

# Rotation Correction for License Plate Recognition

Piyuan Li

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

2017

School of Engineering Computer and Mathematical Sciences



# Abstract

The focus of this thesis is on reliable and robust license plate recognition (LPR). The technology is currently in operation that both the quantity and quality-based approaches are needed. The entire procedure of license plate recognition consists of six steps: (1) image acquisition, (2) image pre-processing, (3) plate locating, (4) character segmentation, (5) character recognition, (6) output.

However, when a road is uneven and slant, a vehicle will be shaky while running; consequently, the plate is also unstable and tilted with an angle of rotation at this moment. Meanwhile, a surveillance camera is very difficult to capture an effective image so that the plate is hard to be located and recognized.

Due to this existing problem, the contributions of this thesis are: (1) plate tilt correction using Hough transform; (2) GNN-based plate number recognition. The novelties of this thesis are to improve the robustness and reliability of license plate recognition through literature review as well as evaluate those existing technologies.

Keywords: plate number recognition, image rotation, image correction, GNN, character recognition

# Table of Contents

Abstract .....	I
List of Figures .....	V
List of Tables .....	VII
List of Algorithms .....	VIII
Attestation of Authorship .....	IX
Acknowledgments .....	X
Chapter 1 Introduction .....	1
1.1 Background .....	2
1.2 Principle .....	3
1.3 Objectives.....	5
1.4 Structure of this thesis .....	6
Chapter 2 Literature review.....	8
2.1 License plate recognition system .....	9
2.2 Applications .....	9
2.3 Triggered methods .....	11
2.4 Genetic neural network .....	13
2.4.1 The definition of genetic algorithm.....	14
2.4.2 The performance of genetic algorithm .....	15
2.4.3 The steps of genetic algorithm .....	15
2.5 Distance classifier .....	16
2.5.1 Euclidean distance classifier .....	17
2.5.2 Cosine distance classifier .....	17
2.6 Neural network classifier .....	17

2.6.1 Neural network.....	18
2.6.2 Characteristics of artificial neural network .....	19
2.6.3 The main research directions of artificial neural network.....	20
2.7 Related work .....	21
Chapter 3 Methodology.....	22
3.1 Introduction .....	23
3.2 Genetic neural network .....	23
3.3 The research question.....	26
3.4 Data acquisition.....	27
3.4.1 Data collection for experiments .....	27
3.4.2 Templates and Datasets for GNN.....	27
3.5 Evaluation methods.....	28
Chapter 4 Design and implementation .....	30
4.1 LPR Design.....	31
4.2 LPR Implementation .....	34
4.2.1 Image pre-processing .....	34
4.2.2 Image correction.....	35
4.2.2.1 Hough transform .....	37
4.2.2.2 The implementation of straight line detection.....	39
4.2.2.3 Straight line detection based on Hough transform .....	41
4.2.3 Image segmentation.....	42
4.2.3.1 Cropping.....	42
4.2.3.2 Edge extraction using morphological operations .....	43
4.2.3.3 Characters segmentation.....	43
4.2.4 Character segmentation .....	44

4.2.5 Character recognition .....	45
Chapter 5 Experimental results and discussions .....	48
5.1 Introduction .....	49
5.2 Image extraction .....	49
5.3 Edge detection .....	50
5.4 Straight line detection.....	52
5.5 Character rotation.....	52
5.6 Character region cropping .....	54
5.7 Character segmentation and recognition .....	54
5.8 Accuracy .....	56
5.9 Discussions.....	58
5.10 Expanded experiments .....	60
5.11 Accuracy of the expanded experiments.....	62
Chapter 6 Conclusion and future work .....	64
6.1 Conclusion.....	65
6.2 Limitations and future work.....	66
References .....	68

# List of Figures

Fig 2.1 The seven steps of LPR.....	9
Fig 2.2 The flowchart of genetic algorithm .....	16
Fig 2.3 Neural network structure.....	18
Fig 3.1 The structure of chromosome.....	24
Fig 4.1 The workflow of license plate recognition.....	31
Fig 4.2 The original video frame.....	35
Fig 4.3 The image after binarization.....	35
Fig 4.4 The tilt in vertical direction.....	36
Fig 4.5 The tilt in horizontal direction.....	36
Fig 4.6 The tilt in both vertical and horizontal directions.....	37
Fig 4.7 The Cartesian coordinate system.....	38
Fig 4.8 The polar coordinate system.....	38
Fig 4.9 The basic schematic diagram of Hough transform .....	39
Fig 4.10 The accumulator .....	40
Fig 4.11 Original image space.....	40
Fig 4.12 Hough transform space.....	41
Fig 4.13 The result of plate correction .....	42
Fig 4.14 The cropped image.....	44
Fig 4.15 The recognition result.....	47
Fig 5.1 The tilt frames of a video.....	49

Fig 5.2 The tilt frames from four videos .....	50
Fig 5.3 The results of edge detection from video frames.....	51
Fig 5.4 The results of straight line detection.....	52
Fig 5.5 The results of rotation correction.....	53
Fig 5.6 The cropped images.....	54
Fig 5.7 Character segmentation and recognition.....	55
Fig 5.8 Recognition results of four sample videos.....	56
Fig 5.9 The tilt frames of a video.....	61
Fig 5.10 The results of edge detection from video frames.....	61
Fig 5.11 The results of rotation correction.....	61
Fig 5.12 Character recognition result.....	62

# List of Tables

Table 2.1 The milestones of neural computing development.....	19
Table 5.1 Correct rate of image rotation correction .....	57
Table 5.2 Accuracy of character recognition using BP and GNN .....	57
Table 5.3 Accuracy of character recognition using BP and GNN for real plates.....	62

# List of Algorithms

Algorithm 3.1 Genetic neural network.....	26
Algorithm 4.1 Character recognition.....	46

# Attestation of Authorship

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



Date 08 October 2017

# Acknowledgments

During this one-year Master degree study, a raft of people have kindly given me a large amount of help in various aspects, especially my supervisor Dr Wei Qi Yan. He imparted me abundant useful knowledge and gave me a great deal of advises to complete my thesis work. Without his guidance and encouragement, this thesis is impossible to be finished. Therefore, I would like to express my deepest gratitude to him. In addition, I also intend to appreciate all my classmates and administration staff who have delivered their help and support to me. I would like to say thanks.

Piyuan Li

Auckland, NZ

08 October 2017

# Chapter 1

## Introduction

*In the first chapter of this thesis, we will give the background, principle and motivation of this research work. After shown the relevant information of the number plate recognition, the objectives will be presented. Finally, the structure of this thesis will be listed.*

## 1.1 Background

Nowadays, with the development of science and technique, computer vision technology is developed rapidly. More and more related techniques and products have been well created, a wide range of applications have been employed to commercial products. The license plate recognition for vehicles is one of the most important achievements which can bring much convenience in traffic control, monitoring and surveillance (Lan et al., 2013).

The license plate recognition requires extraction and identification of the moving number plate from complex background (Qiu, Zhu, Wei, & Yu, 2016). Through license plate detection, image processing, feature extraction, character recognition in computer vision, it is needed to recognize the brand, color and other information of a vehicle (Muammer, 2014). The current techniques of the recognition for letters and numbers can reach nearly the rate at 99.7%; especially, the recognition rate for Chinese characters can get almost 99%.

The plate recognition (VLPR : Vehicle License Plate Recognition) is a technique that can detect the license plate of a vehicle on monitored road and automatically extract the plate information (including Chinese characters, English characters, Arabia Numbers and plate color) (Tian, Wang, Wang, Liu, & Xia, 2015). License plate recognition is playing important role in modern intelligent transportation systems which have already been widespread used. It is based on digital image processing, pattern recognition, computer vision and so on (Türkyilmaz & Kaçan, 2017). Analyzing the vehicle videos or image sequences which are taken by using cameras obtains the number plate of each vehicle so as to complete the identification process. The vehicle number plate is the only “identity” of a vehicle (Xu, Zhang, Lu, Tian, & Wang, 2011). License plate automatic recognition can procure the automatic registration and validation of the vehicle without any special requirements.

By utilizing license plate recognition, we can implement parking management, traffic flow control, vehicle positioning, vehicle anti-theft, highway over-speed management, red-light traffic control, highway toll stations, and so on. Plate number recognition benefits us in traffic and public safety, preventing traffic jams and traffic management.

## 1.2 Principle

### *Identification process*

License plate recognition is a kind of technology of pattern recognition using dynamic videos or static images of a vehicle to recognize the characters, numbers and colors automatically (Guilherme et al., 2017). The hardware normally includes triggering equipment (monitoring whether the vehicle enters the field of view), cameras, lighting facilities, image and video storage, recognition processor (such as a computer). The software encloses the algorithms of license plate locating, license plate character segmentation and optical character recognition (Xia, Chen, Lu, Wang, & Xu, 2015). The license plate recognition also has the function of judging whether there is a car by using the captured video frames. A complete license plate recognition consists of image acquisition, vehicle detection, license plate recognition and other parts (Wang & Liu, 2015). The unit of image acquisition is triggered when the software detects a vehicle arrives, then captures and collects the current video frames. The license plate recognition deals with video frames, locates the position of a license plate, recognizes the characters from license plates and finally outputs the results of recognition (Anagnostopoulos, 2014).

### *Vehicle detection*

Video vehicle detection requires very high processing speed and a good algorithm through image acquisition without frame dropping (Huang & Huang, 2011). If the computing is slow, it will lead to frame dropping so that it is unable to detect the high-speed running vehicles. At the same time, it is difficult to guarantee starting the recognition at a suitable position in favor of the identification that will affect the recognition rate (Meng, 2013). Therefore, there are a great deal of technical difficulties

to combine together the video vehicle detection and license plate automatic identification (Castellanos, Kalva, Marques, & Furht, 2010).

### *Number recognition*

In order to carry out the license plate recognition, it needs the following basic steps:

Step 1. Locate the position of a license plate.

Step 2. Segment the characters out of the license plate.

Step 3. Recognize the segmented characters.

#### *1) License plate location*

In natural environment, the background of an automobile image is complex and the illumination is stable (Asif, Chun, Hussain, Fareed, & Khan, 2017). To accurately determine the region of license plate in the natural background is the key problem for identification. Firstly, a wide range of related search of the captured video images will be applied to find several regions in accordance with its features of the vehicle license plate. Then, these candidate regions are given for further analysis and evaluation (Cao, Zhang, Chen, & Huang, 2014). Finally, the best region is selected as the plate region and will be separated from the image.

#### *2) License plate character segmentation*

After completed the plate region locating, the license plate is split into single characters so as to identify the segmented characters. Generally, vertical projection method is used for character segmentation (Bulan, Kozitsky, Ramesh, & Shreve, 2017). The projection of these characters in the vertical direction is inevitable near the local minima in the character gaps; this position should meet the character writing requirement separated by using a space, the size of license plate character and other conditions or constraints (Li & Connan, 2010). The captured images under the complex environment in character

segmentation have decent results through using vertical projection method (Khan, Thainimit, Kumazawa, Marukatat, & Ghaffar, 2017).

*3) The methods of license plate character recognition based on template matching and artificial neural networks.*

For the template matching, firstly the binarization will be applied to the segmented characters which will be resized to match those template characters in the collected datasets; finally, the best match will be selected as the output (Li & Li, 2015).

For artificial neural network, it is based on two ways: one is feature extraction and the features are used to train the neural networks. The other is to import the image into a neural network directly so as to implement the automatic feature extraction and recognize the plate by using the network (Saleem, Muazzam, Tahir, & Farooq, 2016).

In practical applications, the rate of license plate recognition is closely related to the quality of images and videos (Dallalzadeh & Guru, 2010). The license quality will be affected by various reasons, such as rusting, paint peeling, font fade, license plate obscured, license plate tilt, bright reflective, multiple licenses, fake license and so on.

The actual video-taking process also may be affected by brightness, shooting mode, vehicle speed and other factors. These reasons may reduce the rate of license plate recognition; thus, they are regarded as the difficulties or challenges of plate number recognition. In order to improve the recognition rate, in addition, to continuously improve the recognition algorithm, various lighting and traffic conditions have been utilized to make the captured images useful for final recognition (Kim, Song, Lee, & Ko, 2016).

### 1.3 Objectives

First of all, in this thesis we will introduce applications and methods. It shows that number plate recognition has a wide range of use in our community and can offer more benefits and convenience to those potential users.

Then, related work and useful methods about the plate number recognition will be presented, such as character segmentation, computer vision technique, plate detection and location, neural network algorithm and so on (Sarker, Yoon, & Park, 2014).

Finally, a myriad of problems and challenges will be focused on. The first one is the rotation problem of a license plate, the other is plate recognition problem (Zhao, Yuan, Bai, Liu, & Fang, 2011). When road surface is uneven, the vehicle is shaking while running. So, the plate is also shaking and slant along with the vehicle. At this time, a surveillance camera is too difficult to catch a clear and accurate image so that the plate is hard to be located and recognized (Lalimi, Ghofrani, & McLernon, 2013). To resolve this problem, we used a group of methods to adjust the slant images and stabilize them.

On the other hand, the position of each camera is different. So, when the vehicle appears in the camera, it may come from various directions (Saini & Saini, 2017). For this reason, the position of the plate in the camera image has too many possibilities. It is hard to make sure the plate is horizontal. Therefore, the plate correction appears very important (Fang & Dai, 2012). In addition, various algorithms of character recognition will be tested in our experiment. We compare the accuracy of different algorithms and verify their performances.

In nutshell, we have two contributions in this thesis: one is the automatic correction of plate rotation, the other is the improvement of character recognition.

## 1.4 Structure of this thesis

In Chapter 2, we will discuss the literature related to the algorithms which are used in license plate recognition. The useful information about the plate number recognition will also be presented. At last, we will focus on our research problems and finally solve them.

In Chapter 3, we will explain the methodology of this thesis. The research questions will be mainly discussed in this chapter, the corresponding solutions can be proposed. Finally, the experimental process and settings will be presented.

In Chapter 4, the results of experiments will be compared with the related work. In Chapter 5, the solutions for the research problem will be tested and evaluated. Finally, in Chapter 6, the limitations and challenges of this thesis will be described. Our future work will be envisioned.

## Chapter 2

### Literature review

*In this chapter, a great deal of applications will be presented. It shows us the benefits of number plate recognition. Followed the steps, corresponding methods and existing work will be presented. In conclusion, a majority of considered algorithms and technology are evaluated for achieving the objectives of this thesis.*

## 2.1 License plate recognition system

In this thesis, License Plate Recognition (LPR) will be the focus on. Therefore, first of all, the mechanism of LPR should be introduced. There are six basic steps: image extraction, image pre-processing, plate locating, character segmentation, character recognition and output results (Ahmed, 2015). These six steps are basic ones in traditional LPR as shown in Fig 2.1. However, the plate which is captured by using a camera may not be along with the horizontal direction (Zhang, Li, Yan, & Liu, 2016). So, in our experiment, the special step: plate correction should be added after the plate locating. Therefore, in this research project, our full workflow includes seven steps. The step plate correction will be the new one, it will be deeply investigated.

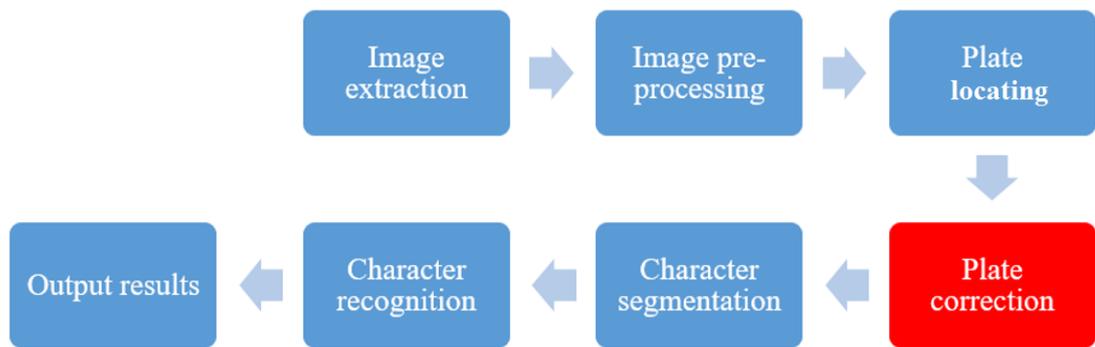


Fig 2.1 The seven steps of LPR

## 2.2 Applications

### 1) Monitoring and alarming

For the vehicles on a black list, such as those wanted vehicles of police, the lost or not registered vehicles, or the vehicles escaped from an accident, the plate should be entered into the motor database; the license plate will be recognized; the results will be compared with the list (Akoz & Karsligil, 2010). Once the specified vehicle is found, an alarming will be made immediately. The error rate of plate number recognition is extremely low. The task can be completed without affecting any ordinary traffic. Actually, the drivers

will not sense the monitoring, it has high confidentiality (Saha, Basu, & Nasipuri, 2015). The applications will greatly improve the efficiency of traffic monitoring and alarm making.

