TRAFFIC SIGN RECOGNITION SYSTEM BASED ON CAPSULE NETWORK

A CHPROJECT REPORT SUBMITTED TO THE AUCKLAND UNIVERSITY OF TECHNOLOGY
IN PARTIAL FULFILMENT OF THE REQUIREMENTS FOR THE DEGREE OF
M ASTER OF COMPUTER AND INFORMATION SCIENCES

Wenyang Hao

School of Engineering, Computer & Mathematical Sciences

Contents

List of Figures

List of Tables

Attestation of Authorship

I hereby declare that this submission is my own work and that, to the best of my knowledge and belief, it contains no material previously published or written by another person nor material which to a substantial extent has been accepted for the qualification of any other degree or diploma of a university or other institution of higher learning.

Signature of candidate Date: 27 May 2022

Acknowledgement

Firstly, I would like to thank my supervisor Wei Qi Yan. He helped me learn relevant knowledge, provided technical support and solved project problems, he has regularly checked the progress of this project, supervised the completion of this report till completion. Furthermore, I would like to thank to the AUT for providing good hardware support.

Abstract

The rapid development of computer vision technology helps us realize intelligent life. It is also becoming more and more widely used. Self-driving technology can help us drive cars with ease and help us anticipate changes on the road. Traffic sign recognition is an important part of self-driving technology. In this report, a model different from the traditional convolutional neural network is applied to recognize traffic signs, which is capsule neural network. Capsule neural network is a neural network which can be used for azimuth detection. Experimental results show that this method has higher recognition accuracy than CNN.

Chapter 1

1. Introduction

1.1 The Overview and Background of Research

With the improvement of living conditions, people's demands for cars are increasing. China's car ownership has grown rapidly from 180 million in 2017 to 260 million in 2021. The effect of this is a decrease in traffic efficiency and an increase in accident rates. In order to improve traffic efficiency and reduce accident rate, more and more attention has been paid to autonomous driving technology. Automatic driving technology is the car according to the driver's command automatically overtaking, lane change, turning and other operations. Traffic sign recognition as a basic technology of automatic driving is the focus of research in recent years. Correspondingly, 70% of the traffic accidents occurred were caused by the inability to clearly identify traffic signs or traffic lights due to weather reason. When we have bad weather or low concentration. We can't distinguish between traffic signs and traffic lights on the road. This leads to our inability to obtain road information in time and may cause hidden dangers to driving safety. Therefore, traffic sign recognition is of great significance.

Traffic sign recognition is also an important part of self-driving technology. The basis for self-driving is that the computer needs to understand speed limit signs, no traffic signs, warning signs and lights. This is a prerequisite for safety. Traffic sign recognition can also be extended to pedestrians and vehicles. The traffic sign recognition system consists of two parts: traffic sign detection and traffic sign recognition. Traffic sign detection determines whether the detected object is a traffic sign. Currently, typical detection methods can be divided into four categories: traffic sign detection based on color, traffic sign detection based on shape, traffic sign detection based on multi-feature fusion and traffic sign detection based on deep learning. Traffic sign recognition is to classify the detected traffic signs. At present, the mainstream traffic sign recognition methods are based on module matching method, machine learning method and deep learning method.

The traffic sign recognition based on module matching is relatively simple. Edge detection was carried out on the obtained image (Jia et al., 2020) color segmentation was conducted to obtain the target image. Finally, the obtained traffic sign image is compared and matched with the image in the template library to obtain the result. Although the method based on module matching is easy to implement, the result depends on the template library and the accuracy is not high. And the update of template library is also a tedious work.

Traffic sign recognition based on machine learning and deep learning is a conventional method. Min-max technique and Z-Score technique are included in SVM prediction model, and k-nearest neighbor algorithm is used for comparison. They were evaluated on the GTSRD dataset. The results show that SVM has higher performance (Margae et al., 2014).

In this project report, the capsule network method is used to identify traffic signs. Capsule network is a new neural network, whose neurons are composed of vectors, the length of which represents the probability of entity existence, and the method represents the instantiation parameters (Sabour et al., 2017). This capsule network has better performance than CNN in recognizing traffic signs. The capsule network can achieve good performance even when the training set is small. The experimental results show that the capsule network training of all samples can achieve up to 99.59% accuracy (Xu, 2021). Even if only 20% of the training samples were used, the accuracy rate could reach 89.92%.

1.2 Research Methodology

The purpose of this study is to use capsule network to realize traffic sign recognition. The first step is to understand the background of traffic sign recognition and to know what technologies are needed to realize the recognition of traffic signs by consulting the data. Through consulting relevant literature, it is found that the current traffic sign recognition model cannot accurately identify traffic signs under special conditions. Therefore, the purpose of this study is to explore a model that can improve the accuracy of traffic sign recognition.

The second step is to review relevant articles and find that these problems can be solved using only the capsule network. The third step is to obtain the training set and test set. TT100K was obtained through Internet search as the experimental dataset. The fourth step is to preprocess the data set, such as binary processing, erosion and dilation processing and edge detection. The fifth step is to build a capsule network model. The sixth step is to train the model and evaluate the results. Finally, the results are analyzed, and the deficiencies are put forward.

1.3 Research Objectives and Major Motivations

Most of the studies on traffic sign recognition have limitations and are not applicable to all use scenarios. The details are listed as follows:

(1) Different countries or regions have different traffic signs. In this case, the trained model is not suitable for all scenarios.

(2) If traffic signs overlap or are covered, they will be unrecognizable. Traffic signs on the road will encounter occlusion and other situations, in this case, the accuracy of recognition is very low.

This study aims to solve the problems are listed as below:

(1) We use different data sets for training in different countries or regions. Because of the different traffic regulations, the traffic signs in each country cannot be unified. Collect data in the country where you do your research.

(2) we harness capsule neural network to recognize traffic signs. Capsule neural network can solve overlapping and occlusion problems well.

1.4 Thesis Organization

The rest of the paper is as follows:

- The second chapter will introduce various types of neural network and capsule network technology, and reviews various types of neural network and capsule network related literature. The second chapter also analyzes the methods in the literature and points out the shortcomings

- The third chapter will introduce the methods used in the research. The methods of data preprocessing, model evaluation and parameter adjustment are introduced in detail. The third chapter also points out the limitations of the research.

- The fourth chapter will analyze the experimental results. The advantages of capsule network and the effect of parameter adjustment on model performance are detailed.

- The fifth chapter will conclude this research. According to the experimental results, the usability of the methods used in the study is concluded. The existing problems in the research and the future work are mentioned in this chapter.

Chapter 2

Literature Review

2.1. Introduction

The rapid development of computer vision technology provides a strong support for the realization of traffic sign recognition. Firstly, this chapter reviews the literature on traffic sign recognition based on deep learning and capsule network. Section 2.2 introduces deep learning techniques. Section 2.3 mainly explains the capsule network technology used in this study. Finally, the limitations of the above literature are summarized.

A new traffic sign recognition method based on convolutional neural network was proposed (Yildiz & Dizdaroglu, 2021). In that article, the input image adopts RGB, CIELab, RIQ, LGI four color space to process, and modify the size of the input image. Then, a CNN model containing 10 convolutional layers, 3 pooling layers and 3 fully connected hidden layers is used for experiments. The results show that the low parameter CNN model achieves high recognition accuracy. However, the maximum size of the input image in this study is 60×60, so it may not be able to guarantee high accuracy if the image is replaced with a large size.

The performance of five target detection algorithms including Faster R-CNN, SSD, RetinaNet, YOLOv3 and YOLOv5 were compared. The data set in this study is the China Traffic Sign Detection Data Set (CCTSDB). Research shows that three algorithms, namely Faster R-CNN, YOLOv3 and YOLOv5, have higher accuracy. SSD algorithm training speed is fast, but accuracy is low (Jiang et al., 2022). However, these target detection algorithms are not good for small targets and dark light conditions. And the CCTSDB data set only divides the data into three categories of mandatory flags, forbidden flags and warning flags, which is too imprecise.

