Blind Spot Warning Using Deep Learning

Yiting Shen

A project report submitted to the Auckland University of Technology

in partial fulfillment of the requirements for the degree of

Master of Computer and Information Sciences (MCIS)

2018

School of Engineering, Computer and Mathematical Sciences

Abstract

This report introduces blind spot warming of vehicles that uses cameras mounted to reduce traffic accidents. In recent years, there have been many new blind spot warning systems designed for vehicles which mainly use radar technology. These warming systems monitor and detect moving object information based on analog signals. In this project, blind spot monitoring using cameras was developed along with clear and intuitive indications. It is generally acknowledged that when driving, observation of blind spots is vital, especially for long and heavy vehicles such as buses. If a bus has several cameras pointed in the blind spots, the driver can view the road clearly in real time.

This report will firstly describe the reason why intelligent surveillance systems need to be used so as to observe blind spots of vehicles through several cameras. This will be followed by the definition and importance of blind spot detection and its technical background. Next, an example will be used to explain based on histogram and SIFT algorithms in blind spot monitoring and HMM is employed to calculate the probability of accidents occurring in the blind spots. After that, RNN will be used as a model of deep learning to predict the cars that will turn up in the blind spots. Finally, the limitations of this technology will be explored and future work will be briefed.

Keywords: Blind spot detection system, intelligent surveillance systems, cameras and car accidents, histogram, SIFT, HMM and RNN.

Table of Contents

Chapter 1 Introduction 6

1.1 Research Questions 8

1.2 Contribution 9

1.3 Objectives of This Report 9

1.4 Structure of This Report 9

Chapter 2 Literature Review 10

2.1 Introduction 11

2.2 Blind Spot Detection 11

Chapter 3 Methodology 14

3.1 Histogram .... **Error! Bookmark not defined.**

3.2 SIFT 20

3.3 Calculate the Number of Cars in Blind Spots 26

3.4 HMM 28

Chapter 4 Results 33

4.1 Data Collection and Experimental Environment 34

4.2 Limitations of the Research 47

Chapter 5 Analysis and Discussions 48

5.1 Analysis 49

5.2 Discussions 49

Chapter 6 Conclusion and Future Work 50

6.1 Conclusion 51

6.2 Future Work 51

References 53

List of Figures

Figure 3.1 The four blind spot zones of the car……………………………….……...17

Figure 3.2 No car and car turn-up…………………..…..……………………………19

Figure 3.3 Car turn-up……………………………………………………………..19

Figure 3.4 Feature points…………………………………………………………..23

Figure 3.5 Feature comparison…...……………..……………………….………...26

Figure 3.6 Number calculation…………………………………………….............27

Figure 3.7 Images cut from the video analysis…….……………………………….28

Figure 3.8 State transition…….…………………………………………………...29

Figure 3.9 HMM structure…………………………………………….....................30

Figure 4.1 Training process of the left blind spot……………………..……………..36

Figure 4.2 Forecast sequence of the left blind spot……………………………….....37

Figure 4.3 Compare the forecast sequence of the left blind spot………………………...37

Figure 4.4 Compare the forecast sequence with updates of the left blind spot…………..38

Figure 4.5 Training process of the right blind spot...39

Figure 4.6 Forecast sequence of the right blind spot……………………………………39

Figure 4.7 Compare the forecast sequence of the right blind spot………………………40

Figure 4.8 Compare the forecast sequence with updates of the right blind spot…………41

Figure 4.9 Training process of the front blind spot……………………………………...42

Figure 4.10 Forecast sequence of the front blind spot…………………………………..42

Figure 4.11 Compare the forecast sequence of the front blind spot………………….......43

Figure 4.12 Compare the forecast sequence with updates of the front blind spot……….44

Figure 4.13 Training process of the behind blind spot…………………………………..44

Figure 4.14 Forecast sequence of the behind blind spot………………………………...45

Figure 4.15 Compare the forecast sequence of the behind blind spot…………………...46

Figure 4.16 Compare the forecast sequence with updates of the behind blind spot……...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: Yiting Shen Date: 01 June 2018

Acknowledgment

Fisrtly, I want to show my great gratitude to my primary supervisor Dr Wei Qi Yan. As I know little about deep learning and never learned it before; Dr Yan taught me the most professional knowledge and gave me help what I needed in this research project. In addition, he was often concerned with my research schedule and progress, gave me a plenty of useful advice. Without his guidance, I cannot finish my study before the final due date. Certainly, I also thank my secondary supervisor Dr. Ruan ji and the school administrators of AUT for their invaluable guidance.

Yiting Shen

Auckland, New Zealand

01 June 2018

Chapter 1
Introduction

This chapter is composed of five parts: the first part introduces the background and motivations; the second part includes the research question, followed by the contributions, objectives, and structure of this report.

Nowadays, cars have become a more necessary transportation than walking due to its great importance in daily lives. However, whilst cars provide a great convenience, they easily cause traffic accidents, affect our ordinary lives and threat property security. In recent years, global number of cars has risen unsustainably and the number of traffic accidents has also increased; both have become a social issue around the world. According to an investigation conducted in 2008, the number of accidents between vehicles and motorcycles has increased by 20% from DGT. A large number of these accidents were caused by drivers who paid not enough attention to their blind spots and possessed no pedestrian awareness (Twisk et al., 2013). It is quite easy for lorry drivers to overlook vehicles in their blind spots when moving between lanes on the road; this is the main motivation for developing a blind spot detection system (BSDS), stated by Blanc, et al. (2007). To reduce the number of accidents effectively, several cameras need to be installed in the blind spots of the lorry so that the intelligent surveillance system can monitor all events in real time. In this way, drivers cannot forget their blind spots, which can lead to traffic accidents. Sotelo and Barriga (2008) put forward a blind spot warning system (BSWS) based on direct vision.

In this report, we will mainly focus on our research of developing a blind spot detection system based on direct vision to detect whether there are any vehicles in the blind spot of another vehicle. This will include the calculations of the occurrence through probability by using HMM. From this, the cars that will enter blind spots can be predicted according to the data collected using RNN.

1.1 Research Questions

1. What techniques can be implemented in real time for a BSWS based on cameras?

2. How does blind spot detection based on vision improve the accuracy compared to traditional methods?

1.2 Contribution

1. *High accuracy*. This BSWS uses digital cameras to detect a vehicle in a quick, accurate and effective manner when compared with systems that use radar or sonar technologies, which easily make mistakes,

2. *Development*. It is generally acknowledged that a BSWS is an important part of an autopilot system. If the BSWS could be developed and used correctly, the autopilot system would follow very quickly after.

1.3 Objectives Of This Report

The main objective is to create a system to reduce the number of accidents resulting from a lack of observations of blind spots. The BSWS will be developed to monitor the blind spots of drivers in real time by installing cameras and calculating the probability of an incident concerning the blind spots.

1.4 Structure Of This Report

The structure of this project report is as follows:

- Chapter 2 will consist of a literature review, a discussion about related studies and research of BSDSs based on different principles.
- Chapter 3 will introduce the research methods, including the design of experiments and how the results will be compared.
- Chapter 4 will implement the proposed algorithms, collect experimental data and demonstrate the research outcomes in the form of figures and tables. Additionally, the limitations of these proposed methods will be explored in detail.
- Chapter 5 will summarize and analyze the experimental results.
- Chapter 6 will conclude the research project and discuss possible future work.

Chapter 2 Literature Review

The topic of this report is to develop a prototype for blind spot warning based on vision, and this chapter will introduce a number of traditional methods and the relevant research outcomes of blind spot monitoring.

2.1 Introduction

With the rapid development of economies and technology, cars have become one of the most common vehicles used instead of walking. At the same time, traffic accidents caused by cars show a continuously increasing trend. To reduce the number of accidents effectively, there have been many kinds of BSWSs developed in current years.

2.2 The Blind Spot Detection

Today, a growing number of traffic accidents have attracted the public's attention in safe driving and assistance systems, it is generally considered that most accidents are caused by the mistakes of drivers (Liu et al., 2017; Kim et al., 2015). Therefore, vehicle advanced driver assistance systems have become quite popular. These systems include forward collision avoidance systems, parking assistance and blind spot warning systems.

Recently, more and more manufactures are including a blind spot warning system in their cars to avoid vehicle collision due to ignore blind spots, like Daimler AG and BMW. There are many kinds of blind spot warning, such as radar-based, ultrasonic-based and camera-based detection systems (Ra et al., 2018). However, a radar-based system, considered by Klotz and Rohling (2000), easily caused errors as it was not sensitive enough to the surrounding environment of the car and sometimes even ignored smaller vehicles like motorcycles. Ultrasonic-based BSDS have relatively low costs, but the detection range tends to be very short and it needs more time to detect cars when in use, its angular resolution is also quite low (Mahapatra et al., 2008). Compared with other systems, a vision-based system is more sensitive to the surrounding environment and the number of false detections will decrease whilst the lateral resolution will increase (Alonso et al., 2008; Blanc, Steux & Hinz, 2007).

According to many studies, sensors and radars are always used in BSDSs to improve detection accuracy. For example, Wong and Qidwai put forward a system that used six ultrasonic sensors and three cameras mounted on a car for collecting data of its surroundings in 2005. This system would predict whether a vehicle collision would happen after the data collected was processed. Visual cameras are adopted mostly which are installed under the wing mirrors on both sides of the vehicle to detect moving objects (Jung, Cho & Kim, 2010). In addition, multi-range radars and cameras are often used to detect different obstacles (Jia, Hu & Guan, 2011). In 2012, a vision-based system was proposed by Milos, and Jan, a vehicle detection system could detect possible vehicles through feature extraction from images collected from the cameras installed in the car. This would help the driver discover potential dangers in blind spots when changing lanes and avoid vehicle collision. Wu, et al. (2013) also supported this by stating that intelligent driving systems can be beneficial in protecting the driver and avoiding accidents as far as possible. Their study described a BSWS which could be used effectively and smoothly no matter when the time of day or night is. The BSWS needs only two cameras installed below the rear-view mirrors on both sides of the car. This intelligent surveillance system could obtain all the road conditions in real time and observe whether there were any vehicle accidents happening near the experimental car. Fernández, et al. (2013) considered blind spot monitoring could be aided by using systems based on passive sensors, like video cameras or active sensors, including radar and laser sensors. In addition, blind spot detection can cover a zone of 20 meters in length behind cameras and 4 meters across on both sides. In 2014, Tseng et al. developed a BSDS based on motion and static features, in which the ground detection zone was divided into four main regions. Baek, et al. (2015) also conducted a vision-based object detection along the sides of a vehicle blind spot, which adapted a HOG cascaded classifier to detect vehicles. To improve the accuracy of the camera-based detection system, Dooley et al. (2016) came up with a new idea, in which the image is divided into three parts and a different algorithm was used to detect objects in each part. According to the investigation of Hane, et al. (2017), cars can be surrounded by several cameras so that the BSWS monitors a full 360 degree field-of-view. Fisheye cameras were thought to be the most suitable cameras as they are always applied to pedestrian detection and parking assistance systems (Bertozzi et al., 2015). Furthermore, multiple cameras can be beneficial for 3D mapping, visually locating and detecting obstacles (Ray & Teizer, 2013). The aim of their study was to build sparse 3D maps for visual navigation, then locate the car and generate an accurate yet dense map that could detect obstacles.

Chapter 3
Methodology

This chapter mainly introduces the research methods, which are used in this report. It mainly contains the details of research methodology for blind spot detection using histogram, SIFT and HMM, which will be explained in details clearly in this chapter.

As we have known, the main four blind spots of a car are located in the front-below zone, the behind-below zone, the left-below zone and the right-below zone. It is therefore reasonable to install four cameras to monitor these blind spot zones individually. Because a car driver sits in a higher driving position, it is quite difficult to observe the surrounding environment which can cause serious crashes (Cheng et al., 2016; McCarthy & Gilbert, 1996). In this scenario, the front-blow zone is considered as the blind spot zone 1. Blind zone 2 refers to the rear-blow zone, Zone 3 and 4 represent the left-blow zone and the right-blow zone, respectively. The right turn for a car can be especially dangerous, even at a low speed, because of its length and different radii of inner wheels, there is a high possibility of crashing into other vehicles and pedestrians which are located in this blind spot (Zhang, Liu & Ma, 2015). As a result, once something occurs in these zones, the cameras could detect them, at the same time, the warning system should warn the driver to take any necessary actions as soon as possible. The most important factor of blind spot detection is speed; otherwise, the BSDS will be of no use. So, there are two main aims of this report, one is to test whether the BSDS can detect obstacles as soon as they occur in the blind spot zones; the other is to test whether the time span of blind spot detection is long enough so that the driver can find the potential danger in time to avoid an accident.

If there are any vehicles, pedestrian or cycle riders in the blind spot zones, the BSWS should detect them and warn the driver using the cameras installed.

To find the most suitable algorithm, four videos were captured from different blind spots including the front, rear, left and right blind spot. To reach the aim of detecting cars in the blind spot zones, there were several steps taken for the system to implement this function:

(1) Find the blind spot zone of the driver and mark it with a rectangle.

(2) Detect whether there are any cars in the blind spot zone of the driver by using different algorithms.

(3) Calculate the number of cars in the blind spots of the driver in real time.

(4) Calculate the probability of the occurrence of accidents when there are several cars

in the blind spots by using HMM.

(5) Predict the number of cars in the blind spots in the future according to experimental data.

3.1 Histogram

Nowadays, image enhancement technology has many different uses such as in traffic and medical technologies as well as others (Gonzalez and Woods, 2010). Histogram transform is a common method of image enhancement technology which was used in this project to solve the relative problems. From the view (Li, Zhang & Zhang, 2014), a histogram is defined as a statistical function between the greyscale levels and the pixel intensity, which is often used to reflect the occurrent times or frequency of different greyscale levels in one image. A histogram is always shown as a two-dimensional image, in which the abscissa represents the grey-scale levels, while the ordinate displays the occurrent times or frequency of the grey-scale levels (Sonka, Hlavac & Blyle, 2003).

To find the image with cars in the blind spot zone, a picture without any cars was chosen and its histogram was acquired to compare with others. There were three possible ways to observe the blind spot using a histogram where the inner product, entropy and HOG are calculated.

