IT Carlow - BSc.
Data Science and Artificial Intelligence

License plate recognition final report

Student Name: Zhe Cui
Student Number: C00266169
Lecturer: Greg Doyle
Date: 07.02.2020

Contents

Table of Figures

1 Introduction

License plate recognition technology refers to the technology that can automatically detect license plates and obtain license plate information. It is a very active research topic in the field of image processing and intelligent transportation system [1]. This technology can be applied in the fields of traffic road dredging, illegal vehicles taking pictures [2], parking lot charge management [3] and so on, in order to reduce labor costs and improve the efficiency of traffic management.

This document contains the license plate recognition system development process, the challenges and achievements encountered in the development process.

2 Tools

2.1 Python

Python is a widely used general-purpose, dynamic, extensible, high-level programming language. It includes high-level data structures, dynamic typing, dynamic binding, and many more features that make it as useful for complex application development as it is for scripting or "glue code" that connects components together. It can also be extended to make system calls to almost all operating systems and to run code written in C or C++. Due to its ubiquity and ability to run on nearly every system architecture, Python is the most suitable languages for this project.

2.2 Pytorch

Pytorch is defined as an open source machine learning library for Python. It is used for applications such as natural language processing, computer vision. It is initially developed by Facebook artificial-intelligence research group, and Uber's Pyro software for probabilistic programming which is built on it. Pytorch provides an excellent platform which offers dynamic computational graphs [4].

2.3 OpenCV

OpenCV is the enormous open-source library for the computer vision, AI, and image processing and now it plays an important n role in real-time operation which is very essential in today's systems. By the usage of it, you'll be able to process images and motion pictures to perceive objects, faces, or even handwriting of a human. When it is included with various libraries with Numpy, python is capable of processing the OpenCV array structure for data analysis.

The first OpenCV version was 1.0. OpenCV is launched beneath a BSD license and subsequently, it's free for both educational and industrial use. It has C++, C, Python and Java interfaces and supports Windows, Linux, Mac OS, iOS and Android. When OpenCV was designed the primary focus was real-time applications for computational efficiency. Everything is written in optimized C/C++ to take advantage of multi-core processing.

3 Implementation

3.1 System design

The license plate recognition system designed by Zhe Cui is composed of license plate acquisition module, license plate localization module, license plate image preprocessing module, license plate character segmentation module, license plate character recognition module and the recognition output of license plate module. The system flow is shown in Figure 1.

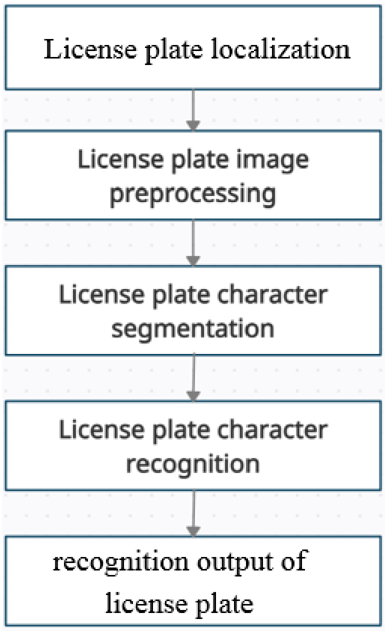


Figure 1 Flowchart of license plate recognition system

3.2 license plate collection

License plate collection is the first step in automatic identification system. The camera is connected to the computer, and the camera in real-time is transferred to the computer and then processed.

3.3 Data set

3.3.1 Web Crawling Tools to get data

First of all, using the Scrapy framework of Python language to code the web crawler, because the Scrapy has a good encapsulation of the web crawler technology, which can complete these tedious work faster. Then the web crawler is used to carry out the preliminary crawling of image data on Google images according to the keywords. After manual screening, 500 high-quality license plate pictures were obtained.

3.3.2 Label data

After data crawling is completed, the images obtained by the crawler should be initially screened, and then label high-quality images by a labeling tool, Labelme. Figure 2 shows the original image. Figure 3 shows the labeled binarization image.

Figure 2 The original image

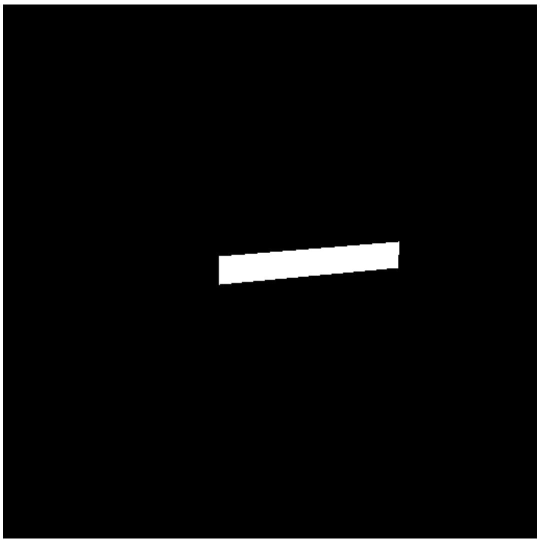


Figure 3 Labeled binarization image

It is necessary to mark each license plate area as accurately as possible when labeling the license plate. As shown in Figure 4, the marked area of the license plate, which contains other information is too big, and can cause some interference to the training the model. It is an example of wrong labeling. As shown in Figure 5, the marked area only contains the license plate, does not contain other interference information. It is a correct sample of license plate annotations.