A real-time traffic sign recognition system (Xu, et al. 2018) was proposed based on convolutional neural network. The dataset in the study was taken with mobile phones and includes videos and photographs. The data were manually marked and put into the constructed AlexNet network to train the model. A GUI interface was designed to display detection time, frame number and other information. Finally, the traffic signs can be recognized. But the research results show that the accuracy is not very high, there are many traffic signs cannot be recognized. It is also time consuming to manually mark all training data.

The algorithm in the study is an improvement of Lenet-5 CNN model, which can gradually and accurately perceive and recognize targets and improve the recognition rate. Gabor's kernel is used in the network, and the normal convolution kernel is used after the pooling layer. German Traffic Sign Identification Benchmark (GTSRB) was used as the data set. The research results show that the recognition accuracy is up to 99% and the efficiency is higher (Prakash et al., 2021).

In the recognition of traffic signs, the traditional convolutional neural network will be affected in complex environment. An improved capsule network (Zhang, 2020) is proposed to solve this problem. It is pointed out that the capsule network can process the physical posture and deformation of the image to improve the accuracy and robustness of recognition. Traffic signs can be correctly recognized even in rainy, snowy and foggy days. The results show that the improved capsule network is more accurate and takes less time than traditional CapsNet and AlexNet.

An improved network is proposed based on lightweight convolutional neural network (Song, 2021). This network model introduces deeply separable convolution and uses Mish activation functions to speed up training. The size and number of convolution kernels can be changed by upgrading the network architecture level. Through the study, it is found that the improved network has higher training speed and accuracy than the network without improvement. However, currently the commonly used traffic sign recognition models are Fast R-CNN or YOLO family, and the research outcomes are only compared with CNN without modification. So there is no way to know the practicability of this improved model.

In this study, YOLOv4 was used to recognize and detect traffic signs. YOLOv4 integrates a deep learning algorithm for target detection. The data is divided into training data and test data. The training data is used to train the model, and the test data is used to test the performance. Throughout the research project, we see that YOLOv4 is suitable for most traffic sign recognition and achieves 95.15% accuracy (Arief et al., 2021). However, YOLOv4 was proposed by the study that cannot solve the problem of occlusion and environmental factors. This needs to be improved.

2.2 Deep Learning

Deep learning is a model that simulates the structure of the human brain (Ren et al., 2022). The aim of this model is to give machines the ability to learn and think like humans. Most of the existing target feature recognition technologies are based on deep learning (Zheng, 2021). Deep learning algorithms include artificial neural network (ANN), convolutional neural network (CNN), recurrent neural network (RNN), attentional mechanism (ATT) and Graph neural network (GNN). Deep neural networks can be used to predict time series events. For example, it is employed to forecast stock prices (Kohara et al., 1997), forecast bitcoin price trends(Apoorva et al., 2019), forecast fuel prices(Lahari et al., 2018).

ANN is the most basic algorithm in deep learning. Artificial neural network technology can extract specific patterns hidden in input data. Because of this performance, artificial neural networks are used in various measurement modes (Jeon et al., 2019).

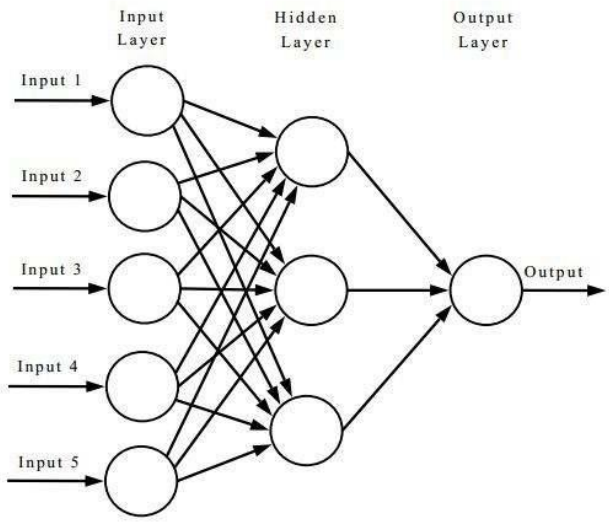

Figure 0.1: The architecture of neural networks

$$y = Wx + b \tag{1}$$

As shown in Figure 2.1, ANN contains an input layer, a hidden layer, and an output layer. There are full links between the layers. The principle of ANN can be understood using linear regression. Eq. (1) shows a linear regression equation with W as the parameter matrix and b as the bias vector. Each layer in the neural network maintains a value of W and b. The model first obtains the predicted value through the forward process of each layer of the sample, and then calculates the error between the predicted value and the real value. Finally, the error is updated W and b of each layer through the Backpropagation algorithm to better fit the sample distribution of the dataset.

The application of ANN is extensive. ANN can be applied to the classification of materials (Vinod et al., 2021) and the recognition of handwritten English numbers (Rath et al., 2021), which can replace the human recognition and classification problems.

Due to ANN has problems of full connection and too many network parameters, an optimized network is proposed, which is CNN. CNN divides the parameter matrix into convolution layer, pooling layer and fully connected layer. Convolution layer is applied to extract local features in the sample. The pooling layer is used to extract the results of the convolution layer. This reduces the number of parameters and prevents overfitting. The full-connection layer splices the results of multiple convolution layers and finally obtains a global feature for classification. Figure 2.2 shows the architecture diagram of the CNN model.

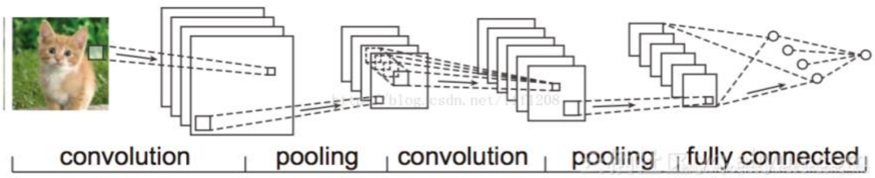

Figure 0.2 Diagram for convolutional neural networks

An improved CNN model (Yahya et al., 2021) was proposed to classify handwritten numbers and achieved good performance. Different from other deep learning models. CNN has built-in invariance, which is especially suitable for image classification tasks (Cheung, 2012).

CNN can be widely used in the field of image recognition. A learning model based on CNN (Hammad & Khotanlou, 2022) was proposed for detecting COVID-19 from chest X-ray. Saranya et al. (2022) introduced a target detection model based on CNN for self-driving. This model can implement lane detection and object detection.

ANN does not consider the temporal relationship between samples, which leads to the failure to capture temporal information in the data with temporal information. Timing information is needed in natural language processing. Zhang et al. (2020) takes advantage of Chinese texts with part of speech as the research object and proposed a part-of-speech synthesis method based on word order rule priority for natural language processing. Therefore, an improved neural network -RNN is proposed. RNN can capture timing information in the data. Unlike common neural networks, neurons in the hidden layer of RNN contain recursive connections(Liang, 2021). In recent years, RNN has achieved great success in language modeling and syntactic parsing.

In RNN, due to the characteristics of RNN, Lieskovska et al. (2022) proposed a time modeling method for RNN to carry out speech emotion recognition. RNN can also be applied to music generation(Sajad et al., 2021).

ATT belongs to the category of computer vision. With the development of deep learning, ATT is being applied to natural language processing to extract important information from statements. The attention mechanism improves the accuracy of the model by giving more weight to key words (Azhar et al., 2021).

GNN uses deep learning to learn directly from graph structure data. Graph neural network is an effective framework for graph structure representation learning. Its main application scenarios are divided into node classification and graph structure classification (Zhou et al., 2022). The essence of GNN is to aggregate the features of adjacent vertices into the central vertex. Then use the graph's adjacency to learn new features. Finally, these features are trained in different neural networks.

1.5 Capsule Network

CNN's inability to process the position and direction of images results in CNN's low accuracy in identifying overlapping or reversed images. The capsule network can solve this problem. Capsule neural network retains the detailed location information between entities in the image and considers the spatial hierarchical relationship between entities(Xi et al., 2017).

Capsule network consists of capsules, which are groups of neurons. The input and output of the capsule network are vectors whose length represents the probability of existence of entities. The direction of the vector contains information about the instantiation parameters(Shi et al., 2021). The fundamental difference between capsule network and traditional artificial neural network lies in the unit structure of the network. Capsule network multiplies the input vector.