3.1.1 Inner Product

In order to detect the conditions of blind spots, the inner product of two vectors needed to be calculated first. An image without any cars in the blind spot zone was selected, as shown in Figure 3.1. This image was transformed into a greyscale image to obtain its histogram, then its vector $I_0(x_0, y_0)$ was determined, so that the length (L_0) could be calculated through equation (3.1).

$$L_0 = \mathrm{sqrt}(\mathrm{sum}(I_0' \cdot I_0)) \qquad (3.1)$$

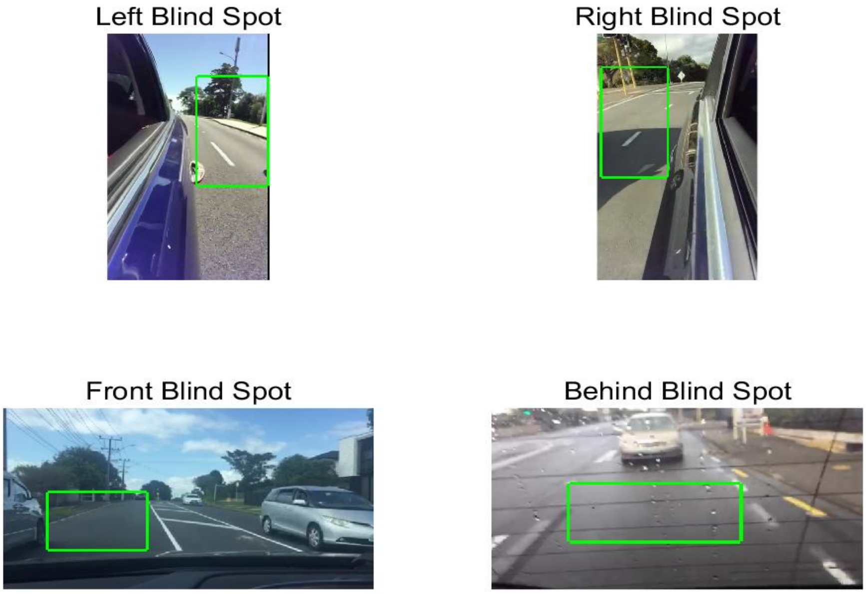


Figure 3.1: The four blind spot zones of the car

Following this, we read the video frames and the vector of each frame was set to I_i, i is the frame number; therefore, I_i can be inserted into equation (3.1) to obtain the length (L_i) of this vector:

$$L_i=\mathrm{sqrt}(\mathrm{sum}(I_i' \cdot I_i))$$

Thirdly, the inner product can be calculated through equation (3.2)

$$IP=\mathrm{dot}(I_0, I_i) \tag{3.2}$$

According to equation (3.3),

$$\mathrm{dot}(I_0, I_i)=L_0 \cdot L_i \cdot \cos\alpha \tag{3.3}$$

the value of $\cos\alpha$ can be calculated by using equation (3.4):

$$\cos\alpha = \frac{\mathrm{dot}(I0,Ii)}{L0 \cdot Li} \tag{3.4}$$

Finally, the value of $\cos\alpha$ of a different frame could be observed to obtain a specific scope [a, b]. If $\cos\alpha \in [a, b]$, it means that there were cars in this blind spot zone; otherwise, there was nothing in the blind spot of the driver.

3.1.2 Entropy

Shannon proposed the concept of information entropy, which allows the measurement of the information content of a given sequence. Miśkiewicz (2016) raised the point that entropy, as a significant parameter representing the state of the system in statistical physics, is defined for a system by a discrete and continuous probability distribution function (PDF), which is based on histogram. It is quite similar to the way the driver's blind spot was detected by using an inner product. After chosen a suitable histogram image (I_0) without any cars in the blind spot zone, it was very important to calculate the entropy of this image (H_0) through the following equation (3.5):

$$H_i = \sum_0^{255} P_i \log_2 P_i \tag{3.5}$$

In equation (3.5), P_i is defined as the probability that a certain greyscale value occurs in this image. Then, the frames of the video were read so that the entropy H_i could be calculated according to equation (3.5). H_0 and H_i were compared to detect whether there were cars in the blind spot of the driver.

3.1.3 HOG And SVM

In 2005, Dalal and Triggs designed a system which combined the HOG algorithm and SVM classifier together. But at that time, the HOG algorithm needs complex computations, so it was time consuming. As an alternative, many researchers began looking into more reliable classifiers and proposed different algorithms to extract the same features with the aim of improving the accuracy and speed of this system. These changes followed the design of an accurate pedestrian detection system using modified HOG and LSVM (Kalshaonkar & Kuwelkar, 2017). There were six steps needed to finish the experiment by using HOG and SVM:

Step 1. Obtain a positive sample set and calculate the HOG feature vector to gain the descriptors. As an example of vehicle detection, the feature descriptors of a vehicle could be extracted from a sample set of vehicles, such as in Figure 3.3.

Step 2. Obtain a negative sample set and calculate the HOG feature vector to gain the descriptors. The images of negative samples could be randomly cut form the images without any detection targets, as Figure 3.2 shows.

Figure 3.2 No car

Figure 3.3 Car turn-up

(From images cut from videos)

Step 3. SVM could be used to focus on the positive and negative samples and obtain individual models.

Step 4. Hard-negative mining by using a model to detect the negative samples from different scales in a focusing set. If the classifier detected images without targets, the images would be put into negative samples.

Step 5. Take the hard-negative samples into account and run the model again.

Step 6. Detect the test set by using the final classifier model and sliding scans for each image in a different scale, then extract the descriptors and make a classification by using a classifier (Žemgulys, et al., 2018). If the target was detected, it would be marked by a box. After scanning the image, NMS (non-maximum suppression) could be used to eliminate the overlapping superfluous targets.

3.2 SIFT

David Lowe put forward the scale-invariant feature transform (SIFT) and improved it in 2004. SIFT is an algorithm based on computing versions used to detect and describe local features. It can calculate the extreme points in the spatial scale and extract their positions, dimensions and rotation invariants. In other words, SIFT can be used to detect a particular object for image matching (Youying & Tadahiro, 2018). SIFT can be widely applied to image registration, there are many different applications of SIFT including object recognition, gesture recognition, image tracking, 3D modelling and machine manipulation and navigation (He, et al. 2018).

The description and detection of local image features could be used to identify objects. The features of SIFT were based on local interest points of the object and have no connection with image size and rotation (Wei, et al., 2018). The accuracy of the detection of partial object occlusion could be very high by using SIFT feature descriptions; sometimes, even only 3 or more SIFT object features could help calculate the position and orientation. SIFT is an algorithm that is quite suitable for fast and accurate computations of massive databases.

3.2.1 Features

There are some main features of SIFT as follows:

(1) SIFT could locate a key points in the image and maintain its invariance and rotation, scale brightness changes, and keep a certain degree of stability for viewing angle changes, refine transformations and noise.

(2) SIFT is very good at distinguishing individual details in large amounts of information and is very suitable for fast and accurate computing of massive databases because of its large amounts of information.

(3) Expandability. It could be quite convenient for the SIFT algorithm to combine with other forms of feature vectors.

(4) Substantiality. A few objects could produce many SIFT feature vectors.

3.2.2 Applications

SIFT could help deal with the issue of image registration and target tracking which was affected by the state of the target object, the environment which the scene was located in, and the image characteristics of the imaging equipment. There were some problems that SIFT could solve to a certain degree:

(1) Rotation, scaling and translation (RST) of the target;

(2) Image Affine and viewpoint transformation;

(3) Illumination effect;

(4) Target occlusion;

(5) Clutter;

(6) Noise.

3.2.3 Main Steps

The main point of using SIFT was to find the corners on different scales of space. These feature points, or key points, found by SIFT were all very specific; not being affected by illumination, Affine and noise, like corner points, edge points, luminous spots in dark areas and dark spots in bright areas. Lowe (2004) divided his SIFT algorithm into four main steps, which spends 99.8% of the time that the SIFT algorithm needs to implement the total process as the following shows (Lalonde et al., 2007):

(1) Scale-space extremum detection. Find out the image locations on all scales and use a Gaussian differentiation function to extract the potential interest point for scale and rotation invariance. Firstly, we should build the Gaussian pyramid. The Gaussian pyramid is always divided into two parts:

(a) Gaussian blur on different scale;

(b) (b) Dot interlace sampling.

A Gaussian convolution kernel is the only linear kernel to attain the scale transformation

(Lindeberg, 1994). The Gaussian blur is an image filter that uses a normal distribution function to calculate the blur model,

$$G(x, y, \partial) = \frac{1}{2\pi\partial^2} e^{-(x^2+y^2)/2\partial^2} \tag{3.6}$$

and uses this model to perform a convolution with the original image $I(x, y)$. The scale space of an image is defined as Gaussian convolution kernel of the variable, described in equation (3.7) as follows:

$$L(x, y, \partial) = G(x, y, \partial) * I(x, y) \tag{3.7}$$

where ∂ is the standard deviation of the normal distribution, with the increase of ∂, the image becomes more smooth. Then, we will build the Gaussian difference pyramid. This is the difference between the two Gaussian pyramids.

Finally, we can detect the extreme points. The extreme value is detected by comparing the intensity value with the 26 neighboring pixels including the previous image and the next one and its corresponding neighboring pixels.

(2). Key point location. At each candidate location, the point and the scale can be determined by a best fit model. The selection of the key points is based on their degree of stability. The extreme points detected are often discrete, but ternary quadratic functions can be used for fitting to determine the position and scale of key points to achieve sub-pixel accuracy. If a desired point is set as the center, the second Taylor expansion of the scale space function $D = (x, y, \text{intvl})$ is expressed by equation (3.8).

$$D(X) = D + \frac{\partial D^T}{\partial X} X + \frac{1}{2} X^T \frac{\partial^2 D}{\partial X^2} X \tag{3.8}$$

In equation (3.8), D is the grayscale value of the desired point, $X=(x, y, \text{intvl})^T$ is the offset center of this point as $D(X)$ is discrete so its derivative can be calculated by differentiation. When $D'(X)=0$, the offset $\hat{X}$ can be obtained, shown in equation (3.9).

$$\hat{X} = -\frac{\partial^2 D}{\partial X^2} \frac{\partial D}{\partial x} \tag{3.9}$$

If $\hat{X}>0.5$ in any dimension, this meant that the extreme point was closer to another point, so the key point was set to one closer to the extreme point. After moving to a new point, the same operation was performed. If the position did not converge after 5 iterations, it means that this point was not a key point. In addition, the points at the edge of the image were not considered key points. In this experiment, the feature points of one car in a blind spot are shown in Figure 3.4.

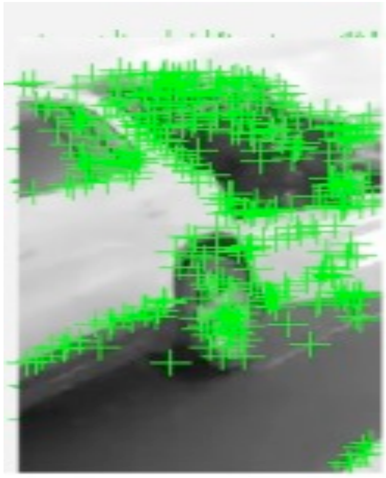


Figure 3.4: Feature points

To locate the key points, we should remove the points with low contrast firstly. Equation (3.10) could be used to determine whether a point was extreme.

$$D(\hat{X}) = D + \frac{1}{2}\frac{\partial D^T}{\partial X}\hat{X} \tag{3.10}$$

If $|D(\hat{X})|$<0.4/s, this point was not considered to be an extreme point. Through this process, the information of the extreme points was saved as data to prepare for a feature construction.

Then we can relieve the edge response. The Gaussian difference function has a strong edge response; therefore, it was important to remove the points which are located at the edge of the image. The feature of these points was to have a large principal curvature in a certain direction and have a small principal curvature in a vertical direction. If r is the ratio of the big principal curvature and the small one, H is the Hessian matrix of the key points, equation (3.11) is obtained.

$$\frac{Tr(H^2)}{Det(H)} = \frac{(r+1)^2}{r} \tag{3.11}$$

If equation (3.11) meets the following condition:

$$\frac{Tr(H^2)}{Det(H)} < \frac{(r_t+1)^2}{r_t}$$

where r_t is a threshold and here r_t=10. It means that r was considered to be quite small, so it could not be in the edge, otherwise this point was removed.

(3) Direction determination. It is based on the local gradient direction of the image with location of each key point having one or more direction. According to the local feature of the key points, each key point could be allocated a direction with rotation invariance. The local feature of the key points could be calculated when detected key points in the Gaussian pyramid image were near the Gaussian difference pyramid image. The gradient could be calculated and directional distribution at neighborhood window of the key points by using equation (3.12) and equation (3.13).