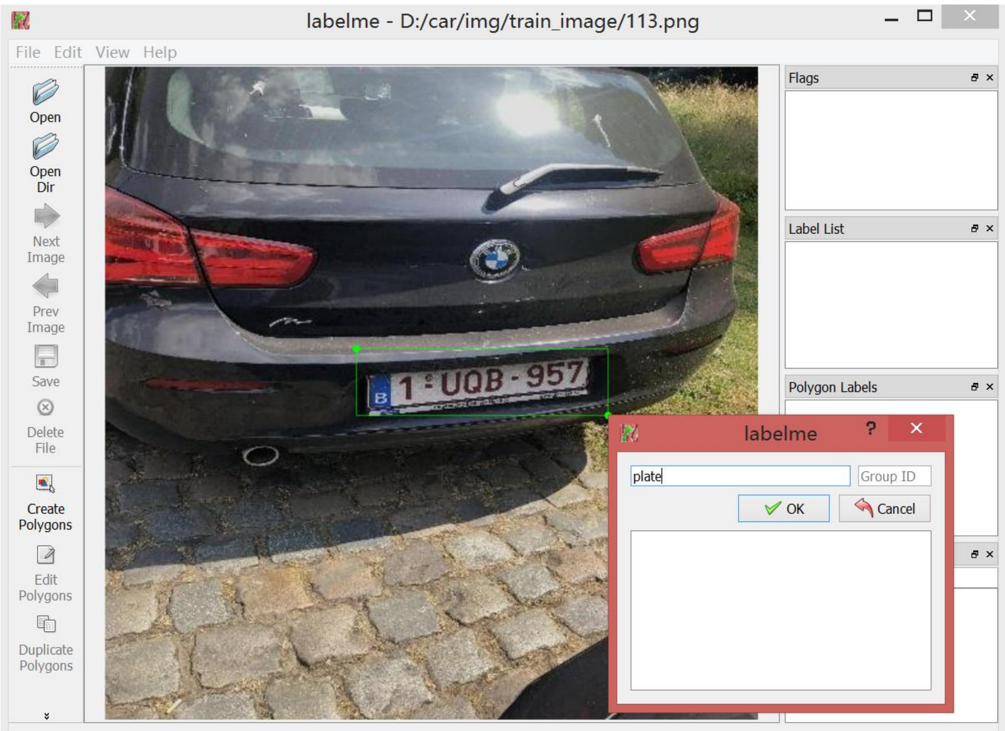


Figure 4 Example of false label of the license plate area

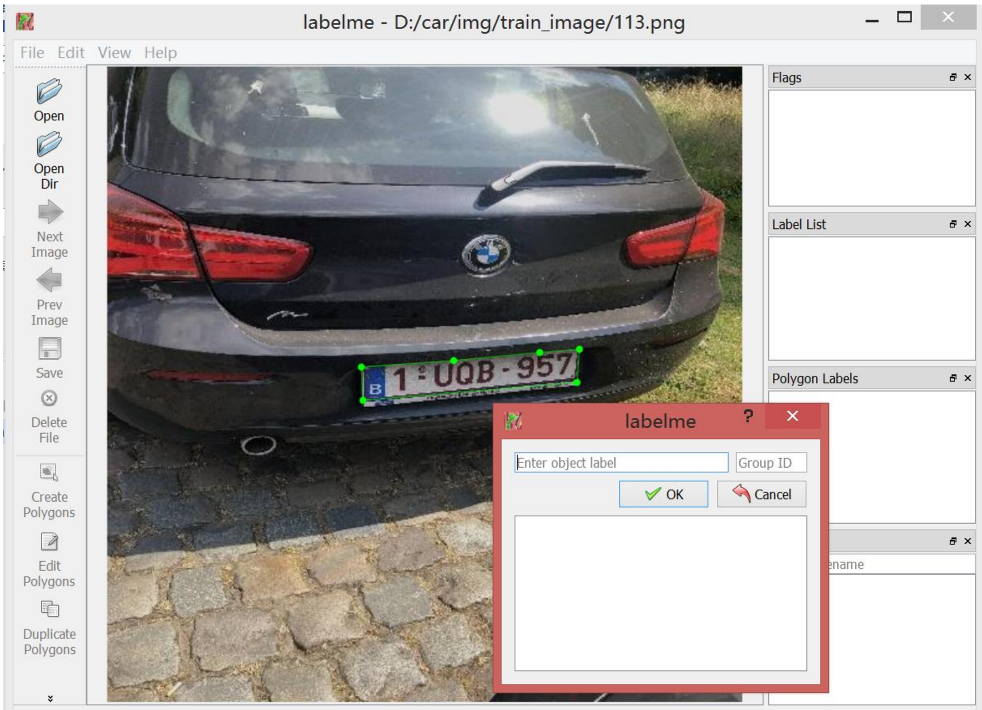


Figure 5 Example of correct label of the license plate area

3.4 License plate localization

License plate localization is the key of license plate recognition, with the wrong localization, the system will not be able to process and recognize the image. This paper adopts U-net network to locate the license plate. U-net is a convolutional neural network, which was originally invented and first used for biomedical image segmentation [5]. The network is based on the fully convolutional network [6] and its architecture was modified and extended to work with fewer training images and to yield more precise segmentation.

As shown in Figure 6, the encoder network can be called as a contraction path and the decoder network can be called as an expansion path. The difference between the contraction and the expansion layers is that the pooling operations in the contraction network are replaced by upsampling operation in the expansion network. Also, the expansion network has more filters which results in high-resolution layers.

