Imaging & Special Effects (ISE)

A·P·U
ASIA PACIFIC UNIVERSITY
OF TECHNOLOGY & INNOVATION

License Plate Recognition System

Student Name:	1. Wahidul Alam Riyad 2. Sylvia Chan Poh Yee
TP Number:	1. TP043338 2. TP050862
Intake Code:	UC2F1811IS
Module Code:	CT029-3-2 ISE
Module Title:	Imaging & Special Effects
Assessment Title:	License Plate Recognition System
Assessment Type:	Group Work
Submission date:	23rd August 2019
Lecturer	Dr Hamam Mokayed

Table of Contents

1.0 Objective

License Plate Recognition (LPR) is an image processing technology which uses number (license) plate to identify the vehicle. The objective is to design an efficient automatic authorized vehicle identification system by using the vehicle number plate. The system is implemented on the entrance for security control of a highly restricted area like military zones or area around top government offices e.g. Parliament, Supreme Court etc. The developed system first detects the vehicle and then captures the vehicle image. Vehicle number plate region is extracted using the image segmentation in an image. Optical character recognition technique is used for the character recognition. The resulting data is then used to compare with the records on a database so as to come up with the specific information like the vehicles owner, place of registration, address, etc. The system is implemented and simulated in Visual Studio, and it performance is tested on real image. It is observed from the experiment that the developed system successfully detects and recognize the vehicle number plate on real images (Qadri, 2018).

2.0 Problem Statement

Automatic vehicle license plate detection and recognition is a key technique in most of traffic related applications and is an active research topic in the image processing domain. Different methods, techniques and algorithms have been developed for license plate detection and recognitions. Approach: Due to the varying characteristics of the license plate from country to country like numbering system, colours, language of characters, style (font) and sizes of license plate, further research is still needed in this area (Maglad, 2016).

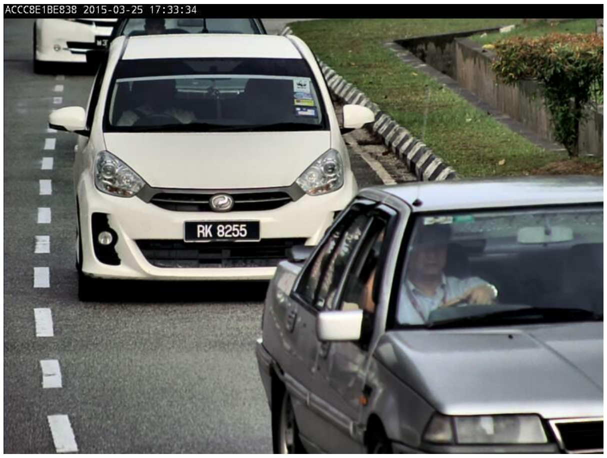


Figure 1 shows License Plate of Malaysia

3.0 Proposed Algorithm

3.1 Greyscale

The reason that converting the colour image which is RGB (Red, Green and Blue) image into a greyscale image is basically to convert every single pixel of the image with a specific width and height of the image. With the converted greyscale image, the programmer able to determine the characters and the numbers that contain on the license plate with same row and the horizontal intensity changes instantaneously.

