License Number Plate Recognition

Himanshu Dahiya **2015csb1013**
Soumyadeep Roy **2015csb1035**

Introduction

In today's world, number of vehicles are increasing day by day, and it getting harder and harder to keep track of them at different places like parking spots, or at toll plaza or any other similar place. But if we have a robust system which detects the **License Plate Number** of every vehicle going in and out of that place and keeps it in database, then it will help in a lot of possible ways.

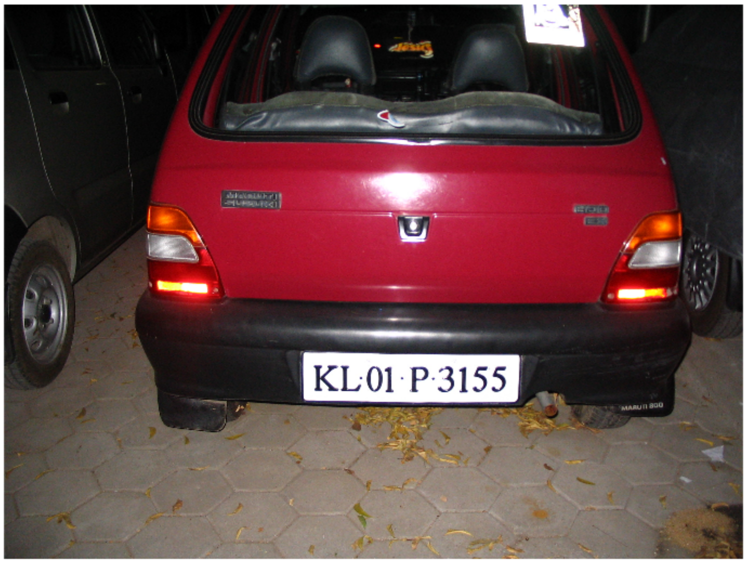
This project is a MATLAB based implementation of how to recognise a license plate in an image. In this project, we used different aspects of **Digital Image Processing** like **Localisation, Histogram Equalisation, Erosion and Dilation,** and **Image segmentation.** Also, to recognise the characters after segmentation of license plate, used **Optical Character Recognition** to detect the characters written on the segmented region of image.

Implementation

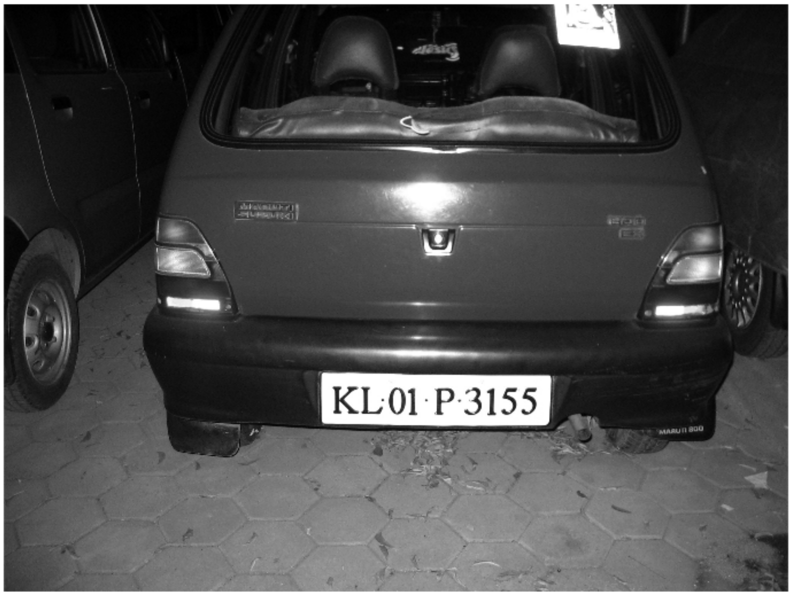
There are various steps involved from image preprocessing to detecting the characters written on license plate. The steps are discussed on the following pages of the report.