### *2) Over speeding*

Number plate recognition combined with other device can be used for vehicle penalties on highway. Setting speed-monitoring site on road, the plates of speeding vehicles could be acquired and identified; through the license plates, those pictures of broke-rule vehicles will be sent to all checkpoints or monitoring centers. With the aid of license plate recognition, we can quickly find the speeding vehicles. Compared with the traditional way of over-speed monitoring, this application can save human power, reduce the work of police and security.

### *3) Road access management*

When the equipment of license plate recognition has been installed at road entrance, it could record the license plate number and access time of a vehicle and manage the vehicles automatically. In a car park, the application can automatically calculate the number of park lots available and give tips to those further-coming drivers. It can greatly save human labor and improve the management efficiency.

When the application is used in intelligent community, it can automatically determine whether the vehicle belongs to the community, timing fees for the vehicles from outside (Baran, Rusc, &Fornalski, 2016). This application can also be integrated with vehicle scheduling systems, objectively record the vehicle access.

When entered the designated license plate information into a database, the access vehicle license plate could be found in the database. It can be used for special units, such as military administrative zones, security facilities, key units, toll road or bridge fee zone, senior residential areas and so on (Zhang, Wang, Fu, Lu, & Xu, 2011).

### *4) Highway management*

Once installed license plate recognition in various high-speed entrances and exits, when a vehicle is passing through the entrance, the plate number could be recognized and saved into the charging system. When the vehicle at an exit, the plate number could be recognized again; by recalling the plate information at the entrance, the relevant fees and costs could be automatically reduced. This application can stop or prevent cheating. The use of license plate recognition is the basic method to resolve such problems (Zhang, Xu, Zhu, Liu, & Lu, 2010).

#### *5) Inference of traffic conditions*

In traffic management, the average time on a road can be treated as a parameter to measure the road congestion. The traffic monitoring center can calculate the average on-road time to predict the status of traffic waiting and congestion (Cui, Li, Chen, & Li, 2011).

#### *6) Automatic registration of license plate number*

Motor control department needs to deal with a large number of illegal vehicles every day. The workload is huge and this easily causes misjudgment. By using automatic recognition, it can greatly improve the processing efficiency and reduce work intensity (Janowski et al., 2014). This could be applied to the electronic police system, traffic monitoring system, and so on.

## 2.3 Triggered methods

License plate recognition has two trigger modes, one is the peripheral triggering, the other is the video triggering.

Peripheral triggering refers to the coil, infrared, or other vehicle detectors which detect the vehicles passing through traffic signals. After received the signal a vehicle triggered, a computer collects vehicle images and recognizes the license plate automatically as well as for subsequently processing. This method has the advantages of high trigger rate and stable performance.

Video trigger mode refers to the license plate recognition using a sequence of dynamic motion for image analysis and processing through checking out vehicle motion on road in real time (Bagheri, Madani, Sahba, & Sahba, 2011). Once triggered, the computer will capture the images of an on-road vehicle, further steps will include recognizing the plate for subsequent process. For the video trigger mode, there is no need to use coil, infrared, or other hardware vehicle detectors (Kieran & Yan, 2010).

The advantage of this method is convenient construction, we do not need to install the car inspection facility; its drawbacks are significant due to the limitations of running the recognition algorithms; the rate of triggering recognition is lower than that of peripheral triggers.

The trigger method also can be grouped into indirect method and direct method. Indirect method refers to identify the license plate and the relevant information by using an IC card or a bar code. Identifying IC card technology has a high accuracy and reliable operation. It can keep working all time, but the whole combo equipment is expensive, the equipment is very complex, so it is not suitable for long-distance operation. Bar code technology requires fast computation, high accuracy, strong reliability and low cost, but the scanner is very demanding. In addition, both of them need to develop a unified standard and can't check whether the car matches the bar code.

Direct method is a technology based on the images of license plate. It is a passive method of license plate recognition. It can obtain the information through non-contact acquisition and real-time identification without any special equipment (Neto, Gomes, Reboucas, & Albuquerque, 2015). Compared with the indirect method, first of all, a direct method saves a lot of money due to equipment placement; secondly, because of the adoption of advanced computer application technology, it can improve the algorithm of recognition and better solve the problem in real time. Moreover, it is based on image recognition and can reduce the identification errors (Schuster, Morzinger, Haas, Grabner, & Van, 2010).

The direct method generally includes image processing technology, pattern recognition technology and artificial neural network technology (Volna & Kotyrba, 2013).

1) Image processing technology has been employed to recognize license plate in the 1980s. The identification process is in use of industrial television cameras taking a front-view picture of the vehicle and sending it to carry on image processing; eventually, it still needs human intervention.

2) The pattern recognition technology refers to the structural characteristic method and statistical characteristic method. In the 1990s, due to the rapid development of computer vision technology, it has been applied to license plate recognition. In 1990s, using computer vision and image processing technology for automatic recognition of license plate came true. The procedure includes image segmentation, feature extraction, character recognition (Htay & Gopalakrishnan, 2016). Through a large number of statistical experiments, a license plate is located by using image histogram; the license plate is segmented into characters based on the corresponding histogram by using a preset standard template, the characters will be recognized.

3) Artificial neural network. In recent years, artificial neural network is explored to resolve the problem of automatic license plate recognition (Deb, Khan, Saha, & Jo, 2012). For example, in 1994, BAM neural network method was used to recognize the characters of a license plate automatically. The BAM neural network is the two-way predictive and single-layer network with the same neurons. Each character template corresponds to a BAM matrix. By comparing with the characters on the plates, the correct number of license plate was recognized (Su & Lu, 2017). The shortcoming of the BAM neural network is that it is difficult to resolve the problem due to the contradictory of its storage capacity and processing speed.

## 2.4 Genetic neural network

A genetic algorithm is a stochastic optimization method, which is a natural selection mechanism combined the evolutionary process of simulated organisms with the survival

of fittest survival of the fittest. It is widely used because it is different from other optimization methods. Since 1975, the theories and methods of genetic algorithms with pioneering significance have been investigated. The algorithms are a highly parallel, random and adaptive which is developed from the natural selection and evolution mechanism of biology.

In other words, it uses the group search technology; the population represents a set of questions based on the current selection, crossover, mutation and a series of operations so as to produce a new generation of population and gradually close to the population which contains the approximate optimal solution of states.

#### 2.4.1 The definition of genetic algorithm

The basic concepts related to the genetic algorithm are:

*Chromosome.* A chromosome is a unit that is constituted by a group of data. Each position on the chromosome is called a gene position, and its length is often denote as  $L$ .

*Population.* The gather of chromosomes that are represented by the data to form a population. The size of a population is the number of chromosomes, called population scale which is represented by  $P$ .

*Fitness.* The ability to measure an individual's ability to adapt to a living environment throughout a population space. In the law of natural selection, a survival of the fittest depends on the fitness value of the individual.

*Selection.* The operation of selecting individuals from a population by a certain probability. In general, the selection is a process based on fitness value.

*Crossover.* One position of two chromosomes is disconnected; two pairs of chromosomes form two new chromosomes. This process is also called genetic recombination.

*Mutation.* When copying a chromosome, it may cause replication errors with small probability; thus, the chromosome mutation can produce new one. These new chromosomes show special characteristics.

*Encoding.* Genetic coding can be seen as the mapping from its phenotype to genetic subtype. It refers to the process of converting the actual problem into a form of the genome of a chromosome. It is a process that the actual problem is arranged in a pattern of a chromosome.

*Decoding.* The transformation of genetic subtype to the phenotype is to map the chromosomes back to the original.

#### 2.4.2 The performance of genetic algorithm

A genetic algorithm is a method to resolve complex optimization problems, which can solve the problem that traditional optimization method cannot work for (Xu, Wang, & Chen, 2016). That is determined by the factors:

- (1) It does not need to know the specific problems of functions. It is not necessary to require the functions to be differentiable, continuity and so on, which are necessary conditions for traditional optimization methods.
- (2) Because it is a global optimization method, the optimal or sub-optimal solution of the problem can be obtained through genetic manipulation in algebra.
- (3) It is a parallel optimization method which can automatically search the solution in the whole solution space. The computing time is relatively short.

Genetic algorithms have been applied to the following areas: function optimization, combinatorial optimization, production scheduling, automatic control, intelligent robot control, image processing, pattern recognition, genetic programming, machine learning and so on. It worked out the problems that cannot be solved by traditional optimization methods.

#### 2.4.3 The steps of genetic algorithm

The steps of a genetic algorithm are shown as Fig 2.2.

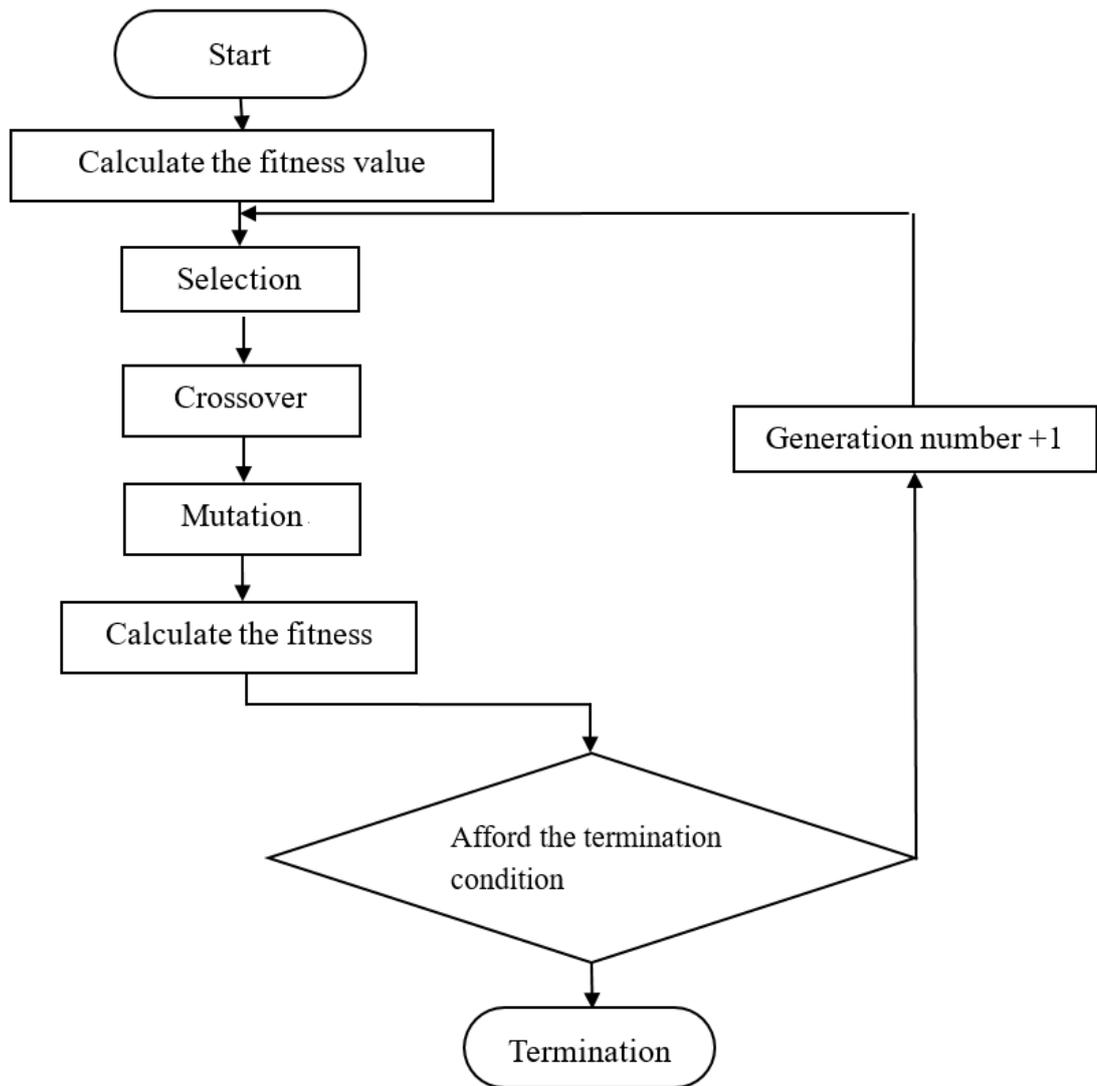


Fig 2.2 The flowchart of genetic algorithm

The genetic algorithm is a function optimization method. Because the requirement of this problem is not so high, the solution of this problem can be obtained easily. Its successful application in TSP problem has shown a new way for solving other problems. Now, genetic algorithm has been widely applied to multiple fields, including image analysis, network analysis, etc., and has been effectively improved in the applications.

## 2.5 Distance classifier

The distance classifier is one of the simple and effective classifiers, which directly calculates the distance between two modes as the basis for model discrimination. If the dimensionality of feature vectors is not very high, the calculation is fast, and the difference between the modes can be distinguished by the distance between them. If the difference between patterns is small enough, it is difficult to distinguish them by using this simple method of distance.

At the same time, the distance between different modes can be distinguished owing to the selection of feature extraction methods. So, the effectiveness of classifications depends on the performance of extracted feature vectors.

### 2.5.1 Euclidean distance classifier

Euclidean distance between two vectors is

$$D(x_i, w_j) = \|x_i - w_j\| \quad (2.1)$$

where  $D$  represents the distance between the components  $i$  and  $j$  in the vector space.  $x_i$  and  $w_j$  are two vectors having the same dimension. The maximum distance of the samples in the feature space is  $\theta_{max}$ . If  $D \geq \theta_{max}$ , at this time, the sample is not the plate image; otherwise, it is.

### 2.5.2 Cosine distance classifier

The cosine distance between two vectors is

$$S(x_i, w_j) = \frac{(x_i \cdot w_j)}{\|x_i\| \cdot \|w_j\|} \quad (2.2)$$

where  $S$  represents the inner product,  $\|\cdot\|$  is the norm,  $x_i$  and  $w_j$  are two vectors with the same dimension. When the unknown image is similar to the plate image, the image is closely related to this type of plate. The greater the similarity is, the closer the vector will be.

## 2.6 Neural network classifier

### 2.6.1 Neural network

Artificial Neural Network (ANNs) is also called Connectionist Model which is the abstraction and simulation of human brain. The artificial neural network is based on physiological research results of the brain, which is designed to simulate the mechanism of human brain.

The study of artificial neural networks can be traced back to the perceptron model in 1957. It started at the same time as artificial intelligence, but it has not been great as success as AI for more than 30 years and experienced a long period of depression. Until the 1980s, it got a practical algorithm of artificial neural network which leads to the revival of the neural networks.

Current research in neural networks has formed multiple genres, the most fruitful research work includes: multi-layer network, BP algorithm, Hopfield network model, the adaptive resonance theory, the theory of self-organizing feature mapping and so on (Wang, Zhu, & Zhang, 2016). Artificial neural networks are proposed on the basis of modern neuroscience. Although it reflects the basic features of human brain function, it is far away from the lifelike description of natural neural networks, it is a kind of simplified abstraction and simulation.

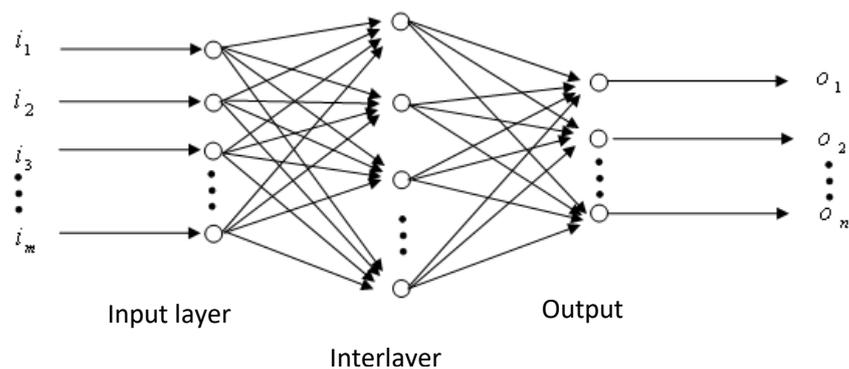


Fig 2.3 The structure of a neural network

Since its birth of the first neural computing model, neuroscience has been developed for 60 years. In the course of more than half a century, neuroscience has experienced a tortuous research and development path.