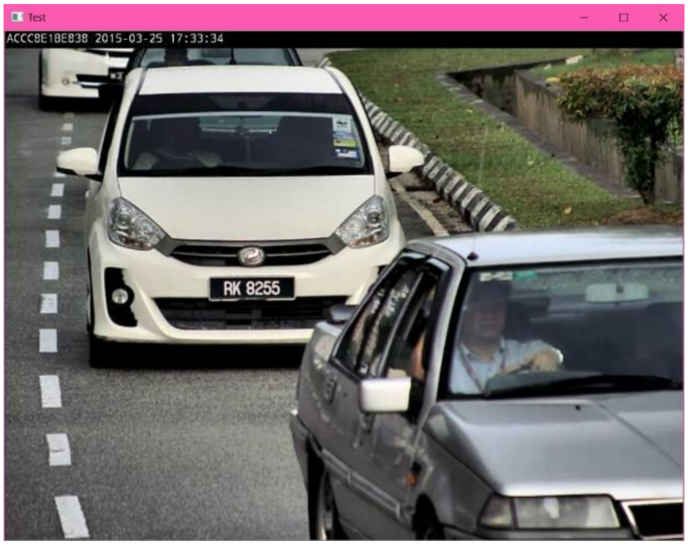


Figure 2 shows the RGB Image

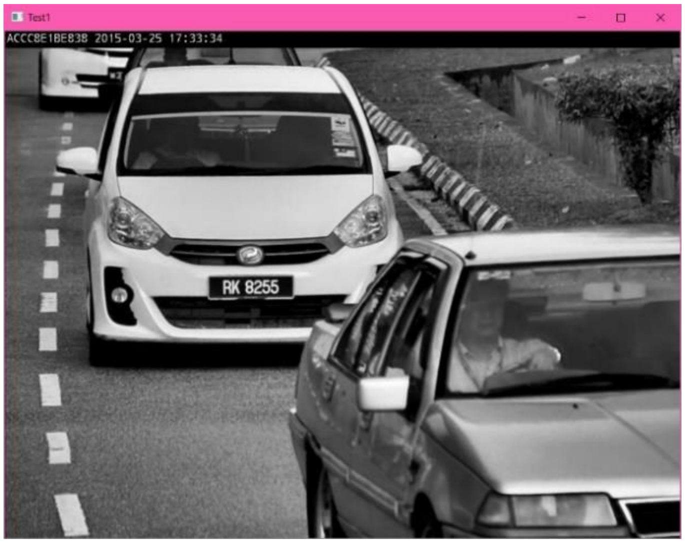


Figure 3 shows the Greyscale Image converted from RGB Image

3.2 Equalize Histogram

Histogram equalization is an image processing method that utilized to refine contrast in the image. It achieves this by successfully distributing out the most frequent intensity values, for example, extending the intensity scope of the image. This technique normally expands the universal contrast of images when its functioning data is signified by the close contrast values. This takes into consideration area of lower local contrast to pick up a higher contrast. (Sudhakar, 2017)

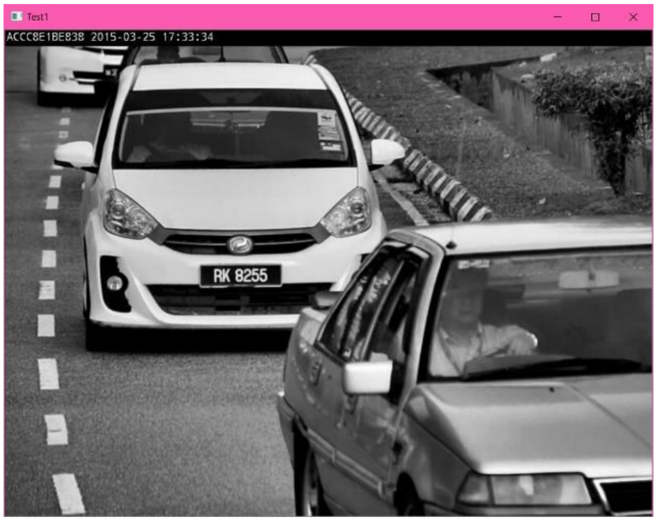


Figure 4 shows the Greyscale Image before applying Equalize histogram

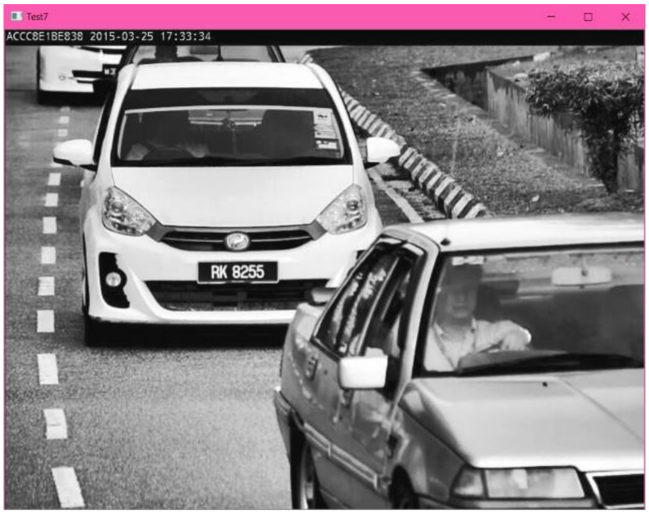


Figure 5 shows the image after applying Equalize histogram

3.3 Blur Image

Blur function is generally giving the smoothening effect to the image to reduce the noise of the image. For instance, a chunk of pixel in the image is selected to be multiply by its equivalent value of the pixel. The centre pixel value is the sum of all the pixel values and divided by the size of the chuck of pixel. In here, the programmers used 3*3 to get the result shown below:

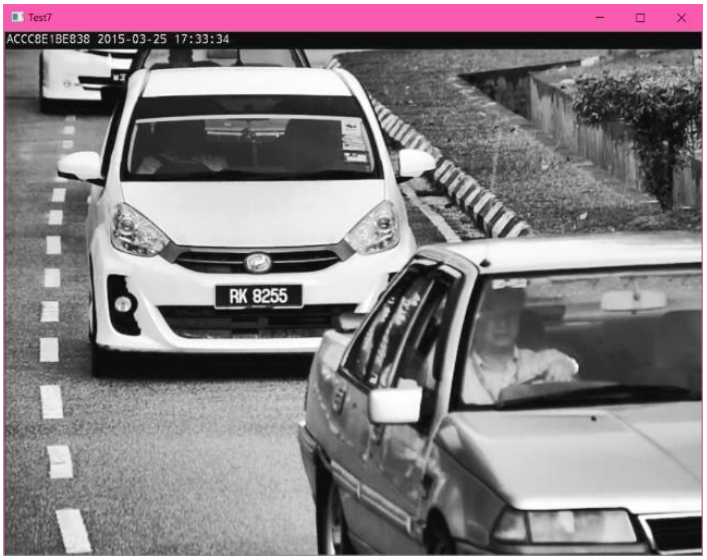


Figure 6 shows the image before Blur

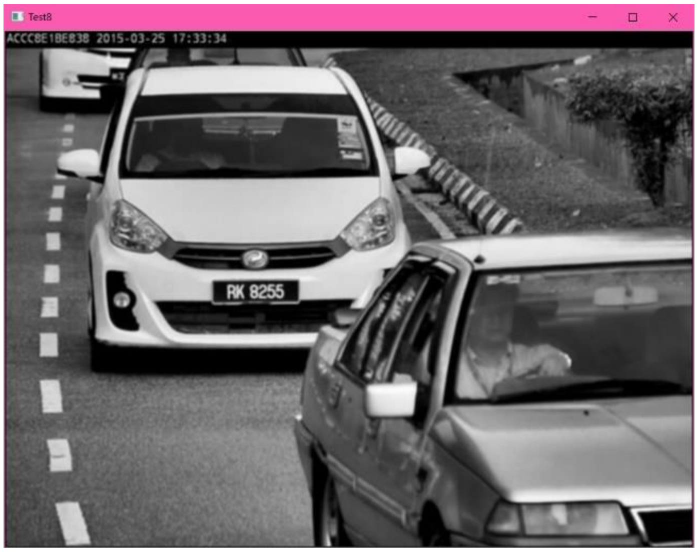


Figure 7 shows the image after Blur

3.4 Edge Detection

The greater part of the shape of a picture is surrounded in edges. First, identify these edges in an image and by utilizing the filters and afterward by upgrading those regions of image which contains edges, sharpness of the image will increase and will becoming clearer. The programmers had applied the specific formula to calculate the average of right and left as well as binarize the image if the both edge average is more than 50.

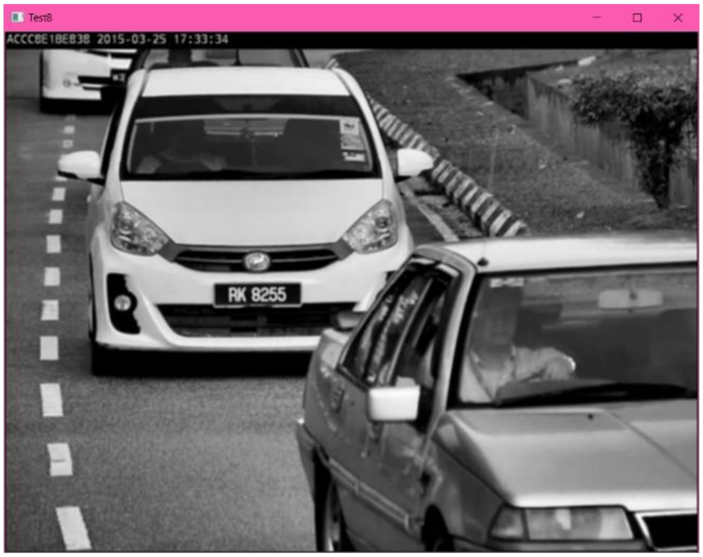


Figure 8 shows the image before Edge Detection

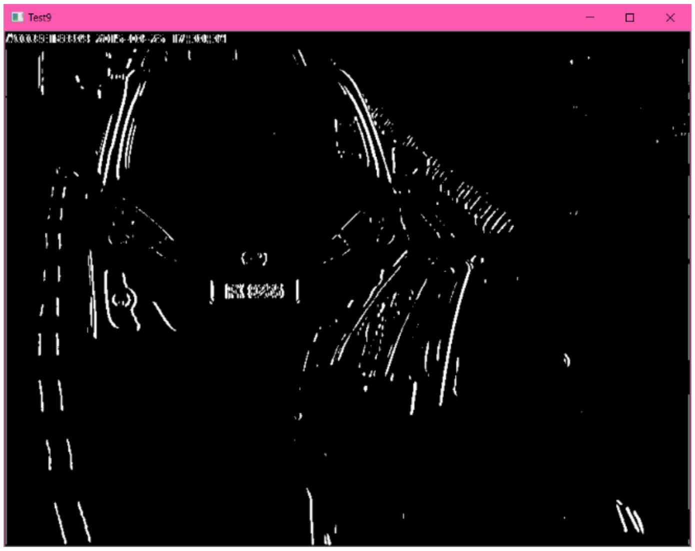


Figure 9 shows the image after Edge Detection

3.5 Dilation

The purpose of implemented dilation is to enrich the edge of the binary image based on the image of edge detection. The range of the bright section will be expanding more as every single pixel has been changed by the maximum value. For instance, the dimensions of bright section will be expanding while the size of dark section reduce. This technique is utilized to compound the number of license plate together as a segment.