1. **Noise Reduction**

The original image is converted to **GrayScale,** applied **Histogram Equalisation** to enhance contrast in the image and then **Noise reduction** is applied by blurring the image and then sharpening it to remove the noise in it so that it does not interfere in further analysis.



Original Image



Noise reduced Image

2. **Processing edges in horizontal direction**

   A histogram is plotted which has sum of differences of consecutive pixels for a row on y-axis and row number on x-axis. The difference is added to sum only if there is a certain minimum intensity difference between consecutive pixels or we can say if the difference crosses a certain threshold, then only it is added to sum.

3. **Passing histogram through low pass filter**

   The histogram in calculated in above part is now passed through a low pass filter to smooth it out so that when we the histogram through dynamic threshold filter, we do not lose essential information.

4. **Processing edges in vertical direction**

   As plotted in part 2, we plot similar histogram but now taking column wise sum of differences instead of row wise.

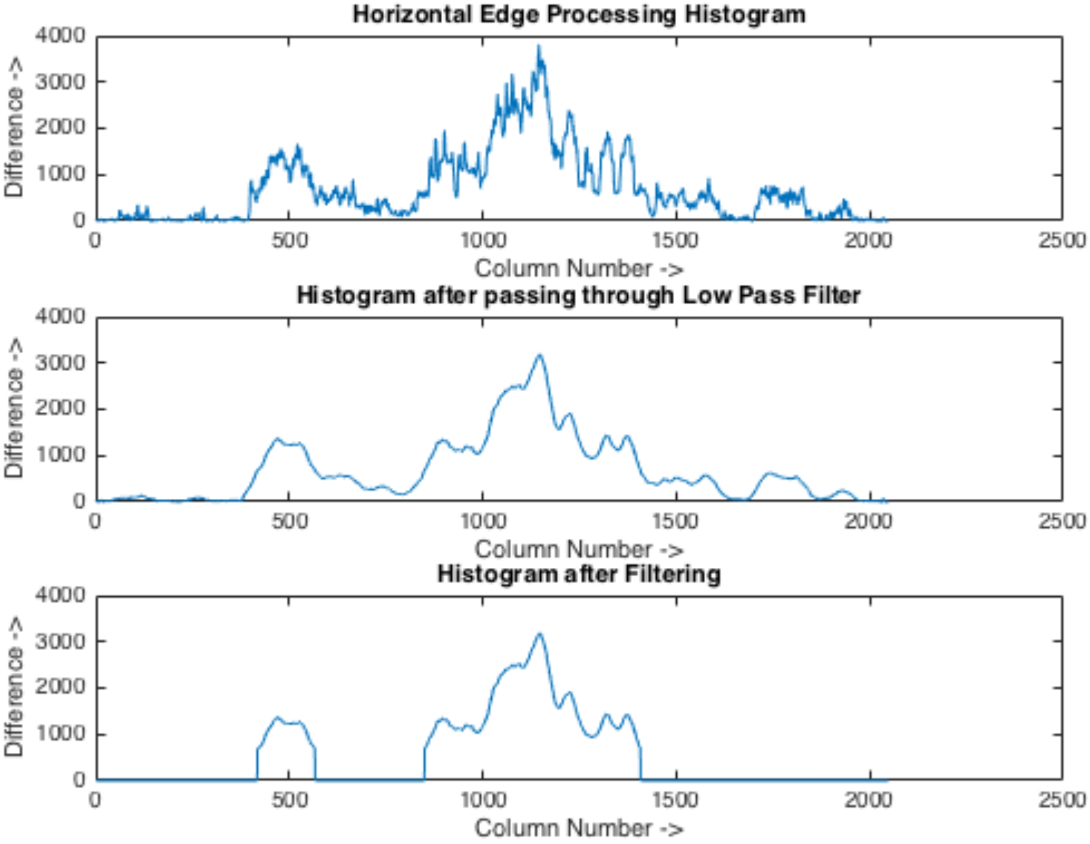
5. **Passing histogram through low pass filter**

   The histogram in calculated in above part is now passed through a low pass filter to smooth it out.