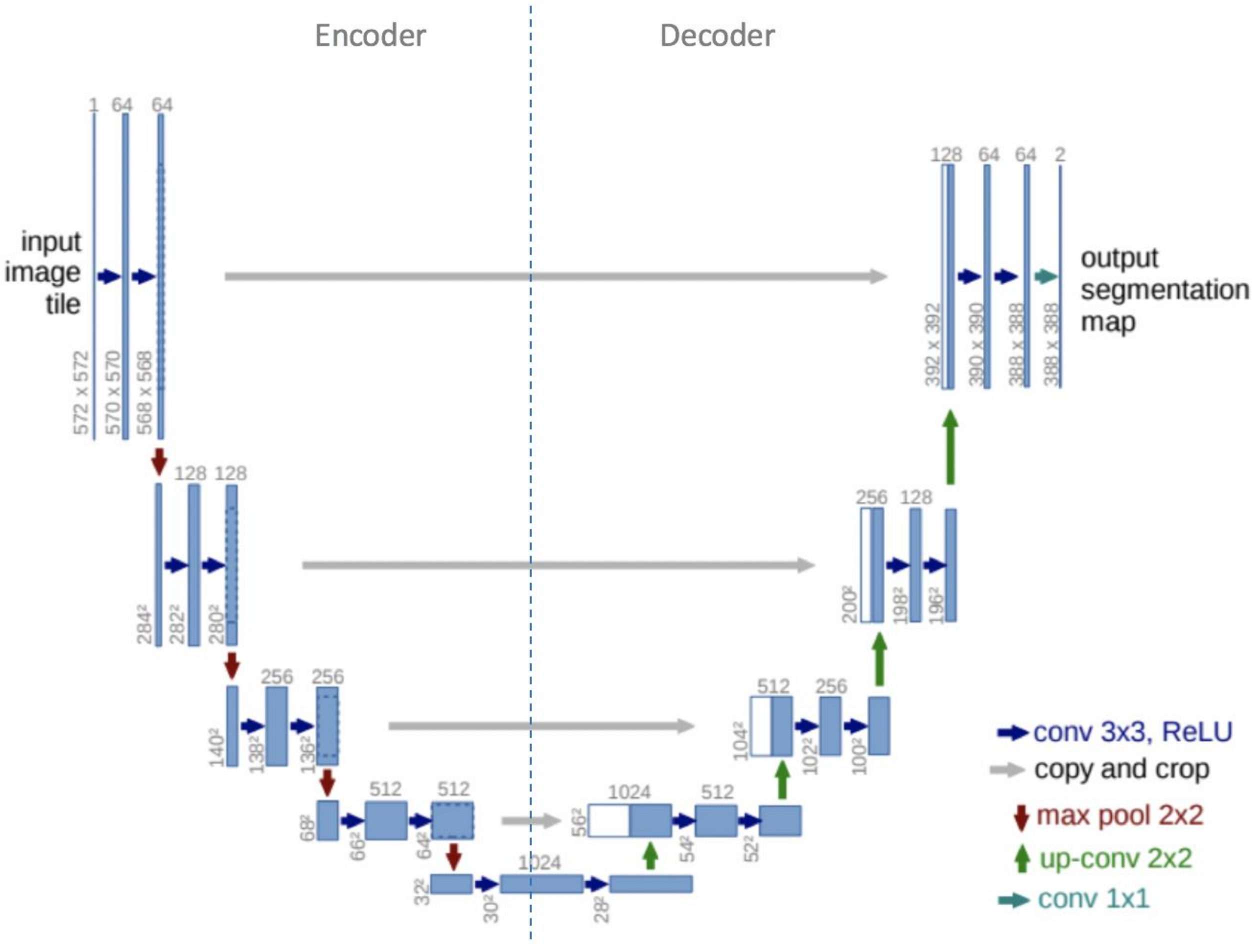


Figure 6 U-Net Expansion-Contraction
Source: https://arxiv.org/abs/1505.04597 (2015) U-Net: Convolutional Networks for Biomedical Image Segmentation

At the beginning, the 512x512x3 input image is passed through two 3x3 convolutional layers (padding='valid' and strides=1) and ReLU activation where the number of channels is increased.

The image is then downsampled as it is sent through a 2x2 max-pooling layer. This process is repeated many times until the image size becomes 28x28x1024. After that, it is sent through a 2x2 transpose convolution layer with strides=2 instead of downsampling.

Then, it is consecutively concatenated with a cropped version of the previous feature map and is sent through two 3x3 convolutional layers like in the contraction path.

U-net's contribution is that concatenate the higher resolution function maps from the encoder network with the upsampled features when upsampling the network, in order to help learn better representations with corresponding convolutions.

The image is segmented by U-net is shown as Figure 7.

Figure 7 Segmentation

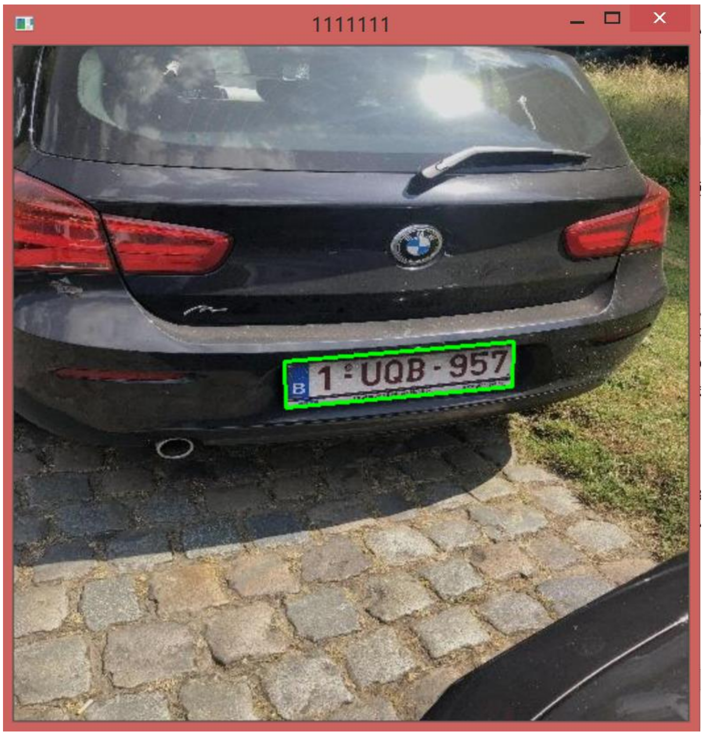


Figure 8 License plate localization

3.5 License plate image preprocessing

3.5.1 Gray processing

The input image is a color image, consisting of red, green, and blue, which is called RGB. When R=G=B, the image is gray, and this process from random RGB values to R=G=B is called gray processing, and the grayscale range is 0 to 255. The common processing methods of color image gray processing include the component method, maximum method and weighted average method.

3.5.2 Binarization

Firstly, setting a threshold, and then transforming the grayscale image which is going to be processed. If the grayscale to be processed has a grayscale value greater than 125, it will become 255, and if it is less than or equal to 125, it will become 0, to achieving binarization. By using graythresh function, the maximum inter-class variance method is used to find the suitable threshold value of the image, and the binarization is completed.

Figure 9 Binarization

3.5.3 Noisy data removal

The image is processed by eroding and diluting, and then the noise such as stains and wear on the license plate are removed by filtering small objects. Open operation and closed operation are used in this paper. The open operation first erodes and then dilutes, which can smooth the contour of the image, truncate the broken place, and remove the edges and corners on the contour. The closed operation first dilutes and then erodes, which can remove the small holes in the image and complete the broken places [7].

3.5.4 Image cropping

After eliminating the noise, the image needs to be cropped in order to improve the subsequent accuracy. The localization of the license plate can be determined through the statistical gray value of the concentrated range. After binarization, the image needs to be inverted to find the black pixels better, and then the number of black dots in the Y direction and X direction is calculated by the projection method. In order to avoid the error caused by the edge, searching from the center position of the license plate to both sides to judge whether the black character is lower than the threshold, if it is lower than the threshold, it is the upper and lower bounds. In order to avoid the influence of the left border, it is necessary to judge whether the number of black pixels at 1/14 in the x-direction is less than 1. If less than 1, it is judged to be the left border, if not, x=1 is regarded as the left border.

3.5.5 Straighten the tilted picture

Due to the influence of external factors when photographing the license plate image, such as the angle between the camera and the license plate is not good, the jitter of the vehicle when driving or the license plate is not hung correctly, resulting in the slanting state of the license plate. If using the most basic fixed angle rotation, the tilt angle of normal license plate pictures or oblique pictures may be larger, causing an irreparable impact on subsequent recognition. In this paper, using Radon transform to rotate images. Different images can be calculated to obtain different rotation angles. Radon transform is shown as Figure 10 and Figure 11.