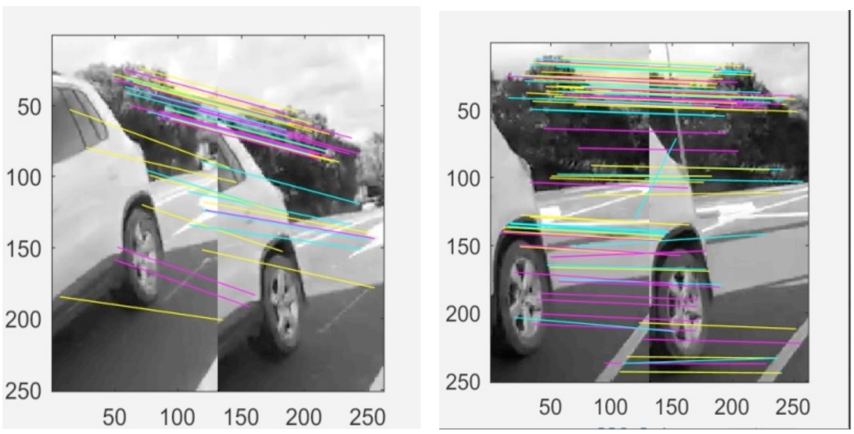
$$\mathrm{m(x,y)} = \sqrt{[L(x+1,y)-L(x-1,y)]^2+[L(x,y+1)-L(x,y-1)]^2} \quad (3.12)$$

$$\theta(\mathrm{x,y}) = \tan^{-1}\{[L(x,y+1)-L(x,y-1)]/[L(x+1,y)-L(x-1,y)]\} \quad (3.13)$$

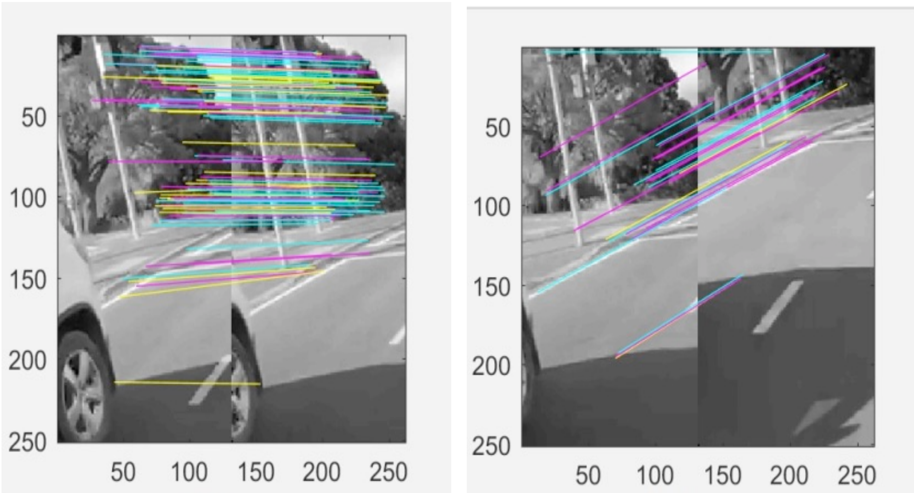
In equation (3.12) and equation (3.13), the positive direction of x is right and that of y is up. L is the greyscale value of the key point after the precise positioning described above. $m(x, y)$ is the amplitude of the gradient, $\theta(x,y)$ is the radian of the gradient and the key point is located in $(\theta(\mathrm{x,y}) \in (-\pi,\pi])$. Then the direction of 360° was divided into 36 bins, and the range of the first area was $\left[\frac{35\pi}{36},\frac{37\pi}{36}\right]$, and the others were divided in a counter clockwise direction. For m(x,y), Gaussian distribution of σ was weighted in the neighboring window to obtain a 36-direction histogram. After being smoothed twice, the histogram weighed each 3 consecutive bins twice with the size of 0.25, 0.5, and 0.25. The direction of the maximum of the histogram was the main direction of the key point. If other peaks were greater than or equal to 80% of the main direction value, one direction was assigned. Therefore, a key point may have many corresponding directions; a feature is defined as the key points with directions; a key point may also have had many features. As the first octave was a double-sized image, the coordinates and scale of the feature were converted to the octave where the original image was located. Finally, the parabolic interpolation could help accurately locate the direction of the feature.

(4) Feature descriptor. Measure the local gradient of the image on the selected scale within the neighbouring pixels around each key point. It often takes 18.75% of the time needed for the descriptor generation according to the survey of Huang et al. (2012). From the views of Lowe (2004), the SIFT descriptor is a $4 \times 4 \times 8 = 128$ dismensional vector. In this report, $h(x, y, \theta)$ was the SIFT descriptor, x and y were the locations of the $4 \times 4 = 16$ images, θ was the gradient direction, with only 8 values. Therefore, the value of $h(x, y, \theta)$ was the gradient in the direction of θ calculated from the image (x, y).

For the aim of descriptor generating, the histogram array was sorted initially and then converted it into a 128D vector. To reduce the influence of any illumination variance, this vector was normalized. Nonlinear illumination variance may have a great effect on the gradient amplitude but have a slight impact on the gradient direction. So, the gradient amplitude was set over the threshold of 0.2 and then normalized. The descriptors were sorted according to the scale of the corresponding Gaussian pyramid images. The four images from Figure 3.5 display the comparison of the key points at different times.



(a) (b)



(c) (d)

Figure 3.5: Feature comparison

3.3 Calculate The Number Of Cars In The Blind Spot Zones

The aim of BSWS was not only to detect whether there were any cars in the blind spot zone of the driver, but also to tell the driver the number of cars in the blind spots. As there were four cameras in total to observe the blind spots there were 4 videos to be analyzed. The BSWS designed should have counted the cars that moved into any blind spot area and subtract the ones that drove away, so that the number of the cars in the driver's blind spot zones could be calculated in real time. Figure 3.6 mainly shows the method to calculate the number of the cars in the left blind spot zone.

```
n1=flag1(k1)-flag1(k1-1);
n2=flag2(k2)-flag2(k2-1);
n3=flag3(k3)-flag3(k3-1);
n4=flag4(k4)-flag4(k4-1);
if (n1>0)
    num1=num1+1;
    u1=u1+1;
end
if (n1<0)
        num1=num1-1;
        h1=h1+1;
end
if ((flag1(k1)==0)&&(flag1(k1-1)==0))
    p1=p1+1;
end
if ((flag1(k1)==1)&&(flag1(k1-1)==1))
    q1=q1+1;
end
```

Figure 3.6 Number Calculation

Figure 3.7 mainly provided us the detailed information about the blind spots of the driver clearly. The green rectangle represented the blind spot zones of the driver, and once there are any cars driving into these zones, this BSDS would display the red words to warn the driver and calculate the total number of the cars in the blind spots at the same time. At this moment, there were two cars driving into the left blind spot and the right blind spot respectively and there was no car in the front blind spot and the behind blind spot, so this BSDS could warn the driver after calculations that the number of the cars in the blind spots is two now.

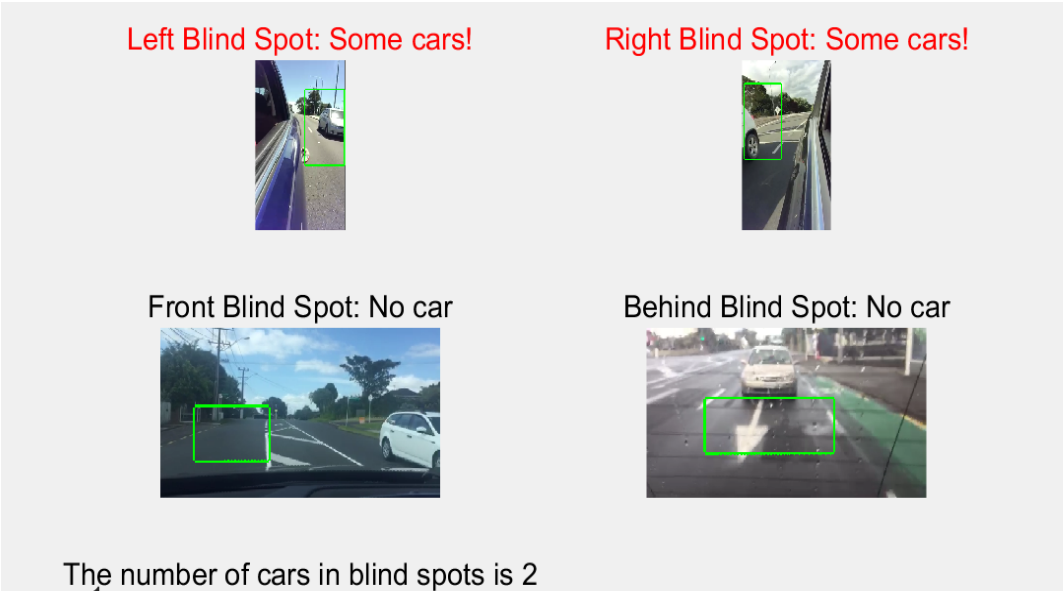


Figure 3.7 Image cut from the video analysis

3.4 HMM

A hidden Markov model (HMM) was firstly put forward by Baum and Petrie (1966). It is a statistical model and often used to describe a Markov process with unobserved states (Yusuf, Brown & Mackinnon, 2015; Toselli et al., 2016). The Markov model is described by five main parameters:

(1) N is the number of the states in this model. If q_t is the state of this model at a certain time (t) then $q_t \in \{S_1, S_2, \dots, S_n\}$, S represents the state.

(2) M is the number of possible observable objects in each state. If M is the observable values then $V_1, V_2, \dots, V_M$, and O_t is the observable value at a certain time (t) then $O_t \in \{V_1, V_2, \dots, V_M\}$.

(3) A is a state transition matrix with size N×N. It is defined as the transition probabilities of different states. A=$(a_{ij})_{N\times N}$, a_{ij} is the transition probability from S_i at t to S_j at t+1, $a_{ij} = P(q_{t+1} = S_j | q_t = S_i)$.

(4) B is an observed state probability matrix. B=$(b_{jk})_{N\times M}$, b_{jk} is the probability to obtain V_k at S_j, $E_{jk} = P(O_t = V_k | q_t = S_j)$.

(5) π is a matrix of the probability of the original states. $\pi = (\pi_1, \pi_2, \dots, \pi_N)$,

($\pi_i = P(q_i = S_i)$, $1 \le i \le N$), which is used to describe the probability distribution of different states of the observable sequence at *t*=1.

In this experiment, according to the assumption that if a car moves into a blind spot, *flag*=1; if a car left the blind spot, *flag*=0. There were four blind spots so in total there were 63 conditions as Table 1 shows.

		Left		Right		Front		Behind	
		0	1	0	1	0	1	0	1
Left	0	√	√	√	√	√	√	√	√
	1	√	√	√	√	√	√	√	√
Right	0	√	√	√	√	√	√	√	√
	1	√	√	√	√	√	√	√	√
Front	0	√	√	√	√	√	√	√	√
	1	√	√	√	√	√	√	×	√
Behind	0	√	√	√	√	√	√	√	√
	1	√	√	√	√	√	√	√	√

Table 1 The transition table

From Table 1, the conditions are marked with a "√" and the condition is marked with a "×". If a car left the front blind spot, it means that it was directly in the line of sight of the driver, this condition was removed.

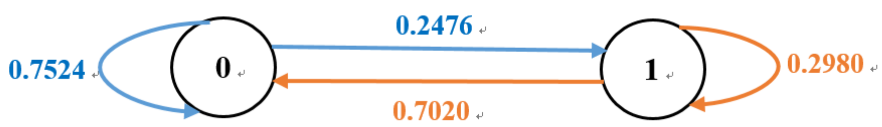


Figure 3.8 State Transition

It was difficult to record four videos of four blind spots at the same time, therefore, for convenience only two states were taken into consideration at a time. Figure 3.8 represented a state with cars and one with no cars, represented by 1 and 0 respectively. Then there were four conditions including 0→0, 0→1, 1→0 and 1→1. Using calculated results from the experiment, the transition probabilities could be obtained: $a_{00} = 0.7524$, $a_{01} = 0.2496$, $a_{10} = 0.7020$, $a_{11} = 0.2980$.

Hence, the transition state matrix **A**:

$$A=\begin{bmatrix}0.7524 & 0.2496\\ 0.7020 & 0.2980\end{bmatrix}$$

Figure 3.9 provides information about the whole structure and the connections between states. According to the Auckland Transport, from 5pm to 7pm in the city center in December, the probability of accidents occurring was 0.28, and the probability of no accidents occurring was 0.72. This is the original state probability matrix $\pi = \begin{bmatrix}0.72\\ 0.28\end{bmatrix}$.

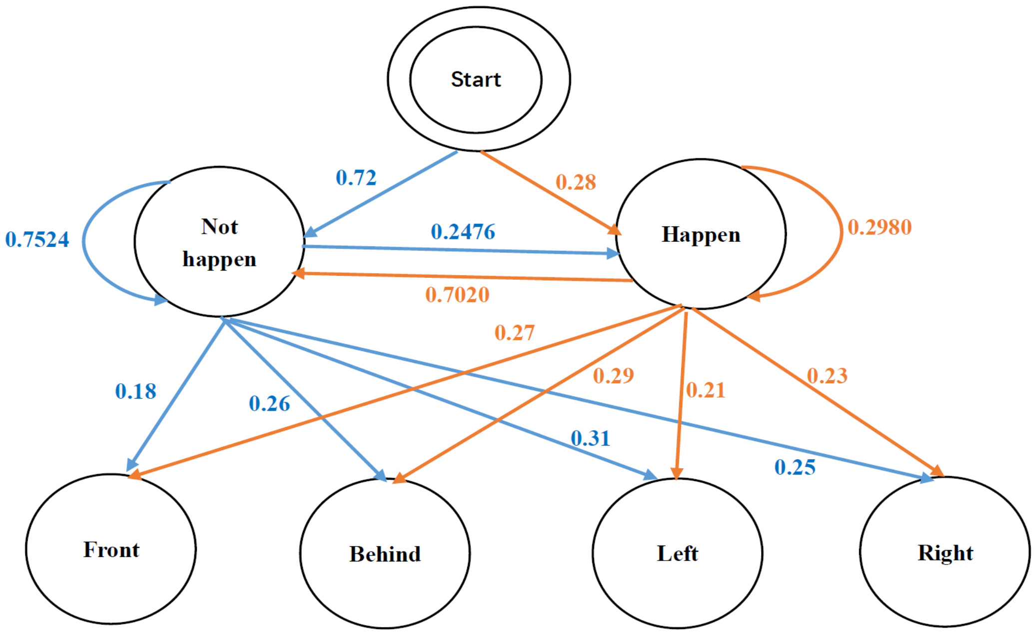


Figure 3.9 HMM Structure

After data collection and calculation, the probabilities of accidents resulting from different conditions in different blind spots could be found from Figure 3.9, which is $B=\begin{bmatrix}0.18 & 0.26\ 0.31 & 0.25\\ 0.27 & 0.29\ 0..21 & 0.23\end{bmatrix}$.

Next, there were two algorithms to deal with the problem:

I. Viterbi algorithm

TRANS: is a transition state matrix, TRANS=A;

EMIS: is a corresponding symbol generated matrix, can also be called emission matrix, which is used to observe symbol probability distribution.

The following functions will be used:

(1) *Hmmgenerate* is used to generate a sequence of states and emissions from a Markov model, which can give a random sequence, *seq* of emission symbols and a random sequence states of states, e.g.,

[*seq*, *states*] = *hmmgenerate* (*length*, *TRANS*, *EMIS*);

(2) *Hmmviterbi* is used to calculate the most probable path for a hidden Markov model. If the transition matrix (TRANS) and the emission matrix (EMIS) are known, the most likely sequence of states can be calculated by using the Viterbi algorithm, e.g.,

likelystates = *hmmviterbi* (*seq*, *TRANS*, *EMIS*);

(3) *P* is the result of dividing number of times that the word part of likely state marked as states in the training corpus by the total number of times that likely state appears in the training corpus, which can be calculated in the experiment using equation (3.14)

$$P = sum(states)/100\times100\% \tag{3.14}$$

Finally, the probability of car crashes in the city center in December could be calculated under the condition that some cars were in blind spots by using the function of Viterbi algorithm.