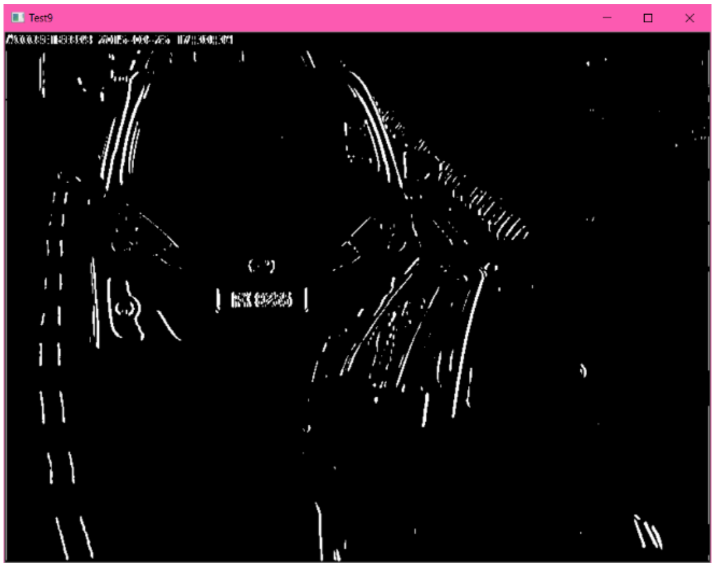


Figure 10 shows the image before Dilation

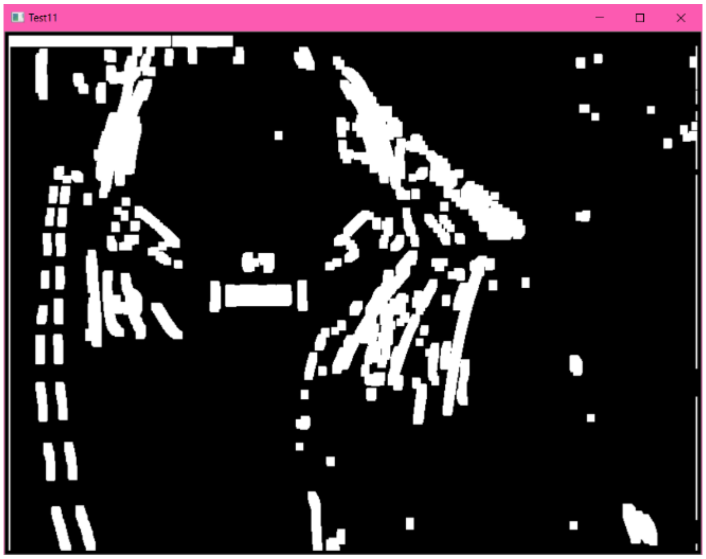


Figure 11 shows the image after Dilation

3.6 Erosion

The reason that the programmer implemented erosion is to decrease the pixel on the image with the binarized image. The quantity of pixels eliminate is depends on the resolution of image. The aim to utilize erosion on the license plate is to cut the size of characters and numbers on the image so it can be separate. Other than that, it is also restrict from the image to stay separately with other redundant fragments after dilation which can cause it filtered through the filtration procedure. Along these processes, the value of the yield pixel is the lowest value of the neighbour pixel value.