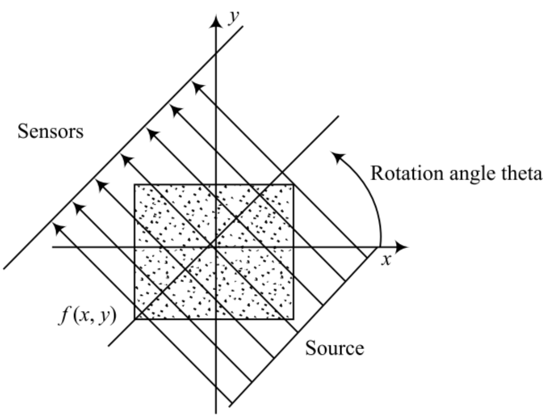


Figure 10 Radon transform

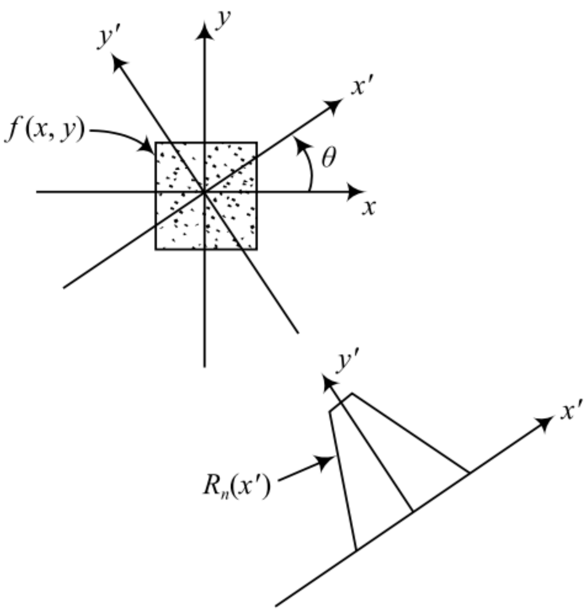


Figure 11 Radon transform

3.6 License plate character segmentation

The methods commonly used for character segmentation include vertical projection, license plate character interval and license plate size detection, license plate character contour detection and so on. The vertical projection method is used in this paper, and the flowchart of License plate character segmentation is shown as Figure 12.

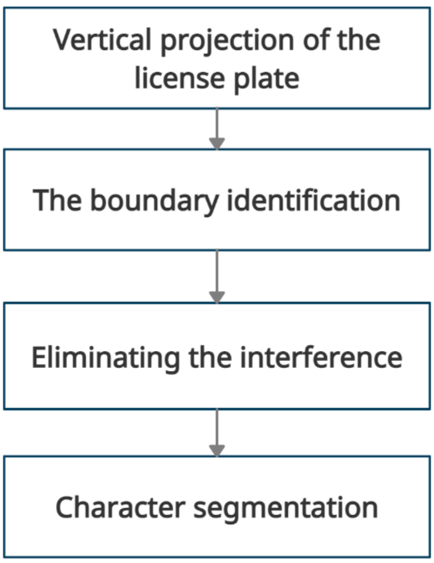


Figure 12 Flowchart of License plate character segmentation

Figure 13 shows the vertical projection of a license plate.

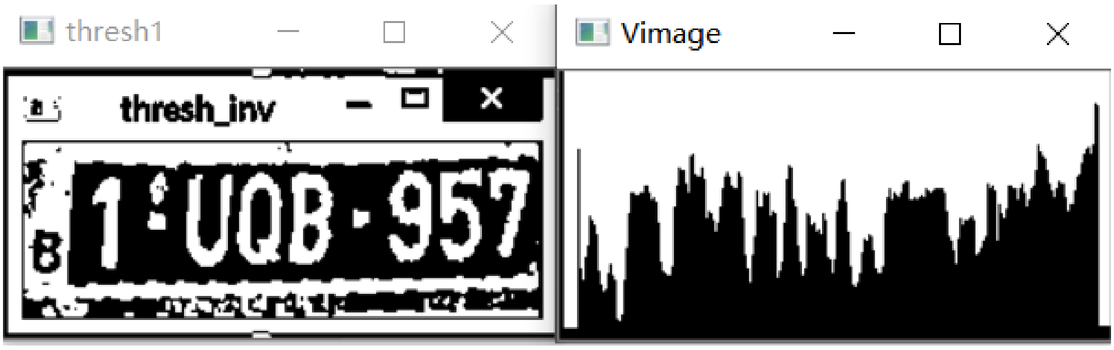


Figure 13 Vertical projection

First, setting a threshold to determine the black and white pixels, and then an array is defined, to count the total number of black pixels in the vertical direction of each column in the image, compare it to the threshold to determine the color of the column. First, judging whether the first column is black. If it is black and there is no interference, it is the boundary of the first character. According to the composition of characters, in the absence of interference, if the gray value of the region boundary ranges from 255 to 0 and then from 0 to 255, then this region is a character. Because the above processing may not be perfect, it may leave some interference, one of which is the interference in front of the letter. In order to remove the interference in front of the letter, the following method is used:

(1) Identifying whether the total of black pixels in the black area is less than the threshold value. If it is less than the threshold value, it means that the black area is not the license plate area.

(2) Whether the identified black area is less than half of the width of the next black area. Normally, the identified black area should be greater than the next one or equal to the next one. If there is interference, the black area is removed through the loop. According to the license plate specification, the segmented characters are classified and recognized subsequently. The result of character segmentation is shown as Figure 14.

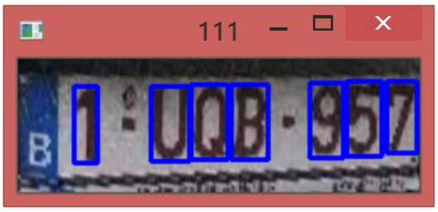


Figure 14 Character segmentation

3.7 License plate character recognition

Histogram of Oriented Gradients (HOG) feature has good anti-interference ability to the change of image light intensity, and has strong robustness to the description of the target [8]. Support vector machine (SVM) algorithm has the advantages of simple system structure, global optimization, short training time and so on. Therefore, the method based on the HOG feature and SVM is used to recognize characters.

3.7.1 Feature extraction based on Histogram of Oriented Gradients (HOG)

The main idea of the HOG feature is to use gradient or edge direction distribution to fully display the contour features of local targets in the image to be detected [9]. HOG extracts the gradient of each pixel point from the cell unit in the local area of the image, so there is no strict limit on clarity, which reduces the requirements for video shooting tools.

The realization process of the HOG feature extraction algorithm is mainly divided into the following four steps:

(1) Standardization of color space

First, the entire image is normalized with Gamma, which represents the compression factor, usually 0.5. This compression process can effectively reduce the sensitivity of the algorithm to illumination and improve the stability of the algorithm to illumination changes. The Gamma compression formula is:

$$I(x, y) = I(x, y)^{\text{gamma}}$$

Formula 1

(2) Calculate image gradient information

After the gradient values of the horizontal and vertical coordinates of the image are respectively calculated, the gradient values of each pixel are also obtained successively. The derivation operation not only gives information about the shape and parts of the texture, but also again reduces the interference caused by the light. The gradient is defined by the operator in the horizontal direction [-1, 0, 1] and the vertical direction $[-1, 0, 1]^T$ as:

$$G_x(x,y) = I(x+1,y) - I(x-1,y)$$

Formula 2

$$G_y(x,y) = I(x,y+1) - I(x,y-1)$$

Formula 3

In the formula, G_x(x, y), G_y(x, y) and I(x, y) respectively represent the horizontal gradient value, vertical gradient value and pixel gray value at the pixel point (x, y) in the input image.

