Liu, X., Nguyen, M., Yan, W. (2019) Vehicle-related scene understanding using deep learning. ACPR'19 Workshop AAPS.
- Liu, Y., & Jin, L. (2017). Deep matching prior network: Toward tighter multi-oriented text detection. In the IEEE Conference on Computer Vision and Pattern Recognition (pp. 1962-1969).
- Liu, Z., Yan, W., Yang, B. Image denoising based on a CNN model. ICCAR, pp.389-393
- Lu, J., Nguyen, M., Yan, W. (2018) Pedestrian detection using deep learning. IEEE AVSS.
- Mishra, A., Alahari, K., & Jawahar, C. (2011). An MRF model for binarization of natural scene text. In the International Conference on Document Analysis and Recognition (pp. 11-16).
- Mishra, A., Alahari, K., & Jawahar, C. (2012). Scene text recognition using higher order language priors. In the British Machine Vision Conference, 127.1--127.11.

- Mohamad, N. S., Hussin, B., Shibghatullah, A. , & Basari, A. (2014). Banknote authentication using artificial neural network. *Science International*, 1865-1868.
- Montufar, G. F., Pascanu, R., Cho, K., & Bengio, Y. (2014). On the number of linear regions of deep neural networks. In the Advances in Neural Information Processing Systems (pp. 2924-2932).
- Nair, V., & Hinton, G. E. (2010). Rectified linear units improve restricted boltzmann machines. In the International Conference on Machine Learning (ICML-10) (pp. 807-814).
- Neumann, L., & Matas, J. (2010). A method for text localization and recognition in real-world images. In the Asian Conference on Computer Vision (pp. 770-783).
- Nomura, S., Yamanaka, K., Katai, O., Kawakami, H., & Shiose, T. (2005). A novel adaptive morphological approach for degraded character image segmentation. *Pattern Recognition* 38(11):1961-1975.
- Parkinson, C., Jacobsen, J. , Ferguson, D. , & Pombo, S. (2016). Instant translation system: Google Patents (US9507772B2).
- Quy Phan, T., Shivakumara, P., Tian, S., & Lim Tan, C. (2013). Recognizing text with perspective distortion in natural scenes. In the IEEE International Conference on Computer Vision (pp. 569-576).
- Redmon, J., Divvala, S., Girshick, R., & Farhadi, A. (2016). You only look once: Unified, real-time object detection. In the IEEE Conference on Computer Vision and Pattern Recognition (pp. 779-788).
- Ren, S., He, K., Girshick, R., & Sun, J. (2015). Faster R-CNN: Towards real-time object detection with region proposal networks. In the Advances in neural information processing systems(pp. 91-99).
- Ren, Y., Nguyen, M., Yan, Y. Real-time recognition of series seven New Zealand banknotes. *IJDCF* 10 (3), 50-66
- Rodriguez-Serrano, J. A., Gordo, A., & Perronnin, F. (2015). Label embedding: A frugal baseline for text recognition. *International Journal of Computer Vision*, 113(3), 193-207.

- Roy, P. , Pal, U., Lladós, J., & Delalandre, M. (2009). Multi-oriented and multi-sized touching character segmentation using dynamic programming. In the International Conference on Document Analysis and Recognition (pp. 11-15).
- Ryan, M., & Hanafiah, N. (2015). An examination of character recognition on ID card using template matching approach. *Procedia Computer Science* 59(520-529).
- Scherer, D., Müller, A., & Behnke, S. (2010). Evaluation of pooling operations in convolutional architectures for object recognition. In the International Conference on Artificial Neural Networks (pp. 92-101).
- Schroth, G., Hilsenbeck, S., Huitl, R., Schweiger, F., & Steinbach, E. (2011). Exploiting text-related features for content-based image retrieval. In the IEEE International Symposium on Multimedia (pp. 77-84).
- Schulz, R., Talbot, B., Lam, O., Dayoub, F., Corke, P., Upcroft, B., & Wyeth, G. (2015). Robot navigation using human cues: A robot navigation system for symbolic goal-directed exploration. In the IEEE International Conference on Robotics and Automation (ICRA) (pp. 1100-1105). IEEE.
- Sengupta, A., Ye, Y., Wang, R., Liu, C., & Roy, K. (2019). Going deeper in spiking neural networks: VGG and residual architectures. *Neuroscience*, 13.
- Shah, A., Kadam, E., Shah, H., Shinde, S., & Shingade, S. (2016). Deep residual networks with exponential linear unit. In Proceedings of the Third International Symposium on Computer Vision and the Internet (pp. 59-65).
- Shen, D., Xin, C., Nguyen, M., Yan, W. (2018) Flame detection using deep learning. ICCAR.
- Shen, Y., Yan, W. (2018) Blind spot monitoring using deep learning. IVCNZ.
- Sheng, F., Chen, Z., & Xu, B. (2019). NRTR: A no-recurrence sequence-to-sequence model for scene text recognition. In the International Conference on Document Analysis and Recognition (ICDAR)(pp. 781-786). IEEE.
- Sheshadri, K., & Divvala, S. K. (2012). Exemplar Driven Character Recognition in the Wild In the BMVC. (pp. 1-10).