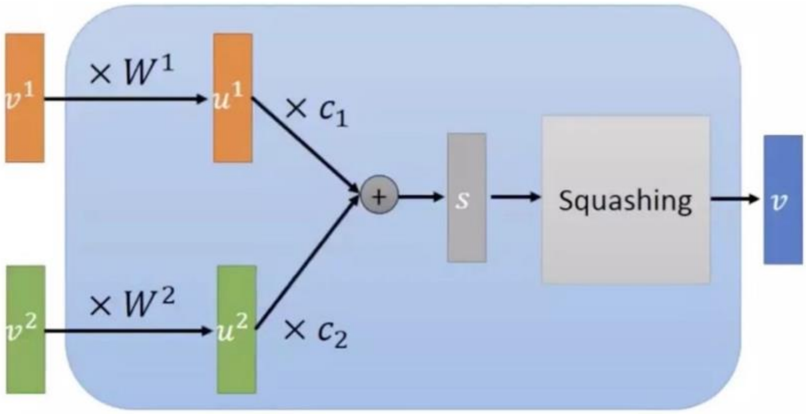

Figure 0.3 Capsule network

As shown in Figure 2.4, a Squash function is used to squash the length of vector S to 0-1 and keep the direction unchanged. The new vector v is the output of the capsule.

$$Squash(s) = \frac{||s||^2}{1+||s||^2} \times \frac{s}{||s||} \tag{2.1}$$

The low-level capsules in the capsule network can predict the activation state of high-level capsules(Liu et al., 2022). The weights of advanced and low-level capsules can be obtained by dynamic routing algorithm. The algorithms are listed as below.

Step 1: input u^i, the layer number of capsule is l, and the number of iterations is r. u^i is the product of input vector v^i and weight w^i mentioned in Figure 2.4.

Step 2: Create a set of temporary variables b^{ij} with an initial value of 0 corresponding to c^{12}. Variables are stored in c^{12} after iteration.

Step 3: set the number of iterations for iterative operation.

Step 4: Use the Softmax function to make all weights c^i non-negative and sum to 1.

Step 5: Perform the operation: $s^i = u^i c^i$

Step 6: Pass s^i into a Squash function to make sure the vector direction keeps unchanged and squash all vectors to a length between 0 and 1.

Step 7: Update vector b^{ij}.

The simple understanding is that when the input vector of the lower capsule is in the same direction as the input vector of the higher capsule, the routing parameter b^{ij} is increased. When the input vector of the lower capsule is in the opposite direction as the input vector of the higher capsule, b^{ij} is reduced. Finally, a set of routing parameter b^{ij} was obtained to judge whether low-level capsules matched with high-level capsules.

Capsule networks can be used in many fields. Capsule networks can play a huge role in medicine. Raje and Jadhav (2022) proposed a classification model consisting of convolutional layer, primary capsule layer and digital capsule layer. The model can automatically detect pneumonia. Afriyie et al. (2021) proposed a simple capsule network for medical diagnosis on complex images. Capsule network

2.3 Summary

The methods and models have some limitations, they are listed as follows:

- Limitations of training data. If the training data parameters are too small or the amount of data is too small, the model will not fit properly and can not accurately identify traffic signs.

- The model cannot guarantee correct recognition under special circumstances. Poor recognition performance or inability to recognize traffic signs in rainy, snowy or foggy weather.

- The model cannot recognize the tilt, occlusion or damage of the target object.

Chapter 3

Methodology

3.1 Introduction

The main purpose of this study was use capsule network to implement the recognition of traffic signs. The data in the study was a Chinese traffic signs dataset. The results were obtained by data preprocessing and training model. Five evaluation methods were used to evaluate the performance of the model. Finally, the model achieved the best performance by adjusting the parameters. This chapter consists of the introductory section, methodological design, data collection and analysis methods, evaluation methods, parameter adjustment and research limitations.

The methodological design contained the research methods used in the research and the whole research process. The data collection and analysis methods section introduced the sources of the datasets and the preprocessing methods used in this study. The image preprocessing method was analyzed in detail. The evaluation methods section introduced five evaluation methods used in the study. The parameter adjustment section described how to adjust the parameters and how to adjust the model for optimal performance. The limitations and solutions of this study are proposed in the research limitations section.

3.2 Methodological Design

The research results were obtained by consulting many relevant literatures and obtaining data for experiments. Firstly, we collected data from the web. Due to the collected data is raw, the data needs to be cleaned. Then the cleaned data is preprocessed to get the data which is needed by the training model. Secondly, the capsule network model was created and trained. Finally, the model parameters were adjusted and trained until the best performance is achieved.

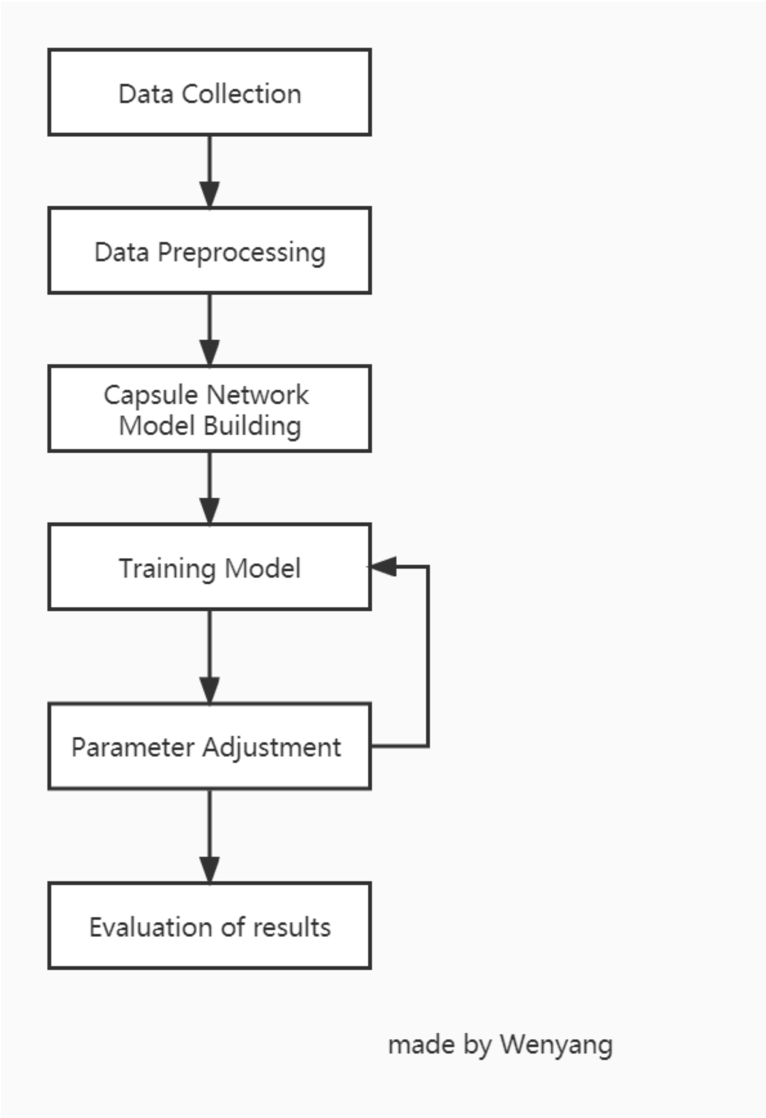

Figure 0.1 The flowchart of our experiments

3.3 Data Collection & Analysis Methods

The dataset used TT100K, a traffic sign dataset co-produced by Tencent and Tsinghua. Zhu et al. (2016) created a library of various traffic signs by collecting 100,000 traffic signs from Tencent Street View. The traffic sign dataset contains 30,000 instances of traffic signs, covering different brightness conditions and weather conditions. The traffic sign dataset contains blocked traffic signs. TT100K dataset contains three major categories. As shown in Figure 3.2, the three categories are indicating signs (i), prohibition signs (p) and warning signs (w). There are 128 subclasses under the three categories to represent the detailed information of each flag.