The gradient amplitude and gradient direction at this pixel are respectively:

$$G(x, y) = \sqrt{G_x(x, y)^2 + G_y(x, y)^2}$$

Formula 4

$$\alpha(x, y) = \arctan\left(\frac{G_x(x, y)}{G_y(x, y)}\right)$$

Formula 5

(3) Histogram of cell gradient direction

The image is divided into multiple cell units of the same size, which are composed of smaller unit pixels. Gradient information of pixels within a cell is collected using a histogram of 9 bins, as shown in Figure 15, each direction block is 20°. In the histogram, all pixel points in the cell are multiplied by the gradient amplitude to project the gradient direction. Then sum their projections, getting a cell gradient direction histogram.

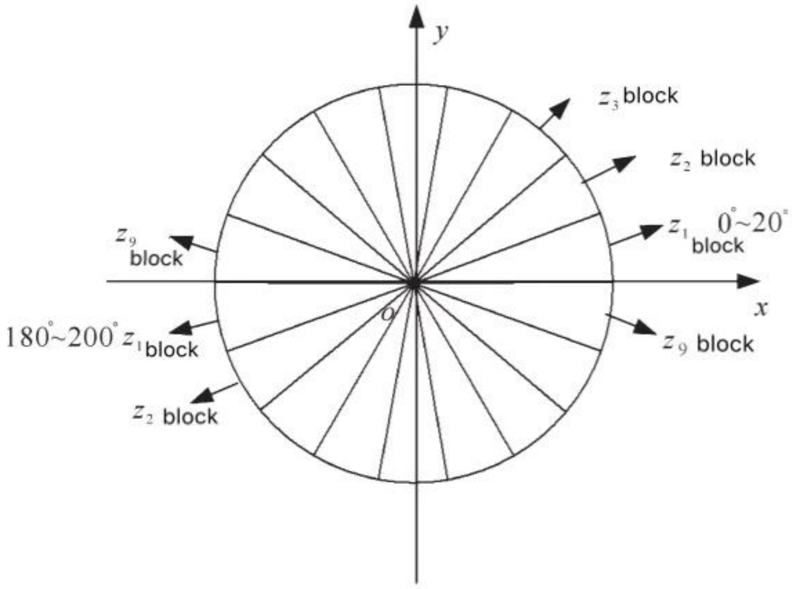


Figure 15 Schematic diagram of gradient direction division

(4) Normalized gradient histogram in blocks

Block is a block composed of several cells, and there can be overlap between different blocks. The HOG feature of the local area of the image is composed of all the blocks it contains. It is necessary to carry out intra-block normalization, mainly in order to avoid large gradient value difference caused by local exposure and other factors. The normalization process is shown in Formula 6. V represents the unnormalized feature vector of a block. F represents the normalized result:

$$f = \frac{\boldsymbol{v}}{\sqrt{\|\boldsymbol{v}\|^2 + \mu^2}}$$

Formula 6

3.7.2 Support vector machine (SVM) classification

The core idea of SVM is to find the separation hyperplane that meets two conditions at the same time. One is that it can accurately divide the training set, the other is that the greatest separation between the classes—it comes no closer to either, such a separation hyperplane is unique [10]. The linear discriminant function in N-dimensional space is shown as Formula 7, and the linear equation of the separating hyperplane is shown as Formula 8 and Figure 16.

$$f(x) = \boldsymbol{w} \cdot \boldsymbol{x} + b$$

Formula 7

$$\boldsymbol{w}^{\mathrm{T}} \cdot \boldsymbol{x} + b = 0$$

Formula 8

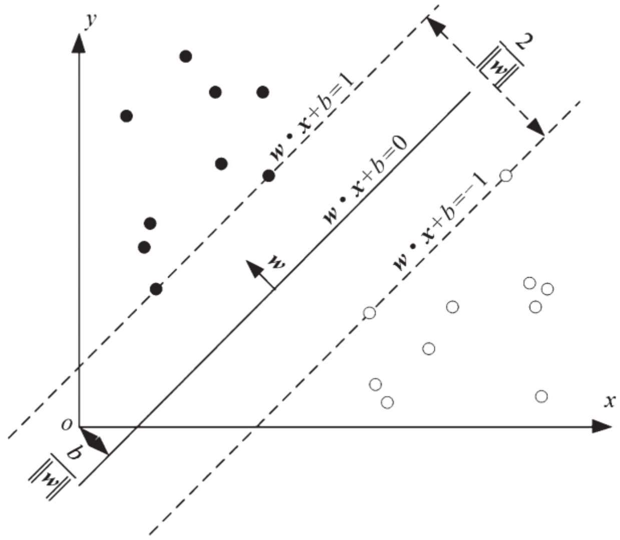


Figure 16 Hyperplane selection

SVM classifier is a binary classifier, so this application uses the one-against-the-others algorithm combined with multiple binary classifiers to convert the multi-class identification problem into a binary identification problem, and each classifier distinguishes only one character from the others. The result of character recognition is shown as Figure 17.

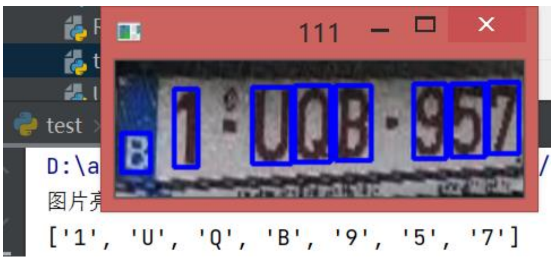


Figure 17 Character recognition

4 Achieved

This report proposes a license plate recognition system, the Uet algorithm is used for license plate localization, vertical projection is utilized for character segmentation, and the HOG +SVM algorithm is employed for character classification. The experimental results show the proposed system could improve the accuracy and calculation speed.

4.1 Data set

4.1.1 License plate data set

Using the Scrapy framework of Python language to code the web crawler, 500 images are crawled according to the keywords, then label them manually.