Table 2.1 The milestones of neural computing development

Milestones	Years	Historical events	Historical people
First	1943	McCulloch-Pitts model	McCulloch, Pitts
Second	1949	Hebb learning law	Hebb
Third	1958	Perceptron	Rosenblatt
Fourth	1969	Monograph "Perceptron"	Minsky, Papert
Fifth	1982	Hopfield network	Hopfield
Sixth	1986	BP learning algorithm	Rumelhart, etc.

### 2.6.2 Characteristics of artificial neural network

The following prominent advantages of artificial neural network have caused great concerns in recent years:

- (1) Sufficient approximation of any complex nonlinear relationship;
- (2) All quantitative or qualitative information is stored in various neurons in the network, so it has strong robustness and fault tolerance.
- (3) The parallel distributed processing method is adopted to make it possible to perform a large number of operations quickly.
- (4) It can learn and adapt the systems that are not known or uncertain;
- (5) It has the ability to handle both quantitative and qualitative knowledge simultaneously.

The characteristics and superiority of artificial neural network are mainly manifested in three aspects:

(a) Self-learning function.

Firstly, only many different samples and the corresponding results have been imported into an artificial neural network; the network will be trained through self-learning function so as to identify similar images. The self-learning function is very important for prediction. It is expected that future artificial neural networks will provide the economic forecast, market forecast and benefit prediction; its prospects are very broad.

(b) Associative storage function.

This association can be achieved with the feedback of artificial neural networks.

(c) The ability to find optimal solutions at a high speed.

Searching for the optimal solution of a complex problem often needs a large amount of calculations. Using a feedback, an optimal solution of the problems of artificial neural network may be found very fast.

### 2.6.3 The main research directions of artificial neural network

The research of neural networks can be divided into two aspects: theoretical research and applied research. Theoretical studies can be grouped into two categories:

(1) Using neurophysiology and cognitive science to study human thinking and intelligent mechanism.

(2) Using mathematical methods to explore functions, algorithms and performance of neural networks, such as stability, convergence, fault tolerance and robustness, etc. and to develop new mathematical theory, such as neural network dynamics, nonlinear neural field and so on. Application research can be grouped into the following two categories:

(1) Research on software simulation and hardware implementation of neural networks;

(2) The research of neural networks in various fields.

These areas mainly include pattern recognition, signal processing, knowledge engineering, expert system, optimization combination, robot control, etc. With the development of neural networks and related theories, the application of neural network will be further explored.

The neural networks have been widely used in the control field due to its merits against interferences and robustness. It has the ability to simulate the thinking procedure of human brain, remember the complicated process, and has a strong tolerance for complex processes.

## 2.7 Related work

The license plate recognition has been investigated for several decades and it already has a large development. Especially, the modules of character segmentation and character recognition are well-investigated with several approaches. Because the plate does not have too much significant features, the edge is used to detect and segmented frequently. Some related work also certifies that it is the effective method. It is also used in this thesis. However, most of related works show that the method of character recognition is BP neural network. It has the good adaptation and robustness in the LPR system. In this thesis, the GNN is used to optimize the results. It also can test the performance of the new approach.

In addition, most of other related work investigated the horizontal and stable license plate. It does not have the influences of angle problem and waggle problem. The state of the plate is good and ideal. If the plate is skew or the plate is shaking when the vehicle runs on an uneven and slant road, it is difficult to capture the effective image. The past work did not have the special module to solve the problem. Without correcting the skew plate, the subsequent steps are difficult to be achieved. Therefore, the plate autocorrection module in this thesis is necessary and helpful. It is the biggest difference from other studies.

# Chapter 3

## Methodology

*In this Chapter, the methodology used in this thesis will be presented. According to the research questions from literature review, the corresponding solutions can be proposed. To afford our objectives, we explain research methods to achieve them. Finally, the experiments and its results also will be described.*

### 3.1 Introduction

With the rapid development of pattern recognition, scene understanding in computer vision knowledge discovery and in artificial intelligence have been dramatically enriched, enhanced and deepened. Pattern recognition and many new methods have been developed to resolve the problems from various applications. Artificial neural networks have been developed as a new classifier for optical character recognition (OCR).

### 3.2 Genetic neural network

The genetic algorithm takes the place of neural networks and BP algorithm. Using the genetic operator in a network could achieve the optimized solution for the network. Genetic algorithm (GA) is used to train the weights of the neural network, it has gained good effects in a majority of applications (Sun, Zhang, Zou, & Yu, 2010).

In this thesis, combined with plate recognition, a plate recognition classifier is designed. The feature of plate recognition is used as the input of a neural network. Since the genetic algorithm in this thesis is used as a search method, the weight of each network is a gene position of the chromosome and the length of chromosome corresponds to all values of the network. In this way, the length of a gene should be considered because the length affects the search process and computation of the chromosomes.

The number of nodes in the input layer of neural networks is determined by the characteristic points. The number of nodes in the middle layer is generally determined by the empirical formula or experimental method. The number of output layers is generally determined by the class of the classifier.

The network structure corresponds to the genetic algorithm. Considered the length of chromosome affects the speed of evolution, the nodes of output layer of the neural network are exported by encoding. Therefore, the number of chromosomes can be significantly reduced which save the time of evolutionary computation of genetic algorithms. The structure of chromosomes is shown in Fig 3.1.

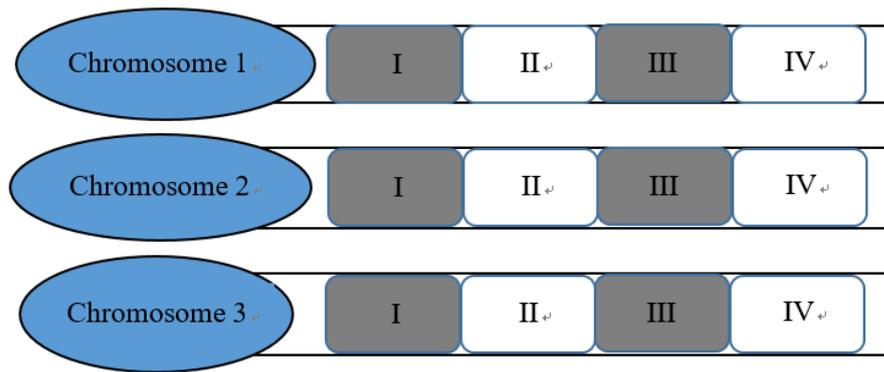


Fig 3.1 The structure of chromosome

In Fig 3.1, part I, II, III and IV correspond to the hidden layer weight, output layer weight, hidden layer threshold and output layer threshold respectively. In the design:

- (1) Determine the coding scheme and generate a set of initial population.
- (2) Initialize population coding and build the structure of the neural network.
- (3) Calculate the fitness of each individual according to the fitness function of the design.
- (4) According to the size of fitness, the genetic manipulation is processed and the evolution of a generation is achieved.
- (5) Calculate the individual fitness and determine whether the accuracy requirement is satisfied. If it is, stop the process; otherwise, switch to step (6).
- (6) Add '1' to the number of generations so as to determine whether the evolutionary algebra of design is achieved. If the maxima is reached, the genetic operation should be stopped; otherwise, switch to (2).
- (7) After trained the GA, the optimized individual is obtained and the neural network can be constructed.
- (8) BP algorithm is used to optimize the network.

Because the selection of encoding scheme affects the design, real value coding is adopted. The coding in real has high precision and is easy to be operated. The output of neural network is encoded and the encoding principle is

$$N = [\log_2 M] + 1 \quad (3.1)$$

where  $[\log_2 M]$  is the integer function,  $M$  is the number of categories of expected output and  $N$  is the number of output nodes of the actual network.

The encoding method can reduce the chromosome length of genetic algorithm and simplify the calculation of genetic operation. The fitness function used in this genetic algorithm is:

$$f = \frac{1}{o\_num * popsize} \sum_{i=1}^{popsize} \left( \sum_{k=1}^{o\_num} (t_{ik} - y_k)^2 \right) \quad (3.2)$$

where  $o\_num$  is the number of output layer node,  $popsize$  is the population size,  $t_{ik}$  is the actual output and  $y_{ik}$  is the expected output.

This chapter firstly introduced two kinds of distance classifier and neural network classifier. Aimed at the problem, such as slow speed convergence, when using the neural network as the plate recognition classifier, the neural network is improved and the genetic neural network classifier is proposed.

---

**Algorithm 1** Genetic neural networks

---

**Input:** Feature and Initialize the network

---

**Output:** Character recognized result

---

Set  $x$  = evolution of numbers**For**  $i = 1:x$ :do    **if**  $bestfitness > newbestfitness$          $bestfitness = newbestfitness$ ;         $bestchrom = individuals.chrom(newbestindex, :)$ ;    **end**     $individuals.chrom(worestindex, :) = bestchrom$ ;     $individuals.fitness(worestindex) = bestfitness$ ;     $avgfitness = sum(individuals.fitness) / sizepop$ ;**End** $w1 = x(1:inputnum * hiddennum)$ ; $B1 = x(inputnum * hiddennum + 1:inputnum * hiddennum + hiddennum)$ ; $w2 =$  $x(inputnum * hiddennum + hiddennum + 1:inputnum * hiddennum + hiddennum + hiddennum * outputnum)$ ; $B2 =$  $x(inputnum * hiddennum + hiddennum + hiddennum * outputnum + 1:inputnum * hiddennum + hiddennum + hiddennum * outputnum + outputnum)$ ; $net.iw\{1,1\} = reshape(w1, hiddennum, inputnum)$ ; $net.lw\{2,1\} = reshape(w2, outputnum, hiddennum)$ ; $net.b\{1\} = reshape(B1, hiddennum, 1)$ ; $net.b\{2\} = B2$ ;

---

Algorithm 3.1 Genetic neural network

### 3.3 The research question

The focus of this thesis is on two major questions related to license plate recognition. Therefore, the two questions related to the recognition is very important. The plate rotation correction module can influence the character image extraction (Dua, Singh, & Parthasarathy, 2016). The quality of the character image can influence character recognition eventually (Liu, Li, & Tu, 2015). Thus, this module is important and the question needs to be answered. Therefore, the main research question of this thesis is:

*Question:*

*What technique should be used to implement the image rotation correction module?*

### 3.4 Data acquisition

#### 3.4.1 Data collection for experiments

Since license plate recognition contains two core modules: rotational correction for tilted images and optical character recognition; correspondingly, two types of data are collected in our experiments. The data for correcting tilted images is a sequence of video frames; an unstable plate was simulated by wagging a white paperboard in front of a camera. These videos are used to investigate the method of image correction. On the other hand, the data for character recognition module is with 100 images that describe a variety of paperboards with various letters. All the videos and images were captured by using 20 million pixels built-in camera of Huawei Mate 9 Pro mobile phone.

The data for rotational correction module is 8 videos captured by using this camera. The shortest video takes 8 seconds and the longest video takes 15 seconds. All the videos have the rate at 30 frames per second; the resolution of each frame is  $1280 \times 720$  in the video. These videos contain two wagging methods. Four videos were waggled from side to side, the others were from back and forth. All the videos are utilized to evaluate the rotational correction of tilted images.

The data for character recognition module consists of 100 images that were captured by the mobile phone camera. All the images were captured in the lab in daytime and the background is white wall. The resolution of these RGB images is  $3968 \times 2976$ . These images were utilized to evaluate the recognition rate.

#### 3.4.2 Templates and Datasets for GNN

The templates used in the experiments contain 36 characters that are the letters from 'A' to 'Z' and the numbers from '0' to '9'. These images were stored with the resolution ratio  $24 \times 32$ . The background of this image is black and the characters are white. All these

templates were used in the character recognition module to evaluate the template matching method.

On the other hand, the data set for character recognition by using GNN contains 100 image samples. The resolution ratio of these samples is  $3968 \times 2976$ . Each of the sample plate contains six characters chosen from 26 letters and 10 numbers randomly. All the samples were used to train the GNN algorithm. The features of the sample were extracted by using PCA. PCA transforms the original data into a set of linearly independent representation of each dimension through linear transformation, which can be used to extract the main characteristic components of data.

### 3.5 Evaluation methods

Evaluation is the essential to estimate the performance of an experiment. There are various evaluations for different experiments. The most common criterion for pattern classification is accuracy (Zheng, He, Samali, & Yang, 2013). The high accuracy means the good performance directly. However, if the accuracies of two algorithms are the same or similar, the consuming time can be another criterion to estimate the performance. With the same accuracy, the shorter time the experiment used, the good performance it has.

In our experiment, we use the accuracy to evaluate the algorithm. In addition, two concepts precision and recall are also introduced. The precision and recall are two measures widely used in information retrieval and statistical classification to evaluate the outcomes. The precision means how many selected items are relevant. The recall defined as how many relevant items are selected.

In addition, there are four terms to be defined, which are True Positives (*TP*), False Positives (*FP*), True Negatives (*TN*) and False Negatives (*FN*). *TP* means the relevant items among the selected items. *FP* means the irrelevant items among the selected items. *TN* means the irrelevant items which are not selected. *FN* means the relevant items which are not selected. Therefore, the equations to calculate the accuracy, precision and recall are,

$$Precision = \frac{TP}{TP+FP} \quad (3.3)$$

$$Recall = \frac{TP}{TP+FN} \quad (3.4)$$

$$Accuracy = \frac{TP+TN}{TP+FP+TN+FN} \quad (3.5)$$

In equation 3.5, the accuracy of character recognition is used in this thesis. The numerator ( $TP+TN$ ) refers to the number of samples which are recognized correctly. The denominator ( $TP+FP+TN+FN$ ) is defined as the sum of all the samples no matter the recognized result is right or wrong. Therefore, the accuracy in our experiment is,

$$Accuracy = \frac{\textit{The number of correct plate}}{\textit{The number of correct plate} + \textit{The number of incorrect plate}} \quad (3.6)$$

Equation 3.6 displays the concept of accuracy in our experiments directly and is easy to be understand.

## Chapter 4

### Design and implementation

*This chapter introduces the components of plate number recognition first. In addition, the details of each component of the LPR are described, especially, the function of plate correction will be showcased in details.*

## 4.1 LPR Design

Our LPR implementation is shown in Figure 4.1. First of all, we use the original image  $I(x, y)$  as the input through relevant pre-processing which contains color to grayscale conversion, brightness transform, median filtering, edge extraction, eliminating image noise and others to get the image  $I_1(x, y)$  (Adatrao & Mittal, 2016).

Then, we crop and rotate the image  $I_1(x, y)$  to get the image  $C(x, y)$ . After this, using the morphology edge extraction, we obtain the image  $E(x, y)$ . The image  $E(x, y)$  needs to be rotated to  $R(x, y)$ . Next, the character segmentation will be used to get  $S(x, y)$ . Finally, we utilize character recognition algorithm to identify and recognize the object  $O(x, y)$ .

Fig. 4.1 shows that the workflow of the LPR.