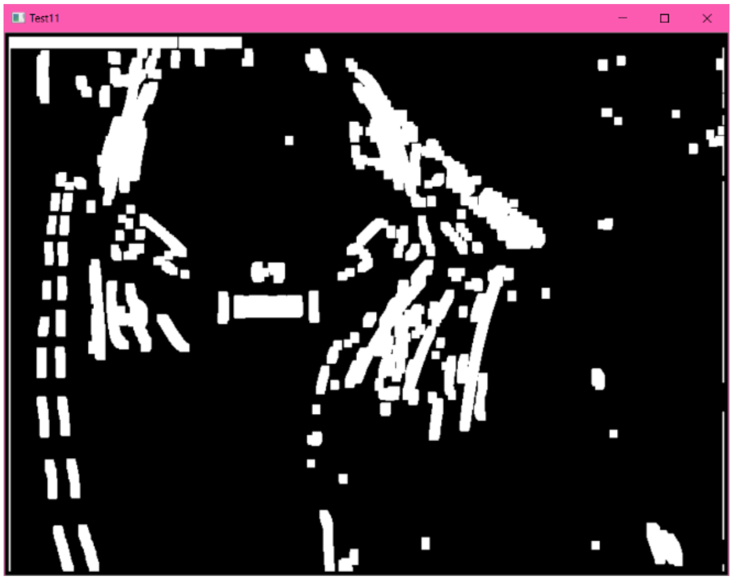


Figure 12 shows the image before Erosion

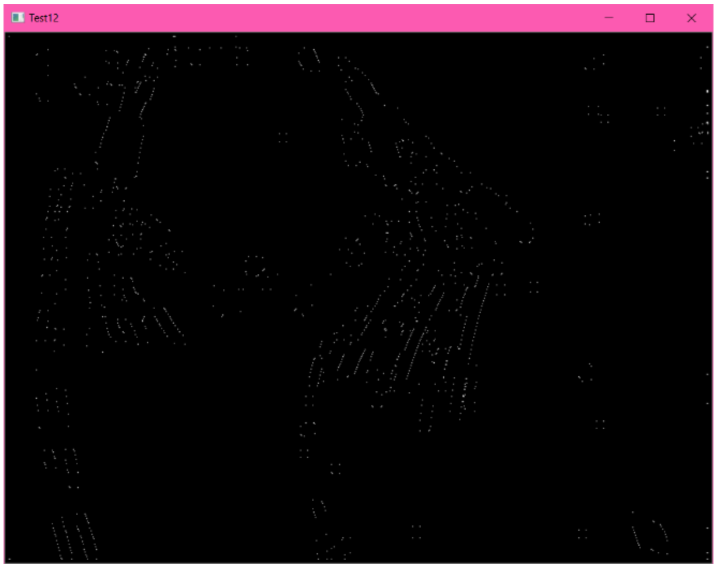


Figure 13 shows the image after Erosion

3.7 Image Segmentation

Every single entity in the image will be emphasized with a lot of different colour to section every single entity in the image. Hence, the programmer had be able to section every single number and character on the license plate based on the erosion image. Image segmentation is generally the progression of isolating the image into several entity or fragment.

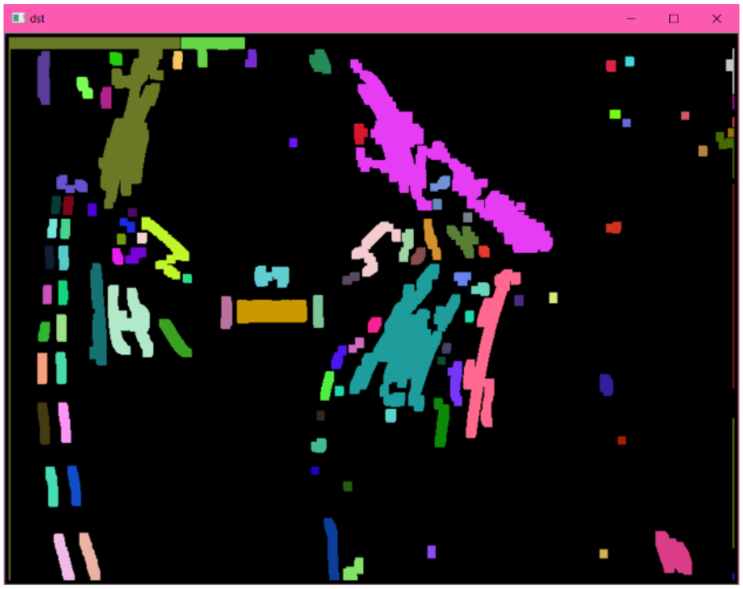


Figure 14 shows the result of Image Segmentation

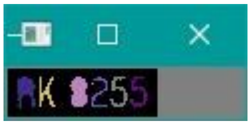
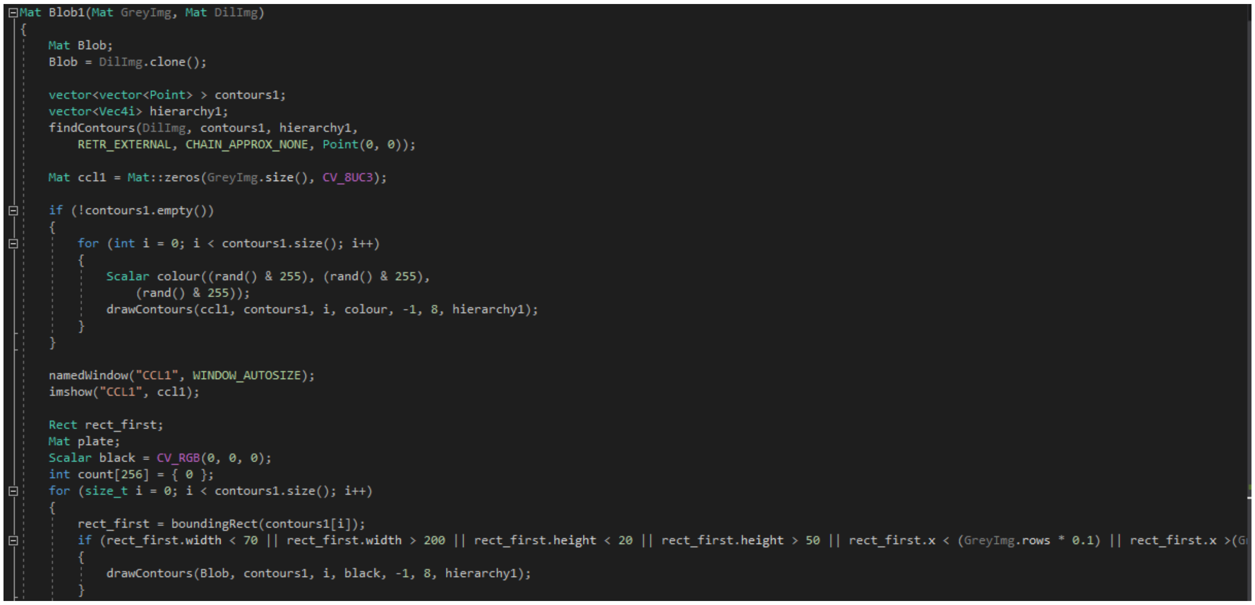