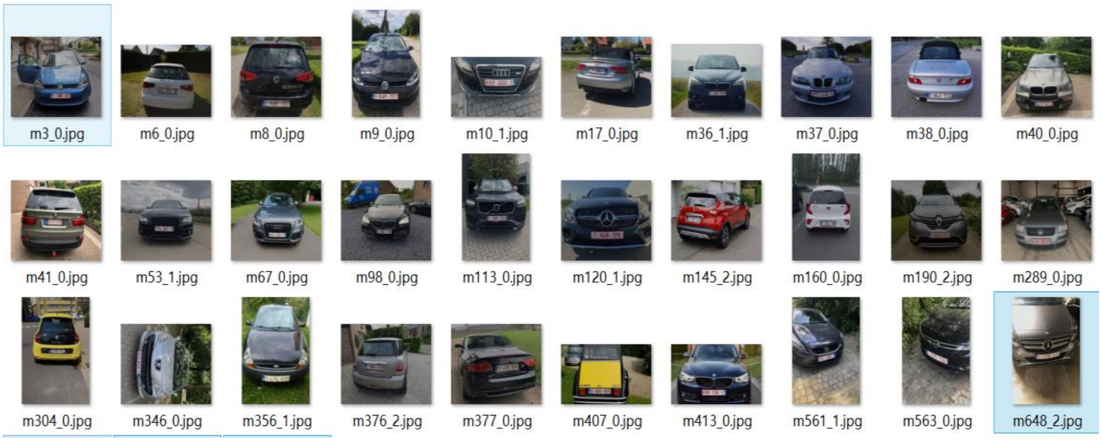


Figure 18 Data set

4.1.2Character data set

Single characters are cropped from license plates and manually labeled.

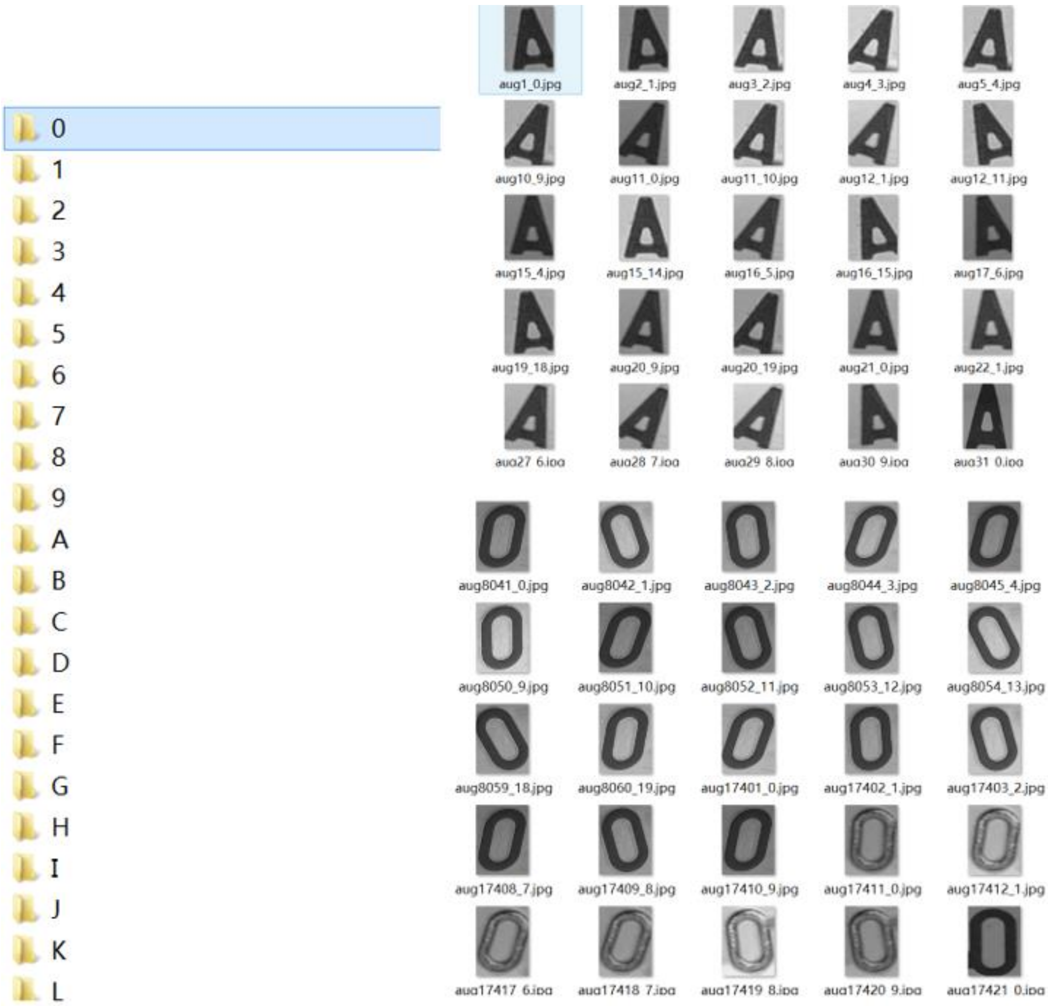


Figure 19 Character data set

4.2 License plate localization

This paper adopts U-net network to locate the license plate, achieving precise positioning of the license plate, and the recognition accuracy can achieve 92%.