Figure 0.2: The number of classes and instances of the dataset

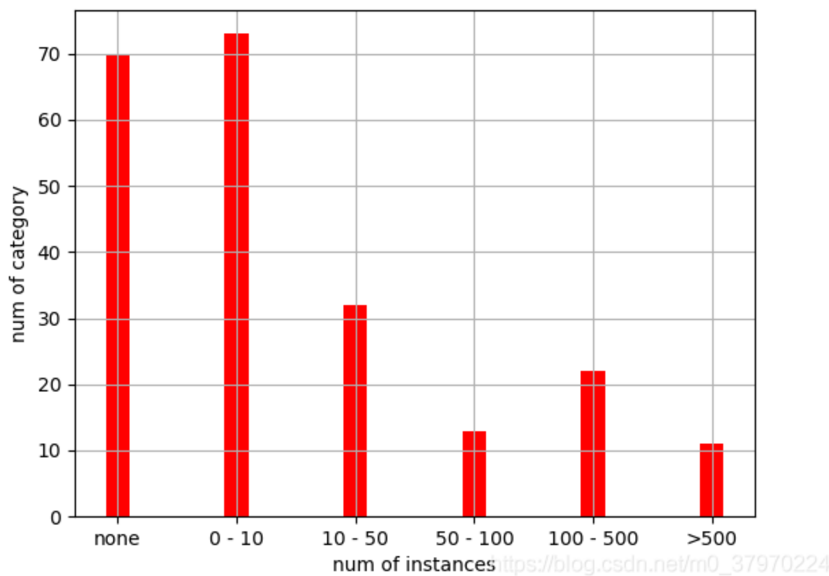

Figure 0.3: The histogram

Figure 0.4: Our samples from dataset which included conditions of different brightness, night and day.

Data preprocessing is important. For image analysis, the quality of image affects the effect of algorithm. In this experiment, binarization processing, edge detection, erosion and dilation these four methods of image preprocessing are used.

3.3.1 Image Binarization

The image is composed of a matrix, in which the RGB value of each point is different, resulting in a different color of the image. However, the 256 color values in image processing is obviously complicated. Image binarization is the process of converting a multi - tuned image into a two - tone image(Fang et al., 2021). The binarization process converts the value of each matrix point to black (0) or white (255), which increases the calculation speed. Yang and Wan (2021) proposed an image binarization based on K-means clustering to solve the height variation between document background and foreground. The binarization processes include simple binarization, adaptive binarization and OTSU binarization.

Simple binarization is to set a threshold that is constant, which is greater than this threshold is 255, less than this threshold is 0.

Adaptive binarization is the algorithm to determine the size of the threshold. The average, maximum and median values of the surrounding pixels are calculated and compared with the pixel values of the target point. The pixel value of the target point is large, it is 255. The pixel value of the target point is small, it is 0. OTSU binarization is also called maximum interclass variance method. Find a threshold that maximizes the sum of the variances of two parts of the pixel that are larger than the threshold and smaller than the threshold. The purpose is to remove as much information as possible and keep only the most basic information.

Figure 3.5 and Figure 3.6 show the results of image processing using global binarization in this experiment. There was original image, grayscale image and binary image.

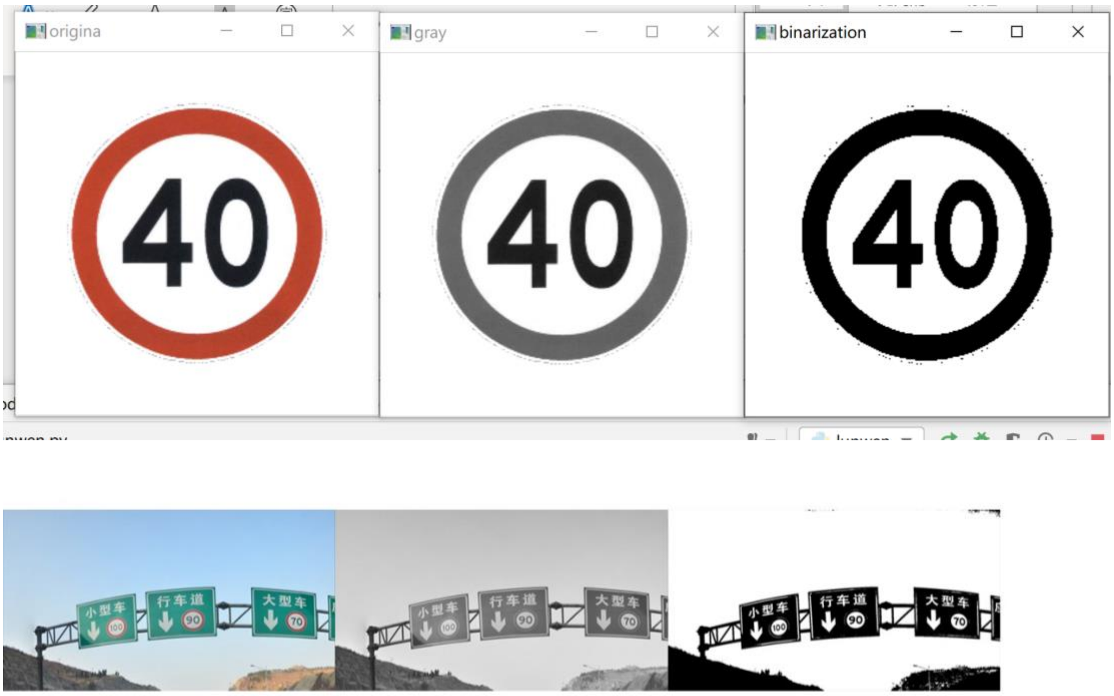

Figure 0.5: Image processing using global binarization

3.3.2 Edge Detection

Edge detection refers to the technique of obtaining edges or boundaries from images(Dhiman & Saroha, 2022). It is a basic problem in image processing. The purpose of edge detection is to mark the points with obvious brightness changes in digital images. The main strategy of edge detection is to reduce the amount of information while preserving the features of structural elements for supplementary processing(Abebe Berwo et al., 2021). There are three methods of edge detection. The first is the classical edge detection algorithm, such as differential operator, optimal operator. The second method is global extraction. The third is the image edge extraction method based on wavelet transform and mathematical morphology.

Edge detection algorithm is a filtering algorithm in essence. The basic idea of edge detection is to perform two filtering operations along x-axis and y-axis, obtain the image gradient of the current pixel point by summing the obtained results and adding the square root of the operation. The basic edge detection filter consists of Sobel operator, Prewitt operator and Roberts operator. The filter is shown in Figure3.7, Figure3.8, and Figure 3.9. Take Sobel operator as an example, where S_x and S_y represent edge detection operators for x-axis and y-axis respectively. From the structure of S_x operator, we see that this filter is to calculate the difference between the gray value of 8 connected pixels on the right and left of the current pixel point. However, the edge calculated directly by the basic edge operator has some problems, such as noise is not eliminated. Therefore, the more advanced Canny operator has become the main edge detection operator.

There are four steps for edge detection by Canny operator. Gaussian filtering is first used to remove noise. The second step is to calculate gradient image and angle image. Canny uses gaussian filter to calculate the gradient to get the filter, and the result is similar to Sobel operator. The closer the pixels are to the center point, the more weight they have. The third step is non-maximum suppression of gradient image. This will remove a large number of non-edge pixels. Finally, double threshold is used for edge detection. This removes the false edges. The edge obtained by canny operator is the optimal edge. Figure 3.6 shows the image before and after edge detection using canny operator.

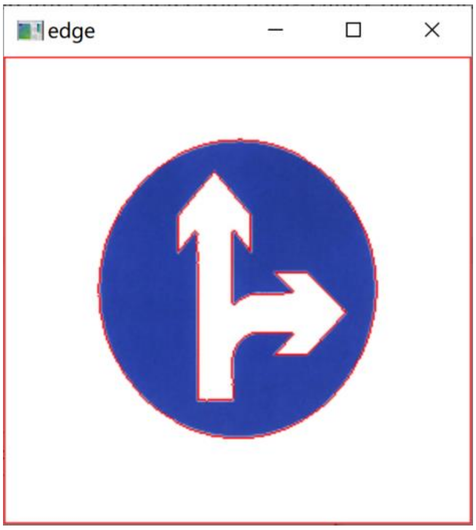

Figure 0.6: Example of edge detection using canny operator

3.3.3 Image Erosion and Dilation

Erosion and dilation are basic operators of mathematical morphology (Xinyu, 2009). Erosion is the enhancement of dark areas, mainly used to remove very bright noise points. Dilation is the enhancement of brighter areas, mainly used to connect areas of similar color or intensity. Erosion and dilation focus on the establishment of the convolution kernel. Different convolution kernels lead to different results after erosion and dilation. In general, it is a 3x3 matrix as shown in Figure 3.11.

