

CO543 : IMAGE PROCESSING
FINAL PROJECT EVALUATION REPORT
Group F

TITLE :

Sri Lankan Vehicle Number Plate Recognition System

AUTHORS :

E/17/212 : MORAIS K.W.G.A.N.D.

E/17/219 : NAWARATHNA K.G.I.S.

E/17/284 : RATHNAYAKA R.L.D.A.S.

ABSTRACT

In this study, we have proposed a method for recognition of vehicle license plates. The objective is to design an efficient recognition system for distorted and blurred number plates. Number plate recognition usually contains three steps, namely license plate detection/localization, character segmentation and character recognition. When reading characters on a license plate one by one after the license plate detection step, it is crucial to accurately segment the character. In particular, character segmentation becomes difficult as the acquired vehicle images are seriously degraded. We have used pytesseract in python as optical character recognition to recognize the extracted characters. Therefore, it is not necessary to do character segmentation. There are many methods for character segmentation and recognition, including advanced and complex deep learning algorithms. We have used a simple approach. The plate area is further pre-processed before extracting the characters. The blurry and noisy images of number plates are enhanced by applying preprocessing techniques, such as resizing, removing the noise of the images and filtering to reduce the blurriness of the images and to detect the edges. This system is developed based on digital images and can be easily applied to commercial car park systems for the use of documenting access of parking services, secure usage of parking houses and also to prevent car theft issues.

INTRODUCTION

Vehicle license plate recognition is frequently used for parking management and speed limit enforcement systems. The closed-circuit television (CCTV) cameras installed on roads or buildings are used to store and transmit videos and images. However, the data acquired through this system is not being fully utilized. It requires much time and human resources to properly utilize and analyze the images captured by CCTV, which becomes a limiting factor in quick response to an accident or crime. These images are distorted due to several reasons such as a limitation of the resolution, motion blur, extensive lighting conditions and perspective distortion due to various environmental changes and camera installation

locations. It is desired to have an image processing technology that can accurately recognize the characters smaller than the image size.

The proposed system is taking a distorted image of the number plate and then returning the vehicle number in the terminal. Therefore, There are two main stages of our project.

1. Preprocessing the distorted image
2. Character recognition and return to the terminal

Initially the image is taken as the command line argument and read it. Then convert it into a grayscale image. Afterwards, suitable filters are applied to reveal the characters on the license plate and edges of the number plate is detected. This all tasks are done for extracting the number plate and it is moved to the character recognition part. Number plate recognition uses optical character recognition (OCR) on images to recognize the number plates of vehicles. After finding the characters, validation of this number is checked and the vehicle number is printed on the terminal.

This is the overall plan of our project.