- Shi, B., Bai, X., & Belongie, S. (2017). Detecting oriented text in natural images by linking segments. In the IEEE Conference on Computer Vision and Pattern Recognition (pp. 2550-2558).
- Shi, B., Bai, X., Yao, C. (2016). An end-to-end trainable neural network for image-based sequence recognition and its application to scene text recognition. *IEEE PAMI* 39(11), 2298-2304.
- Shi, B., Wang, X., Lyu, P., Yao, C., & Bai, X. (2016). Robust scene text recognition with automatic rectification. In the IEEE Conference on Computer Vision and Pattern Recognition (pp. 4168-4176).
- Shi, C., Wang, C., Xiao, B., Zhang, Y., Gao, S., & Zhang, Z. (2013). Scene text recognition using part-based tree-structured character detection. In the IEEE Conference on Computer Vision and Pattern Recognition (pp. 2961-2968).
- Shin, H., Roth, H., Gao, M., Lu, L., Xu, Z., Nogues, I., Summers, R. (2016). Deep convolutional neural networks for computer-aided detection: CNN architectures, dataset characteristics and transfer learning. *IEEE Transactions Medical Imaging*, 35(5), 1285-1298.
- Shivakumara, P., Bhowmick, S., Su, B., Tan, C. L., & Pal, U. (2011). A new gradient based character segmentation method for video text recognition. In the International Conference on Document Analysis and Recognition (pp. 126-130).
- Simonyan, K., & Zisserman, A. (2014). Very deep convolutional networks for large-scale image recognition, In ICLR.
- Storkey, A. (2009). When training and test sets are different: characterizing learning transfer. *Dataset Shift in Machine Learning*, 3-28.
- Szegedy, C., Ioffe, S., Vanhoucke, V., & Alemi, A. (2017). Inception-v4, Inception-ResNet and the impact of residual connections on learning. In the AAAI Conference on Artificial Intelligence (pp. 31-32).

- Szegedy, C., Liu, W., Jia, Y., Sermanet, P., Reed, S., Anguelov, D., Rabinovich, A. (2015). Going deeper with convolutions. In the IEEE Conference on Computer Vision and Pattern Recognition (pp. 1-9).
- Tao, C., Gao, S., Shang, M., Wu, W., Zhao, D., & Yan, R. (2018). Get The Point of My Utterance! Learning Towards Effective Responses with Multi-Head Attention Mechanism. In the IJCAI (pp. 4418-4424).
- Tian, Z., Shen, C., Chen, H., & He, T. (2019). FCOS: Fully convolutional one-stage object detection. In the IEEE International Conference on Computer Vision (pp. 9627-9636).
- Tsai, S. S., Chen, H., Chen, D., Schroth, G., Grzeszczuk, R., & Girod, B. (2011). Mobile visual search on printed documents using text and low bit-rate features. In the IEEE International Conference on Image Processing (pp. 2601-2604).
- Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., Polosukhin, I. (2017). Attention is all you need. In the Advances in neural information processing systems (pp. 5998-6008).
- Wakahara, T., & Kita, K. (2011). Binarization of color character strings in scene images using k-means clustering and support vector machines. In the International Conference on Document Analysis and Recognition (pp. 274-278).
- Wang, F., Jiang, M., Qian, C., Yang, S., Li, C., Zhang, H., Tang, X. (2017). Residual attention network for image classification. In the IEEE Conference on Computer Vision and Pattern Recognition (pp. 3156-3164).
- Wang, J. Bacic, B. Yan, W. An effective method for plate number recognition. *Multimedia Tools and Applications*, 77 (2), 1679-1692
- Wang, K., Babenko, B., & Belongie, S. (2011). End-to-end scene text recognition. In the International Conference on Computer Vision (pp. 1457-1464).
- Wang, T., Wu, D. J., Coates, A., & Ng, A. (2012). End-to-end text recognition with convolutional neural networks. In the International Conference on Pattern Recognition (ICPR2012), pp. 3304-3308.