0	1	0
1	1	1
0	1	0

Figure 0.7: An example of 3x3 example

Erosion is the removal of small gully details. From a mathematical point of view, the image is convolved with the kernel, the minimum value of the pixel point in the covered area of kernel B is calculated, and the minimum value is assigned to the element specified by the reference point. This will gradually reduce the area of highlight in the image.

The action of dilation is the opposite of erosion. Dilation is the operation of finding a local maximum. From a mathematical point of view, it is to convolve the image with the kernel, calculate the maximum value of pixels in the area covered by kernel B, and assign this maximum value to the element specified by the reference point. This will cause the highlighted areas in the image to gradually grow.

Erosion and dilation can eliminate noises and unnecessary noise points. It can also split up individual image regions and connect adjacent elements in the image. Erosion and dilation can also be used to find regions of maximum and minimum values in the image and to find the gradient of the image. Figure 3.12(a) was an unprocessed image. There were many interferences information in the image, which is not conducive to model training.

(a)


(b)

Figure 0.8: An example (a) original image (b) binary image

Figure 3.8 (b) was the image after erosion treatment. Most of the interference information in the picture has been removed after corrosion treatment. Valid information for identification processing was retained.

3.4 Evaluation Methods

In this experiment, accuracy, error rate, recall rate, ROC curve and loss function were used to evaluate the traffic sign recognition model. These evaluation methods are based on confusion matrix. The confusion matrix was shown in Table 3.1.

Table 3.1: Confusion matrix

	Positive	Negative
True	TP	TN
False	FP	FN

In Table 3.1, TP is true positive, positive samples are predicted to be positive samples, TN refers to true negative, negative sample is predicted to be negative sample, FP means false positive, negative samples are predicted to be positive samples, FN represents false negative, positive sample is predicted to be negative sample. Accuracy is the proportion of models correctly classified in the total sample. High accuracy indicates that the model has good performance as shown in eq. (3.1),

$$accuracy = \frac{TP+TN}{P+N} \tag{3.1}$$

Error rate means the proportion of model classification errors in the total sample. Low error rate indicates that the model has good performance as shown in eq.(3.2).

$$error\ rate = \frac{FP+FN}{P+N} \tag{3.2}$$

Recall rate refers to the percentage of the total that is actually positive that is correctly classified as positive. A high recall rate means more samples are misclassified as shown in eq. (3.3).

$$recall = \frac{TP}{TP+FN} \tag{3.3}$$

Precision is how many of the results judged to be true which are real true. High precision means that the model has good predictive ability as shown in eq. (3.4).

$$precision = \frac{TP}{TP+FP} \tag{3.4}$$

A loss function is a function that maps a random event to a non-negative real number to represent the "risk" or "loss" of that random event. The smaller the loss function is, the more accurate the model is. The loss functions include mean absolute error (MAE), mean square error (MSE), *Binary_CrossEntropy* and *Categorical_CrossEntropy.* The selection of appropriate loss functions is important for the stability and generalization of deep learning-based applications (Kuppala et al., 2022).

MAE and MSE are loss functions used for regression task learning. *Binary_CrossEntropy* is a loss function for dichotomous learning that describes the difference between the tag and the predicted value. *Categorical_CrossEntropy* is a loss function used for learning multiple categories. The loss function used in this experiment is the *Categorical_CrossEntropy* loss function.

3.5 Parameter adjustment

Parameter adjustment is an indispensable step in machine learning model. An evaluation index of the model can be improved by adjusting parameters. Bokaba et al. (2022) improved the performance of road traffic accident classifier by adjusting parameters. In this experiment, the accuracy of the model is improved by adjusting parameters. The model was iterated for 25 and 35 times respectively with other parameters unchanged and discussed the relationship between iteration times and accuracy.

3.6 Research Limitations

This project report has the following limitations:

- The samples are not abundant. There were only eleven categories with more than 500 instances in the TT100K dataset. Most of the categories in the sample had no more than 500 instances, and even 70 categories had no instances.

- The performance of the model was affected by hardware devices. The model could achieve the best performance by using GPU to recognize traffic signs on a computer. However, the speed of traffic sign recognition on a computer without GPU was slow and the model could not achieve optimal performance.

Chapter 4

Analysis and discussion

4.1 Results Analysis and Discussion

The experimental results in this chapter are obtained by the experimental methods in Chapter 3. The experimental results will be analyzed and discussed in this chapter. The loss function, accuracy, recall rate and accuracy rate were used to evaluate the model during training. There were 32 neurons in the model and a total of 25 iterations. Figure 4.1 shows the graph of loss function and iteration times. As the number of iterations increases, the Categorical_Crossentropy loss function becomes smaller and smaller.

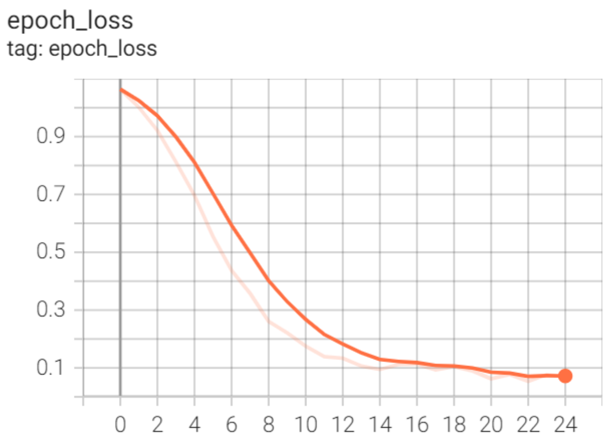

Figure 0.1: Relationship between loss function and iteration times

Figure 4.2 shows the chart of accuracy and iteration times. As the number of iterations increases, the accuracy rate becomes higher and higher. This indicates that the model is getting better and better at recognizing traffic signs.

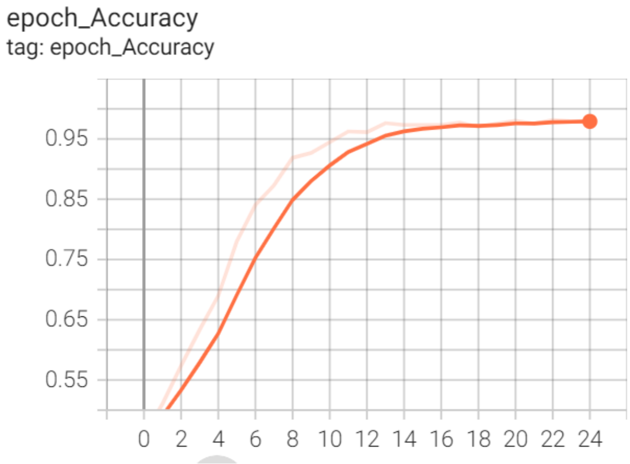

Figure 0.2: Relationship between accuracy and iteration times

Figure 4.3 reflects the relationship between recall rate and number of iterations. The recall rate increases with the number of iterations and finally approaches 1. This shows that the model is getting better and better.

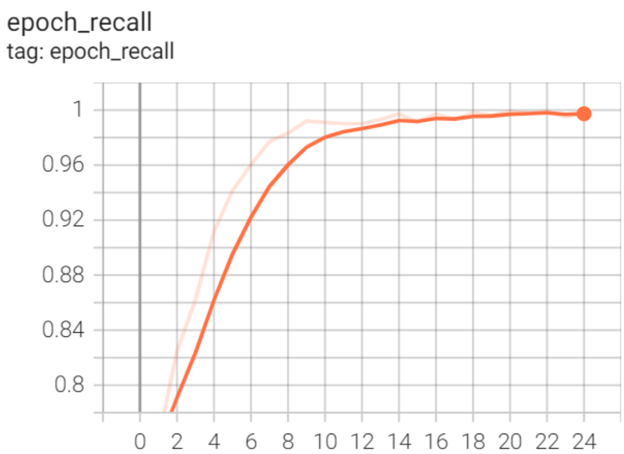

Figure 0.3 Relationship between recall rate and number of iterations

Figure 4.4 is the chart of precision and iteration times. The precision is positively correlated with the number of iterations. The increase in the number of iterations improves the precision.

