Vehicle License Plate Recognition

Zhijie Lan 201990309

Computer Engineering
Memorial University of Newfoundland
St. John's, NL
zlan@mun.ca

Abstract

Vehicle License Plate Recognition (VLPR) plays an essential role in contemporary society. This research will study the publications related to VLPR and reproduce the VLPR function based on KNN (K nearest neighbors). The results of this study show that the segmentation and positioning algorithms are highly accurate, but the training data of KNN is still needed to be improved.

1. Introduction

Vehicle License Plate Recognition (VLPR) is an image Recognition technology that automatically locates the License Plate area and extracts License Plate information from one or a series of digital images. VLPR, based on digital image processing, pattern recognition, and computer vision, is an essential part of the modern intelligent transportation system. This technology is widely used in daily life, including parking fee management, vehicle access management, automatic release, traffic flow control measurement, auto theft, highway speed supervision, electronic police, highway toll booths, and so on.

Commonly used VLPR systems usually include vehicle detection, image acquisition, and license plate recognition. Its hardware base includes trigger equipment (such as ground sensor coil, infrared ray), camera equipment, lighting equipment, image acquisition equipment, processing machine (such as computers) for recognizing license plate numbers, etc. The core of the software includes license plate location algorithms, character segmentation algorithms, and optical character recognition algorithm. Its basic principle is that when the vehicle arrives, the vehicle detection part triggers the image acquisition unit to collect the current video image. The license plate recognition unit processes the image, locates the license plate position, and then divides and recognizes the characters in the license plate to generate license plate information output. The design and implementation of VLPR can provide an in-depth understanding of various image processing technologies.

2. Workflow

The following flowchart (in the next page) shows the process of license plate recognition. The entire program can be roughly divided into two parts, framed in red and blue respectively. The program uses the OpenCV library for image processing.

The function of the part 1, the red part, is to generate areas in the image that may be license plates. The first step is, after the original image has been read in, it will undergo grayscale conversion, using function as below:

```
cvtColor(imgOriginal, img_Value, CV_BGR2GRAY);
```

Here, set "code" parameter as CV_BGR2GRAY to indicate a grayscale conversion. After the grayscale image is obtained, the program will use morphology function to obtain the top-hat and black-hat of the grayscale image to maximize the contrast of this image.

```
morphologyEx(imgGrayscale, img_TopHat,\
    CV_MOP_TOPHAT, img_structuringElement);
morphologyEx(imgGrayscale, img_BlackHat,\
    CV_MOP_BLACKHAT, img_structuringElement);
```

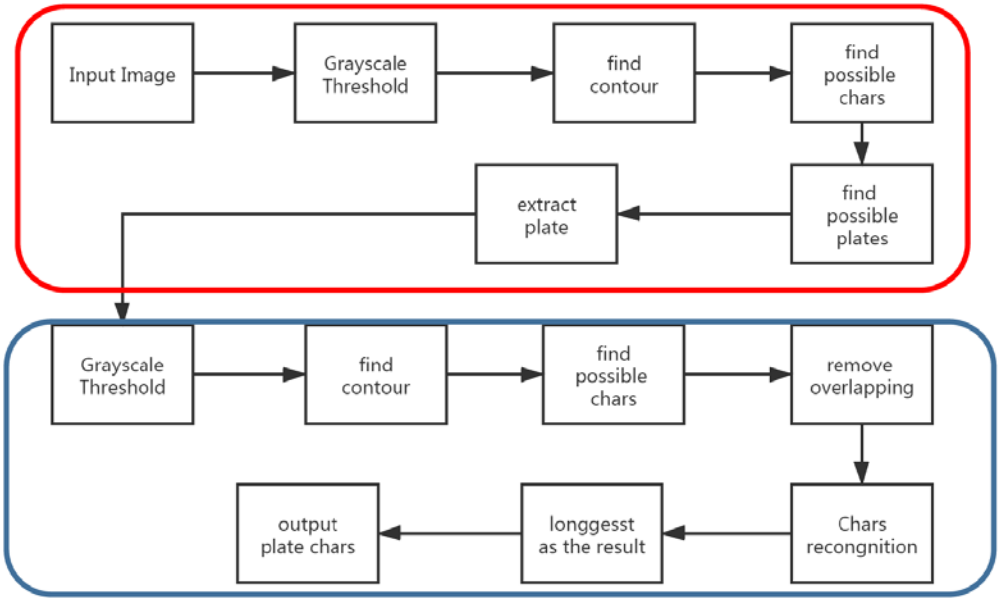


Figure 1. Flow chart of plates recognition

Then a Gaussian filter will be applied to eliminate possible noise in the picture in preparation for the subsequent binarization processing.

```
GaussianBlur(img_Max_Contrast_Grayscale,\
    img_Blurred, GAUSSIAN_SMOOTH_FILTER_SIZE, 0);
```