- Weinman, J., Learned-Miller, E., & Hanson, A. (2007). Fast lexicon-based scene text recognition with sparse belief propagation. In the International Conference on Document Analysis and Recognition (ICDAR 2007) (Vol.2, pp.979-983).IEEE
- Wenhong, L., Wenjuan, T., Xiyan, C., & Zhen, G. (2010). Application of support vector machine (SVM) on serial number identification of RMB. In the World Congress on Intelligent Control and Automation (pp. 34-36).
- Xu Q., Lam Y.P., L., & Suen Y., C. (2003). Automatic segmentation and recognition system for handwritten dates on Canadian bank cheques. In the International Conference on Document Analysis and Recognition. (pp. 12-16).
- Yadav, S., & Shukla, S. (2016). Analysis of k-fold cross-validation over hold-out validation on colossal datasets for quality classification. In the IEEE International Conference on Advanced Computing (IACC), pp. 78-83.
- Yan, W. (2019). *Introduction to Intelligent Surveillance: Surveillance Data Capture, Transmission, and Analytics*. Springer, (pp. 9-15).
- Yan, Z., Piramuthu, R., Jagadeesh, V., Di, W., & Decoste, D. (2019). HD-CNN: Hierarchical deep convolutional neural network for image classification. In the ICCV (pp. 3-9)..
- Yao, C., Bai, X., Liu, W., Ma, Y., & Tu, Z. (2012). Detecting texts of arbitrary orientations in natural images. In the IEEE Conference on Computer Vision and Pattern Recognition (pp. 1083-1090).
- Yao, C., Bai, X., Shi, B., & Liu, W. (2014). Strokelets: A learned multi-scale representation for scene text recognition. In the IEEE Conference on Computer Vision and Pattern Recognition (pp. 4042-4049).
- Yi, C., & Tian, Y. (2011). Text string detection from natural scenes by structure-based partition and grouping. *IEEE Transactions on Image Processing*, 20(9), 2594-2605.
- Yin, X., Yin, X., Huang, K., Hao, H. (2013). Robust text detection in natural scene images. *IEEE Transactions on Pattern Analysis and Machine Intelligence* 6(5), 970-983.
- Yin, X., Goudriaan, J., Lantinga, E. A., Vos, J., & Spiertz, H. (2003). A flexible sigmoid function of determinate growth. *Annals of Botany* 91(3), 361-371.

- Zhang, Q. (2018). *Currency Recognition Using Deep Learning*. Auckland University of Technology (pp. 12-25).
- Zhang, Q., Yan, W. (2018) Currency recognition using deep learning. IEEE AVSS 2018.
- Zhang, Y., Tian, Y., Kong, Y., Zhong, B., & Fu, Y. (2018). Residual dense network for image super-resolution. In the IEEE conference on computer vision and pattern recognition (pp. 2472-2481).
- Zhao, T., Zhao, J., Zheng, R., & Zhang, L. (2010). Study on RMB number recognition based on genetic algorithm artificial neural network. In the International Congress on Image and Signal Processing (Vol. 4, pp. 1951-1955).
- Zheng, K., Yan, W., Nand, P. Video dynamics detection using deep neural networks. IEEE Transactions on Emerging Topics in Computational Intelligence, 2(3): 224 - 234.
- Zhiwei, Z., Linlin, L., & Lim, T. (2010). Edge based binarization for video text images. In the International Conference on Pattern Recognition (pp. 133-136). IEEE.
- Zhong, Z., Zheng, L., Kang, G., Li, S., & Yang, Y. (2020). Random erasing data augmentation. In the AAAI.
- Zhou, X., Yao, C., Wen, H., Wang, Y., Zhou, S., He, W., & Liang, J. (2017). EAST: An efficient and accurate scene text detector. In the IEEE conference on Computer Vision and Pattern Recognition (pp. 5551-5560).
- Zhu, Y., Yao, C., & Bai, X. (2016). Scene text detection and recognition: Recent advances and future trends. *Frontiers of Computer Science* 10(1), 19-36.