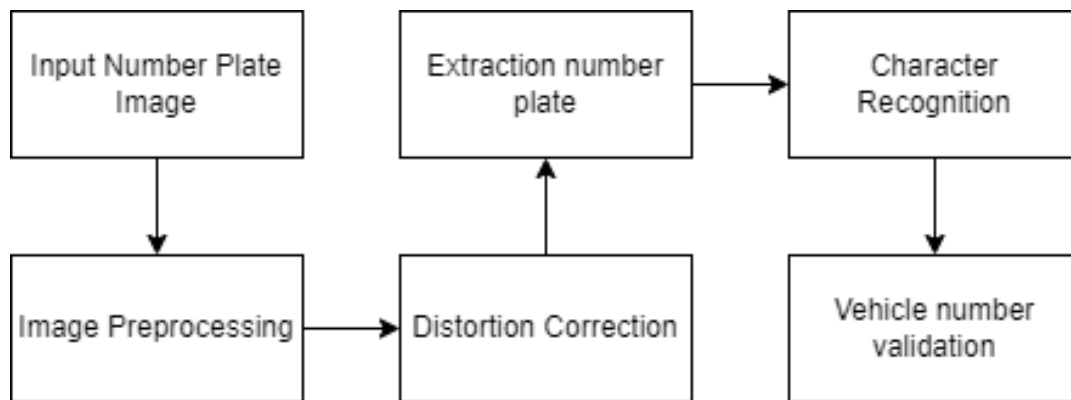


Figure 1 : Number plate recognition system

BACKGROUND

The development of current life has prompted an expansion in the number of road vehicles. This has led to a phenomenal level in civil problems being vehicle identification and traffic control. Because of the rising test of traffic lights, there emerges a need to manage traffic better. The recognition of the number plate of vehicles is a significant part of the Intelligent Transportation System (ITS). ITS was created to execute data and correspondence innovations in the field of street transport, some of which incorporate vehicle the executives, traffic the board as well as connecting with different frameworks of transportation. A lot of researchers have tried to develop ITS systems with different levels of success stories. We are designing a system to recognize plate numbers using Python and OpenCV. The plan was carried out to perceive the characters in the number plate after applying to the program. The framework utilized the python programming and picture handling calculations in the OpenCV library to perceive the picture. It was also used in preprocessing to resize the image, perform error checking, and convert image to grayscale. Threshold algorithms were used to binarize the image to retain the quality of image.

Various techniques have been proposed in the literature on vehicle license plate recognition. Plate localization is the first step of these methods. A color based technique is used to extract plate regions from an image in a Sri Lankan context.