II. Baum-Welch algorithm

The Baum-Welch algorithm is a member of the unsupervised learning models (Yang, Wainwright & Balakrishnan, 2017) and the aim of unsupervised learning is to calculate the model parameter λ to attain the maximum of the probability of $P(O|\lambda)$ under this parameter. That is the maximum likelihood estimation, but it does not mean that $P(O|\lambda)$ is isolated; on the contrary, it is associated with its hidden states.

Using the test data, the observed sequence was set as the observed data **O** and the state sequence was set as the hidden data **I**, under the condition of including **S** observed sequences $\{O_1, O_2, \dots, O_S\}$, of which the size was T without the corresponding state sequences. Therefore, HMM was a probability model with hidden probabilities according to equation (3.15).

$$\mathrm{P}(\mathrm{O}|\lambda) = \sum_I P(O|I,\lambda)P(I|\lambda) \tag{3.15}$$

There are three main steps of Baum-Welch algorithm:

(1) Initialization

Setting n=0, returns $a_{ij}^{(0)}$, $b_j(k)^{(0)}$and $\pi_i^{(0)}$, which then obtains the model:

$$\lambda^{(0)} = (A^{(0)}, B^{(0)}, \pi^{(0)})$$

(2) Recursion

When n=1, 2, …

$$a_{ij}^{(n+1)} = \frac{\sum_{t=1}^{T-1} \xi_t(i,j)}{\sum_{t=1}^{T-1} \gamma_t(i)}$$

$$b_j^{(n+1)}(k) = \frac{\sum_{t=1,O_t=V_k}^{T-1} \gamma_t(j)}{\sum_{t=1}^{T} \gamma_t(j)}$$

$$\pi_i^{(n+1)} = \gamma_1(i)$$

(3) Termination

The model parameter was calculated to be $\lambda^{(n+1)} = (A^{(n+1)}, B^{(n+1)}, \pi^{(n+1)})$.

According to calculation, the range of λ was approximately 0.31 to 0.40.

Chapter 4
Results

This chapter shows the experimental results analyzed by using RNN based on deep learning. It will also make a discussion in terms of the limitations of this report.

4.1 Data Collection And Experimental Environment

4.1.1 RNN

Recurrent neural network (RNN) is one method of deep learning which is now applied to many different aspects widely as its unique network structure is quite helpful and beneficial when dealing with sequence data (Wang & Zhang, 2018). In 1986, Rumelhart, Hinton and Williams proposed the definition of standard RNN firstly. In this project, the deep learning model based on RNN encoder and decoder was used to analyze the related data of the experiments and predicted the number of faults in assess software reliability.

The RNN structure has the current Hidden layer connected to that of the next step. Compared with other traditional multilayer sensor systems, RNN can be impacted over time, so the next step can be affected by the current time, as stated by Graves (2012). To explain in more detail, there is an unidirectional flow of information from the input unit to the hidden unit, whilst there is another unidirectional flow of information from the hidden unit to the output one (Park & Yoo, 2017). However, under some conditions, RNN will break these restrictions and force the information flow from the output units back to the hidden units, which are known as back projections. In addition, the input of the hidden layer also contains the state of the previous hidden layer; the nodes of the hidden layer can be self-connected or interconnected (Wang et al., 2016), this is LSTM (Long Short-Term Memory). Hochreiter and Schmidhuber firstly introduced the definition of LSTM in 1997. LSTM is to avoid the long-term dependency problem.

If X is the input layer, O is the output layer, t is the number of times, s is the hidden layer and V, W and U are all weights, the state of the hidden layer at a certain time can be calculated according to equation (4.1)

$$S_t = f(U * X_t + W * S_{t-1}) \tag{4.1}$$

If there was a sequence of inputs $x_1, x_2, \ldots\ldots, x_T$, each in R^n, and the sequence of hidden states calculated by the network was $h_1, h_2, \ldots\ldots, h_T$, each in R^m, and the sequence of predictions was $\widehat{y_1}, \widehat{y_2}, \ldots\ldots, \widehat{y_T}$, each in R^k, the following equations could be obtained through iterating the equations below (Martens, J., & Sutskever, I., 2011):

$$t_i = W_{hx}x_i + W_{hh}x_{i-1} + b_h \tag{4.2}$$

$$h_i = e(t_i) \tag{4.3}$$

$$s_i = W_{yh}h_i + b_y \tag{4.4}$$

$$\widehat{y_i} = g(s_i) \tag{4.5}$$

where W_{hx}, W_{hh} and W_{yh} are the weight matrices; the sequence of t_i represents the inputs to the hidden units, and the sequence of s_i represents the inputs to the output units; b_h and b_y are bias vectors; e and g are the pre-defined vector valued functions.

4.1.2 Results

In this experiment, Matlab R2018a was used to run the Time Series Forecasting method by using deep learning to predict the sequence of the numbers of the cars in the blind spot zones (Sorkun, Paoli & Incel, 2017), which can be searched in the latest official website of MathWorks, and the results are as follows:

As discussed, when *flag*=1, there was a car moving into the blind spots, and when *flag*=0, there were no cars in the blind spots. LSTM was used to analyze the number of cars in the different blind spots from the four videos.

The four pictures (Figure 4.1, Figure 4.2, Figure 4.3 and Figure 4.4) mainly provide the information about the analysis of the number of cars in the left blind spot.

Figure 4.1 describes the training progress that the RMSE (root-mean-square-error) kept the trend of decline slow after two fluctuations when the number of iterations was less than 25, while it saw a small rising fluctuation and then finally decreases. The loss trend is quite similar to that of RMSE, but its peak is only about 0.5 while that of RMSE is 1.

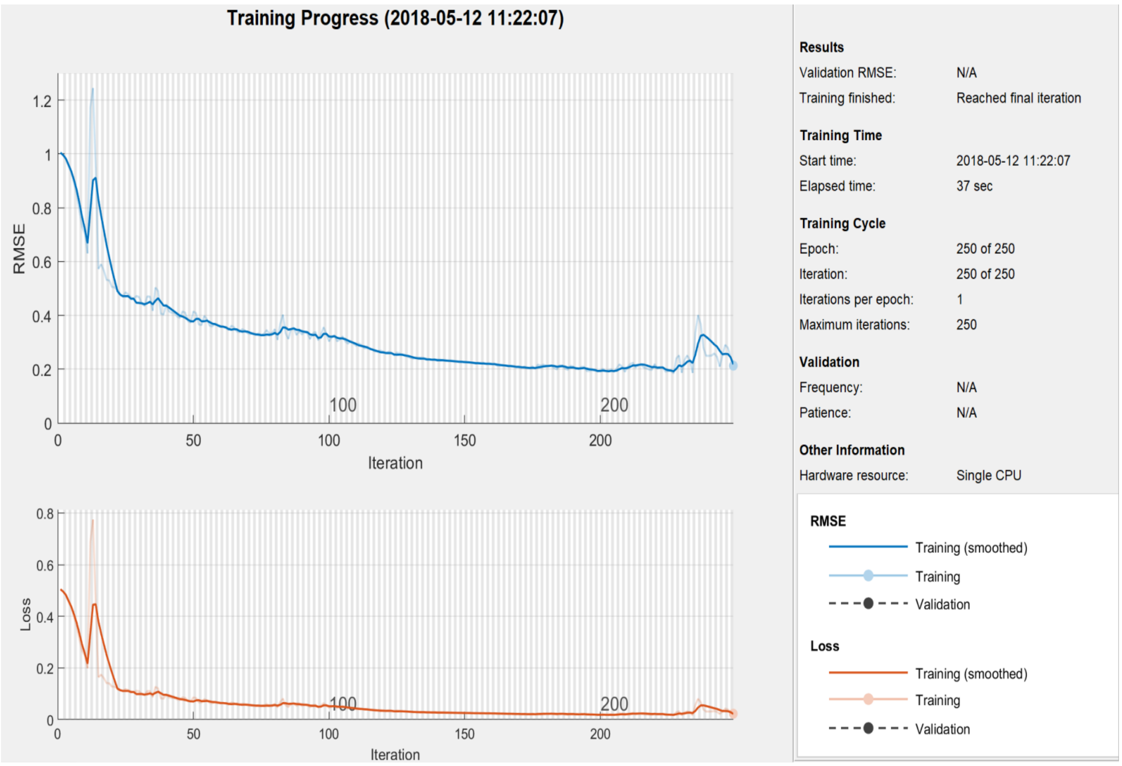


Figure 4.1 Training process of the left blind spot

Figure 4.2 displays the sequence of the observed numbers at each second and the forecast numbers at the final 50 seconds. According to the previously observed numbers in 450 seconds, the following trend of the number of the cars in blind spots could be estimated.

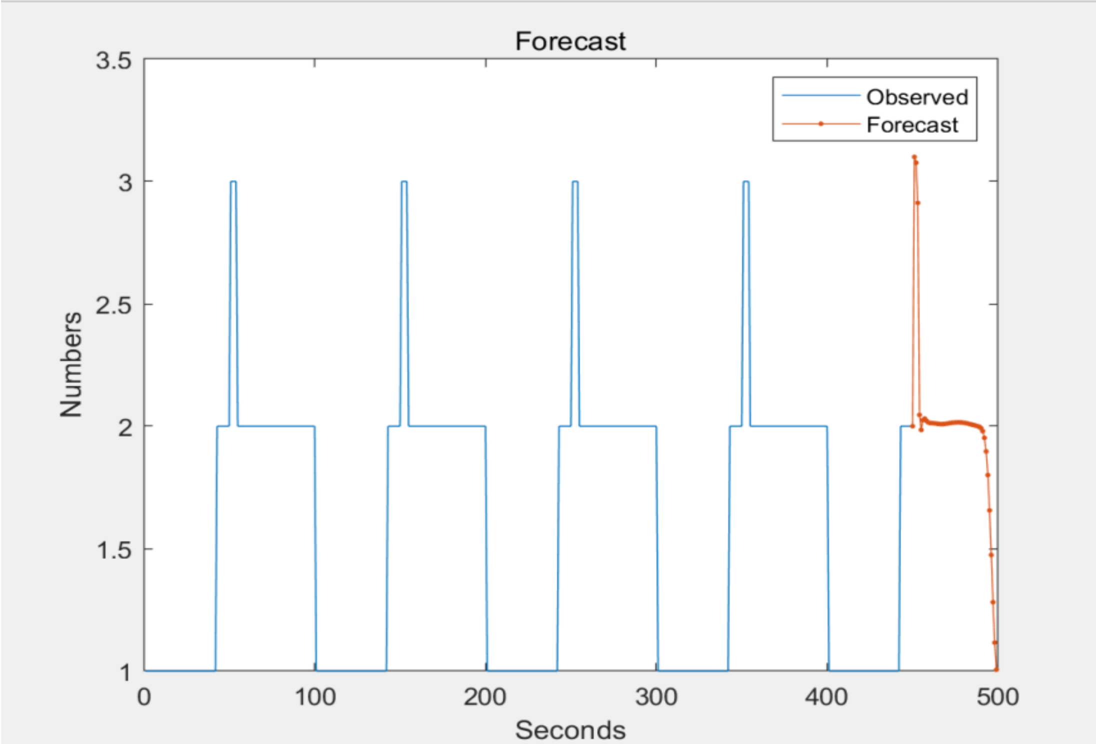


Figure 4.2 Forecast Sequence of the left blind spot

Figure 4.3 compares the observed sequence and the forecast sequence and calculates the error of each second and obtains the RMSE of 0.23764.

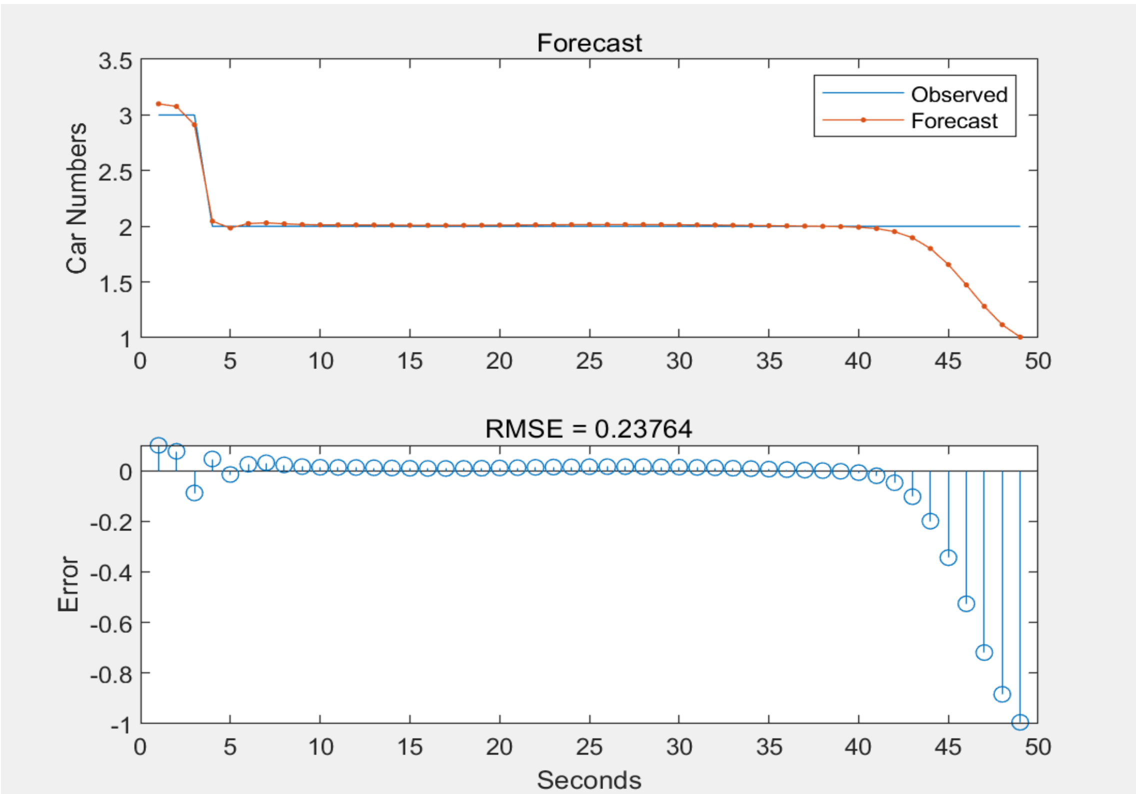


Figure 4.3 Compare the forecast sequence of the left blind spot

If the network state was reset after initializing it to avoid the effect from previous predictions, the new sequence could be predicted. Comparing with Figure 4.4, the RMSE is only 0.12146, approximately half of the upstate.