Figure 15 shows the result of license plate implemented Image Segmentation

3.8 Noise Filtration

The reason of utilizing noise filtration is to filter or remove the shape for the license plate on the image and only take the last yield of character and number on the image. The noise of the image has been triggered by the corrupted pixel which is needed to be filtered. The condition of filter will be based on width, height, x-axis, y-axis, as well as density.



```
Mat Blob1(Mat GreyImg, Mat DilImg)
{
    Mat Blob;
    Blob = DilImg.clone();

    vector<vector<Point> > contours1;
    vector<Vec4i> hierarchy1;
    findContours(DilImg, contours1, hierarchy1,
        RETR_EXTERNAL, CHAIN_APPROX_NONE, Point(0, 0));

    Mat ccl1 = Mat::zeros(GreyImg.size(), CV_8UC3);

    if (!contours1.empty())
    {
        for (int i = 0; i < contours1.size(); i++)
        {
            Scalar colour((rand() & 255), (rand() & 255),
                (rand() & 255));
            drawContours(ccl1, contours1, i, colour, -1, 8, hierarchy1);
        }
    }

    namedWindow("CCL1", WINDOW_AUTOSIZE);
    imshow("CCL1", ccl1);

    Rect rect_first;
    Mat plate;
    Scalar black = CV_RGB(0, 0, 0);
    int count[256] = { 0 };
    for (size_t i = 0; i < contours1.size(); i++)
    {
        rect_first = boundingRect(contours1[i]);
        if (rect_first.width < 70 || rect_first.width > 200 || rect_first.height < 20 || rect_first.height > 50 || rect_first.x < (GreyImg.rows * 0.1) || rect_first.x >(G
        {
            drawContours(Blob, contours1, i, black, -1, 8, hierarchy1);
        }
```

Figure 16 shows part of the Noise Filter Algorithm

The prime state of the calculation utilized the average of width and height of the license plate to first filter. Every one of the fragments that exceed or lower than the range will be filtered. The subsequent condition filters the fragments at the outskirt of the license plate (10% the entire image height, width y and 90% the entire image height, width). The final condition is the thickness of the images. The thickness can be utilized to decide if the images is truly fit in something.

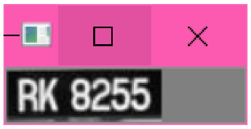
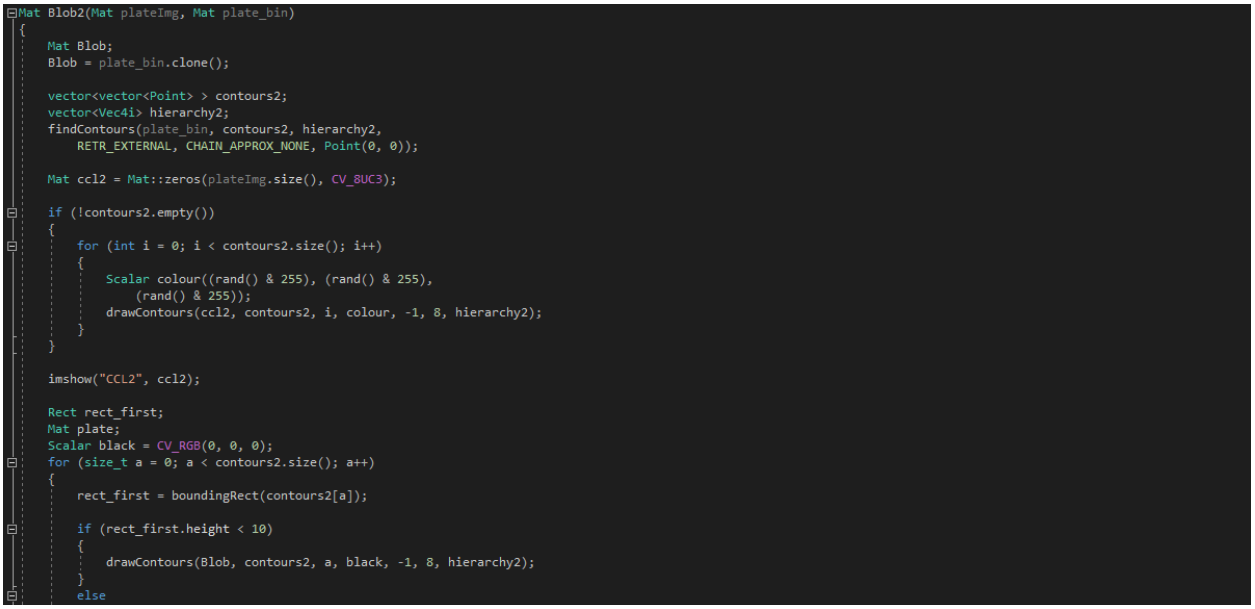