APPROACH

1. Image Acquisition.

Input the vehicle number plate images

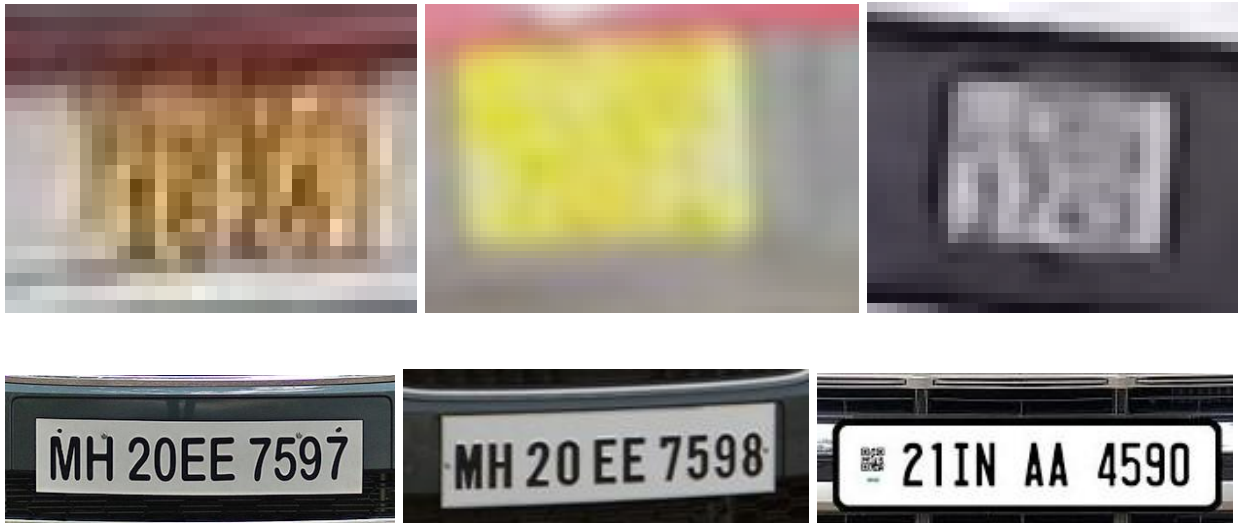


Figure 2 : Input number plate images

- Initially the image is read using cv2.imread() function

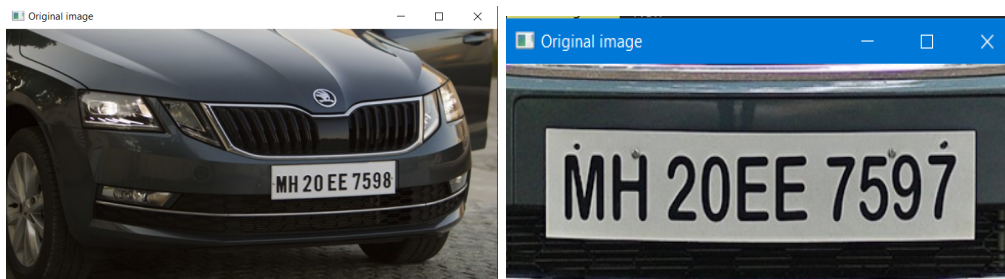


Figure 3 : input images

2. Character recognition.

- Then the image was converted to grayscale using the `cv2.bilateralFilter()`

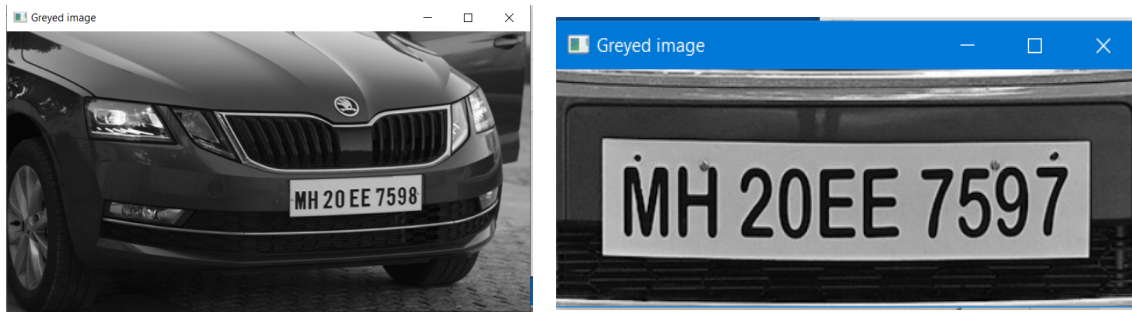


Figure 4 : Grayscale images

- Next the edges were detected using the `cv2.Canny()` filter.

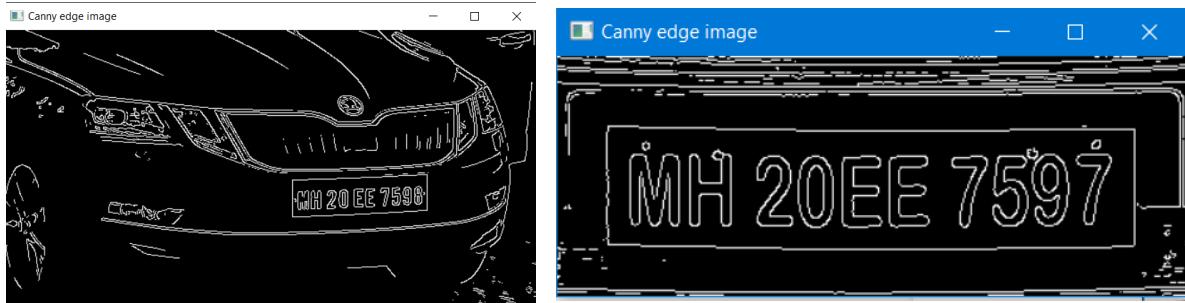


Figure 5 : Canny edge images

Number plate recognition

- After the preprocess the contours (the curve joining all the continuous points) were detected from the `cv2.fondContours()`. To obtain a higher accuracy when finding the contours, binary images were used. For that in the pre process the Canny edge detection filter was used.