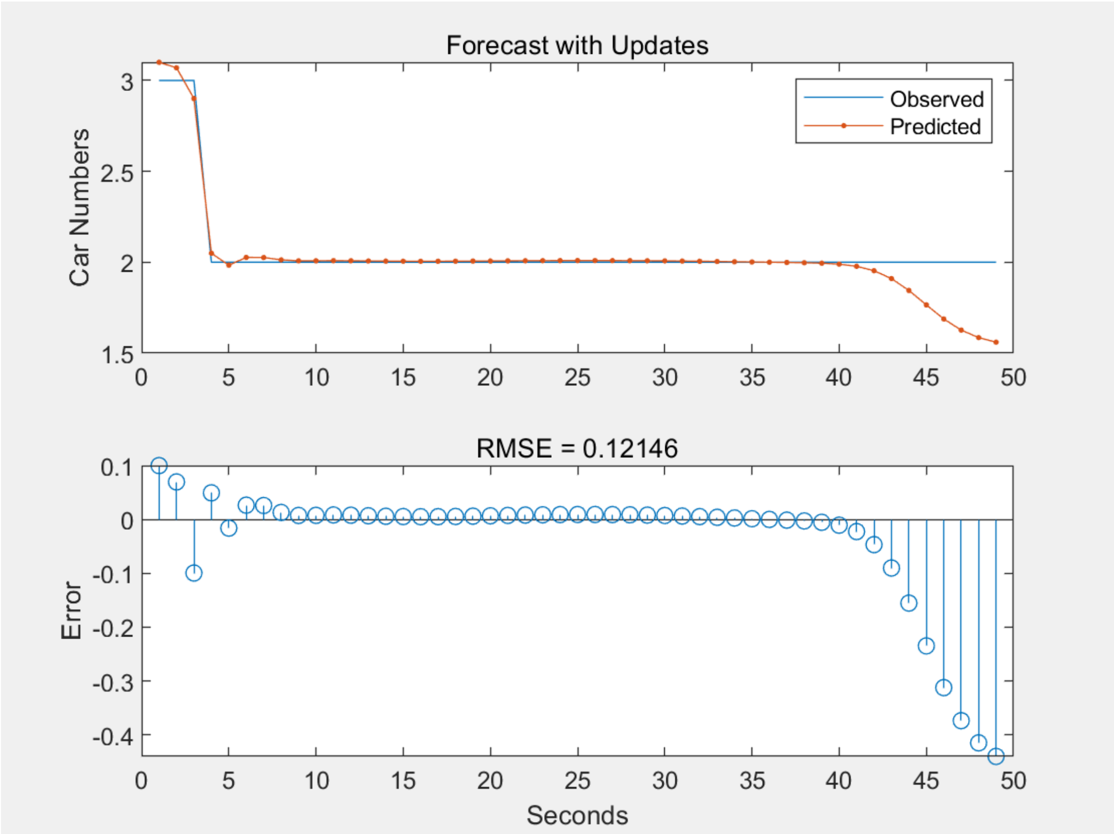


Figure 4.4 Comparison of the forecast sequence with updates of the left blind spot

Figure 4.5, 4.6, 4.7 and 4.8 show the related data about the right blind spots. In Figure 4.5, there are three relatively sharp fluctuations in this training progress, except for a steady trend of both RMSE and loss. As a result, the RMSE is always fluctuating around 0.87, peaking at 1.4 when the number iteration is 15, and the value of loss is fluctuating around 0.4, with a peak of 1 at the same progress of iteration.

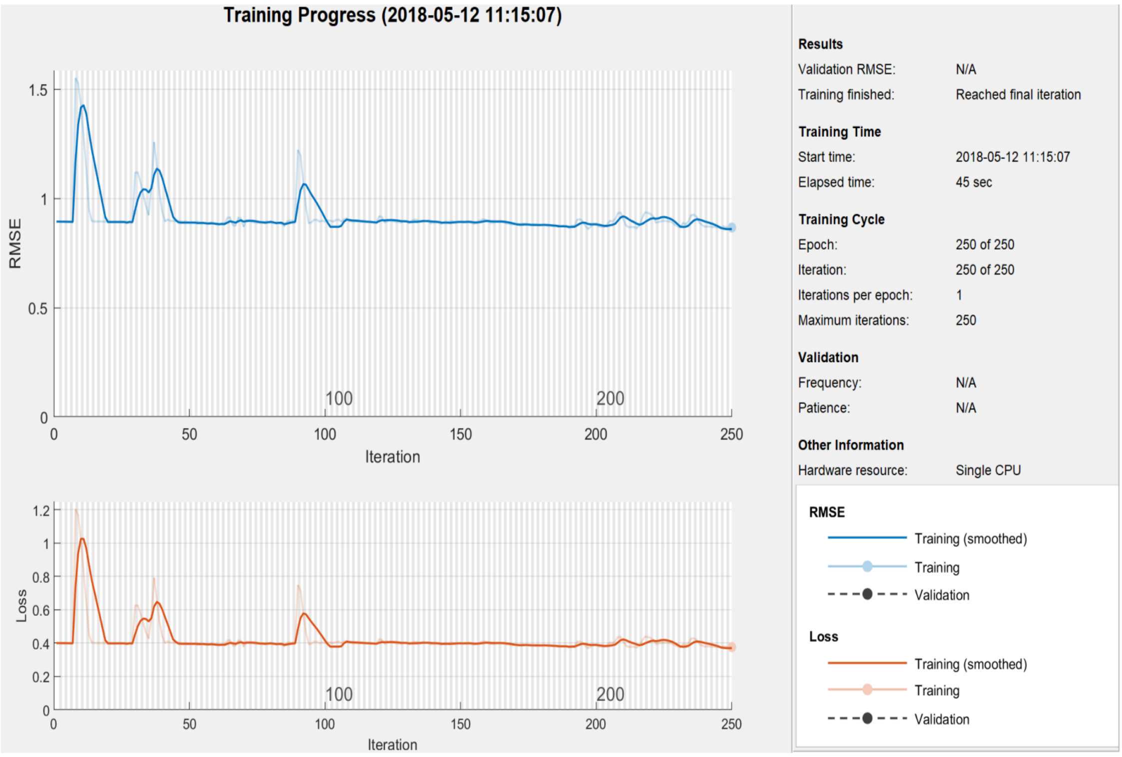


Figure 4.5 Training process of the right blind spot

According to Figure 4.6, it is clear that the observed numbers of cars in the blind spots fluctuate around 1 and 2. The forecast sequence is predicted to be around 2.

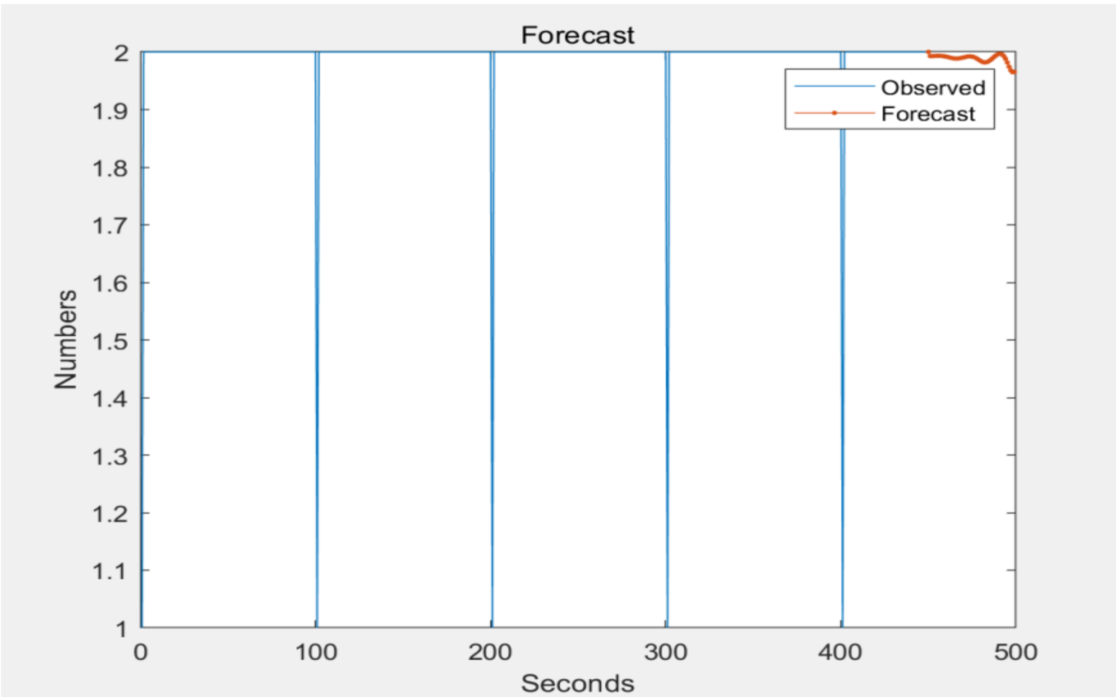


Figure 4.6 Forecast Sequence of the right blind spot

In Figure 4.7, the errors between the observed values and the forecast number are very small, less than 0.04; the RMSE is only 0.013759, much less than that in the left blind spot.

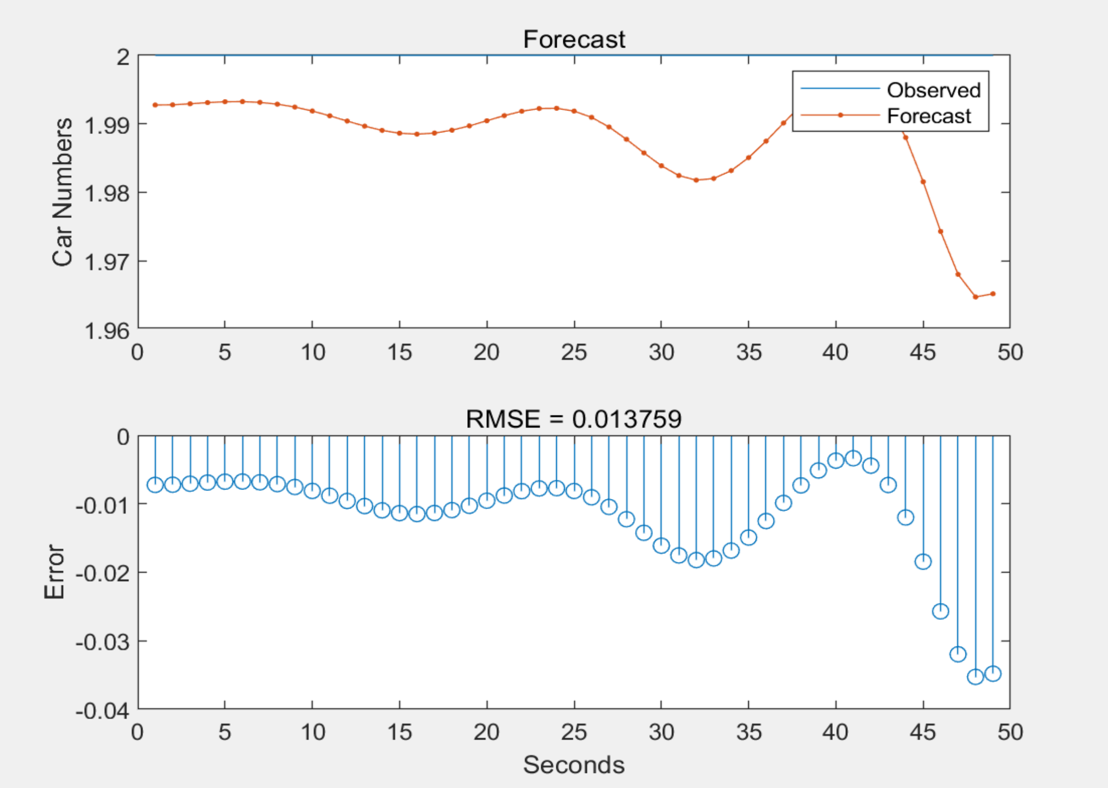


Figure 4.7 Comparison of the forecast sequence of the right blind spot

However, once the network state was reset and the sequence was predicted again, the errors increased to 0.9; from Figure 4.8, the RMSE became 0.02526 as twice as much before the updates in Figure 4.7.

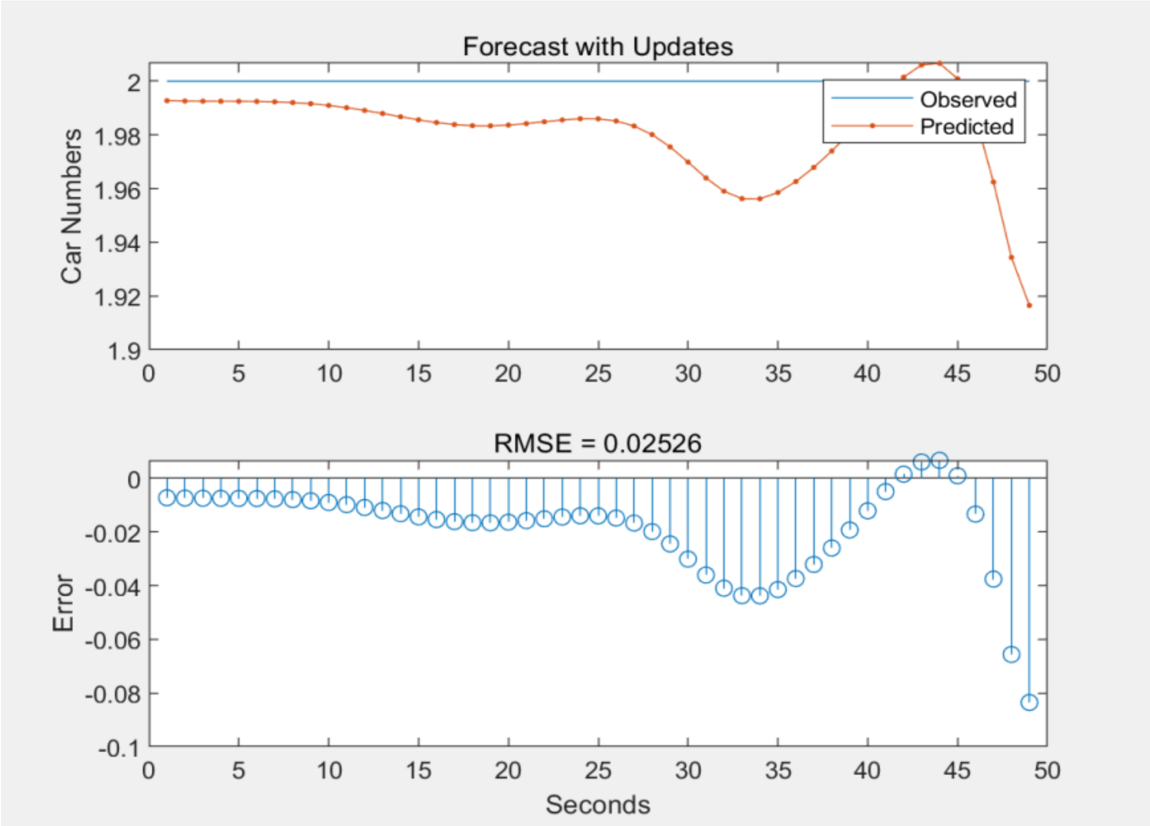


Figure 4.8 Comparison of the forecast sequence with updates of the right blind spot

The conditions of the front blind spots are shown below in the four charts of Figure 4.9, Figure 4.10, Figure 4.11 and Figure 4.12. From Figure 4.9, the RMSE shows a falling trend, but there are three relatively big waves, in the three iteration ranges ([60, 80], [210, 230] and [235, 250]), respectively. The final value of the RMSE is 0.4 and that of loss is 0.08.