Figure 17 shows the result of license plate implemented Noise Filtration

The entire algorithm is essentially founded on the average width and thickness of the character. In the event that the characters are gathering together, at that point the license plate need to dissolve more so as to part them. On the off chance that the forms thickness is high, this mean the thickness of the character is excessively fat and furthermore need to disintegrate more than before so as to part them successfully.



```
Mat Blob2(Mat plateImg, Mat plate_bin)
{
    Mat Blob;
    Blob = plate_bin.clone();

    vector<vector<Point> > contours2;
    vector<Vec4i> hierarchy2;
    findContours(plate_bin, contours2, hierarchy2,
        RETR_EXTERNAL, CHAIN_APPROX_NONE, Point(0, 0));

    Mat ccl2 = Mat::zeros(plateImg.size(), CV_8UC3);

    if (!contours2.empty())
    {
        for (int i = 0; i < contours2.size(); i++)
        {
            Scalar colour((rand() & 255), (rand() & 255),
                (rand() & 255));
            drawContours(ccl2, contours2, i, colour, -1, 8, hierarchy2);
        }
    }

    imshow("CCL2", ccl2);

    Rect rect_first;
    Mat plate;
    Scalar black = CV_RGB(0, 0, 0);
    for (size_t a = 0; a < contours2.size(); a++)
    {
        rect_first = boundingRect(contours2[a]);

        if (rect_first.height < 10)
        {
            drawContours(Blob, contours2, a, black, -1, 8, hierarchy2);
        }
        else
```

Figure 18 shows the part of Noise Filtration implemented

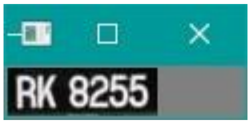


Figure 19 shows the result before Noise Filtration

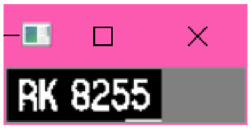


Figure 20 shows the result after Noise Filtration

3.9 OTSU

The purpose of OTSU formula is to compute the threshold value of pixel in forefront and background. In the wake of getting both side of threshold value, the threshold value will be binarize to acquire a uniform histogram of threshold value.

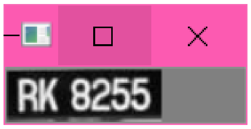


Figure 21 shows the image before OTSU

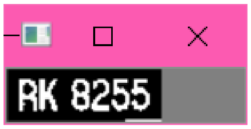


Figure 22 shows the image after OTSU

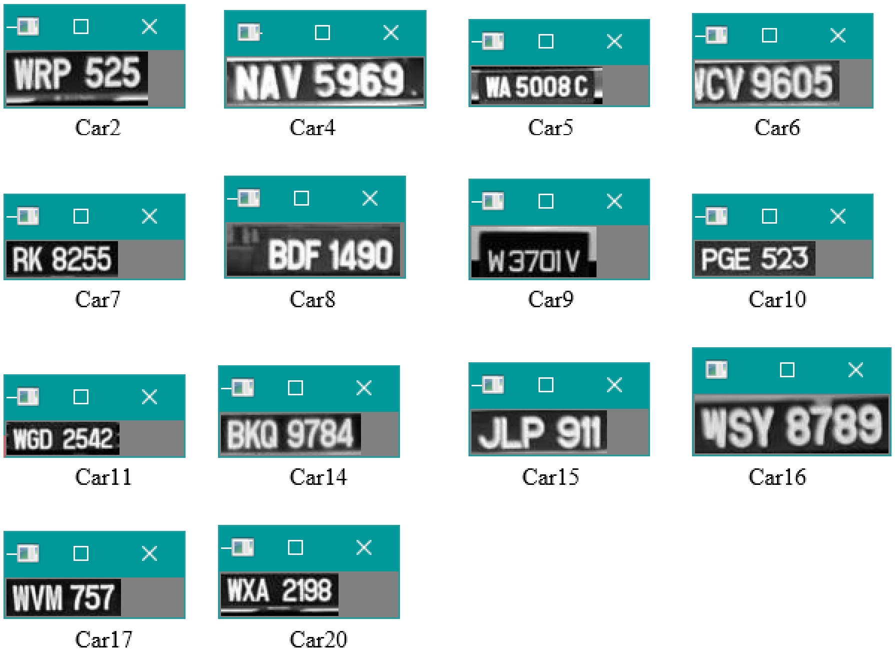
3.10 OCR

OCR known as Optical Character Recognition. The OCR that the group is utilizing is Tesseract OCR with incorporation of API to visual studio. The license plate that the group has done will be embedded to tesseract OCR motor to perceive the character and number. The yield of the character and number will be show in the console based. The tesseract OCR motor comprise of a system where it contains preparing information and library that should be arrange and fabricate utilizing vc package.

3.11 Experimental Results

The License Plate Recognition system had detected 14/20 license plate images and successfully recognized 13 out of 14 images effectively. There are total 8 images show the recognized characters are separating, showing the logo of the vehicle or misconstrue one or two characters.