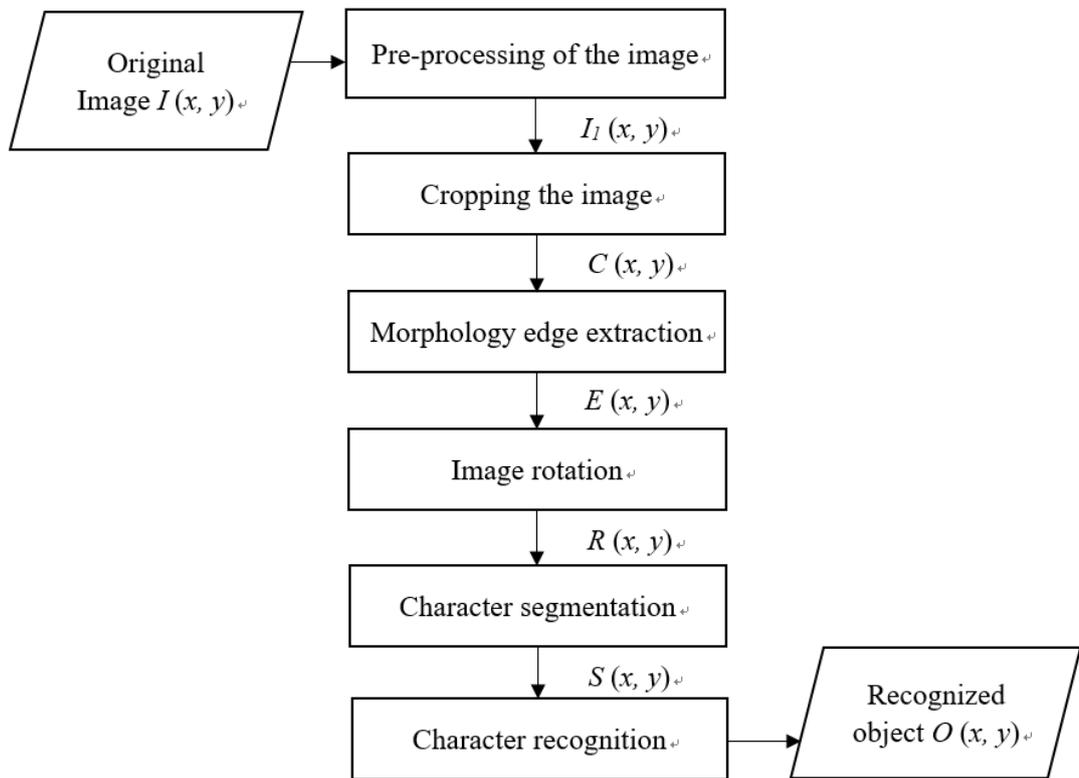


Fig 4.1 The workflow of number plate recognition

Step 1. Extract any frames in the input video.

Step 2. Pre-processing to the extracted frame

1) Transform the RGB images into a grayscale one. It is convenient to extract the high contrast information from the image.

2) Brightness transform. It can eliminate the noises caused by changes of illumination.

3) Median filtering. It will filter out and eliminate various noises.

4) Edge extraction. To identify the image edges.

Step 3. Determine the image rotation angle.

Using Hough transform to extract straight lines from the image region with characters, we calculate the rotation angles of the straight lines and the horizontal lines so as to determine the rotated angle of the image (Xing, Li, Xie, Liao, & Zeng, 2016). We rotate the image back to the oriental direction with the angle obtained. The horizontal characters can be received after this operation. The bilinear interpolation method is employed after rotation.

Step 4. Morphology processing and locating characters.

1) Processing the rotated image with corrosion (Ktata & Benzarti, 2012). The corrosion is a kind of “refinement” or “shrink” operation. It can get rid of the details of irrelevant regions, such as the thumb in the video frames because it may cause the problems of wrong edges.

2) Using the close operation to the image, we eliminate the edges which do not belong to the rectangle of video frames. The regions, we obtained after the character corrosion, present the shape of this block (Gou, Wang, Yao, & Li, 2016).

3) After deleted the connected regions which are very smaller in the experiment, the procedure sets a parameter as the threshold so that the connected regions, which are smaller than 400 pixels, are considered as the noises.

Step 5. Segmentation of the character region.

1) The numbers of non-zero pixels in each line of the image are counted. If the number is bigger than a threshold, say 5, we think that this line may locate within the character zone. On the other hand, if the number is smaller than this threshold, it is considered that the line may locate in the non-character region. Because of the binarization for edge detection, the pixels of edge regions are set as 1 as well as the others are 0 (Ha & Shakeri, 2016).

2) If all of the lines in the character region are continuity, it means that the line at the most top is the top edge of these characters. Meanwhile, the line at the bottom is the bottom edge of the characters.

3) If all the lines in the character region are discontinuity but form a number of different blocks comparing to the height of these blocks, we then consider the highest block as the character region. The topside line of this block is the border line of these characters. The bottom line of this block is the lowest line of these characters. Thus, these lines can accurately locate the upper and lower edges of these characters (Manzanera, Nguyen, & Xu, 2016). In addition, the determination of left and right edges is as same as the top and bottom edges. Normally, the length of these height lines is lower than that of the wide lines (Chen & Luo, 2012).

Step 6. According to the range of top, bottom, left and right lines, we deal with the characters of the rotated image and segment each character out.

Step 7. Pre-processing the characters

1) Transform a RGB image into grayscale one.

2) The brightness transform eliminates the noises caused by changes of illumination.

3) Image binarization. Due to separate the characters from their background, we would like to convert all grayscale images into binary.

4) Median filtering. For the reason of eliminating various noises, we filter them out using median filter.

5) Expand operations. In order to connect the disconnected regions, we use the morphological operations, such as open and close operations.

Step 8. Single character segmentation.

In both vertical and horizontal directions, we count the numbers of non-zero pixels of the images. Then, we determine the top, bottom, left and right lines of a single character. If the number of non-zero pixels along horizontal or vertical directions of a character region is not zero, the boundary of these characters can be affirmed. We therefore set the foreground pixels as 1, and the background ones as 0.

Step 9. After separating those individual characters of the image, we compare the characters to the templates stored in the database. Then, we obtain the maximum similarity characters and output them as the results.

## 4.2 LPR Implementation

### 4.2.1 Image pre-processing

It is the first step to eliminate the noise from the original video frame. Because many factors may influence the analysis of images, such as changes of illumination, color, background and others, the original video frame is converted into a grayscale one.



Fig 4.2 The original video frame

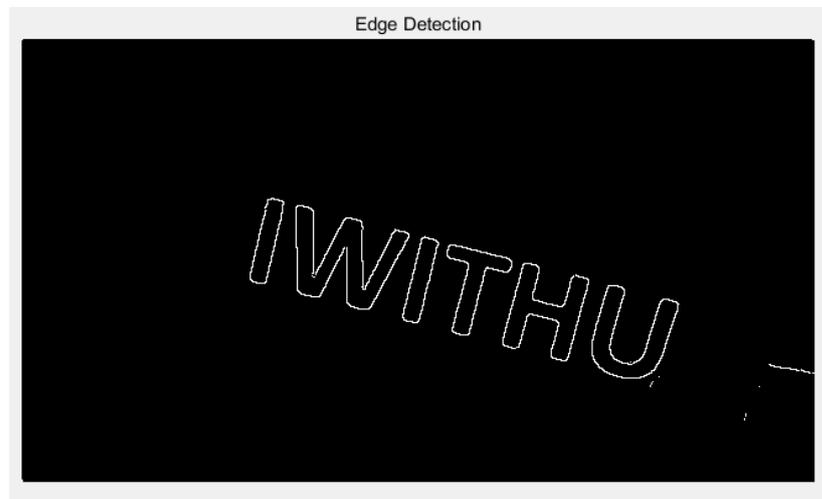


Fig 4.3 The image after binarization

#### 4.2.2 Image correction

When the tilt angle of a license plate is too large, it may affect the accuracy of plate locating; this even leads to the failure of locating. At the same time, the subsequent segmentation and recognition of character are very sensitive to this tilt. Therefore, it is necessary to correct the tilted image (Zhu & Liang,2012). Thus, we have to detect the tilt angle first.

Normally, there are three possible ways that a number plate may have, the first is the vertical tilt as shown in Fig 4.4, the second one is the horizontal tilt in Fig 4.5 as well as the last one with both in Fig 4.6.

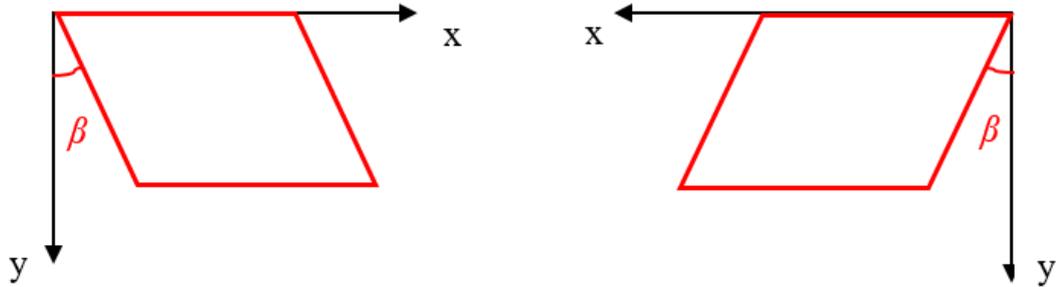


Fig 4.4 The tilt in vertical direction

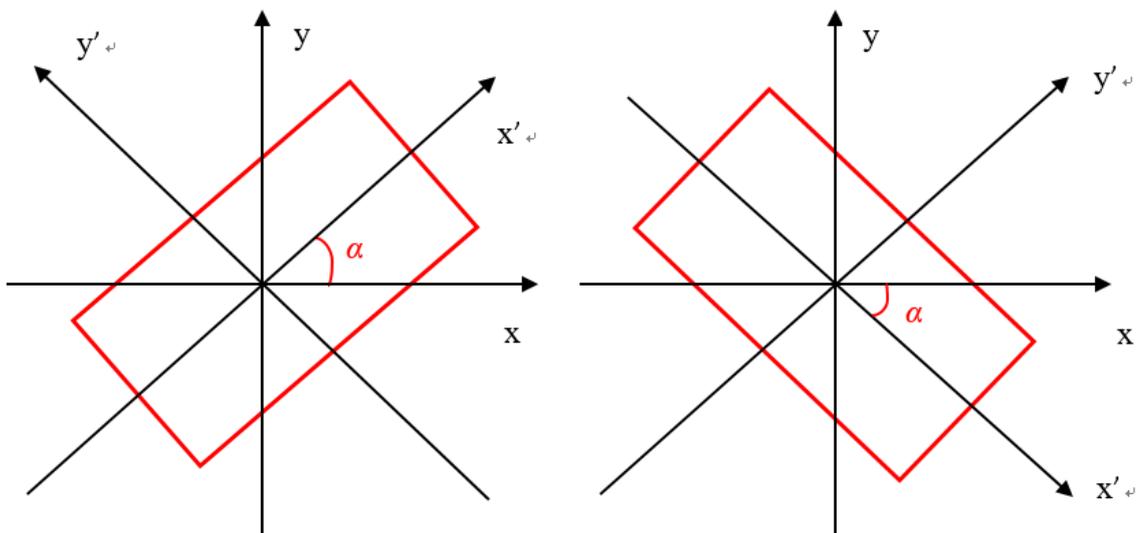


Fig 4.5 The tilt in horizontal direction

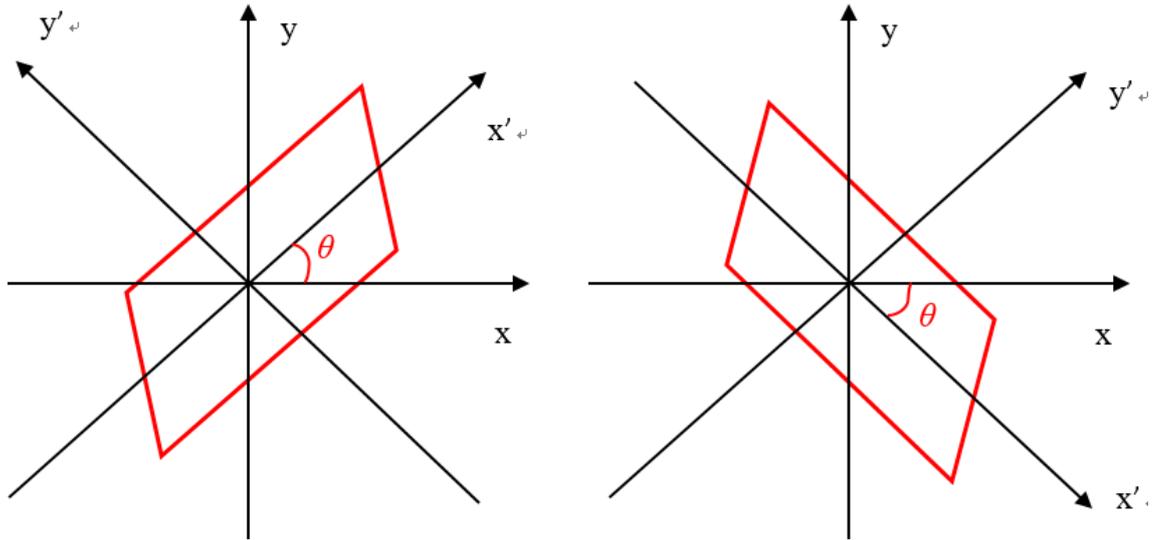


Fig 4.6 The tilt in both vertical and horizontal directions

The existing tilt correction algorithm, which has been verified to be effective and important, is Hough transform. In this thesis, we use Hough transform to detect the frame of a license plate, then calculate the tilt angle of the plate region. After that, the coordinates of the four corners of a plate region are calculated (Patel, Desai, & Panchal, 2015). Through the bilinear interpolation, we correct the distorted images of number plate.

#### 4.2.2.1 Hough transform

The basic idea of Hough transform is to use the duality theorem of the points and lines. It means that the collinear points of an image correspond to the intersection points of several curves in the parameter space (Arulmozhi, Perumal, Mohan, & Nallaperumal, 2012). Likewise, all the curves, which intersect at one point in the parameter space, thus have a line which has all the collinear points in the image space. Thus, a problem of curve detection in the image space can be converted to the problem finding the peaks in the parameter space. In the image space  $x$ - $y$ , a straight line can be noted as:

$$y = kx + b \quad (4.1)$$

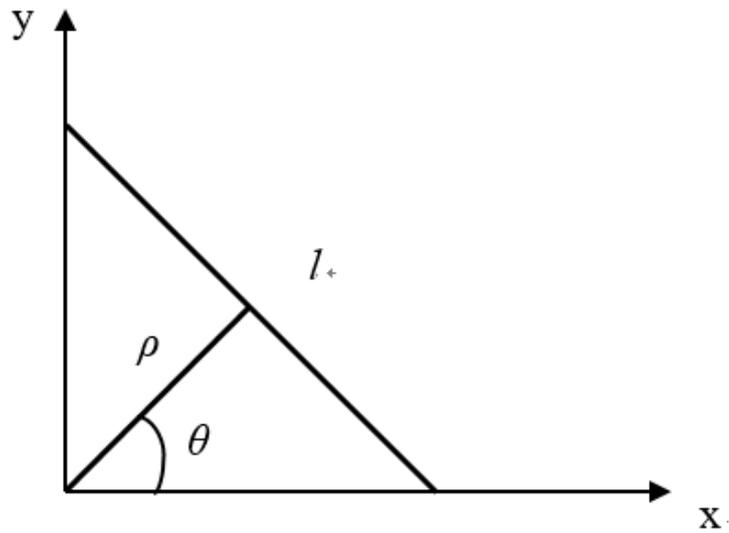


Fig 4.7 The Cartesian coordinate system

As shown in Fig 4.7,  $l$  is a straight line in the image space,  $\theta$  is the angle which passes through the origin and is perpendicular to the straight line  $l$ .  $\rho$  is the distance from the origin to the straight line  $l$ . This straight line is expressed by using the polar coordinate equation

$$\rho = x \cos(\theta) + y \sin(\theta) \quad \theta = (0, 180) \quad (4.2)$$

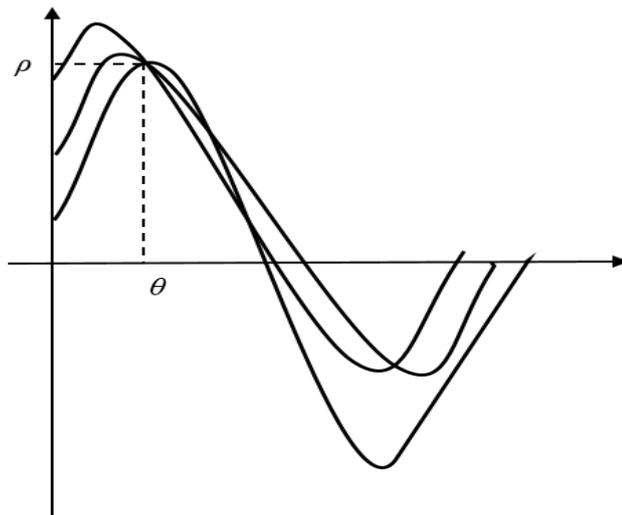


Fig 4.8 The polar coordinate system

The parameter space is denoted as  $\rho$ - $\theta$  as shown in Fig 4.8. From the two Figures, we see that a point in  $x$  -  $y$  space corresponds to one curve in  $\rho$ - $\theta$  space. Thus, a straight line in  $x$ - $y$  space can map to a point where all curves will intersect in  $\rho$  -  $\theta$  space.