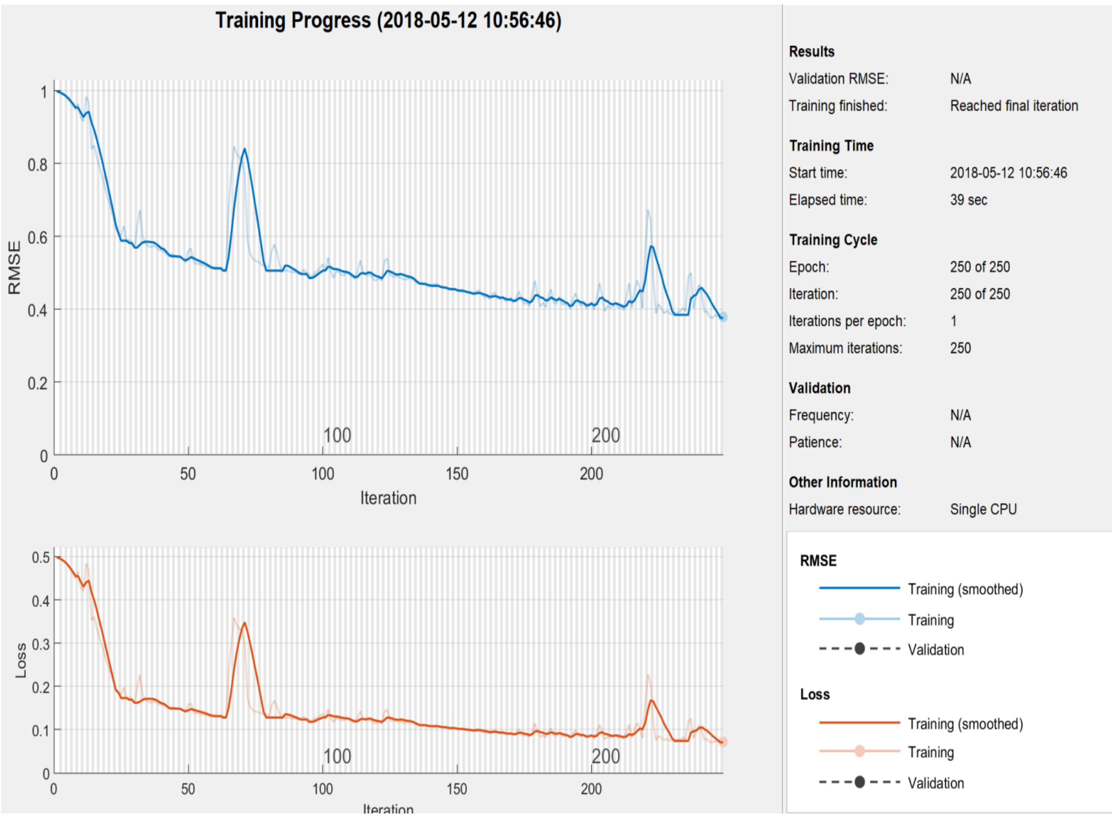


Figure 4.9 Training Process of the front blind spot

In Figure 4.10, the observed car numbers in the front spot fluctuates from 1 to 2, and the predicted car number is 2 at the beginning and then becomes 1.

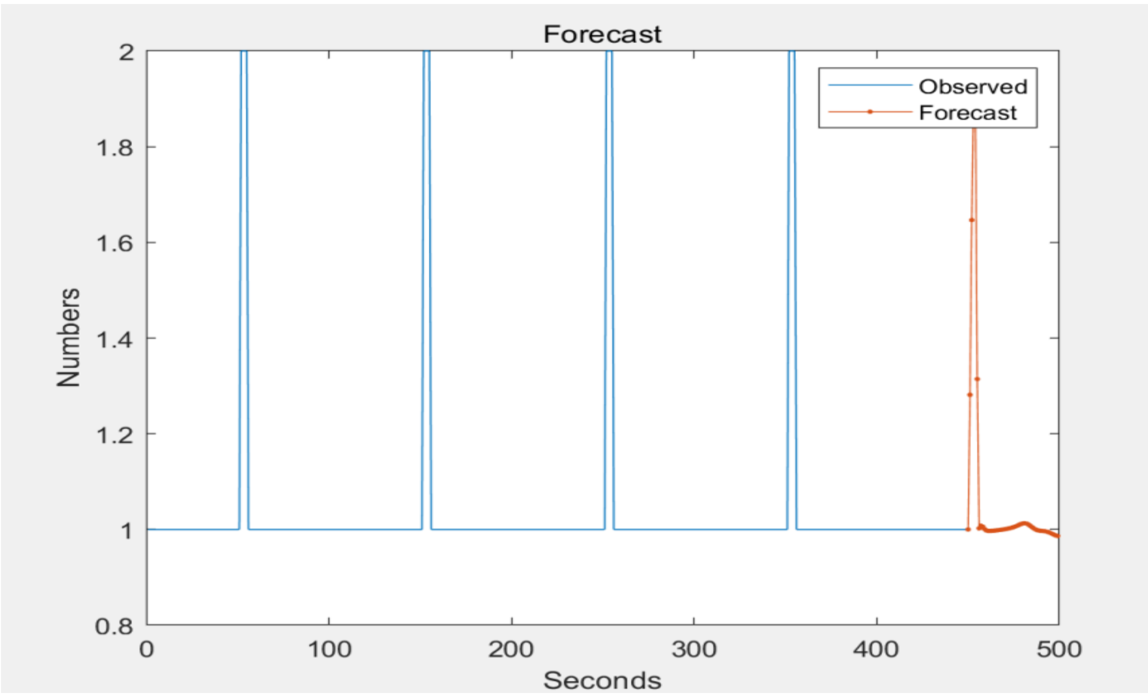


Figure 4.10 Forecast Sequence of the front blind spot

It is obvious that in Figure 4.11, after 6 seconds, the forecast sequence is as same as the observed one; therefore, the RMSE is very small, only 0.12462, and the range of errors is from -0.8 to 0.38.

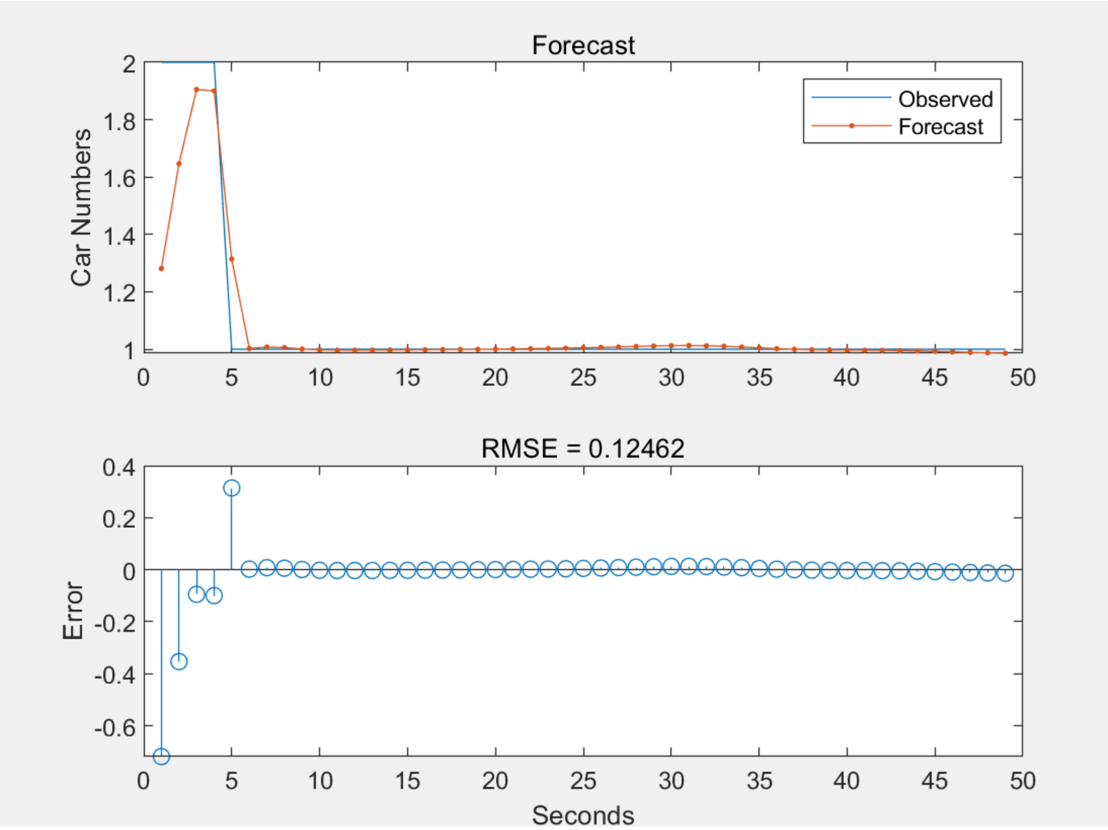


Figure 4.11 Comparisons of the forecast sequence of the front blind spot

After initializing and resetting the network state, the new forecast sequence can be obtained in Figure 4.12, and the RMSE is 0.10331, a little less than before the reset. At the same time, it can also narrow the range of errors [-0.7, 0.05].

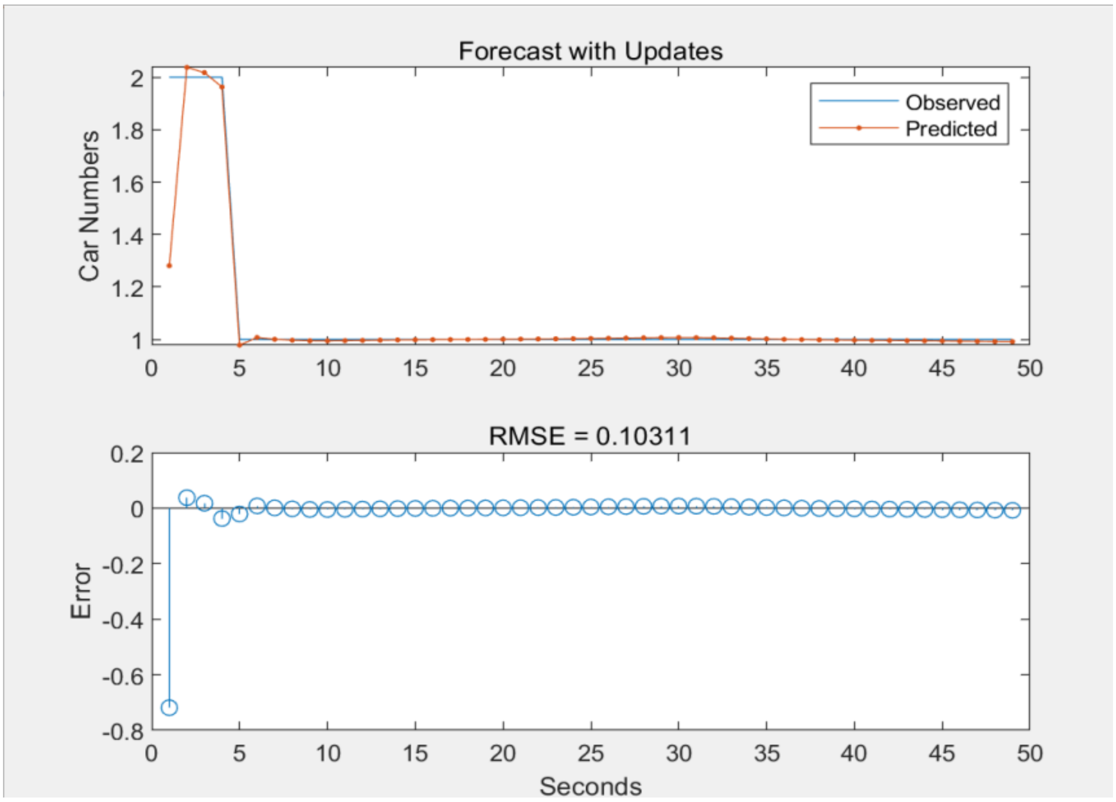


Figure 4.12 Compare the forecast sequence with updates of the front blind spot

The analysis of the results of the condition in the rear spot is shown in the following 4 figures. In Figure 4.13, it obvious fluctuations occur before the iteration number is 80; after that, it steadily decreases. The RMSE ends at 0.39 and the error ends at 0.08.