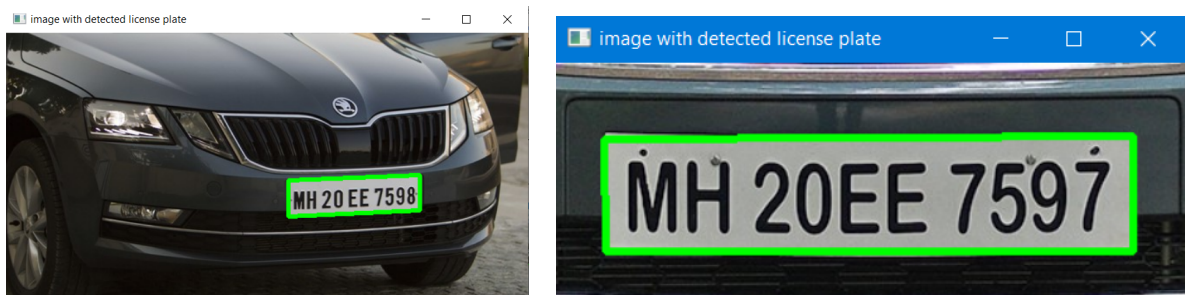


Figure 6 : Image with detected license plate

- Then the contours were sorted in reverse direction, using the `contourAres` as the key, to get the best contour. Because the best contour is the License plate.



Figure 7 : Extracted Number plate images

- After sorting the contours, they were saved to a directory as .png files. Then the best contour which has the highest file name according to the naming, used to read the license plate number.
- To read the License plate number, the selected file name is given as the input to the pytesseract, which is used to read the characters from the images.
- After reading the License plate the validity of the license plate was checked.

Check the validity of the Recognized number

- After recognizing the characters on the number plate, the validity of the License Number is checked. In that case, the main number of characters on the License number except provincial code, should be between seven and five, last four characters should be numeric characters, if the first part of character are numeric the number should be smaller than or equal 325, if the first part of characters are English letters, for three-character numbers and for two-character numbers there are specific patterns. They were checked at the last stage and displayed the validity of the License number.
- The **“Sri Lankan Vehicle Number Plate Recognition System”** is specially developed for recognizing the vehicle License plates in Sri Lanka. Therefore as the final step it checks whether the recognized License plate is valid in Sri Lanka.
 - For the License plate number : ND - 4957 (A valid number in Sri Lanka)

```
#####
Sri Lankan Vehicle Number Plate Recognition System
#####

Vehicle Licence Plate: ND - 4957

-----
The validity
=====
Number is valid
```

Figure 8 : Validation results 1

For the License plate number : MH 20 EE 7598 (A invalid number in Sri Lanka)


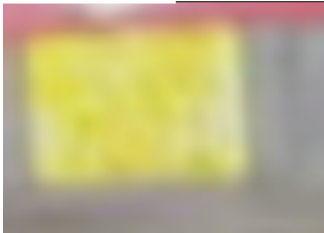




```
#####  
Sri Lankan Vehicle Number Plate Recognition System  
#####  
  
Vehicle Licence Plate: MH 20 EE 7598  
  
-----  
The validity  
-----  
Number is not Valid in Sri Lanka
```

EXPERIMENT

- First a series of experiments were performed to get the best enhancement for the image in the preprocessing steps. Below are the steps of preprocessing experiments.

1. Converting the image into a gray scaled image

This is done in order to increase the readability of the image. When there are pixels in the image that have rgb values, the algorithm might not extract the important features of the image due to various patterns generated by the rgb values. Instead, if the image is gray scaled the pattern recognition algorithm can identify the characters in the image in a much more detailed manner.