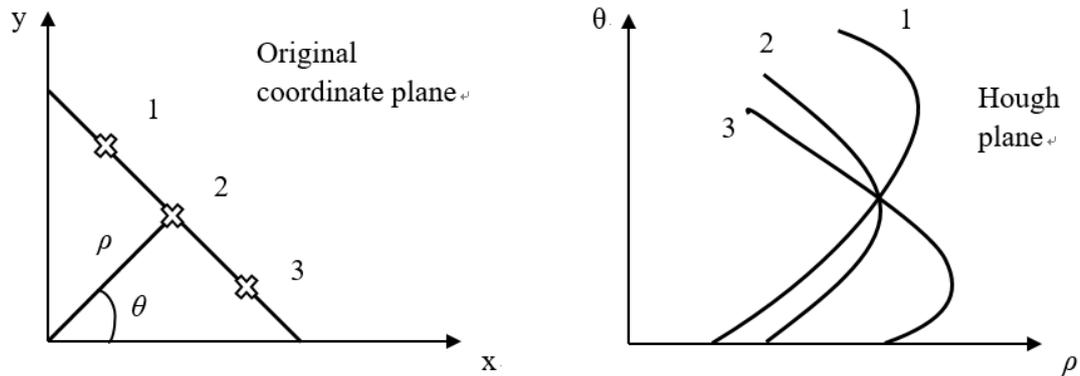


Fig 4.9 The basic schematic diagram of Hough transform

#### 4.2.2.2 The implementation of straight line detection

The essence of straight line detection based on Hough transform is to convert the search problem of straight lines in an image space to the problem of searching local maxima in the parameter space (Mukhopadhyay & Chaudhuri, 2015).

The specific steps include:

Firstly, the parameter in the space  $(\rho, \theta)$  is quantized into several small lattices according to the requirements. Secondly, we initialize an accumulator  $A(\rho, \theta)$  as Fig 4.10, the initial value is 0. The quantization number of the  $\theta$  direction is 180 degree and the quantization number of the  $\rho$  direction is the number of pixels on the image diagonal.

Then, each pixel  $(x, y)$  in the image is detected. According to the equation  $\rho = x \cos(\theta) + y \sin(\theta)$ , we calculate  $\rho$  which corresponds to each  $\theta$ . At the same time, we add 1 to the corresponding accumulator  $A(\rho, \theta)$ .

Finally, the maximum  $A(\rho_m, \theta_m)$  in the accumulator  $A(\rho, \theta)$  is found, where  $\rho_m$  and  $\theta_m$  refer to the longest straight line in the image.

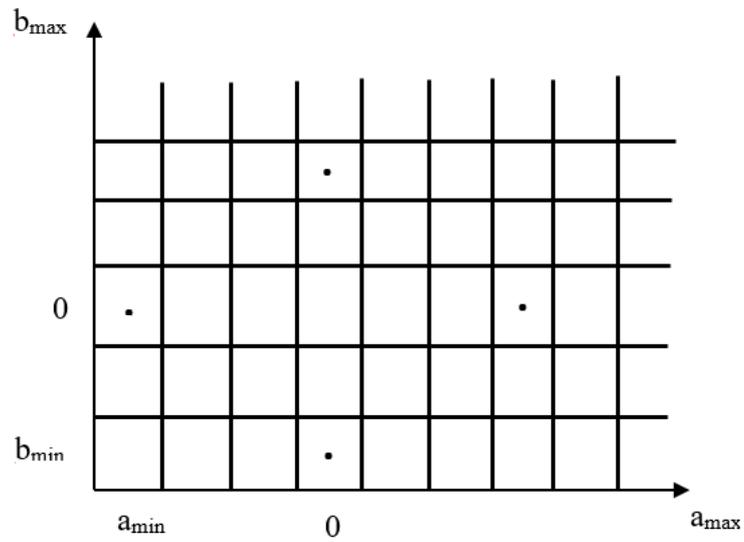


Fig 4.10 Accumulator

The maximum  $\theta_m$  in the parameter space is the horizontal angle of this straight line in the image space. The tilt angle of horizontal border of this plate has been detected shown as Figure 11 and Figure 12.

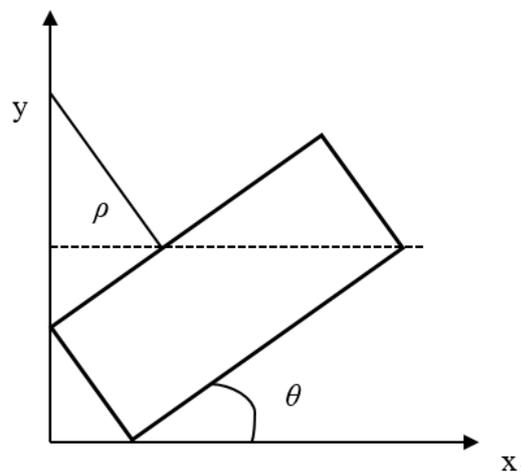


Fig 4.11 Original image space

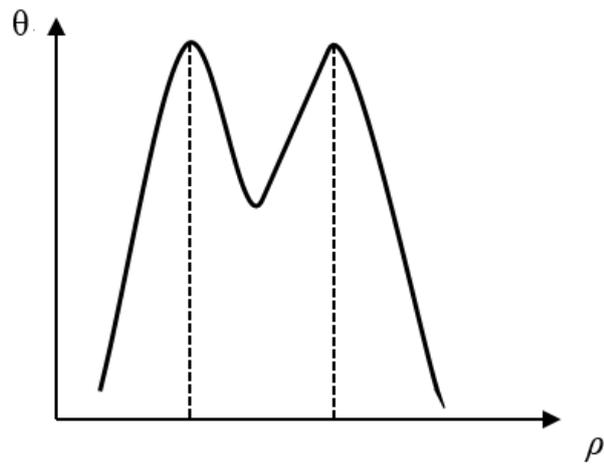


Fig 4.12 Hough transform space

#### 4.2.2.3 Straight line detection based on Hough transform

Before the straight line detection, it is necessary to get the coordinates of a point in Hough transform. The number plate contains two statuses: one is the plate has the border, the other is that it does not have the border.

For the license plate which has the border, it is easy to get the coordinates of detected points according to the border. However, for the plate without border, only the edges of top and bottom contour of the characters can reflect the tilt angle of the plate (Al-Ghaili, Mashohor, Ramli, & Ismail, 2013). Considered that the plate frame may not exist, the most effective method is, through the top and bottom edges of the characters, to calculate the useful data that the Hough transform needs (Arulmozhi, Perumal, Deepak, & Nallaperumal, 2012).

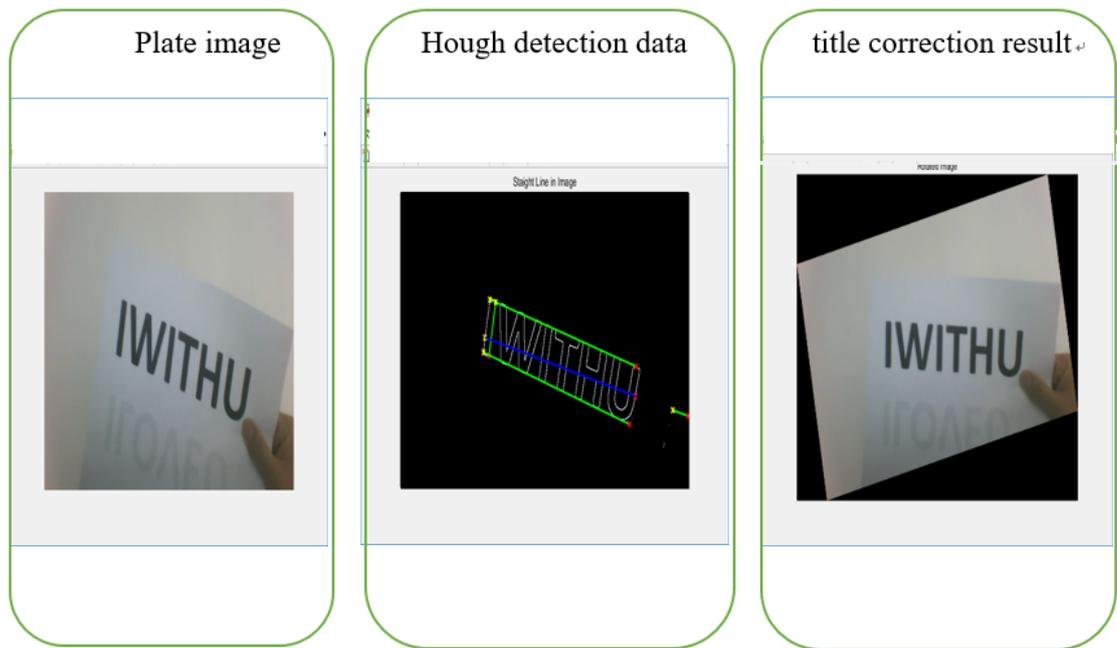


Fig 4.13 The result of plate correction

Fig 4.13 shows the result of license plate correction. From Fig 4.13, it is obvious that almost all the white points in Hough transform are used for calculating the tilt angle. The useless points are less. Therefore, it not only improves the accuracy of the plate correction but also reduces the costs significantly.

The failure of this license plate correction has a large variant of influence on the subsequent work. Firstly, it will affect the character segmentation. Secondly, it may dilute the accuracy of character recognition. Therefore, the performance of this plate correction is very important and is significant. Thus, it is the most important step of this algorithm.

#### 4.2.3 Image segmentation

##### 4.2.3.1 Cropping

After corrected the image, it is able to obtain the rotated image. However, in the rotated image, there is noise needed to be deleted. Therefore, the image needs to be cropped to a precise image which only contains the useful information for further processing. This can be treated as the rough and simple processing which only can delete the obvious and big

range of the unwanted information. The cropped image may also contain some small objects.

#### 4.2.3.2 Edge extraction using morphological operations

First of all, it should transform the cropped image to a binary image which only has black and white (1 or 0) color. We use morphological operations to extract the edges of binary images. The morphological edge extraction contains two basic morphological operations: erosion and dilation (Paunwala & Patnaik, 2010). Through selecting different sizes and extensions, we transform the cropped image to a grayscale image. Then, we delete the unwanted regions according to the basic morphological operations (Roy, Hossen, & Nag, 2016). In the process, firstly we connect pixels together to build a group connected objects. Then, we remove the edges that do not belong to the same rectangle in the image. After the erosion, the character region is able to form blobs.

#### 4.2.3.3 Characters segmentation

Firstly, the number of non-zero pixels in each line of the image are counted. Then, if all the lines in the character region are continuity, it means that the top side of these lines is the top border of these characters. The line at the bottom is the bottom border of these characters. On the other hand, if all the lines in the character region are discontinuity, a number of blocks will be formed. Compared with the height of these blocks, the highest block is the character region. The top side of this block is the top border of these characters. The bottom line of this block is treated as the bottom of the characters.

According to the result, we are able to obtain the top, bottom, left and right edges. Then, we segment the obtained characters from the rotated image. Fig 4.14 shows the result of cropped image from the rotated image.



Fig 4.14 The cropped image

#### 4.2.4 Character segmentation

At present, there are a lot of segmentation algorithms for the characters of number plate, we list them below.

*(1) The character segmentation method based on template*

The algorithm employs a template for each character; then, letting the template slide in the area of the license plate, we calculate the number ratio between the pixels within the characters and the pixels within the character template. When the ratio reaches the minima or maxima, the location of current template is determined as the split point of the character. In this method, the prior knowledge of this character is its width; usually, we cannot get this width very easily.

*(2) The character segmentation method of the plate based on vertical projection of a license plate.*

The algorithm first needs to obtain the curve of integral projection of a license plate in vertical direction; then, based on the curve peaks and valleys of the characters to deal with the character segmentation of license plate. This approach usually obtains the better results, but when there are adhesions between the license plate characters, the division quality will be dramatically reduced.

*(3) The method of character segmentation for a license plate based on the connected domain.*

The algorithm first binarized the license plate image, then removes the connected regions. At the same time, the fracture of license plate characters is merged. Afterwards, the region

markup method is used to get the region of each character. Finally, the character segmentation of a number plate is implemented. This method has high demand for noise removal and integrity of optical characters.

#### *(4) The character segmentation method of a license plate based on colors*

The algorithm is based on the difference between background color and character color of a license plate. The analysis of color discontinuity is carried out to realize the character segmentation (Ashtari, Nordin, & Fathy, 2014). This method utilizes the colors of a license plate; the calculation is based on color image processing (Biao, Qin, & Ruien, 2014).

These methods have a good segmentation result on license plates, each of them has their own features. Among them, the second and third methods are the most stable ones and the requirements for segmenting license plates are relatively low. It is recommended the second method should be combined with the first one. In this thesis, the third method has been adopted and the ideal results are obtained. Because a printed paper having characters of number plate is used to replace a real plate in the experiment, the characters are very clear and there are not adhesions between these characters. So, the method is able to get the ideal results.

#### 4.2.5 Character recognition

The method of character recognition mainly has two methods. One is based on the template matching and the other is on the genetic neural network (GNN). Based on the template matching, the first algorithm converted the segmented characters into binary colors and resized it as same as that of the character templates. With all template matching, we choose the best match as the result. It is important to establish the dataset in the process of license plate recognition. The accuracy of number plate recognition is the key element of evaluation (Uddin et al., 2014).

There are two kinds of algorithms based on artificial neural networks. One is to extract the character features and then use the acquired features to train the neural network. The other is to input the image into the network directly and automatically until the results are identified.

The template matching algorithm is simple, when the characters are much regular, the adaptability of these images is strong; this leads to the high recognition rate. In the experiment, the paper plate is used to replace a real one; the characters on the paper are very clear and regular. Therefore, this thesis treats the template matching as the main one of the license plate recognition algorithms. In addition, another character recognition experiment was based on the GNN algorithm so as to compare the performance of template matching.

Template matching is the most representative method in recognition. It extracts character features from the image  $f(x, y)$  to compare with the characters in the template  $T(x, y)$ , and calculates the normalized cross-correlation between them. The maxima of cross-correlation means the similarity is the highest. Hence, the image can be grouped into the corresponding class. We calculate the distance between the image and the template so as to determine the class with the minimum distance.

However, the images for matching are different and they may include a large amount of noises. After preprocessing and normalization treatment, the changes of pixels may be occurred. Considering the interference caused by the processing such as noise and displacements, we create a template based on image invariant features. Therefore, it can avoid the above problems significantly. Fig 4.15 shows the result which we achieved in this experiment. The cropped image in Fig 4.14 is used to locate the characters. Then, the characters are segmented into individual one. Finally, we compare these characters with the samples in the template. The character which has the highest similarity is regarded as the output.

---

**Algorithm 2** Character Recognition

---

**Input:** Frames of video

**Output:** every character shown in the video

- 1: reading a frame `rgb_image` from video
  - 2: preprocessing of the image, including filtering, enhancement and edge detection (obtaining the `edge_image`)
  - 3: rotating the `edge_image` (see *Function* Hough Transformation)
  - 4: segmenting the useful objects from the `edge_image` and extracting the edge of characters
  - 5: cropping the characters from the `edge_image`
  - 6: character recognition:
  - 7: **for** each character **do**
  - 8:     comparing it with each image in standard character dataset
  - 9:     **if** matching **then**
  - 10:         character confirmed
  - 11:     **else**
  - 12:         comparing it with next image in standard character dataset
  - 13: **End for**
  - 14: **return** recognition result
  - 15:
  - 16: *Function* Hough Transformation:
  - 17: **for** each edge pixel  $(x,y)$  in image **do**
  - 18:     **for**  $\theta = -90 : 90$  **do**
  - 19:          $\rho = x \cos \theta + y \sin \theta$
  - 20:          $++ H(i\theta, j\rho)$
  - 21: choosing a set of  $(\tilde{\theta}, \tilde{\rho})$  can make  $\max(H(i\theta, j\rho))$  as the optimal rotating angel, and rotating the `edge_image`
- 

Algorithm 4.1 Character recognition

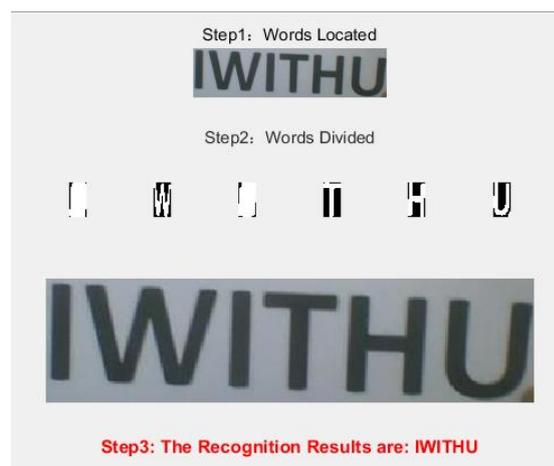


Fig 4.15 The recognition result

# Chapter 5

## Experimental results and discussions

*In this chapter, the experimental results and corresponding analysis will be presented. The results will be described step by step in details according to the workflow. Then, the results of two different methods will be compared and analyzed. Finally, we will present the discussion of this chapter.*

## 5.1 Introduction

In Chapter 4, the relevant experimental approaches and algorithms have been presented. This chapter will describe the corresponding results and discussion of the experiment. The facility of our experiments is Intel Core i5-4210H CPU 2.90GHz laptop and the software is MATLAB R2015b. All the videos were captured by using the laptop camera and the resolution is 640×360.