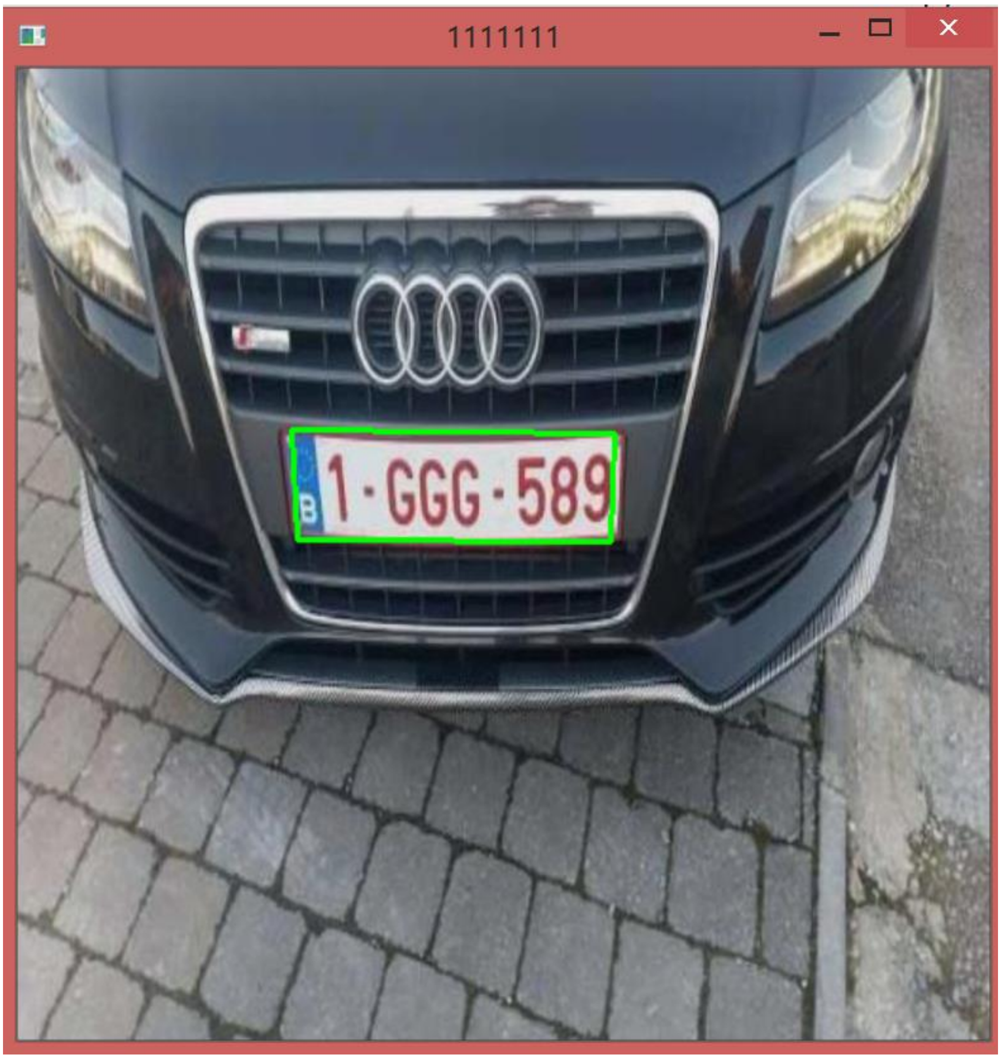


Figure 20 License plate localization

4.3 License plate character segmentation

The vertical projection method is used for character segmentation, it works well for majority license plates.

In the experiment, the vertical projection method and Canny operator edge detection method are used for character segmentation, the experimental results show that the vertical projection is not effective, so this report chooses Canny operator character for character segmentation. As shown in Figure 21, for the license plate which is located very precisely, there is no interference from other parts when the character segmentation, character splitting results can be good. But for the license plate which is not located very accurately, as shown in Figure 22, the localization results include not only the license plate, but also the other parts of the vehicle. In the process of vertical projection, other parts will cause interference, and there will be no obvious boundary between multiple characters, then forming a connected domain, leading to unable to cut out the characters.

Vertical projection method determines the horizontal position of characters based on the vertical projection distribution of binary images. The character boundary is considered to be the less white pixel projection (near zero or zero). This puts forward high requirements for the binarization, geometric distortion correction and filtering of license plate image (especially for the integrity, continuity and noise removal after the binarization of license characters). If these aspects are not handled well, the license plate image can not show the projection distribution law required by the method, which makes the judgment decision behind lose the basis. Due to the complexity and diversity of license plate image acquisition environment, it is difficult to meet the requirements of this method in practical application at present, which makes it impossible to determine the horizontal position of characters using this method in many cases.

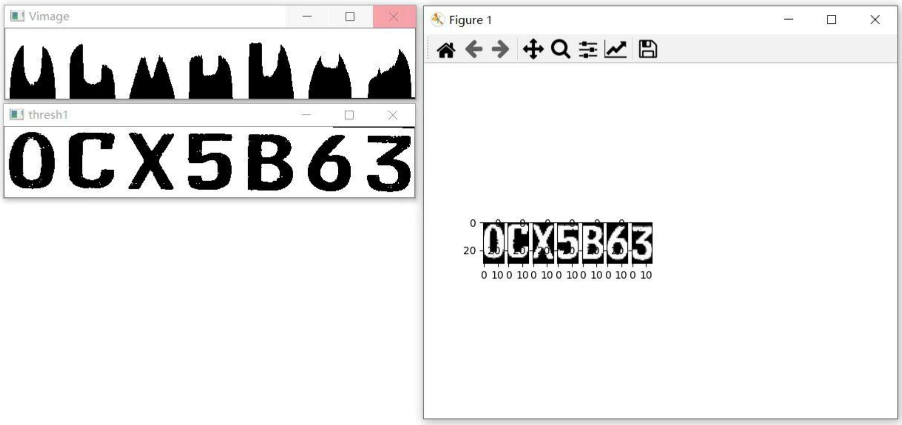


Figure 21 Successful character segmentation

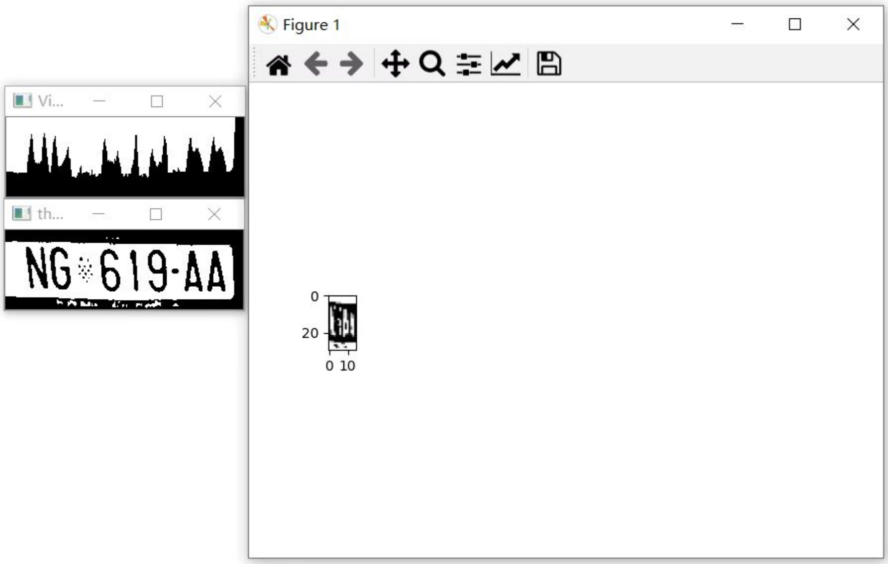


Figure 22 Unsuccessful character segmentation

The Canny operator edge detection method follows three criteria for edge detection:

(1) To ensure that the edge is successfully detected, there should be a strong response to the weak edge.

(2) Ensuring good edge positioning.

(3) To ensure that an edge can only be detected once.

The Canny edge detection algorithm can be broken down into 4 steps:

(1) Two-dimensional Gaussian filter template is convolved with gray image to reduce the impact of noise.

(2) Using derivative operator (such as Prewitt operator and Sobel operator) to find derivative G_x and G_y of image gray along two directions, and find the magnitude and direction of the gradient.

$$|G| = \sqrt{G_x^2 + G_y^2}, \quad \theta = \arctan\left(\frac{G_y}{G_x}\right)$$

Formula 9

(3) Non-maximum suppression

Traversing the image, if the gray value of a pixel is not the largest compared with the gray value of the two pixels before and after it in the gradient direction, the pixel value is set to 0, which means it is not the edge.