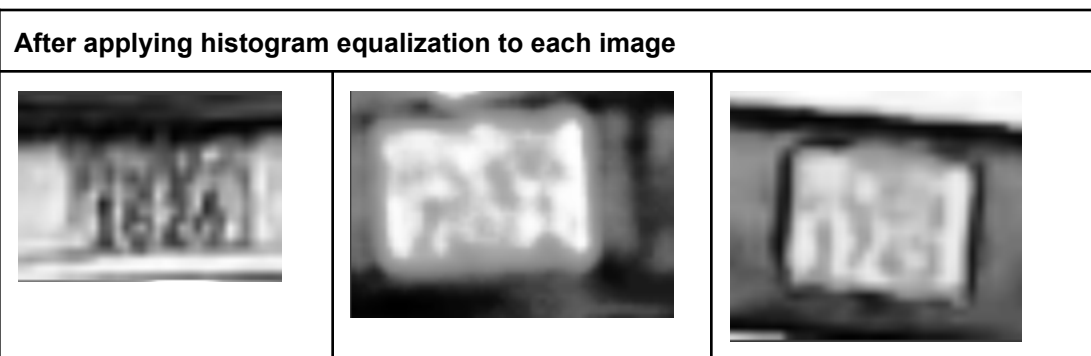
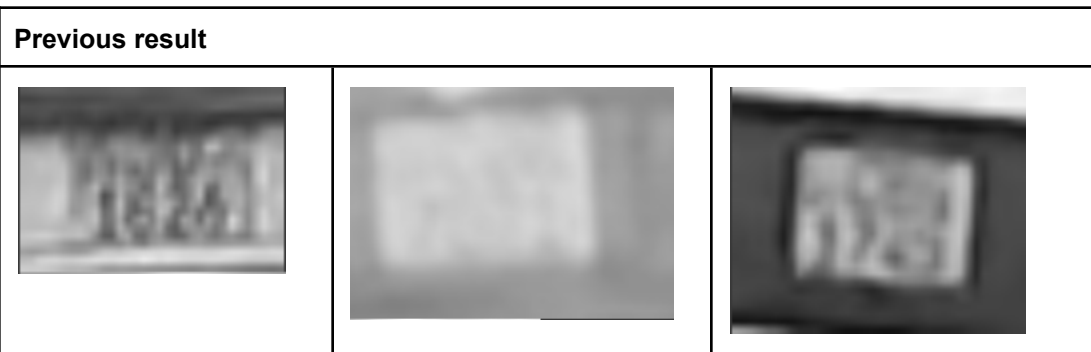
Original Image		
		
After making it gray scaled		
		

-We can see the first image and the third image has already started to show some improvements.

2. Applying the histogram equalizations to the image

One of the most obvious observations that can be made after observing the images is that the contrast of the images is not in a healthy manner. This will make the images so unreadable. To improve the contrast of each image, one of the best methods is to use image equalization.

-This is done via histogram equalization. Since the spread of the original image is not that good, histogram equalization extends the spread of the histogram and that will improve the image readability.



- We can see the contrast of each image is much improved. The second image also has a noticeable result.

3. Applying transformations

- The image needs to be enhanced more because it is clear that the image is not still in very unreadable format. Therefore to further enhance the image we can use transformations. Two of the best transformations are,

1. Power Transformation.
2. Log Transformation.

-Experimentation.









- There were two experiments done in order to select the best method.

1. Which is the best transformation to apply first? Is it the power transformation or log transformation?
2. What is the best value for the gamma value?









Below steps were carried out to find answers to these questions.

- Steps.