3.11.1 Detected Car Images



Car2 Car4 Car5 Car6

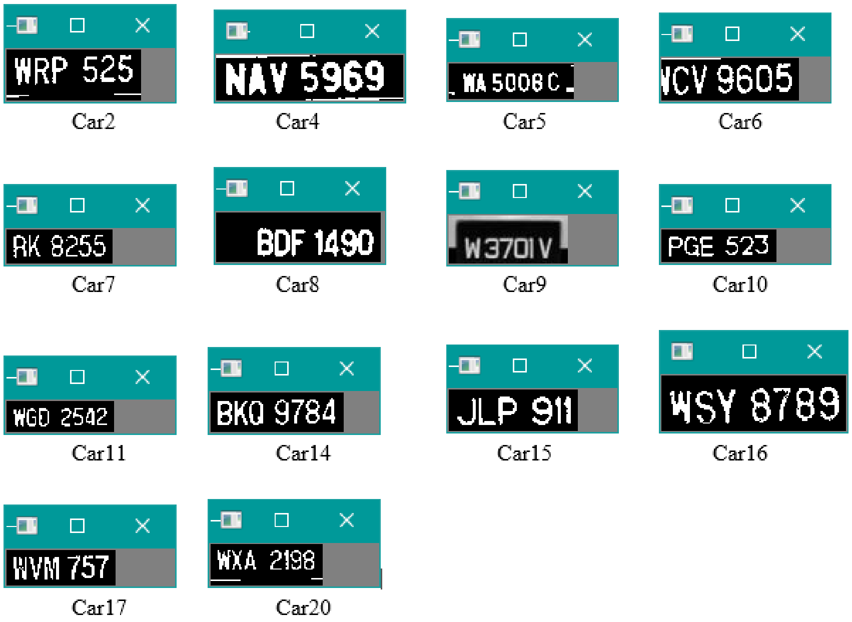
Car7 Car8 Car9 Car10

Car11 Car14 Car15 Car16

Car17 Car20

Figure 23 shows the license plates successfully detected

3.11.2 Noise Filtered License Plate using Tesseract OCR



Car2 Car4 Car5 Car6

Car7 Car8 Car9 Car10

Car11 Car14 Car15 Car16

Car17 Car20

Figure 24 shows the result of the license plates after suing Tesseract OCR

3.11.3 Results of Tesseract OCR

WRP525 – 25P5WR
NAV5969 – 69V59NA
WA5008C – CWA5008
WCV9605 – 5V960VC
RK8255 – RK8255
BDF1490 – BDF1490
W3701V – W3701V
PGE523 – 3PGE52
WGD2542 – WGD2542
BKQ9784 – 4978BKQ
JLP911 – P911JL
WSY8789 – 9878YWS
WVM757 – 757WVM
WXA2198 – WXA2198

4.0 Critical Analysis

4.1 License Plate Detection

There are two reasons why the license plate cannot be recognized 100%. The first image that can't be distinguished is a result of the license plate portion is excessively circulated during edge discovery. It is too dark for the image, even the dilation and erosion been tested so many times but still not getting the result. The dilation value which connected for all license plate image is not sufficient to make the sections associated. The subsequent image adheres to the background portions which make it such a major fragment and neglect to go through the license plate filtration process. For the license plate with two rolls cannot be detected by the LPR system, even the results were not showing in the console, the programmers had changed the value on the width, height, x-axis and y-axis several times, but it still remain nothing.

4.2 License Plate Recognition

While the fact that there are still a few characters in the images are misread or missing, however on the off chance that the programmer can keep on training the framework with the information, the Tesseract should ready to recognize it all the more precisely and reliably. Other than that, there is one image which additionally could not fit in the given algorithm. The result of this image (car 17) is cutting the character separately with the character of “M” is cutting in half. The image (car 12 and car 19) show the license plate is blur, it only can detect its car logo. In this manner, the filtration technique ought to alter like point of confinement the thickness of the character and the edge detection should be improve as well as check again whether the characters are staying together or not and pick the reasonable erosion value based on the outcomes.

4.0 Conclusion

In conclusion, the programmers had implemented the algorithms of greyscale, equalize histogram, convert to blur image, edge detection, dilation, erosion, image segmentation, noise filtration, OTSU and OCR to recognize the license plate from the image. They had fulfilled the requirement as the given assignment question, and they had learnt to implement all the algorithms to get the results that has been written down in the report.

6.0 Workload Matrix

TASK ID	TASK NAME	Wahid TP043338	Sylvia TP050862
1	TABLE OF CONTENTS	50%	50%
2	OBJECTIVE	50%	50%
3	PROBLEM STATEMENT	50%	50%
4	PROPOSED ALGORITHM	50%	50%
5	CRITICAL ANALYSIS	50%	50%
6	CONCLUSION	50%	50%
7	REFERENCES	50%	50%
	SIGNATURE		

References

Maglad, K. W., 2016. A Vehicle License Plate Detection and Recognition System. *Journal of Computer Science,* 8(https://www.researchgate.net/publication/290348089_A_Vehicle_License_Plate_Detection_and_Recognition_System), pp. 301-3015.

Qadri, M. T., 2018. Automatic Number Plate Recognition System for Vehicle Identification Using Optical Character Recognition. *IEE Explorer,* Issue https://ieeexplore.ieee.org/document/5169511.

Sudhakar, S., 2017. *Histogram Equalization.* [Online]
Available at: https://towardsdatascience.com/histogram-equalization-5d1013626e64
[Accessed 1 August 2019].