(4) Calculating two thresholds using image cumulative histogram.

A pixel whose gray value is greater than the high threshold is an edge, a pixel whose gray value is less than the low threshold is not an edge, a pixel between two thresholds is an edge if its adjacent pixel has a gray value greater than the high threshold, otherwise, it is not the edge.

The result of edge extraction by Canny operator method is shown in Figure 23. For the edge extraction of license plate image, the Canny operator method can be used to better extract the continuous character contour edge, and at the same time has a good noise suppression. What's more, the Canny operator method automatically selects corresponding thresholds according to the cumulative histogram of specific gray images, and can generally obtain stable edge extraction results for license plate images in different scenes.

Figure 23 Edge extraction by Canny operator

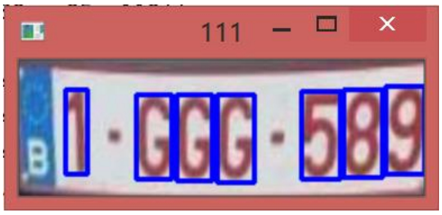
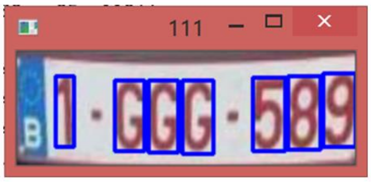


Figure 24 Character segmentation

4.4 License plate character recognition

The method based on the HOG feature and SVM is used to recognize characters. HOG feature has good anti-interference ability to the change of image light intensity, and has strong robustness to the description of the target [8]. SVM algorithm has the advantages of simple system structure, global optimization, short training time and so on. The identification results are shown in Figure 25.




```
['B', '1', 'G', 'G', 'G', '5', '8', '9']
```

Figure 25 Character recognition

4.5 Detecting the license plate that appears in the video

The original plan was to recognize license plates in pictures, later in the experiment, the recognition of license plates in videos and accumulated the time that license plates appears in the video are implemented. The number in the top left of the video represents how long the license plate is visible in the video. The video is divided into frames and the license plate in each frame is detected separately. Since the license plate generally has three or more numbers, it is timed when three or more digits are detected in a picture.

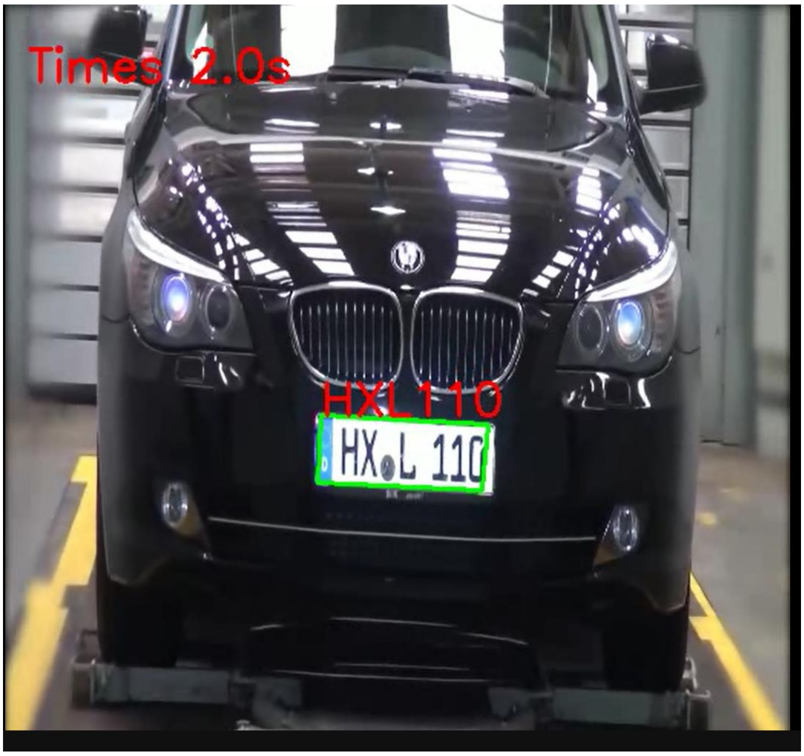


Figure 26 Video detection

5 Future work

Conclusion

This report proposes a license plate recognition system, the Uet algorithm is used for license plate localization, vertical projection is used for character segmentation, and the HOG +SVM algorithm is used for character classification. The proposed system shows decent accuracy and calculation speed. However, due to the time and capacity limited, there are still many deficiencies that need to be improved:

(1) More quality data is needed to make the system work accurately.

(2) It is difficult to identify damaged and dirty license plates.

(3) It is difficult to identify a license plate with a large tilt angle.

(4) The initial plan was to develop an Android-based license plate recognition system, which will need to be ported to Android in the future.

(5) B and 8 are very similar, the image becomes blurred after stretching. For blurred images, the system cannot distinguish B and 8. The next step is to update the image preprocessing method and character classification algorithm to improve the recognition accuracy.

Bibliography

[1] Duan, T.D., Du, T.H., Phuoc, T.V. and Hoang, N.V., 2005, February. Building an automatic vehicle license plate recognition system. In *Proc. Int. Conf. Comput. Sci. RIVF* (Vol. 1, pp. 59-63).

[2] Liu, L. and Jiang, T., 2003. Research on image acquisition technology in intelligent license plate recognition system (Doctoral dissertation).

[3] Peng, W., Ren, C. and Zhou, Y., 2017. Research on License Plate Recognition System Based on MATLAB. *Electronic manufacture*, *22*(11).

[4] Tutorialspoint.com. 2021. *PyTorch - Introduction - Tutorialspoint*. [online] Available at: <https://www.tutorialspoint.com/pytorch/pytorch_introduction.htm> [Accessed 5 February 2021].

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

[6] Long, J., Shelhamer, E. and Darrell, T., 2015. Fully convolutional networks for semantic segmentation. In *Proceedings of the IEEE conference on computer vision and pattern recognition* (pp. 3431-3440).

[7] Chen, C. and Zhao, W., 2011. MATLAB Application Case Introduction: Image Processing and GUI Design

[8] Han, S., Han, Y. and Hahn, H., 2009. Vehicle detection method using Haar-like feature on real time system. *World Academy of Science, Engineering and Technology*, *59*, pp.455-459.

[9] Rybski, P.E., Huber, D., Morris, D.D. and Hoffman, R., 2010, June. Visual classification of coarse vehicle orientation using histogram of oriented gradients features. In *2010 IEEE Intelligent vehicles symposium* (pp. 921-928). IEEE.

[10] Noble, W.S., 2006. What is a support vector machine?. *Nature biotechnology*, *24*(12), pp.1565-1567.