CO543 : Image Processing

Mid Project Evaluation Report

TITLE :

Sri Lankan Vehicle Number Plate Recognition System

GROUP F

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.

INTRODUCTION

The number of vehicles is increasing day by day in our life. The vehicle may violate traffic rules and cause accidents. Therefore, the number plate recognition system plays a significant role in identifying these vehicles. But the problem is, inability to clearly identify the vehicle number using the captured photos. Because mostly that photos are captured when the vehicles are moving. Then the photos can be not focused well and can be blurred. Also due to different weather and lighting conditions the captured images can include different noises.

Number plate recognition uses optical character recognition (OCR) on images to recognize the number plates of vehicles. Initially, the blurry 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. After that the characters in the number plate can be extracted.

PROBLEM STATEMENT

Problem :

- In real world situations, number plates of vehicles will be captured automatically by CCTV cameras and cameras that are carried by police officers.

- These images are distorted due to several reasons. Normally these types of number plates are captured from moving vehicles. Since the vehicles are moving, the captured photo will obviously generate some noise in the image and that will result in a bad , unreadable image.
- The CCTV cameras which produce these types of images often do not have a good sensor to capture quality images hence from that factor also produce bad images.
- Other than that, environmental conditions such as raining, shining, low light situations, high exposed lighting will also directly contribute to producing a bad image.
- Due to these factors, more often than not number plate images will not be in good shape. Therefore, the problem of this project can be separated into two main parts.

1. Preprocessing the image such that the content in the number plate is visible to the naked eye.
2. Then applying techniques to convert the image content into a string(ASCII code).

Dataset :

- Dataset to be used : All the images of the given test cases will be used to test and evaluate the performance of the system.

Results :

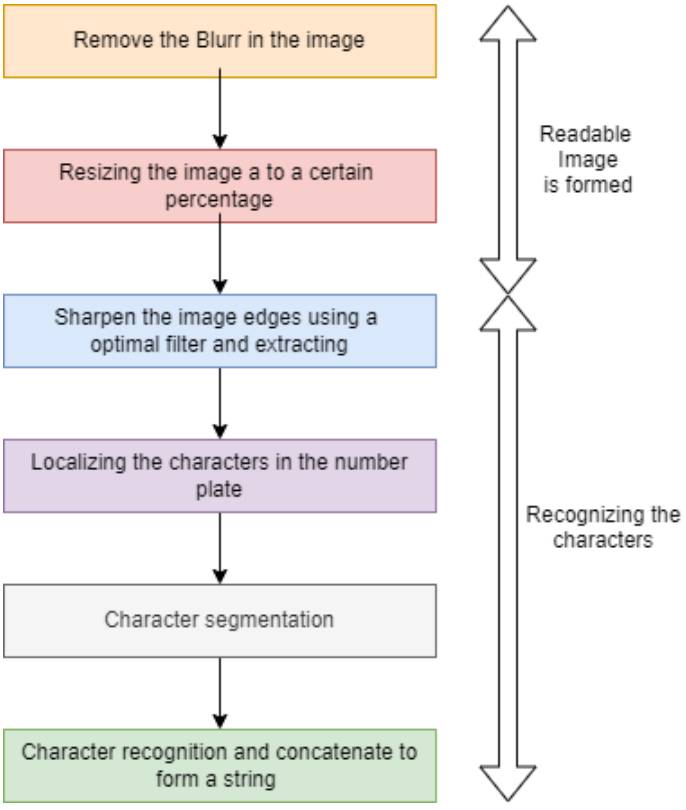
- Intermediate output : Image of a number plate that has a readable content in it.
- The output of the system is to identify the characters in the number plate accurately.

Evaluation:

- After image preprocessing is done, the blur will be removed, and the characters inside the number plate can be clearly visible to the naked eye.
- Then the images of number plates will be inputted to the system, and then the output of the system will be compared with the actual images of number plates and we can determine whether the number plate is clearly identified by the system or not.
- Also libraries such as Pytesseract to test the system, will be considered.

TECHNICAL APPROACH

- The main approach of how to achieve the main objective is given below.



1. Remove the blur in the image

-For deblurring the image, the following equation would be used.

$$g(m,n) = h(m,n) \times f(m,n) + u(m,n)$$

-Here f(m,n) is the original image and it has gone through a multiplication operation to produce an image with blur. u(m,n) makes the noise in the image.

-To get the f(m,n) , image deconvolution techniques will be used.