## 5.2 Image extraction

First of all, we need to capture the frames of a video. Fig. 5.1 shows the selected frames of a sample video. Because the printout for our experiment keeps shaking, we extract the distinct frames for our further experiments.

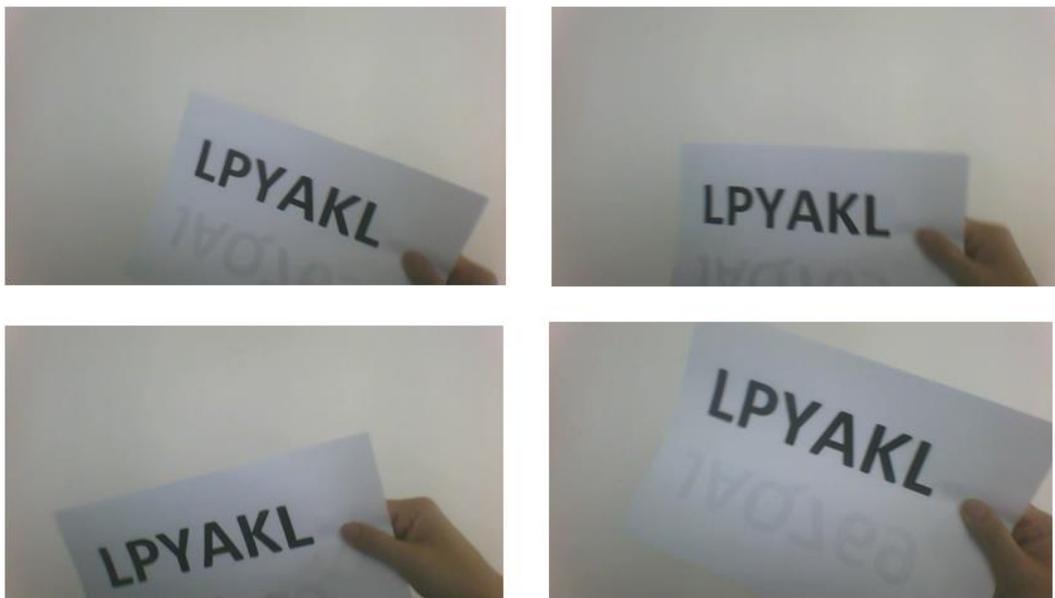


Fig 5.1 The tilt frames of a video

In our experiment, four videos were used. All the shaking was random. The characters include 26 English letters 'A'~ 'Z' and 10 Arabic numbers '0' ~ '9'. Fig 5.2 presents the four extracted frames from the four videos.

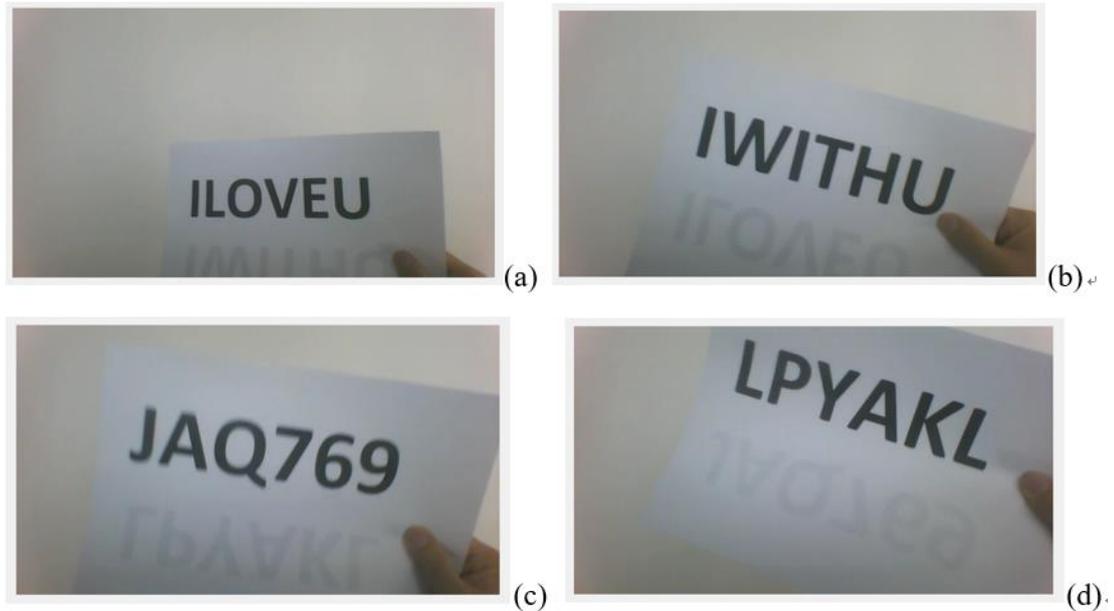


Fig 5.2 The tilt frames from four videos

### 5.3 Edge detection

In our experiments, a printout is used to replace a real license plate. The real plate is difficult to be captured and collected; however, the plate is easily to be printed on a normal A4 paper. The focus of our experiment is on the plate which does not have a frame. Fig 5.3 shows that experimental results of edge extraction of the four video frames.

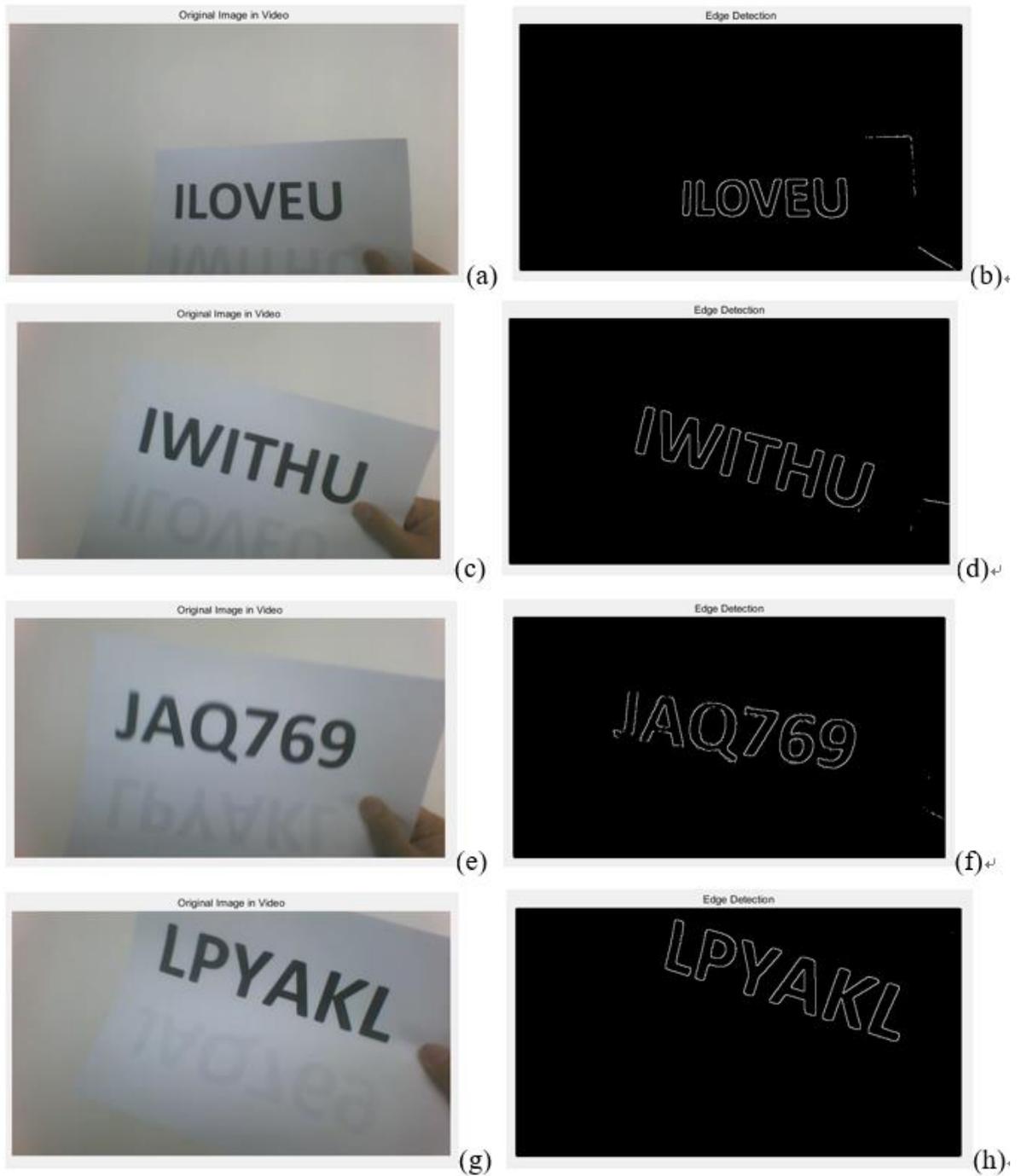


Fig 5.3 The results of edge detection from video frames

From Fig 5.3, it is clear that the edges of the characters have been extracted. A human thumb at the right corner has been successfully removed; therefore, it has less influence on the final results.

## 5.4 Straight line detection

In our implementation, we use the characters at the top and bottom boundaries to locate the horizontal line; thereby, we calculate the tilt angle of image rotation. Fig 5.4 shows the results of straight line detection from the four video frames as shown in Fig 5.3.

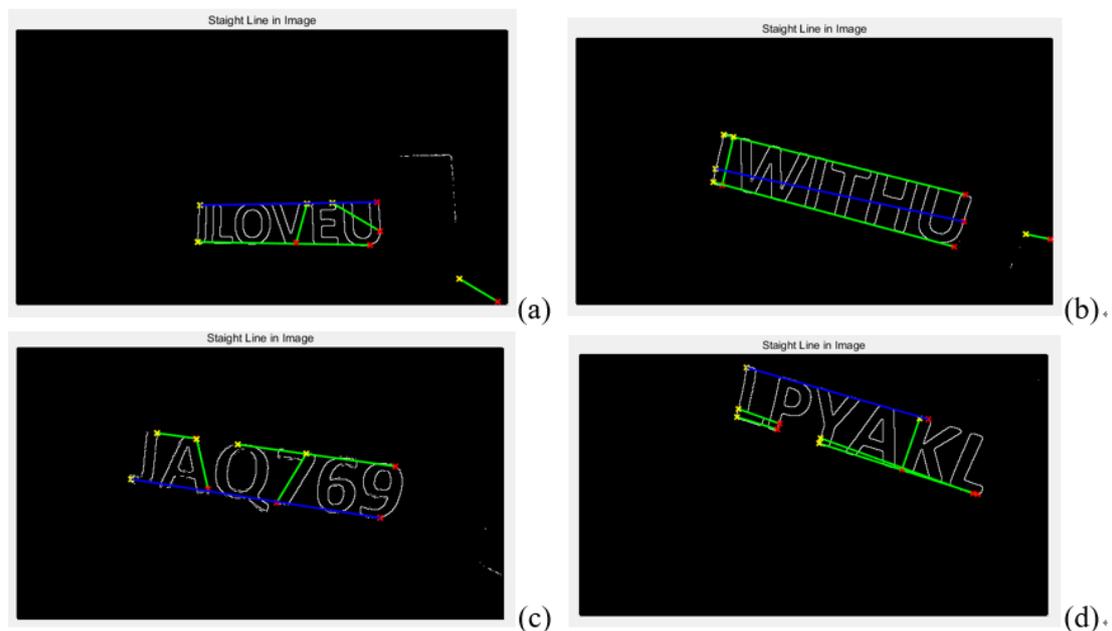


Fig 5.4 The results of straight line detection

In Fig 5.4, the blue and green lines are the boundary of those characters that are used as the input of Hough transform. Although there are other lines appeared, the longest line is the outcome of our experiment. Because the width of a number is always longer than the height, the shorter lines in the image will be ignored by the algorithm.

## 5.5 Character rotation

After received the results of straight line detection, the character tilt angle can be calculated through Hough transform. Afterwards, the image can be rotated back to the horizontal orientation based on the tilt angle. The horizontal character can be obtained.

Fig 5.5 demonstrates the rotated horizontal characters of the four video frames.

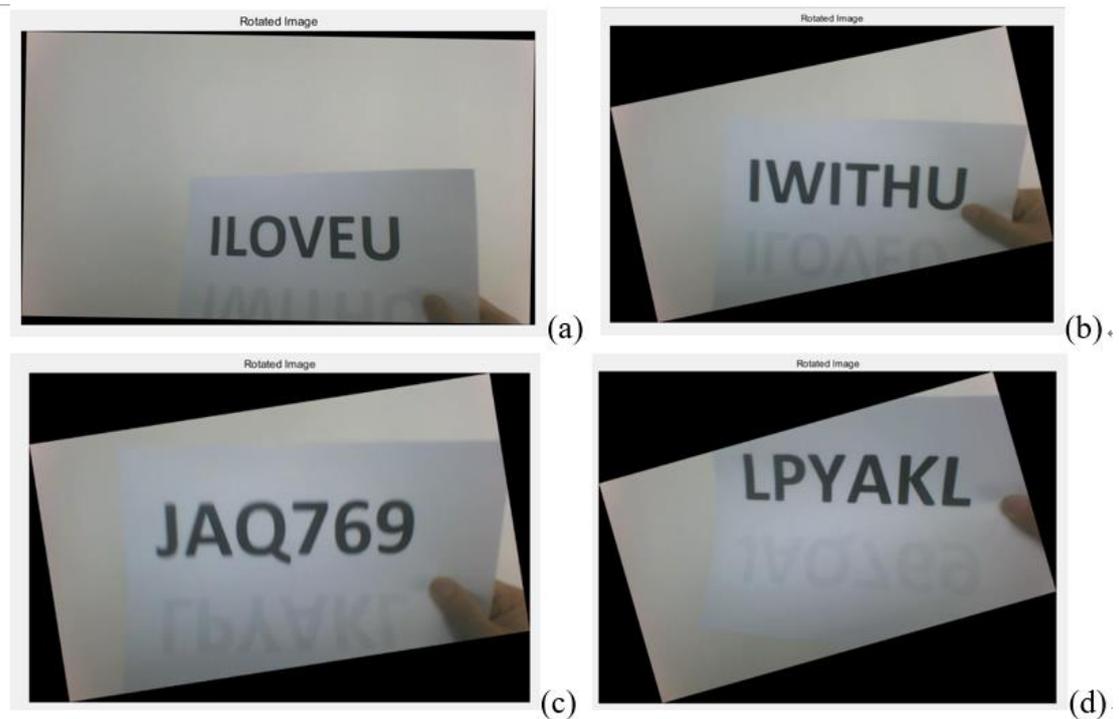


Fig 5.5 The results of rotation correction

In Fig 5.5, it is very clear that the characters have been rotated back to the horizontal orientation. The most important goal of our experiment has been achieved. It brings the following operations; thereafter, it plays a major role to guarantee the success of those final results.

## 5.6 Character region cropping

After rotated the image in Fig 5.5, the next step is to crop the character region. We tackle the rotated images with opening and closing operations in image erosion. It can get rid of the details of irrelevant target area, such as the interferences of a human finger and background in the experiments. Then, we procure the character region. According to the up, down, left and right edges, we crop the character region. Fig 5.6 shows the results of image cropping from the videos.



Fig 5.6 The cropped image

The results show that in the cropped images, there is not any irrelevant objects in the cropped images, this will greatly improve the accuracy of number plate recognition.

## 5.7 Character segmentation and recognition

When obtained the cropped image, the characters thus are able to be located by using the cropped images through binarization, brightness adjustment, median filtering and dilation operations, etc. Then, its boundary of each character can be determined. According to the segmentation, a single character could be separated. Fig 5.7 describes the results of character segmentation and recognition from those four videos.