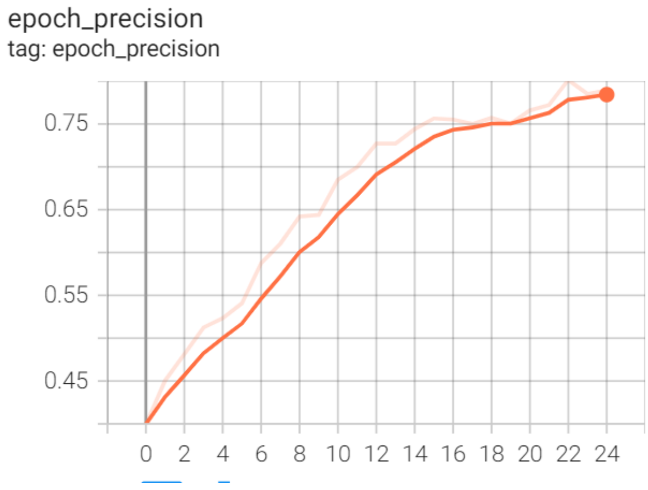

Figure 0.4: Relationship between precision and iteration times

Figure 4.5 showed that the accuracy of the model can reach 98.12% if the number of iterations is 25.

```
emoved in a future version. Please use  Model.fit , which supports generators.
  model.fit_generator(training_set,epochs=25,callbacks=[tensorboard_callback])

Epoch 1/25
32/32 [==============================] - 14s 394ms/step - loss: 1.0640 - Accuracy: 0.4579 - recall: 0.7473 - precision: 0.3998
Epoch 2/25
32/32 [==============================] - 12s 369ms/step - loss: 1.0015 - Accuracy: 0.5104 - recall: 0.7611 - precision: 0.4504
Epoch 3/25
32/32 [==============================] - 12s 371ms/step - loss: 0.9215 - Accuracy: 0.5738 - recall: 0.8246 - precision: 0.4812
Epoch 4/25
32/32 [==============================] - 12s 368ms/step - loss: 0.8127 - Accuracy: 0.6333 - recall: 0.8622 - precision: 0.5124
Epoch 5/25
32/32 [==============================] - 12s 368ms/step - loss: 0.6975 - Accuracy: 0.6898 - recall: 0.9118 - precision: 0.5233
Epoch 6/25
32/32 [==============================] - 12s 370ms/step - loss: 0.5536 - Accuracy: 0.7800 - recall: 0.9415 - precision: 0.5407
Epoch 7/25
32/32 [==============================] - 12s 371ms/step - loss: 0.4369 - Accuracy: 0.8404 - recall: 0.9604 - precision: 0.5876
Epoch 8/25
32/32 [==============================] - 12s 367ms/step - loss: 0.3577 - Accuracy: 0.8731 - recall: 0.9772 - precision: 0.6101
Epoch 9/25
32/32 [==============================] - 12s 371ms/step - loss: 0.2604 - Accuracy: 0.9187 - recall: 0.9832 - precision: 0.6421
Epoch 10/25
32/32 [==============================] - 12s 374ms/step - loss: 0.2214 - Accuracy: 0.9267 - recall: 0.9921 - precision: 0.6437
Epoch 11/25
32/32 [==============================] - 12s 371ms/step - loss: 0.1749 - Accuracy: 0.9445 - recall: 0.9911 - precision: 0.6849
Epoch 12/25
32/32 [==============================] - 12s 369ms/step - loss: 0.1388 - Accuracy: 0.9623 - recall: 0.9901 - precision: 0.6996
Epoch 13/25
32/32 [==============================] - 12s 375ms/step - loss: 0.1336 - Accuracy: 0.9613 - recall: 0.9901 - precision: 0.7271
Epoch 14/25
32/32 [==============================] - 13s 392ms/step - loss: 0.1062 - Accuracy: 0.9762 - recall: 0.9931 - precision: 0.7271
Epoch 15/25
32/32 [==============================] - 13s 392ms/step - loss: 0.0946 - Accuracy: 0.9732 - recall: 0.9970 - precision: 0.7441
Epoch 16/25
32/32 [==============================] - 16s 492ms/step - loss: 0.1119 - Accuracy: 0.9732 - recall: 0.9911 - precision: 0.7564
Epoch 17/25
32/32 [==============================] - 18s 557ms/step - loss: 0.1129 - Accuracy: 0.9732 - recall: 0.9970 - precision: 0.7553
Epoch 18/25
32/32 [==============================] - 16s 500ms/step - loss: 0.0933 - Accuracy: 0.9772 - recall: 0.9931 - precision: 0.7500
Epoch 19/25
32/32 [==============================] - 18s 552ms/step - loss: 0.1043 - Accuracy: 0.9703 - recall: 0.9980 - precision: 0.7571
Epoch 20/25
32/32 [==============================] - 15s 469ms/step - loss: 0.0897 - Accuracy: 0.9752 - recall: 0.9960 - precision: 0.7506
Epoch 21/25
32/32 [==============================] - 15s 462ms/step - loss: 0.0625 - Accuracy: 0.9802 - recall: 0.9990 - precision: 0.7660
Epoch 22/25
32/32 [==============================] - 15s 483ms/step - loss: 0.0774 - Accuracy: 0.9752 - recall: 0.9980 - precision: 0.7722
Epoch 23/25
32/32 [==============================] - 15s 463ms/step - loss: 0.0536 - Accuracy: 0.9812 - recall: 0.9990 - precision: 0.8013
Epoch 24/25
32/32 [==============================] - 15s 469ms/step - loss: 0.0772 - Accuracy: 0.9792 - recall: 0.9950 - precision: 0.7850
Epoch 25/25
32/32 [==============================] - 18s 567ms/step - loss: 0.0702 - Accuracy: 0.9802 - recall: 0.9980 - precision: 0.7892

: <keras.callbacks.History at 0x1ff7ded2500>
```

Figure 0.5: The accuracy of the model can reach 98.12% if the number of iterations is 25

4.2 Summary

The research and results show that the capsule network has achieved good performance on TT100K dataset. And capsule network can accurately identify traffic signs with an accuracy of 98%. Compared with the traditional traffic sign recognition model, the capsule network has better performance. The traffic sign recognition model in this study can accurately identify the blocked and damaged traffic signs. And achieved good performance in rainy and foggy days. It proves the practicability of capsule network in image recognition field.

After the model training is completed, the evaluation result of the model is improved by adjusting the number of iterations. If the number of iterations is 25, the loss function is 0.0702. If the number of iterations is increased to 35, the loss function continues to decrease to 0.0471. When the number of iterations is 25, the accuracy of the model reaches 98%. Then the number of iterations was increased to 35, and the accuracy was not significantly improved. If the number of iterations is 25, the recall rate of the model reaches 0.9802. Then the number of iterations was increased to 35, and the recall rate reached the maximum value of 1. When the number of iterations is 25, the precision of the model is up to 80%. Later, when the number of iterations was increased to 35, the precision dropped to 73.97%. This proves that the number of iterations should be adjusted within a reasonable range. Excessive elevation will make the model overfit and recognize some pictures similar to traffic signs as traffic signs. If the number of iterations is too low, the model will not fit properly, and the best performance will not be achieved.

Chapter 5

Conclusion

5.1 Conclusion

Traffic sign recognition technology has been improved with the development of autonomous driving technology. Traffic sign recognition can be accomplished by using a variety of neural network algorithms. Because of the complexity of the environment and the randomness of the road, some problems of traffic sign recognition have not been solved yet. Therefore, in this report, capsule neural network was be used to recognize traffic signs to achieve better recognition effect. Traffic sign recognition is of great significance for L5 level autonomous driving, which is the only way for the development of artificial intelligence. This paper introduced the architecture and routing algorithm of capsule network in detail. Through data collection, data preprocessing, model building and model training to identified traffic signs and achieved excellent performance, the accuracy rate exceeds CNN. The experimental results showed that the capsule network can effectively recognize the occluded or tilted images, and this network model with vector as input and output solves the problem that the traditional neural network cannot identify the orientation.