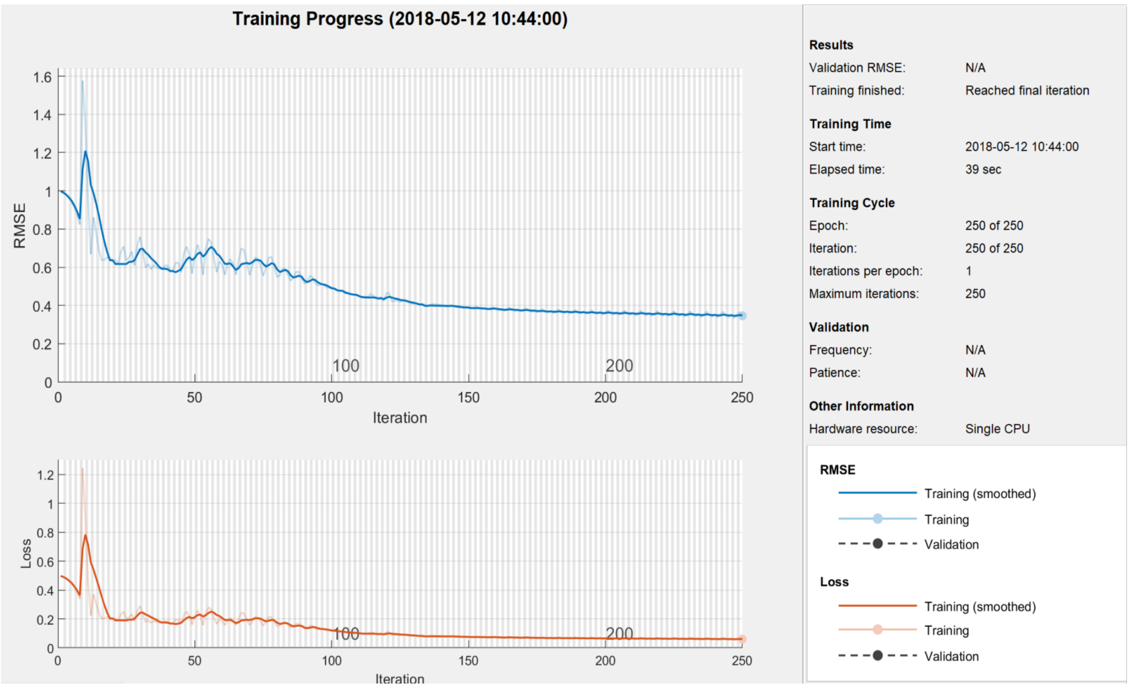


Figure 4.13 Training Process of the behind blind spot

Figure 4.14 illustrates that the numbers of cars in the behind spot are between 1 and 2 according to the observed information, and it also predicts the following car numbers with a red mark.

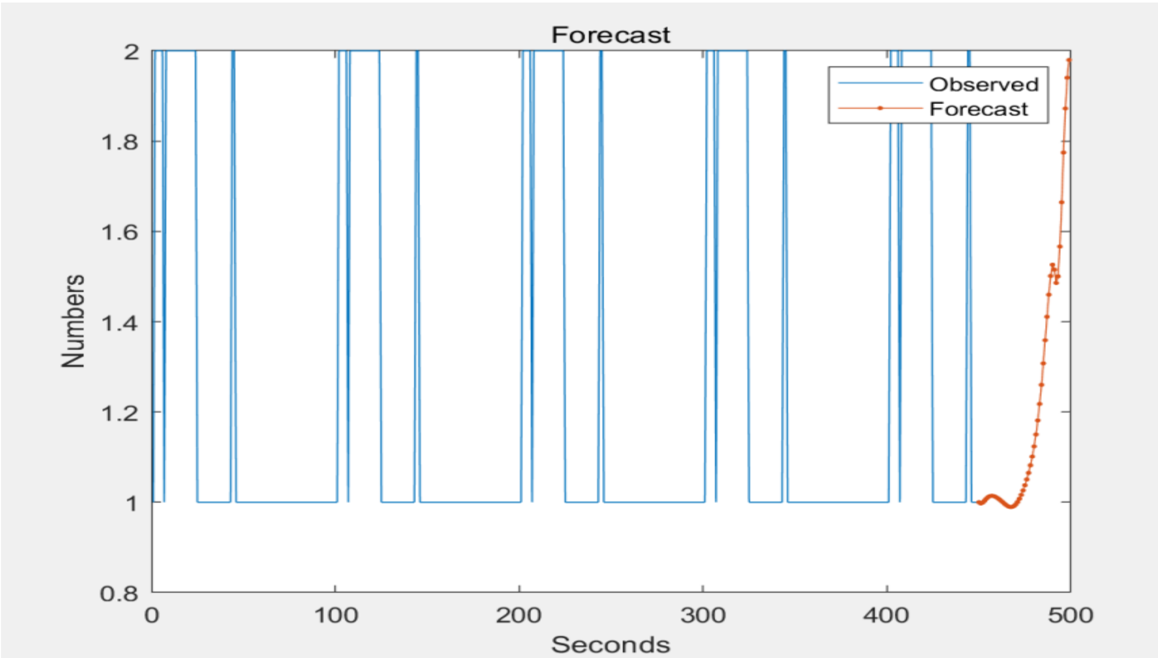


Figure 4.14 Forecast Sequence of the behind blind spot

It is easy to see that the forecast data that follows the previous sequence is different to the observed data and that the errors between them steadily increase, reaching a maximum at 1 from Figure 4.15. In addition, the RMSE calculated is up to 0.3517, which is the highest among the four blind spots.

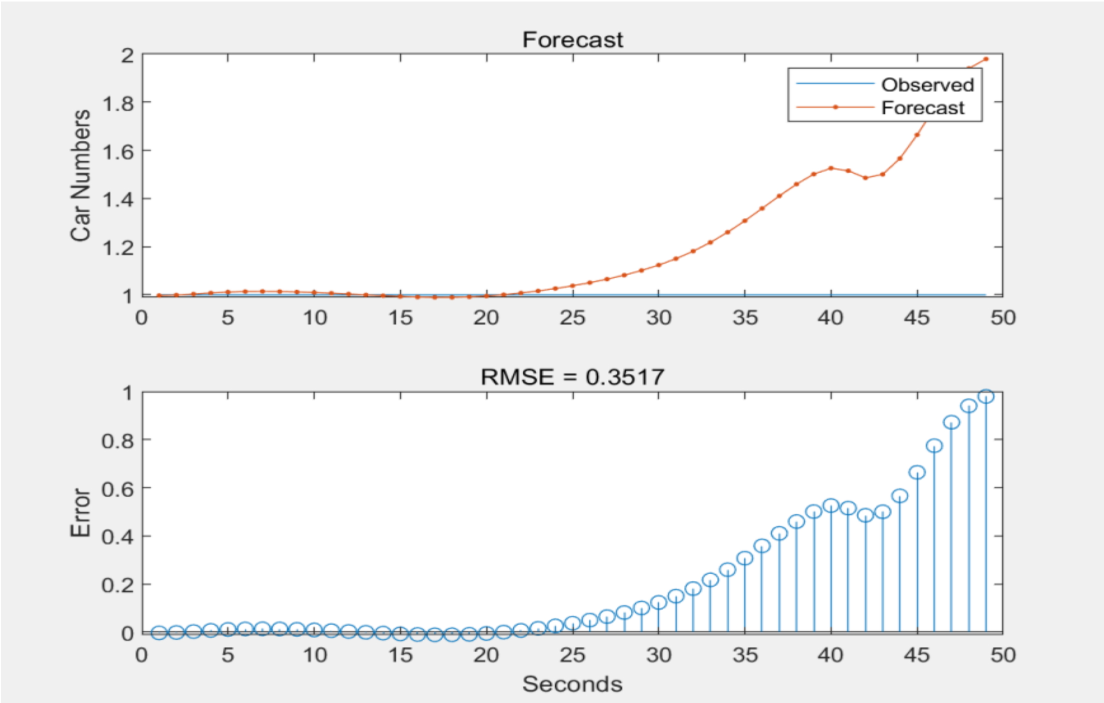


Figure 4.15 Compare the forecast sequence of the behind blind spot

To avoid the influence of the previous sequence, the network was initialized and reset state to predict the sequence again. The errors were found in Figure 4.16 to steadily decrease with the range of the errors being -0.01 to 0.06, the RSME was only 0.028362, much less than the previous one.

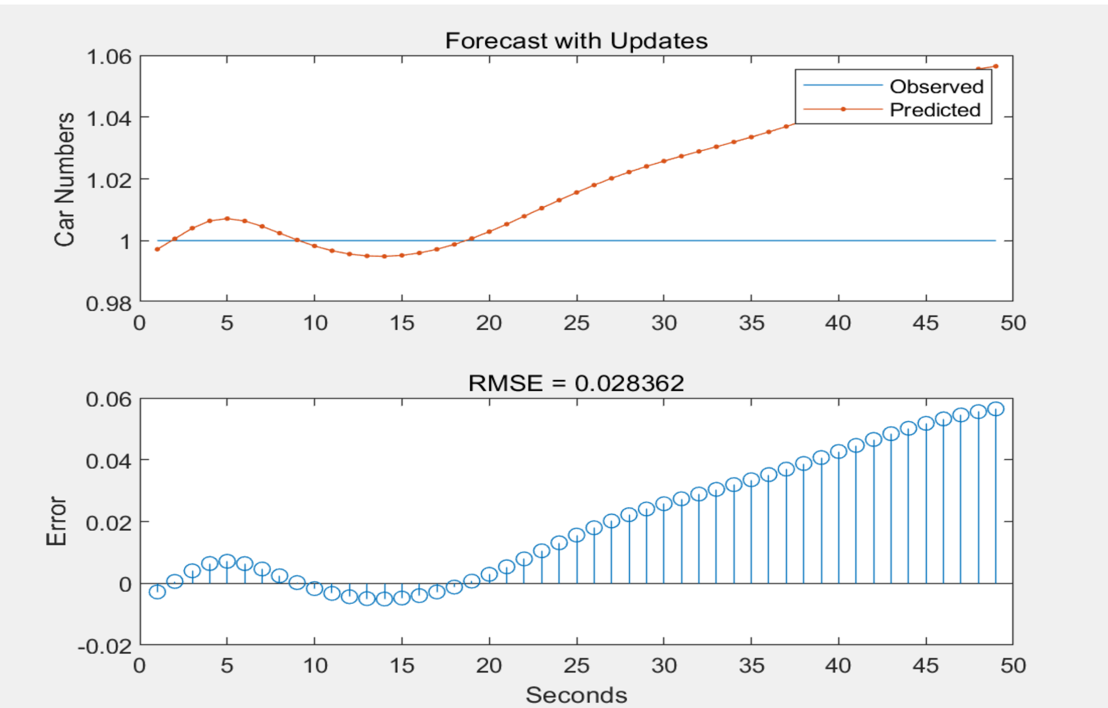


Figure 4.16 Compare the forecast sequence with updates of the behind blind spot

4.2 Limitations Of The Research

4.2.1 Limitations of Histograms

It is known that histograms are one of the most significant applications of digital images, which are effective and easy to implement. There are some limitations which are often found in histogram algorithms:

(i) The actual intensity range of the output image is difficult to reach within the maximum greyscale range allowed by the image format.

(ii) The greyscale level of the output image may be excessively merged, and the image information may be lost due to the phagocytosis of the grey scale.

(iii) Although the histogram of the greyscale distribution of the output image is evenly distributed, its value may still have a large difference from the ideal value of $1/n$, which is not an optimal value.

(iv) For some images, such as those whose histograms have peaks, the contrast is unnaturally over-stretched after processing.

(v) The greyscale level of the transformed image may be reduced because some details to disappear.

4.2.2 Limitations of SIFT

Although SIFT has many advantages including invariant feature extraction (Volckaert, et al., 2016), it still has many limitations as the follows:

(i) The SIFT algorithm runs too slowly and it is a little difficult for SIFT to implement real-time blind spot detection.

(ii) There are not enough feature points in some cases so it is hard to compare them.

(iii) It is not easy to extract the feature points for the objects with smoothened edges accurately.

Chapter 5
Analysis and Discussions

In this chapter, experimental results are analyzed and compared. Comparisons of the results under various conditions will be mentioned.

5.1 Analysis

In conclusion, HMM was used to calculate the probability of accidents under the condition that cars entered blind spots using data collected and experimental data from this project. The Viterbi algorithm and Baum-Welch algorithm were mainly employed to generate the results. RNN was used to predict the number of cars that would enter blind spots according to the previous sequence and the RMSEs of each blind spot was also calculated. The accuracy of the forecast data in the right blind spot was the highest because the RMSE value was the lowest one, only 0.013759. However, it seems that the most errors occurred in the rear blind spot as its RMSE values went up to 0.3517.

5.2 Discussion

In this report, the algorithms of histograms and SIFT were compared in order to detect whether there were any cars in blind spots. The histogram algorithm appears superior to the SIFT algorithm, as the latter runs too slowly and cannot be monitored in real time. Hence only the data from the histogram algorithm was used to calculate the probability of accidents happening when there were cars in the blind spots; this was done by using the two algorithms, Viterbi and Baum-Welch. According to the results of HMM, the probability of accidents occurring, when cars are present in blind spots, was not very high (less than 0.4), which is the average of the results obtained from the two HMM algorithms.

To predict the number of the cars in the blind spots, RNN, a model of deep learning was adopted in this report. To obtain the forecast sequence and calculate the RMSE, the range of errors at the same time, two methods were used and compared. The first was to predict the following sequence by using the previous sequence and the other was to reset the network state and estimate the sequence from new.

Chapter 6
Conclusion and Future Work

In this chapter, we will make a summary about our subject and method of this project, and put forward the future research direction according to the results and limitations of the experiment.

6.1 Conclusion

This report introduces a method to detect and alert a driver when cars appear in their blind spots, using videos from cameras installed in the car. In this report, two algorithms based on histogram were compared to detect the conditions of the blind spots. HMM was then used to obtain the maximum probability of this state occurring. After the data was collected, RNN was used to analyze the results, predict future sequences and calculate the RMSEs.

The comparison between the Histogram and SIFT algorithm found that the former had the most advantages due to its high calculation speed. It would also allow to monitor blind spots, while the SIFT algorithm could not support this due to its slow speed. Finally, the analysis results of RNN showed that the RMSE is very low, ranging from 0.013759 to 0.3517, which means the accuracy of this blind spot detection system is very high.

6.2 Future Work

Although this report proposed a BSWS and described an algorithm to reduce vehicle accidents, this blind spot monitoring system still has many aspects that required improvement. For further work, the most necessary and important tasks are as follows:

(1) To improve the accuracy and the speed of this BSWS, it is better to use a tracking algorithm based on video analysis, which can provide the driver with enough time to take any actions to avoid an accident (Chen and Chen, 2009).

(2) It is easy for this BSDS to ignore obstacles located in the edge of the image, creating false results.

(3) It is quite significant for developers to pay attention to embedded algorithm performance optimization.