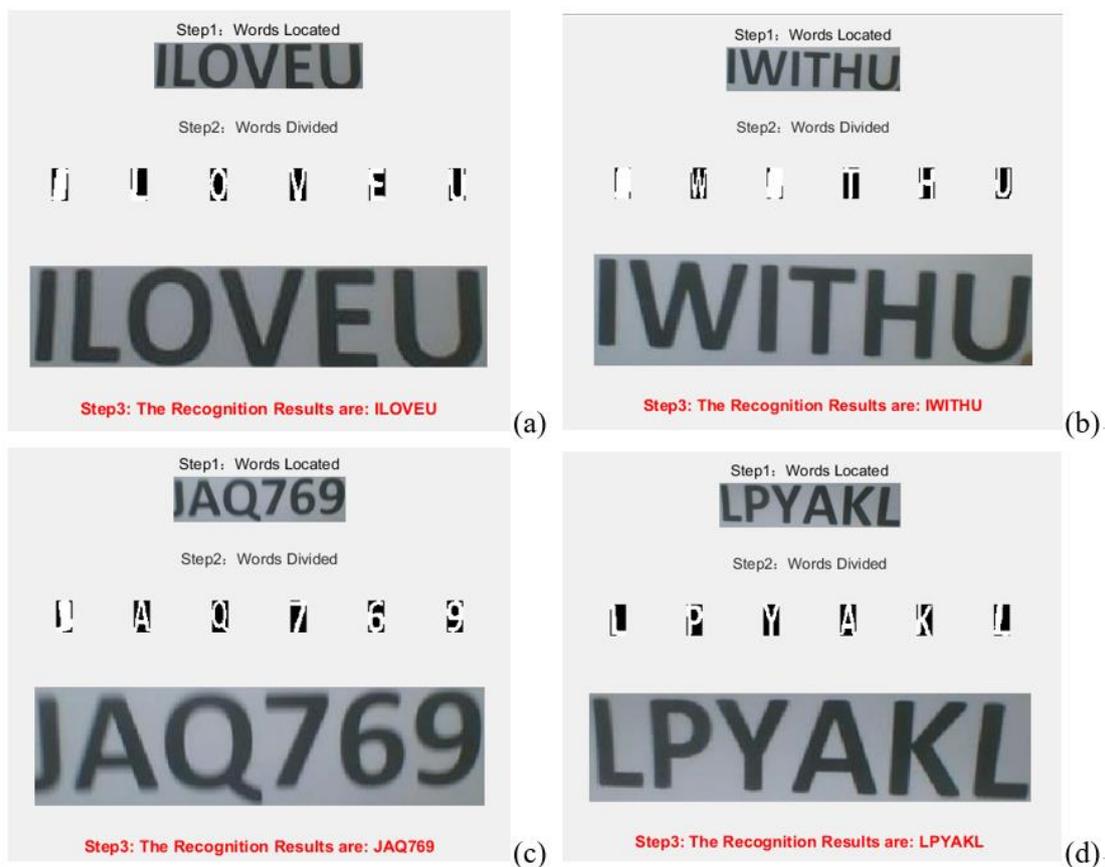


Fig 5.7 Character segmentation and recognition

After obtained the single character, it can be compared to the characters from the character dataset. The maximum similarity of these characters is treated as the outcome. For the GNN algorithm, the recognition was conducted according to the training data and the training results. The features of all the samples were extracted and have been

applied to the inputs of GNN algorithm. Finally, according to the training, the results of character recognition were output as the result.

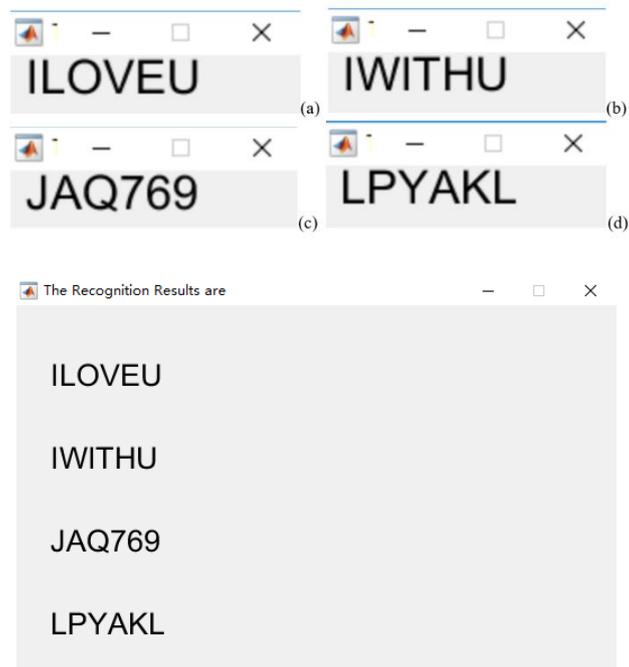


Fig 5.8 Recognition results of four sample videos

Fig 5.8 presents each result of the corresponding video and the results were integrated together as the final output of our experiment. The goal of this thesis has been achieved.

## 5.8 Accuracy

There are six videos used in the experiment to test the correct rate of image rotation correction and character recognition. We parse each video into frames and remove the blurred images. All the plates in our videos contains 6 characters. But the content and the sequence of characters are different. All the correct rates of rotation correction are shown as Table 5.1 and the results of plate recognition using BP and GNN are provided in Table 5.2.

Table 5.1 Correct rate of image rotation correction

Videos	Numbers of video frames	Numbers of corrected frame	Rate of correct correction
Video 1	20	19	95%
Video 2	30	29	96.7%
Video 3	40	38	95%
Video 4	50	48	96%
Video 5	60	57	95%
Video 6	80	76	95%

Table 5.2 Accuracy of character recognition using BP and GNN

Video	Number of frames	Correct recognized frame using BP	Accuracy of BP	Correct recognized frame using GNN	Accuracy of GNN
Video 1	20	14	70%	15	75%
Video 2	30	20	66.7%	23	76.7%
Video 3	40	26	65%	30	75%

Video 4	50	32	64%	37	74%
Video 5	60	37	61.7%	44	73.3%
Video 6	80	48	60%	58	72.5%

With the BP algorithm, the accuracy is around 65%; however, with the GNN algorithm, the images can be recognized increase around 72.5%. We find that the accuracy of BNN is that always higher than BP. It means GNN has the better performance. In our experiment, if the number of samples can be increased, the accuracy may be much higher. From these tables, we find that Hough transform takes a great role for line detection in image rotation correction.

## 5.9 Discussions

In this chapter, the details of our experiment have been introduced step by step. In each module, the data and results of our experiments have been presented. At the same time, corresponding explanation has been described.

Section 5.1 shows the environment of our experiments. Both the camera and the laptop are normal equipment. So, it means the experiment can be conducted under normal condition. It does not have the special requirement about the equipment.

In Section 5.2, we describe the image extraction. We extract any frames of the sample videos for processing. There is not special requirement. Any video frames have not negative influence on the processing.

The edge extraction has the direct influence on straight line detection. Because the straight line detected in this step, will not be the top or bottom line of the character region if the character edges are not exact. Then, it will lead to the wrong angle of the tilted image. The image thus cannot be rotated back to the right position. Thus, the character segmentation and recognition take a pivotal role in number plate recognition. Therefore, this step is an important one amongst those of the whole experiment.

In Section 5.4, we introduce the straight line detection based on top or bottom edges of the character region. In the experiment, the background is simple and there are not so many interferences in the image. Therefore, we can detect the straight line quickly and exactly. However, if there is a similar shape or object at the background, the detected result will be affected and lead to a low accuracy of the straight line detection. These interferential lines may be treated as the edge of this region.

Section 5.5 shows the character rotation. According to the straight line detected in Section 5.4, we use Hough transform to calculate the tilt angle. We rotated the image to the opposite direction with the tilt angle and receive the adjusted image. This is the core step of our experiment. According to the result, it is obvious that Hough transform has a good performance (Polishetty, Roopaei, & Rad, 2016).

In Section 5.6, the image can be processed with opening and closing operations in erosion to reduce or remove the interferences such as the human finger. Thus, the characters can be cropped out.

According to the cropped images in Section 5.6, it provides convenience to the character segmentation and recognition in Section 5.7. Each character can be segmented from the character region and compared to the character sample in the character dataset. Finally, the character, which matches the highest similarity with the sample in the template, is regarded as the output.

Section 5.8 gives two tables for presenting the image rotation accuracy and the accuracy of BP and GNN, respectively. From those, we find that Hough transform is effective for image rotation. Both BP and GNN algorithms have good performance in the character recognition. However, GNN has stronger robustness than BP.

## 5.10 Expanded experiments

Because the experiments are to simulate license plate using a printed paper instead of a real plate, the practicability of these experiments has been doubted. Therefore, the expanded experiment using a real plate needs to be presented. There are two videos used in the experiments to verify the practicability of the LPR system. In each of the videos, the frames of a real plate is captured by using a camera. We mosaicked the plates for the purpose of privacy preservation.

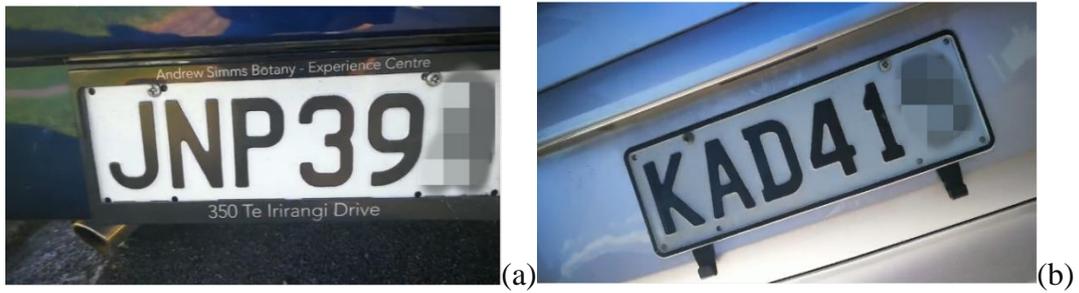
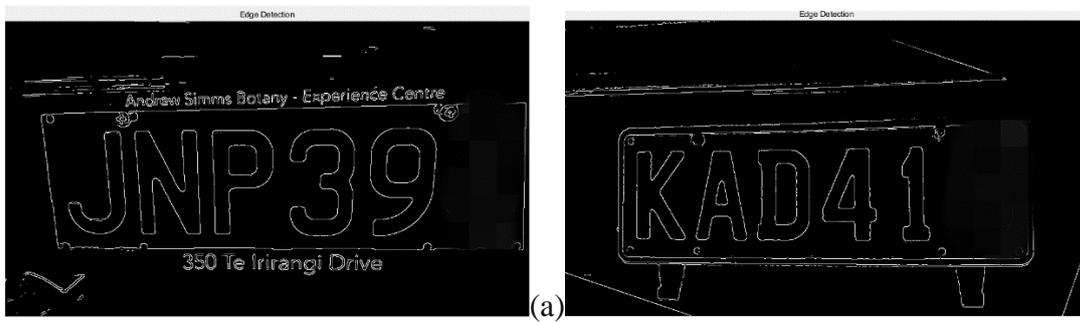


Fig 5.9 The tilt frames of a video

A frame of the video will be chosen randomly. So, the plate may be at any position and state.



(b)

Fig 5.10 The results of edge detection from these video frames

The edges of these plates and the characters can be detected; the plate will be rotated back according to the edge rotation of these plates.

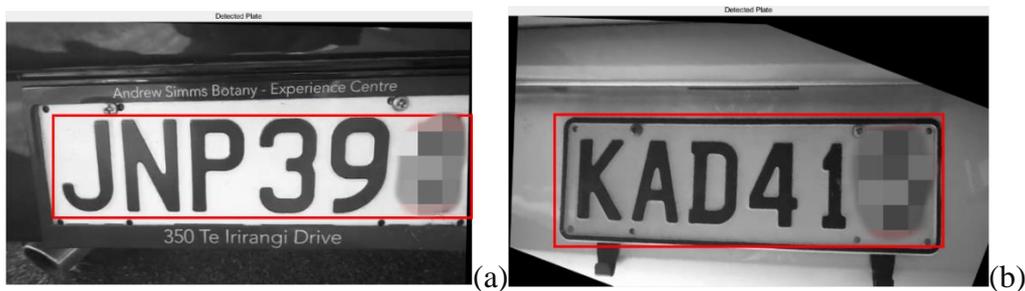


Fig 5.11 The results of rotation correction

The results of this plate rotation are presented. The plate is at the stable and horizontal position.

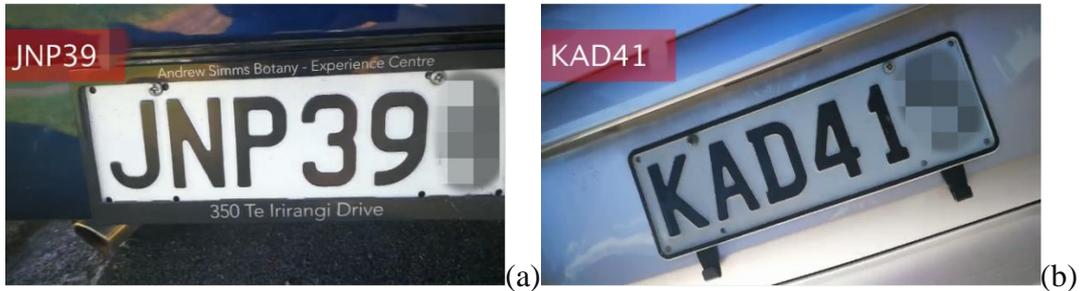


Fig 5.12 The results of character recognition.

With these two expanded experiments, we verify that the video of real plate can be realized with this LPR system. It gets the further success.

### 5.11 Accuracy of the expanded experiments

In the expanded experiments, we also used 6 videos to compare the different accuracies between BP and GNN. The details are presented below:

Table 5.3 Accuracy of character recognition using BP and GNN for real plates

Videos	Number of frames	Correct recognized frame using BP	Accuracy of BP	Correct recognized frame using GNN	Accuracy of GNN
Video 1	20	12	60%	13	65%
Video 2	30	17	56.7%	20	66.7%

Video 3	40	22	55%	26	65%
Video 4	50	27	54%	32	64%
Video 5	60	33	53.3%	38	63.3%
Video 6	80	42	52.5%	50	62.5%

From Table 5.3, we find that the accuracy of GNN is still higher than BP algorithm.

With the real plate, the performance of GNN is better than BP. However, the holistic accuracy using the real plate is lower than the paper plate in the previous experiments.

It seems that the real plate in the real environment has some interferences so that the results are not as good as the previous experiments. But even so, the performance of GNN is good and stable. It indicates the plate recognition of two situations in real.

## Chapter 6

# Conclusion and future work

*In this thesis, each step of a license plate recognition has been implemented. The corresponding methods and results also have been detailed. We achieved the objective and received the desired results. In this chapter, a conclusion of this thesis is drawn and the limitations and future work are also presented.*

## 6.1 Conclusion

In the first chapter, the background of license plate recognition was introduced. We could know that this is the major constituent part of intelligent transportation (Dalida et al., 2016). Then, we introduced the basic principle of LPR. The hardware and software were presented. In addition, we described each module of the LPR system which contains image extraction, image pre-processing, plate locating, character segmentation and character recognition. Finally, our objective has been raised. Because road surface may not always be flat, the vehicle will keep shaking when it runs on these uneven roads. The plate in the video footages, captured by a camera, is also shaking. Therefore, it has the influence to the detection and recognition in the LPR. Meanwhile, most of roads have a gradient slope; the cameras monitoring the road always have slant angles to capture the plate. All these reasons affect the accuracy of the LPR. Therefore, our objective was to detect, correct the plate and rotate the plate back to the horizontal orientation, then complete following steps of our experiment.

Chapter 2 showed literature review about the LPR. The basic steps had been described again. Then, we introduced a number of domains which can use the LPR and its applications. In addition, several factors which affect the accuracy of this recognition have been pointed out. Among those factors, in this thesis we only select the two factors: shaking and tilting. Furthermore, we presented the GNN algorithm and its relevant methods which can be used in the LPR for our experiment. Finally, the state-of-the-art technologies of the LPR have been shown. In addition, many kinds of ongoing projects and relevant researchers also had been briefed.

In chapter 3, the methodology of this thesis has been presented. According to the questions in the literature review, the corresponding solutions were justified. To achieve the objectives, we explained the research methods to resolve these problems. The image correction module and character recognition module have been detailed. Finally, the experiments and our expected results also have been described.