5.2 Future Work

There are may limitations in this study. Firstly, there are not enough instances in the dataset. Most of the categories in TT100K dataset have less than 500 instances, and some categories have no instances. As a result, the model cannot accurately identify all traffic signs. Secondly, the model has high requirements for hardware. The model can achieve optimal performance under GPU rendering. But the performance is mediocre on the devices without GPU. The monotonicity of data and the inadequacy of equipment and hardware affected the results of the study.

Future work will mainly focus on these two directions. Through continuous data collection to enrich the data set, increase the number of instances of types, and update the data set in accordance with traffic laws and regulations in time. The time cost and money cost can be reduced by adjusting parameters and optimizing network model, so that the model can achieve better performance on common equipment and increase the universality of the model.

Reference

Abebe Berwo, M., Fang, Y., Mahmood, J., & Afele Retta, E. (2021). Automotive engine cylinder head crack detection: Canny edge detection with morphological dilation. *Asia-Pacific Signal and Information Processing Association Annual Summit and Conference (APSIPA ASC)*.

Afriyie, Y., Weyori, B. A., & Opoku, A. A. (2021). Exploring optimised capsule network on complex images for medical diagnosis. *IEEE International Conference on Adaptive Science and Technology, ICAST*.

An, N., Yan, W. (2021) Multitarget tracking using Siamese neural networks. ACM Transactions on Multimedia Computing, Communications and Applications.

An, N. (2020) Anomalies Detection and Tracking Using Siamese Neural Networks. Master's Thesis. Auckland University of Technology, New Zealand.

Apoorva, A., Isha, G., Novesh, G., & Anurag, G. (2019). Deep learning approach to determine the impact of socio economic factors on bitcoin price prediction. *International Conference on Contemporary Computing (IC3)*

Arief, R. W., Nurtanio, I., & Samman, F. A. (2021). Traffic signs detection and recognition system using the YOLOv4 algorithm. *International Conference on Artificial Intelligence and Mechatronics Systems*.

Azhar, I. al, Ahmed, S., Islam, M. S., & Khatun, A. (2021). Identifying author in Bengali literature by Bi-LSTM with attention mechanism. *International Conference on Computer and Information Technology, ICCIT 2021*.

Bokaba, T., Doorsamy, W., & Paul, B. S. (2022). Comparative study of machine learning classifiers for modelling road traffic accidents. *Applied Sciences (Switzerland)*, *12*(2).

Chaitanya Lahari, M., Dhakate Hardik, R. (2018). Fuel price prediction using RNN. *International Conference on Advances in Computing, Communications and Informatics (ICACCI)*

Chen, L., Huang, S., Yang, C., & Chen, Q. (2020). Analyzing factors that Influence expressway traffic crashes based on association rules: Using the Shaoyang–

Xinhuang Section of the Shanghai–Kunming expressway as an example. *Journal of Transportation Engineering, Part A: Systems*, *146*(9), 05020007.

Cheung, B. (2012). Convolutional neural networks applied to human face classification. *International Conference on Machine Learning and Applications, 2*, 580–583.

Cui, W., Yan, W. (2016) A scheme for face recognition in complex environments. International Journal of Digital Crime and Forensics (IJDCF) 8 (1), 26-36.

Cui, W. (2015) A Scheme of Human Face Recognition in Complex Environments. Master's Thesis, Auckland University of Technology.

Dhiman, A., & Saroha, V. (2022). Detection of severity of disease in paddy leaf by integrating Edge detection to CNN-based model. *International Conference on Computing for Sustainable Global Development (INDIACom)*, 470–475.

Fang, S., Chang, X., & Wu, S. (2021). Histogram-based fuzzy C-means clustering for image binarization. *IEEE Conference on Industrial Electronics and Applications,* 1432–1437.

Gu, Q., Yang, J., Kong, L., Yan, W., Klette, R. (2017) Embedded and real-time vehicle detection system for challenging on-road scenes. Optical Engineering, 56 (6), 063102.

Gu, Q., Yang, J., Yan, W., Klette, R. (2017) Integrated multi-scale event verification in an augmented foreground motion space. Pacific-Rim Symposium on Image and Video Technology (pp.488-500)

Gu, Q., Yang, J., Yan, W., Li, Y., Klette, R. (2017) Local Fast R-CNN flow for object-centric event recognition in complex traffic scenes. Pacific-Rim Symposium on Image and Video Technology (pp.439-452)

Hammad, H., & Khotanlou, H. (2022). Detection and visualization of COVID-19 in chest X-ray images using CNN and Grad-CAM (GCCN). *Iranian Joint Congress on Fuzzy and Intelligent Systems (CFIS)*, 1–5.

Jeon, M., Do, M., Kawashima, D., & Takei, M. (2019). Application of artificial neural network model to a voltage-current system in the estimation of void-fraction in a gas-liquid flow. *IEEE Sensors Journal*, *19*(13), 5098–5109.

Jia, L., Shi, X., & Wei, L. (2020). Design of traffic sign detection and recognition algorithm based on template matching. *International Conference on Civil Aviation Safety and Information Technology*, 237–240.

Jiang, J., Yang, J., & Yin, J. (2022). Traffic sign target detection method based on deep learning. *International Conference on Computer Information Science and Artificial Intelligence (CISAI).*

Kohara, K., Japan, T. U., & Fukuhara, Y. (1997). *Stock price prediction using prior knowledge and neural networks Tsutomu Ishikawa*. *Intelligent Systems in Accounting, Finance and Management*

Kuppala, K., Banda, S., & Imambi, S. S. (2022). Hybrid loss functions for infrared and visible patch matching using Siamese CNN. *International Conference on Computing for Sustainable Global Development (INDIACom)*, 731–736.

Liang, H. (2021). Research on pre-training model of natural language processing based on recurrent neural network. *IEEE International Conference on Information Systems and Computer Aided Education,* 542–546.

Li, F., Zhang, Y., Yan, W., Klette, R. (2016) Adaptive and compressive target tracking based on feature point matching. International Conference on Pattern Recognition (ICPR), (pp.2734-2739).

Li, P. (2018) Rotation Correction for License Plate Recognition. Master's Thesis, Auckland University of Technology, New Zealand.

Li, P., Nguyen, M., Yan, W. (2018) Rotation correction for license plate recognition. International Conference on Control, Automation and Robotics.

Lieskovska, E., Jakubec, M., & Jarina, R. (2022). RNN with improved temporal modeling for speech emotion recognition. *International Conference Radioelektronika* 1–5.

Liu, J., Zhu, M., Cai, T., Kong, D., & Xing, Y. (2022). Intelligent tablet detection system based on Capsule neural network. *International Conference on Power, Electronics and Computer Applications (ICPECA),* 665–669.

Liu, X., Nguyen, M., Yan, W. (2019) Vehicle-related scene understanding using deep learning. Asian Conference on Pattern Recognition.

Liu, X. (2019) Vehicle-related Scene Understanding Using Deep Learning. Master's Thesis, Auckland University of Technology, New Zealand.

Liu, X., Yan, W. (2020) Vehicle-related scene segmentation using CapsNets. International Conference on Image and Vision Computing New Zealand.

Liu, X., Yan, W. (2021) Traffic-light sign recognition using Capsule network. *Springer Multimedia Tools and Applications.*

Luo, Z., Nguyen, M., Yan, W. (2022) Kayak and sailboat detection based on the improved YOLO with Transformer. ACM ICCCV.

Luo, Z., Nguyen, M., Yan, W. Sailboat detection based on automated search attention mechanism and deep learning models. International Conference on Image and Vision Computing New Zealand.

Mehtab, S., Yan, W. (2021) FlexiNet: Fast and accurate vehicle detection for autonomous vehicles-2D vehicle detection using deep neural network. International Conference on Control and Computer Vision.

Mehtab, S., Yan, W. (2022) Flexible neural network for fast and accurate road scene perception. Multimedia Tools and Applications.