(4) SIFT is a functional algorithm, but in this experiment, its calculation speeds were too slow. It is worthwhile considering how to accelerate feature detection using

SIFT in order to provide more accurate feature matching whilst reducing the number of outliers (Alhwarin et al., 2008). In other words, one should aim to improve the speed of SIFT.

References

Alhwarin, F.,Wang, C., Risti-Durrant, D., & Gräser, A. (2008). Improved SIFT-features matching for object recognition. *International Aca Demic Conference–Visions of Computer Science,* pp. 179–190.

Alonso, J. D., Vidal, E. R., Rotter, A., & Muhlenberg, M. (2008). Lane-change decision aid system based on motion-driven vehicle tracking. *IEEE Transactions on Vehicular Technology*, *57*(5), 2736–2746. doi:10.1109/TVT.2008.917220

Baek, J., Lee, E., Park, M., & Seo, D. (2015). Mono-camera based side vehicle detection for blind spot detection systems. 2015 Seventh International Conference On Ubiquitous & Future Networks, 147-149. doi:10.1109/ICUFN.2015.7182522

Baum–Welch algorithm. (2018). Retrieved from https://en.wikipedia.org/wiki/Baum%E2%80%93Welch_algorithm

Baum, L.E., & Petrie, T. (1966). Statistical Inference for Probabilistic Functions of Finite State Markov Chains. *Ann. Math. Statist*, *37*(6), 1554-1563. doi:10.1214/aoms/1177699147

Bertozzi, M., Castangia, L., Cattani, S., Prioletti, A., & Versari, P. (2015). 360° Detection and tracking algorithm of both pedestrian and vehicle using fisheye images. *2015 IEEE Intelligent Vehicles Symposium (IV)*, 132-137. doi:10.1109/IVS.2015.7225675

Blanc, N., Steux, B., & Hinz, T. (2007). Larasidecam: A fast and robust vision-based blindspot detection system. *2007 IEEE intelligent vehicles symposium*, 480 - 485. doi:10.1109/IVS.2007.4290161

Chen, C.T., & Chen, Y.S. (2009). Real-time approaching vehicle detection in blind-spot area. *2009 12th International IEEE Conference on Intelligence Transport System*, 1. doi:10.1109/ITSC.2009.5309876

Cheng, Y. K., Wong, K. H., Tao, C. H., Tam, C. N., Tam, Y. Y., & Tsang, C. N. (2016). Front blind spot crashes in Hong Kong. *Forensic Science Internationa*l, *266*, 102-108. doi:10.1016/j.forsciint.2016.05.013

Dalal, N., & Triggs, B. (2005). Histograms of Oriented Gradients for Human Detection. *2005 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'05)*, *2*, 886-893. doi:10.1109/CVPR.2005.177

Dooley, D., McGinley, B., Hughes, C., Kilmartin, L., Jones, E., & Glavin, M. (2016). A blind-zone detection method using a rear-mounted fisheye camera with combination of vehicle detection methods. *IEEE Transactions On Intelligent Transportation Systems,* 17(1), 264-278. doi:10.1109/TITS.2015.2467357

En.wikipedia.org. (2018). Hidden Markov model. [online] Available at: https://en.wikipedia.org/wiki/Hidden_Markov_model [Accessed 20 Apr. 2018].

Fernández, C., Llorca, D., Sotelo, M., Daza, I., Hellín, A., & Álvarez, S. (2013). Real-time vision-based blind spot warning system: Experiments with motorcycles in daytime/nighttime conditions. *International Journal of Automotive Technology*, *14*(1), 113-122. doi:10.1007/s12239-013-0013-3

Gonzalez, R. C., & Woods, R. E.(2010). Digital image processing. *Upper Saddle River, N.J. : Prentice Hall, [2008].*

Graves, A. (2012). Supervised Sequence Labelling with Recurrent Neural Networks. *Textbook, Studies in Computational Intelligence, Springer.*

Hane, C., Sattler, T., Pollefeys, M., Heng, L., Lee, G. H., Fraundorfer, F., & Furgale, P. (2017). 3D visual perception for self-driving cars using a multi-camera system: Calibration, mapping, localization, and obstacle detection. *Image And Vision Computing*, *68*,14-27.

He, Y., Deng, G., Wang, Y., Wei, L., Yang, J., Li, X., & Zhang, Y. (2018). Full length article: Optimization of SIFT algorithm for fast-image feature extraction in line-scanning ophthalmoscope. *International Journal for Light and Electron Optics*, *152*, 21-28. doi:10.1016/j.ijleo.2017.09.075

Hidden Markov Models (HMM)- MATLAB & Simulink- MathWorks Australia. (2018). Retrieved from https://au.mathworks.com/help/stats/hidden-markov-models-hmm.html

Hochreiter, S., & Schmidhuber, J. (1997). Long Short-Term Memory. *Neural Computation,* 9(8), pp. 1735-1780.

Huang, F., Huang, S., Ker, J., & Chen, Y. (2012). High-Performance SIFT Hardware Accelerator for Real-Time Image Feature Extraction. *IEEE Transactions On Circuits And Systems For Video Technology, 22*(3), 340-351.

Jia, X., Hu, Z., & Guan, H. (2011) A new multi-sensor platform for adaptive driving assistance system (ADAS). *2011 9th World Congress on Intelligent Control and Automation*, 1224. doi:10.1109/WCICA.2011.5970711

Jung, H.G., Cho, Y.H., & Kim,J. (2010). Integrated side or rear safety system. *Int. J. Automot. Technol*, *11* (4), pp. 541–553.

Kalshaonkar, R., & Kuwelkar, S. (2017). Design of an accurate pedestrian detection system using modified HOG and LSVM. *Proceeding - IEEE International Conference On Computing, Communication And Automation*, 957-962. doi:10.1109/CCAA.2017.8229945

Kim, D., Choi, J., Yoo, H., Yang, U., & Sohn, K. (2015). Rear obstacle detection system with fisheye stereo camera using HCT. *Expert Systems With Applications, 42,* 6295-6305. doi:10.1016/j.eswa.2015.04.035

Klotz, M., & Rohling, H. (2000). 24 GHZ radar sensors for automotive applications. *13th International Conference on Microwaves, Radar and Wireless Communications,* 359. doi:10.1109/MIKON.2000.913944

Lalonde, M., Byrns, D., Gagnon, L., Teasdale, N., & Laurendeau, D. (2007). Real-time eye blink detection with GPU-based SIFT tracking. *Proceedings - Fourth Canadian Conference On Computer And Robot Vision, CRV 2007,* 481-487. doi:10.1109/CRV.2007.54

Li, K., Zhang, Y., & Zhang, Y. (2014). The Research of Histogram Enhancement Technique Based on Matlab Software. *Sensors & Transducers, Vol 177, Iss 8, Pp 171-176 (2014)*, (8), 171.

Lin, B., Lin, Y., Fu, L., Hsiao, P., Chuang, L., Huang, S., & Lo, M. (2012). Integrating Appearance and Edge Features for Sedan Vehicle Detection in the Blind-Spot Area. *IEEE Transactions On Intelligent Transportation Systems, 13*(2), 737-747. doi:10.1109/TITS.2011.2182649

Liu, G., Zhou, M., Wang, L., Wang, H., & Guo, X. (2017). Original research article: A blind spot detection and warning system based on millimeter wave radar for driver assistance. *International Journal for Light and Electron Optics*, *135,*353-365. doi:10.1016/j.ijleo.2017.01.058

Lowe, D.G. (1999). Object Recognition from Local Scale-Invariant Features. ICCV 99.

Lowe, D.G. (2004). Distinctive Image Features from Scale-Invariant Key-points. *International Journal of Computer Vision*, 60(2), 91-110.

Mahapatra, R. P., Kumar, K. V., Khurana, G., & Mahajan, R. (2008). Ultra sonic sensor based blind spot accident prevention system. *Proceedings - 2008 International Conference On Advanced Computer Theory And Engineering, ICACTE 2008*, 992-995. doi:10.1109/ICACTE.2008.165

Martens, J., &Sutskever, I. (2011). Learning Recurrent Neural Networks with Hessian-Free Optimization. *University of Toronto, Canda*.

McCarthy, M., & Gilbert, K. (1996). Cyclist road deaths in London 1985-1992: Drivers, vehicles, manoeuvres and injuries. *Accident Analysis And Prevention, 28*(2), 275-279. doi:10.1016/0001-4575(95)00061-5

Milos, S., & Jan, L. (2012). The new approach of evaluating differential signal of airborne FMCW radar-altimeter. *Aerospace Science And, 17*(1), 1–6.

Miśkiewicz, J. (2016). Improving quality of sample entropy estimation for continuous distribution probability functions. *Physica A: Statistical Mechanics and Its Applications, 450*, 473-485. doi:10.1016/j.physa.2015.12.106

Park, H., & Yoo, C. (2017). Melody extraction and detection through LSTM-RNN with harmonic sum loss. *2017 IEEE International Conference on Acoustics, Speech, and Signal Processing, ICASSP 2017 - Proceedings*, 2766-2770. doi:10.1109/ICASSP.2017.7952660

Ra, M., Jung, H. G., Suhr, J. K., & Kim, W. (2018). Part-based vehicle detection in side-rectilinear images for blind-spot detection. *Expert Systems with Applications*, *101*, 116-128.

Ray, S. J., & Teizer, J. (2013). Computing 3D blind spots of construction equipment: Implementation and evaluation of an automated measurement and visualization method utilizing range point cloud data. *Automation in Construction*, *36*, 95-107. doi:10.1016/j.autcon.2013.08.007

Robert, J. Aumann. (1974). Subjectivity and correlation in randomized strategies. *Journal of Mathematical Economics*, *1*(1), 67-96.

Rumelhart, D.E., Hinton, G.E., & Williams, R.J. (1986). Learning representations by back-propagating errors. *Nature*, *323*(6088), 533–536. doi:10.1038/323533a0

Shannon, C.E. (1948). A mathematical theory of communication. *Bell Syst. Tech.* 27, 379-423.

Sonka, M., Hlavac,V., & Blyle,R. (2003). Image processing, analysis, and machine vision. *People's Posts and Telecommunications Press, Beijing*.

Sorkun, M., Paoli, C., & Incel, Ö. (2017). Time series forecasting on solar irradiation using deep learning. *2017 10Th International Conference On Electrical And Electronics Engineering*, *ELECO 2017,* 151-155.

Sotelo, M. A., & Barriga, J. (2008). Blind spot detection using vision for automotive applications. *J. Zheijiang University Science A, 9*(10), 1369–1372.

Toselli, A. H., Vidal, E., Romero, V., & Frinken, V. (2016). HMM word graph based keyword spotting in handwritten document images. *Information Sciences*, *370-371*, 497-518. doi:10.1016/j.ins.2016.07.063

Tseng, D., Hsu, C., & Chen, W. (2014). Blind-Spot Vehicle Detection Using Motion and Static Features. *International Journal of Machine Learning and Computing*, *4*(6), 516-521.

Twisk, D., Vlakveld, W., Mesken, J., Shope, J.T., & Kok, G. (2013). Inexperience and risky decisions of young adolescents, as pedestrians and cyclists, in interactions with lorries, and the effects of competency versus awareness education. *Accident Analysis And Prevention, 55,* 219-225. doi:10.1016/j.aap.2013.02.038

Volckaert, D., Heynderickx, P. M., Fathi, E., & Van Langenhove, H. (2016). SIFT-MS a novel tool for monitoring and evaluating a biofilter performance. *Chemical Engineering Journal*, *304*, 98-105. doi:10.1016/j.cej.2016.04.138

Wang, J., & Zhang, C. (2018). Software reliability prediction using a deep learning model based on the RNN encoder–decoder. *Reliability Engineering and System Safety, 170,* 73-82. doi:10.1016/j.ress.2017.10.019

Wang, M.S., Song, L., Yang, X.K., & Luo, C.F. (2016). A parallel-fusion RNN-LSTM architecture for image caption generation. *International Conference on Image Processing, ICIP, 2016-August*. 4448-4452. doi:10.1109/ICIP.2016.7533201

Wang, Y., Ban, X.J., Chen, J., Hu, B., & Yang, X. (2015). License plate recognition based on SIFT feature. *International Journal For Light And Electron Optics, 126*, 2895-2901. doi:10.1016/j.ijleo.2015.07.040

Wei, Y., Liu, Z.J., He, T., Qing, G.R., & Zhang, D.F. (2018). Content Based Image Retrieval Method Based on SIFT Feature. *2018 International Conference on Intelligent Transportation, Big Data & Smart City (ICITBS),* 649. doi:10.1109/ICITBS.2018.00169

Wong, C.Y., & Qidwai, U. (2005). Intelligent surround sensing using fuzzy inference system. *Proceedings of the Fourth IEEE Conference on Sensors 2005,* 1034-1037. doi:10.1109/ICSENS.2005.1597879

Wu, B.F., Huang, H.Y., Chen, C.J., Chen, Y.H., Chang, C.W., & Chen, Y.L. (2013). A vision-based blind spot warning system for daytime and nighttime driver assistance. *Computers And Electrical Engineering, 39(Special issue on Image and Video Processing)*, 846-862. doi:10.1016/j.compeleceng.2013.03.020

Yang, F., Balakrishnan, S., & Wainwright, M. J. (2017). Statistical and Computational Guarantees for the Baum-Welch Algorithm. *Journal of Machine Learning Research, 18*(124/149), 1-53.

Youying, Z., & Tadahiro, K. (2018). New idea for SIFT to evaluate the matching result. *2017 4Th International Conference On Image Information Processing*, 548-551. doi:10.1109/ICIIP.2017.8313777

Yusuf, S. A., Brown, D. J., & Mackinnon, A. (2015). Application of acoustic directional data for audio event recognition via HMM/CRF in perimeter surveillance systems. *Robotics And Autonomous Systems*, *72*, 15-28. doi:10.1016/j.robot.2015.04.004

Žemgulys, J., Raudonis, V., Maskeliūnas, R., & Damaševičius, R. (2018). Recognition of basketball referee signals from videos using Histogram of Oriented Gradients (HOG) and Support Vector Machine (SVM). *Procedia Computer Science, 130*, 953. doi:10.1016/j.procs.2018.04.095

Zhang, R.S., Liu, J.C., & Ma, L. (2015). A Typical Blind Spot Danger Pre-warning Method of Heavy Truck under Turning Right Condition. *2015 6th International Conference on Intelligent Systems Design and Engineering Applications, ISDEA 2015*, 93-96. doi:10.1109/ISDEA.2015.33