Chapter 4 first introduced the LPR implementation procedure. After that, the details of each step have been presented. In addition, we introduced the algorithm correspondingly. Because these are the key parts of this thesis, we used a large portion of content to describe each step of the algorithms including Hough transform. It was the core algorithm in the experiment which has the large and direct connection with the image correction.

Chapter 5 is our experimental part. In this chapter, a large amount of the experimental data and screenshots of the results have been demonstrated. Each step of the experiment has been detailed; all the results and screenshots have been explained. It could make the experiments crystal clear. Finally, we showed two tables to describe the accuracy of image rotation and character recognition with two different algorithms. The experiments with the simulative plates which have not frames differing from the real plate. Even if so, the result still demonstrated that Hough transform has a good performance in our experiment. Therefore, we could achieve the objectives and receive the expected results of this thesis.

Therefore, in this thesis, we have two contributions: one is the plate rotation correction using Hough transform with the plate, but without the frame. The other is the new GNN-based algorithm for plate recognition with outperformed results.

## 6.2 Limitations and future work

Although the whole experiment has been implemented successfully, there are still several limitations in the experiment which needs to be improved in the near future.

Firstly, we used numbers and letters printed on an A4 paper to simulate the real number plate. Because it is difficult to get the real vehicle which is shaky in a video footage from the roadside camera, we print out several characters on the paper and keep swinging the paper before a laptop camera to simulate the scenario. In addition, the paper is clean

and the characters are regular on the paper. However, in practice, some license plates may have been stained, warped or wrecked.

Secondly, because we used the paper plate to simulate the real plate before the laptop camera in indoor, the background we used is simpler than the outdoor background. If the camera located beside a road, the background must be very complicated. Those objects may influence the plate recognition.

In future, we will use real license plates in our experiments; the environment of the experiment can be shifted from indoor to outdoor. It may have a sophisticated background of real scenarios. In addition, we can change the number plates to test the plate recognition.

On the other hand, the current work needs to input the video footages which was captured before the experiments. The plate recognition was not carried out in real time. Therefore, in future, we will add the real-time functionality to our plate recognition and make it more effective and practical.

## References

Adatrao, S., & Mittal, M. (2016). *An analysis of different image preprocessing techniques for determining the centroids of circular marks using Hough transform*. International Conference on Frontiers of Signal Processing, 110-115.

Ahmed, S. (2015). *Shadow aware license plate recognition system*. Soft computing, 19(1), 225-235.

Akoz, O., & Karsligil, M. E. (2010). *Video-based traffic accident analysis at intersections using partial vehicle trajectories*. IEEE Signal Processing and Communications Applications Conference (SIU), 499-502.

Al-Ghaili, A. M., Mashohor, S., Ramli, A. R., & Ismail, A. (2013). *Vertical-edge-based car-license-plate detection method*. IEEE Transactions on Vehicular Technology, 62(1), 26-38.

Anagnostopoulos, C. N. E. (2014). *License plate recognition: A brief tutorial*. IEEE Intelligent Transportation Systems Magazine, 6(1), 59-67.

Arulmozhi, K., Perumal, S. A., Deepak, T. T., & Nallaperumal, K. (2012). *A centroid based Hough transformation for Indian license plate skew detection and correction of IR and color images*. IEEE International Conference on Computational Intelligence and Computing Research, 1-4.

Arulmozhi, K., Perumal, S.M., Mohan, M. V., & Nallaperumal, K. (2012). *Skew detection and correction of Indian vehicle license plate using polar Hough transform*

*research*. IEEE International Conference on Computational Intelligence and Computing research, 1-4.

Ashtari, A. H., Nordin, M. J. & Fathy, M. (2014). *An Iranian license plate recognition system based on color features*. IEEE Transactions on Intelligent Transportation Systems, 15(4), 1690-1705.

Asif, M. R., Chun, Q., Hussain, S., Fareed, M. S., & Khan, S. (2017). *Multinational vehicle license plate detection in complex backgrounds*. Journal of Visual Communication and Image Representation, 176-186.

Bagheri, M., Madani, M., Sahba, R., & Sahba, A. (2011). *Real time object detection using a novel adaptive color thresholding method*. In ACM International Workshop on Ubiquitous Meta User Interfaces, 13-16.

Baran, R., Rusc, T., & Fornalski, P. (2016). *A smart camera for the surveillance of vehicles in intelligent transportation systems*. Multimedia Tools & Applications, 75(17), 10471-10493.

Biao, Y., Qin, Y., & Ruien, K. (2014). *Two-row license plate extraction based on CIE Lab color space in a digital image*. Journal of Digital Information Management, 12(5), 343-347.

Bulan, O., Kozitsky, V., Ramesh, P., & Shreve, M. (2017). *Segmentation- and annotation-free license plate recognition with deep localization and failure identification*. IEEE Transactions on Intelligent Transportation Systems.

Cao, L., Zhang, X., Chen, W., & Huang, K. (2014). *License plate localization with efficient Markov chain Monte Carlo*. In International Conference on Internet Multimedia Computing and Service, 295.

Castellanos, R., Kalva, H., Marques, O., & Furht, B. (2010). *Event detection in video using motion analysis*. In proceedings of the first ACM International Workshop on Analysis and Retrieval of Tracked Events and Motion in Imagery Streams, 57-62

Chen, H. L., Hu, B., Yang, X., Yu, M., & Chen, J. (2014). *Chinese character recognition for LPR application*. International Journal for Light and Electron Optics, 125(18), 5295-5302.

Chen, R. B., & Luo, Y. F. (2012). *An improved license plate location method based on edge detection*. International Conference on Applied Physics and Industrial Engineering 2012, Physics Procedia, 1350-1356.

Cui, L., Li, K., Chen, J., & Li, Z. (2011). *Abnormal event detection in traffic video surveillance based on local features*. International Congress on Image and Signal Processing (CISP), 362-366.

Dalida, J. P. D., Galiza, A. J. N., Godoy, A. G. O., Nakaegawa, M. Q., Vallester, J. L. M., & Cruz, A. R. D. (2016). *Development of intelligent transportation system for Philippine license plate recognition*. IEEE Region 10 Conference, 3762-3766.

Dallalzadeh, E., & Guru, D. S. (2010). *Feature-based tracking approach for detection of moving vehicle in traffic videos*. In International Conference on Intelligent Interactive Technologies and Multimedia, 254-260.

Deb, K., Khan, I., Saha, A., & Jo, K.H. (2012). *An efficient method of vehicle license plate recognition based on sliding concentric windows and artificial neural network*. International Conference on Computer, Communication, Control and Information Technology, 812-819.

Dua, S., Singh, J., & Parthasarathy, H. (2016). *Rotation angle estimation of an image using least square method*. International Conference on Signal Processing and Integrated Networks, 296-300.

Fang J., & Dai S. G. (2012). *A vehicle license plate location and correction method based the characteristics of license plate*. In the 10th World Congress on Intelligent Control and Automation Intelligent Control and Automation, 42-46.

Gou, C., Wang, K. F., Yao, Y., J., & Li, Z. X. (2016). *Vehicle license plate recognition based on extremal regions and restricted Boltzmann machines*. IEEE Transactions on Intelligent Transportation Systems, 17(4),1096-1107.

Guilherme, L. C., Francisco, A. D. S., Danilo, R. P., Leandro, L. D. A., Almir, O. A., Joao, P. P., ... Helton, M. S. (2017). *A new method for automatic vehicle license plate detection*. IEEE Latin America Transactions, 15(1), 75-80.

Ha, P. S., & Shakeri, M. (2016). *License plate automatic recognition based on edge detection*. *Artificial Intelligence and Robotics*, 170-174.

Htay, M. M., & Gopalakrishnan, A. K. (2016). *Localization and recognition of a Myanmar license plate based on partially cut character structure*. *International Conference on ICT and Knowledge Engineering*, 38-43.

Huang, Y. F., & Huang, J. J. (2011). *Semantic event detection in baseball videos based on a multi-output hidden Markov model*. In *ACM Symposium on Applied Computing*, 929-936.

Janowski, L., Kozłowski, P., Baran, R., Romaniak, P., Glowacz, A., & Rusc, T. (2014). *Quality assessment for a visual and automatic license plate recognition*. *Multimedia Tools & Applications*, 68(1), 23-40.

Khan, S., Thainimit, S., Kumazawa, I., Marukat, S., & Ghaffar, F. (2017). *Character segmentation on traffic panels using visual appearance from roadside imagery*. *Asian Conference on Defence Technology*, 65-69.

Kieran, D., & Yan, W. (2010). *A framework for an event driven video surveillance system*. *International Conference on Advanced Video and Signal Based Surveillance*, 97-102.

Kim, D., Song, T., Lee, Y., & Ko, H. (2016). *Effective character segmentation for license plate recognition under illumination changing environment*. *IEEE International Conference on Consumer Electronics*, 532-533.

Ktata, S., & Benzarti, F. (2012). *License plate detection using mathematical morphology*. International Conference on Sciences of Electronics, Technologies of Information and Telecommunications, 735-739.

Lalimi, M. A., Ghofrani, S., & McLernon, D. (2013). *A vehicle license plate detection method using region and edge-based methods*. Special Issue on Image and Video Processing, Computers and Electrical Engineering, 39(3), 834-845.

Lan, J., Guo, M., Liu, X., Sun, X., Aibibu, T., & Ran, B. (2013). *Real-time detection algorithm for moving vehicles in dynamic traffic environment*. IEEE International Conference on Electro/Information Technology, 1-6.

Li, P., & Connan, J. (2010). *Number plate detection using double segmentation*. In proceedings of the 2010 Annual Research Conference of the South African Institute of Computer Scientists and Information Technologists, 386-389.

Li, S. S., & Li, Y. M. (2015). *A recognition algorithm for similar characters on license plates based on improved CNN*. International Conference on Computational Intelligence and Security, 1-4.

Liu, P., Li, G. H., & Tu, D. (2015). *Low-quality license plate character recognition based on CNN*. International Symposium on Computational Intelligence and Design, 53-58.

Manzanera, A., Nguyen, T. P., & Xu, X. L. (2016). *Line and circle detection using dense one-to-one Hough transforms on greyscale images*. Journal on Image & Video Processing, 2016(1), 1-18.

Meng, Z. (2013). *PPCA-based license plate detection algorithm*. ACM SigSoft software Engineering Notes, 38(3), 1-4.

Muammer, C. (2014). *Car license plate recognition based on EKE-Poisson transform*. Journal of Intelligent & Fuzzy Systems, 27(4), 2023-2028.

Mukhopadhyay, P., & Chaudhuri, B. B. (2015). *A survey of Hough transform*. Pattern Recognition, 48(3), 993-1010.

Neto, E.C., Gomes, S.L., Rebouças, F. P. P., & De Albuquerque, V. H.C. (2015). *Brazilian vehicle identification using a new embedded plate recognition system*. Measurement: Journal of the International Measurement Confederation, 36-46.

Patel, H. N., Desai, K., & Panchal, T. (2015). *An algorithm for automatic license plate detection from video using corner features*. International Conference on Information Processing, 301-306.

Paunwala, C., & Patnaik, S. (2010). *An improved license plate extraction technique based on gradient and prolonged Haar wavelet analysis*. In the International Conference and Workshop on Emerging Trends in Technology, 618-622.

Polishetty, R., Roopaei, M., & Rad, P. (2016). *A next-generation secure cloud-based deep learning license plate recognition for smart cities*. IEEE International Conference on Machine Learning and Applications, 286-293.

Qiu, J. N., Zhu, N. D., Wei, Y., & Yu, X. Q. (2016). *An optimized license plate recognition system for complex situations*. International Conference on Audio, Language and Image Processing, 586-590.

Roy, A. C., Hossen, M. K., & Nag, D. (2016). *License plate detection and character recognition system for commercial vehicles based on morphological approach and template matching*. International Conference on Electrical Engineering and Information Communication Technology, 1-6.

Saha, S., Basu, S., & Nasipuri, M. (2015). *ILPR: An Indian license plate recognition system*. Multimedia Tools and Applications, 74(23), 10621-10656.

Saini, M. K., & Saini, S. (2017). *Multiwavelet transform based license plate detection*. Journal of Visual Communication and Image Representation, 128-138.

Saleem, N., Muazzam, H., Tahir, H. M., & Farooq, U. (2016). *Automatic license plate recognition using extracted features*. International Symposium on Computational and Business Intelligence, 221-225.

Sarker, M. M. K., Yoon, S., & Park, D. S. (2014). *A fast and robust license plate detection algorithm based on two-stage cascade AdaBoost*. KSII Transactions on Internet and Information Systems, 8(10), 3490-3507.

Schuster, R., Mörzinger, R., Haas, W., Grabner, H., & Van Gool, L. (2010). *Real-time detection of unusual regions in image streams*. In Proceedings of the International Conference on Multimedia, 1307-1310.

Su, B., & Lu, S. J. (2017). *Accurate recognition of words in scenes without character segmentation using recurrent neural network*. Pattern Recognition, 397-405.

Sun, G. M., Zhang, C. H., Zou, W. W., & Yu, G. Y. (2010). *A new recognition method of vehicle license plate based on Genetic Neural Network* IEEE Conference on Industrial Electronics and Applications, 1662-1666.

Tian, J. M., Wang, R., Wang, G. Y., Liu, J. G., & Xia, Y. C. (2015). *A two-stage character segmentation method for Chinese license plate*. Computers and Electrical Engineering, 539-553.

Türkyilmaz, I. & Kaçan, K. (2017). *License plate recognition system using artificial neural networks*. Special issue: Software and Digital Content Technologies, 39(2), 163-172).

Uddin, J., Rahman, M. M., Rakib, M., Ferdous, M. J., Kabir, S., & Azam, M. A. (2014). *Low complexity offline character recognition of license plate using perspective view*. International conference on: Computer and Information Technology, 185-190.

Volna, E., & Kotyrba, M. (2013). *Vision system for license plate recognition based on neural networks*. International Conference on Hybrid Intelligent Systems (HIS), 140-143.

Wang, C. M., & Liu, J. H. (2015). *License plate recognition system*. International Conference on Fuzzy Systems and Knowledge Discovery, 1708-1710.

Wang, N. G., Zhu, X. W., & Zhang, J. (2016). *License Plate Segmentation and Recognition of Chinese Vehicle Based on BPNN*. International Conference on Computational Intelligence and Security, 403-406.

Xia, Y. J., Chen, J. L., Lu, X. D., Wang, C. H., & Xu, C. (2015). *Big traffic data processing framework for intelligent monitoring and recording systems*. Big Data Driven Intelligent Transportation Systems, Neurocomputing, 139-146.

Xing, J. J., Li, J., Xie, Z. F., Liao, X. X., & Zeng, W. (2016). *Research and implementation of an improved radon transform for license plate recognition*. International Conference on Intelligent Human-Machine Systems and Cybernetics, 42-45.

Xu, T. L., Wang, Y. B., & Chen, K. (2016). *Tailings saturation line prediction based on genetic algorithm and BP neural network*. Journal of Intelligent & Fuzzy Systems, 30 (4), 1947-1955.

Xu, W., Zhang, Y., Lu, J., Tian, Y., & Wang, J. (2011). *A framework of simple event detection in surveillance video*. Intelligent Computing and Information Science, 556-561.

Zhang, T., Xu, C., Zhu, G., Liu, S., & Lu, H. (2010). *A generic framework for event detection in various video domains*. In the International Conference on Multimedia, 103-112.

Zhang, W. L., Li, H. B., Yan, X., & Liu, Z. X. (2016). *A method of recognizing curve direction based on Hough transform*. International Symposium on Computational Intelligence and Design, 3-6.

Zhang, Y., Wang, J., Fu, W., Lu, H., & Xu, H. (2011). *Specific vehicle detection and tracking in road environment*. In the third International Conference on Internet Multimedia Computing and Service, 182-186.

Zhao, Y., Yuan, Y., Bai, S., Liu, K., & Fang, W. (2011). *Voting-based license plate location*. IEEE Conference on Intelligent Transportation Systems (ITSC), 314-317.

Zheng, L. H., He, X. J., Samali, B. J., & Yang, L. T. (2013). *An algorithm for accuracy enhancement of license plate recognition*. International Journal of Computer and System Sciences, 79(2), 245-255.

Zhu L. H., & Liang S. F. (2012). *An algorithm based on geometric region to correct tilt license plates*. International Conference on Industrial Control and Electronics Engineering, 665-667.