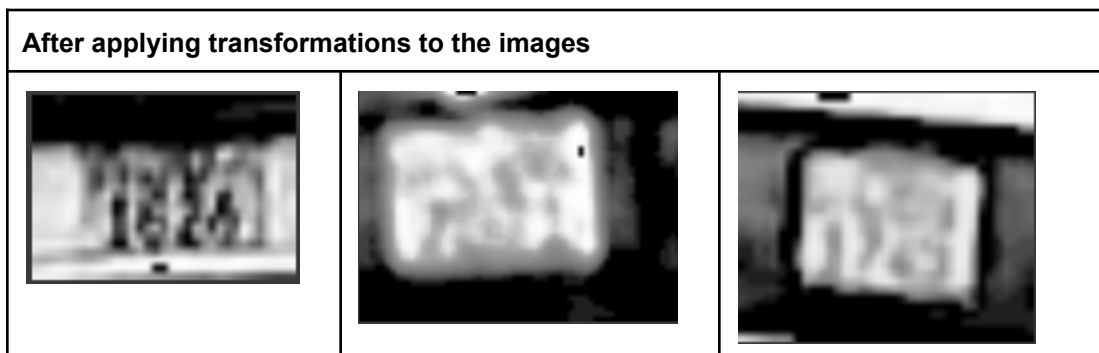
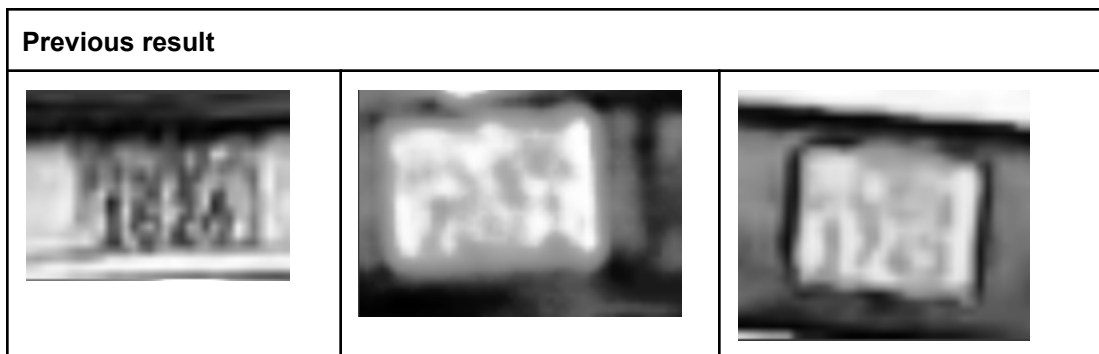
1. First the log transformation was applied to the output of the previous step.
2. Then as for the first option, log transform was applied to the image then power transform was applied.
3. The power transform was applied using different gamma values. Depending on that the best alpha value can be chosen. Below are the results.

Gamma	0.2	0.5	4	6
Log Transformation				
Power Transformation				

4. Below are the results when power transformation first and the log transformation next method.

Gamma	0.2	0.5	4	6
Power Transformation				
Log Transformation				

- Observation of the results :
- We can see that the second method is the most suitable method given that the characters on the number plate are much more readable when using the 2nd option which is power transformation first and then the log transformation.
- Also in the second method we can see if we set gamma to be 4 the image is not dark enough and also if gamma is 6 the image is darker than it should be. Therefore the gamma = 5 chosen as the final value.
- Now we can compare the images of the previous step with the most updated ones.



- We can see that the images are much more improved than they were before. However the 2nd image state is still not that good.