2.Image Resizing

-Since the images given in the test cases are too small, the original images need to be resized to a certain percentage.(for ex : resizing the original image to 120%)

-But there is an issue with this also. Since the image resizing uses techniques like image interpolation, the original pixel arrangement will be distorted and the original content of the image will be affected by it.

-Therefore this step might have to be skipped.

3.Sharpen the image edges using an optimal filter

-Then the image will be filtered to detect edges.

-For this several edge detection techniques will be tried. For example : Prewitt edge detection, Canny edge detection are good solutions for edge detection. The final output after the edge detection of the image is an image which has white edges in the black background.

4.Localizing the characters in the number plate

-This is mainly to identify the boundaries of the number plate and the characters. By identifying the boundaries, it is able to identify if there is any rotation in the number plate itself. If there are any, the rotated number plate will be brought back to a vertical position using geometrical transformations. Orthogonal projections and other methods in the localization are used to identify the number plate alignment.

5.Character Segmentation

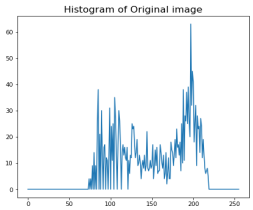
- For this the black and white image of the number plate is used and histogram projection will be used. Since this is a binarized image, we can use background and foreground pixels to identify the characters.
- To identify the height of the character, horizontal histogram projection will be used and there we count the number of pixels along rows, and assign those values to an array. The length of that array is the height.
- To identify the width of the character , vertical histogram projection will be used where we count the number of pixels in the columns and assign that to an array. The length of this array will be the width.

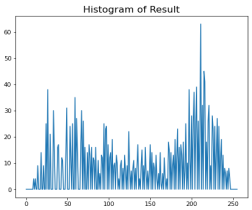
6.Character recognition and concatenate into a string

- After identifying each character , those values will be concatenated to form the string.
- Also the Pytesseract, OpenCV libraries will be considered to implement this step.

INTERMEDIATE/ PRELIMINARY RESULTS

Initially the images that had low contrast are enhanced using contrast stretching function to get good pixel value distribution.





Then, spatial filters were applied to the contrast stretched images.

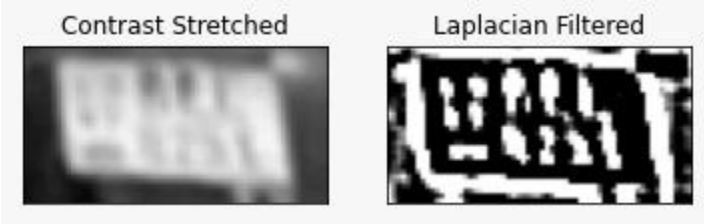
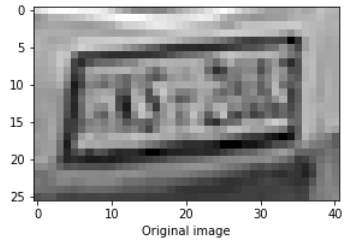
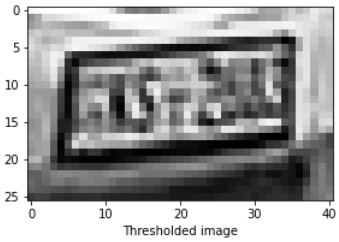
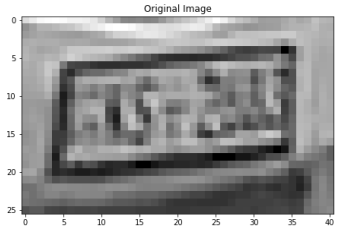


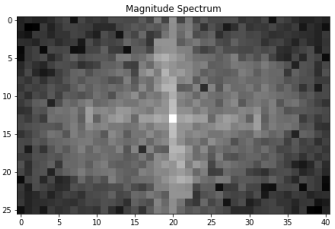
Image thresholding filters were also applied as the basic segmentation to partition an image into a foreground and background. These are the results of some of them.

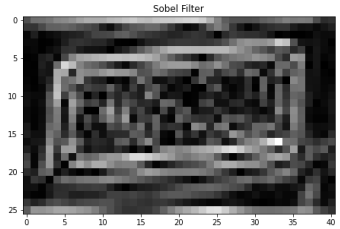




Frequency domain filters were also applied to the images.







REFERENCES

1. https://www.mygreatlearning.com/blog/introduction-to-edge-detection/
2. https://link.springer.com/content/pdf/10.1007%2F978-3-642-24088-1_46.pdf