Mehtab, S. Yan, W., Narayanan, A. (2022) 3D vehicle detection using cheap LiDAR and camera sensors. International Conference on Image and Vision Computing New Zealand.

Ming, Y., Li, Y., Zhang, Z., Yan, W. (2021) A survey of path planning algorithms for autonomous vehicles. International Journal of Commercial Vehicles.

Pan, C., Yan, W. (2018) A learning-based positive feedback in salient object detection. International Conference on Image and Vision Computing New Zealand.

Pan, C., Yan, W. (2020) Object detection based on saturation of visual perception. *Multimedia Tools and Applications*, 79 (27-28), 19925-19944.

Pan, C., Liu, J., Yan, W., Zhou, Y. (2021) Salient object detection based on visual perceptual saturation and two-stream hybrid networks. *IEEE Transactions on Image Processing.*

Qin, Z., Yan, W. (2021) Traffic-sign recognition using deep learning. *International Symposium on Geometry and Vision.*

Raje, N. R., & Jadhav, A. (2022). Automated diagnosis of Pneumonia through Capsule network in conjunction with ResNet50v2 model. *International Conference on Emerging Smart Computing and Informatics (ESCI)*, 1–6.

Rath, A., Sarangi, P., Rath, A., & Panda, G. (2021). Development and performance assessment of bio-inspired based ANN model for handwritten English numeral recognition. *IEEE International Conference on Applied Electromagnetics, Signal Processing, and Communication.*

Ren, P., Xiao, Y., Chang, X., Huang, P. Y., Li, Z., Gupta, B. B., Chen, X., & Wang, X. (2022). A survey of deep active learning. *ACM Computing Surveys* (Vol. 54, Issue 9).

Sabour, S., Frosst, N., & Hinton, G. E. (2017). *Dynamic routing between capsules*. http://arxiv.org/abs/1710.09829

Sajad, S., Dharshika, S., & Meleet, M. (2021). Music generation for novices using recurrent neural network (RNN). *IEEE International Conference on Innovative Computing, Intelligent Communication and Smart Electrical Systems*.

Saranya, M., Archana, N., Reshma, J., Sangeetha, S., & Varalakshmi, M. (2022). Object detection and lane changing for self driving car using CNN. *International Conference on Communication, Computing and Internet of Things (IC3IoT)*, 1–7.

Shen, D., Xin, C., Nguyen, M., Yan, W. (2018) Flame detection using deep learning. *International Conference on Control, Automation and Robotics.*

Shen, H., Kankanhalli, M., Srinivasan, S., Yan, W. (2004) Mosaic-based view enlargement for moving objects in motion pictures. *IEEE ICME'04*.

Shen, J., Yan, W., Miller, P., Zhou, H. (2010) Human localization in a cluttered space using multiple cameras. *IEEE International Conference on Advanced Video and Signal Based Surveillance.*

Shen, Y., Yan, W. (2019) Blind spot monitoring using deep learning. *International Conference on Image and Vision Computing New Zealand*.

Shi, L., Xu, H., Ji, W., Zhang, B., Sun, X., & Li, J. (2021). Real-time human activity recognition system based on Capsule and LoRa. *IEEE Sensors Journal*, *21*(1), 667–677.

Shuang Xu, Deqing Niu, Bo Tao, & Gongfa Li. (2018). Convolutional neural network based traffic sign recognition system. *International Conference on Systems and Informatics (ICSAI)*

Song, G. (2021). An improved traffic sign recognition algorithm based on deep learning. *International Conference on Intelligent Transportation, Big Data and Smart City, ICITBS 2021*, 481–484.

Suriya Prakash, A., Vigneshwaran, D., Seenivasaga Ayyalu, R., & Jayanthi Sree, S. (2021). Traffic sign recognition using deep learning for autonomous driverless vehicles. *International Conference on Computing Methodologies and Communication,* 1569–1572.

Vinod, P., Babu, M. S., Sarathi, R., & Kornhuber, S. (2021). Classification of polluted silicone rubber micro nanocomposites based on ESDD using ANN. *IEEE International Conference on Power Systems: Developments towards Inclusive Growth for Sustainable and Resilient Grid*.

Wang, J., Yan, W., Kankanhalli, M., Jain, R., Reinders, M. (2003) Adaptive monitoring for video surveillance. International Conference on Information, Communications and Signal Processing.

Wang, J., Kankanhalli, M., Yan, W., Jain, R. (2003) Experiential sampling for video surveillance. *ACM SIGMM International Workshop on Video surveillance* (pp.77-86).

Wang, J., Yan, W. (2016) BP-neural network for plate number recognition. International *Journal of Digital Crime and Forensics* (IJDCF) 8 (3), 34-45.

Wang, J. (2016) *Event-driven Traffic Ticketing System*. Master's Thesis, Auckland University of Technology, New Zealand.

Wang, J., Ngueyn, M., Yan, W. (2017) A framework of event-driven traffic ticketing system. *International Journal of Digital Crime and Forensics* (IJDCF) 9 (1), 39-50.

Wang, J., Bacic, B., Yan, W. (2018) An effective method for plate number recognition. *Multimedia Tools and Applications,* 77 (2), 1679-1692.

Xi, E., Bing, S., & Jin, Y. (2017). *Capsule network performance on complex data*. http://arxiv.org/abs/1712.03480

Xin, C. (2020) *Detection and Recognition for Multiple Flames Using Deep Learning.* Master's Auckland University of Technology, New Zealand.

Xin, C., Nguyen, M., Yan, W. (2020) Multiple flames recognition using deep learning. *Handbook of Research on Multimedia Cyber Security*, 296-307.

Xing, J., Yan, W. (2021) Traffic sign recognition using guided image filtering. *International Symposium on Geometry and Vision.*

Xing, J., Nguyen, M., Yan, W. (2022) The improved framework of traffic sign recognition by using guided image filtering. *Springer Nature Computer Science.*

Xing, J. (2022) *Traffic Sign Recognition from Digital Images Using Deep Learning.* Master's Thesis, Auckland University of Technology, New Zealand.

Xinyu, C. (2009). Fast binary dilation/erosion algorithm using reference points. *International Conference on Networking and Digital Society, ICNDS 2009*, *2*, 87–90.

Xu, H., & Xu, F. (2021). Multi-scale capsule network with coordinate attention for SAR automatic target recognition. *Asia-Pacific Conference on Synthetic Aperture Radar*.

Yan, W. (2019) Introduction to Intelligent Surveillance: Surveillance Data Capture, Transmission, and Analytics. Springer London.

Yan, W. (2021) Computational Methods for Deep Learning: Theoretic, Practice and Applications. Springer London.

Yahya, A. A., Tan, J., & Hu, M. (2021). A novel handwritten digit classification system based on convolutional neural network approach. *Sensors*, *21*(18).

Yang, X., & Wan, Y. (2021). Non-uniform illumination document image binarization using K-means clustering algorithm. *IEEE International Conference on Information, Communication and Networks,* 506–510.

Yildiz, G., & Dizdaroglu, B. (2021). Convolutional neural network for traffic sign recognition based on color space. *International Informatics and Software Engineering Conference*.

Zhang, H. (2020). The method of recognizing traffic signs based on the improved capsule network. *International Conference on Computer Engineering and Intelligent Control,* 22–26.

Zhang, H. L., Shao, Y. bin, Long, H., & Du, Q. (2020). A Chinese text synthesis method based on word order rules. *Procedia Computer Science*, *166*, 512–516.

Zheng, G. (2021). A novel attention-based convolutional neural network for human activity recognition. *IEEE Sensors*, *21*(23), 27015–27025.

Zhou, J., Chen, L., & He, J. (2022). SIMGAT: A sentiment analysis model based on graph attention mechanism. *International Seminar on Computer Science and Engineering Technology (SCSET),* 133–136.

Zhu, Y., Yan, W. (2022) Traffic sign recognition based on deep learning. *Multimedia Tools and Applications.*

Zhu, Z., Liang, D., Zhang, S., Huang, X., Li, B., & Hu, S. (2016). Traffic-sign detection and classification in the wild. *IEEE Conference on Computer Vision and Pattern Recognition*.