Then, Threshold processing is performed on it to obtain a binary image. Here, the adaptive threshold function is used, and the function method is set to CV_ADAPTIVE_THRESH_GAUSSIAN_C for better results.

```
adaptiveThreshold(img_Blurred, img_Thresh, 255.0, \
    CV_ADAPTIVE_THRESH_GAUSSIAN_C, CV_THRESH_BINARY_INV, \
    ADAPTIVE_THRESH_BLOCK_SIZE, ADAPTIVE_THRESH_WEIGHT);
```

Next, contour detection processing is performed on the binary image.

```
findContours(img_ThreshCopy, v_contours,\
    CV_RETR_LIST, CV_CHAIN_APPROX_SIMPLE);
```

The parameter CV_RETR_LIST used here means to detect all contours, including inner and outer contours, but the detected contours do not establish a hierarchical relationship. CV_CHAIN_APPROX_SIMPLE means that only the inflection point information of the contour is saved, and all the points at the inflection point of the contour are saved in the contours vector, and the information points on the straight line between the inflection point and the inflection point are not retained.

Here, all the image preprocessing is accomplished. The program will perform rough character detection on the obtained contour vector by length and angle comparison to filter out some useless contours. After rough filtering of round one, the possible contours are then filtered for the second round. The primary method for this is still about the distance and angle relationship among contours, and whether multiple contours can form a license plate area. Here, some license plate area candidates can be calculated based on the data of the second round filtering and part 1 finished.

In part 2, the blue part, each possible license plate areas obtained from part 1 will be treated as an individual image and preprocessed in the same way as part 1 for the original image, which includes grayscaling, morphological processing, Gaussian bluring, adaptive thresholding, finding contours, and finding possible characters. After that, the possible characters in each possible plate areas will be performed character recognition using KNN training data. KNN is a pattern recognition technology. The specific training method is not the focus of this article. This research uses open-source training data. Finally, the longest results will be the final plate area.

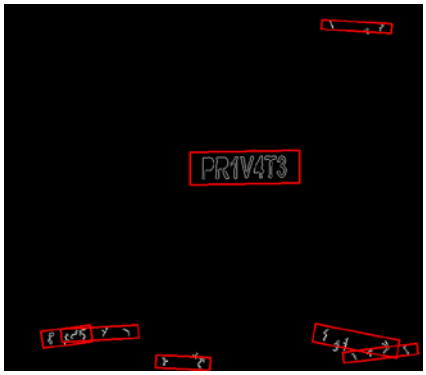
3. Examples of workflow


Original -> Grayscale ->


-> Maximized contrast -> Gaussian blurring ->

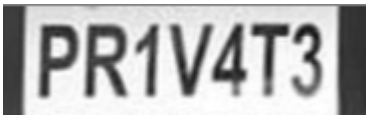

-> Adaptive threshold -> Find contour ->


-> Round 1 filter -> Round 2 filter ->



-> Extract possible plate area ->

-> Processing possible plate area ->



-> Grayscale ->

-> Maximized contrast ->

-> Gaussian blurring ->

-> Adaptive threshold ->

-> Find contour and 2 rounds filter ->

-> Character recognition ->

-> Final result

4. Generation of results

Several images form the internet have been tested:

It can be seen from the test results that the positioning accuracy of the license plate is very high, but the recognition rate of the license plate does not reach the desired result. The main error is that one of the characters will be recognized as a similar error character, such as Y and V, Z and 2.

5. Discussion and conclusion

This research successfully reproduced the license plate recognition function and achieved a very high accuracy of license plate positioning. However, the character recognition rate needs to be improved. This is mostly because the KNN training data used has insufficient feature amount and is not specially trained for license plate fonts. In addition, factors such as the size, clarity, and distance of vehicle images obtained from the internet will affect the pre-processing requirements before binarizing the images, such as whether needed to blurring, thereby affecting the final result.