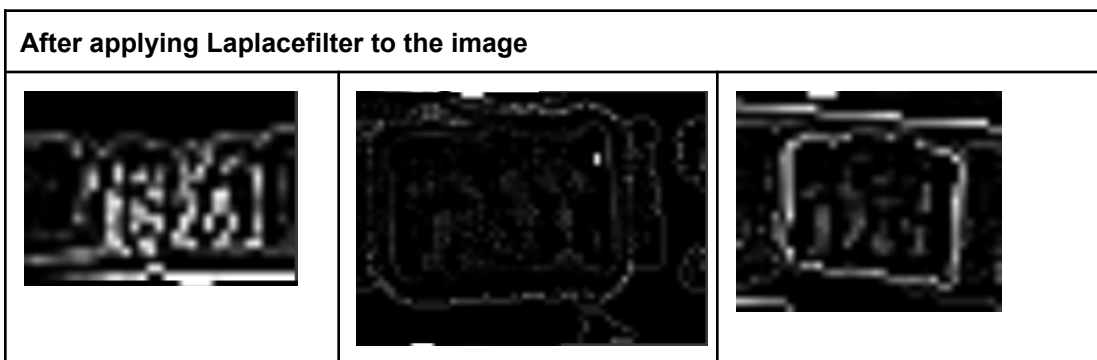
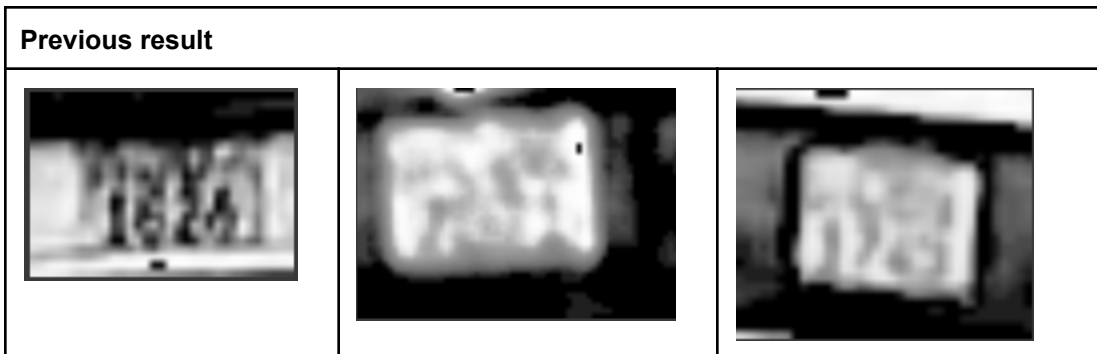
4. Apply filters to improve the image readability

The next step is to add a filter to detect edges in the image. For this Laplace and Sobel Filters are used.

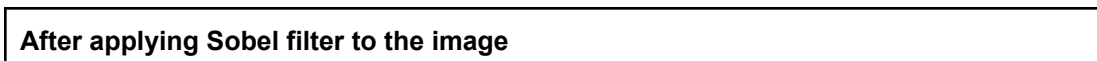
It is also important to notice that the images still do not show the character information properly. Therefore applying the filters might make the image more unreadable. However it is important to detect edges in the images therefore applying those filters is a necessary step. This step is also carried out as an experiment to see if the edge detection is done correctly on the images.

Experiment.

1. Apply the Laplacian Filter to the previous image outputs.



2. We can see the image edge detection does not behave as intended. Let's use the Sobel filter on top of this.





It can be seen that the images are not in a good shape. Therefore skipping this step can also be considered.

There were a lot of experiments to restore the number plate images that were distorted badly.

5. Read the characters from the license plate

To extract the characters from the recognized License plate, initially it was required to find the contours. In that case due to defects and certain distortions on the images, sometimes it was unable to find the correct contours.



Figure 9 : Eros in Contour Detection

Then the different image processing techniques in the spatial and the frequency domain applied again on the images to enhance the features more.

RESULT

Input Image	Result
	<pre>##### Sri Lankan Vehicle Number Plate Recognition System ##### Vehicle Licence Plate: ND - 4957 ----- The validity ===== Number is valid</pre>



CONCLUSION

We have learned how image processing relates to real world applications. Therefore, we can use image processing techniques to solve efficient solutions for day to day problems. We learnt some basic filters to reduce the noise of an image and OCR methods to extract the data on an image.

There is also so much future work to do this project. We are going to build a real time mobile application to capture vehicle number plates and recognize the characters on them. According to the vehicle number, all details of the vehicle are fetched from the Sri Lanka Motor Traffic department and presented.

REFERENCES

1. https://www.nsbm.ac.lk/wp-content/uploads/2021/08/ICOB1_2018_Vehicle-License-Plate-Recognition-System.pdf
2. <https://ieeexplore.ieee.org/document/5169511>
3. <https://www.ijert.org/research/number-plate-recognition-using-segmentation-IJERTV1IS9020.pdf>