6. **Passing both horizontal and vertical histograms through dynamic threshold filter after smoothing**

   This is done to to remove unwanted part from the image. After passing through a threshold, the region in which there were not much difference in intensities is converted to black as license plate consists of characters so it is more likely to switch intensities then the regions which consists of vehicle body. The threshold is taken as sum of total values of histogram upon total number of rows for horizontal histogram and similarly sum of total values upon total columns for vertical histogram.

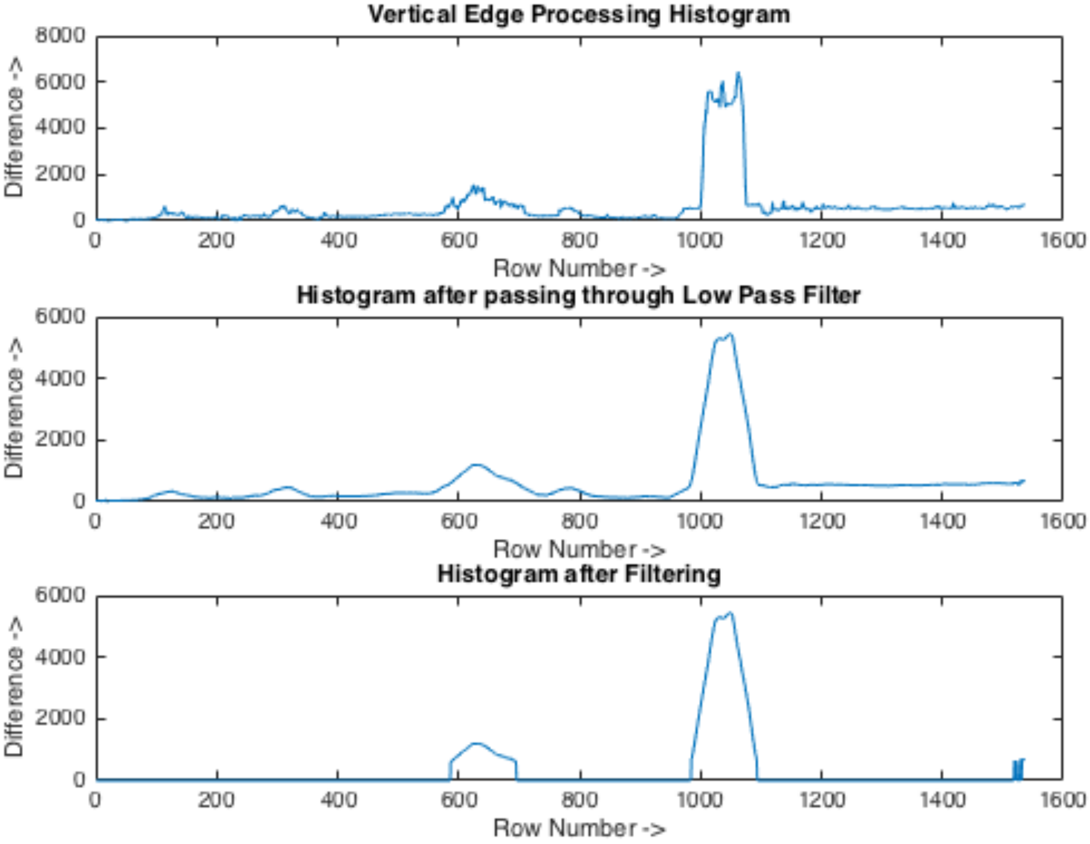
After this step, we are left with a few rectangular blocks in the image which are having maximum intensity variations one of which is probably license plate. Now our next step is to determine which one is license plate.



Horizontal Edge Processing Histogram

Histogram after passing through Low Pass Filter

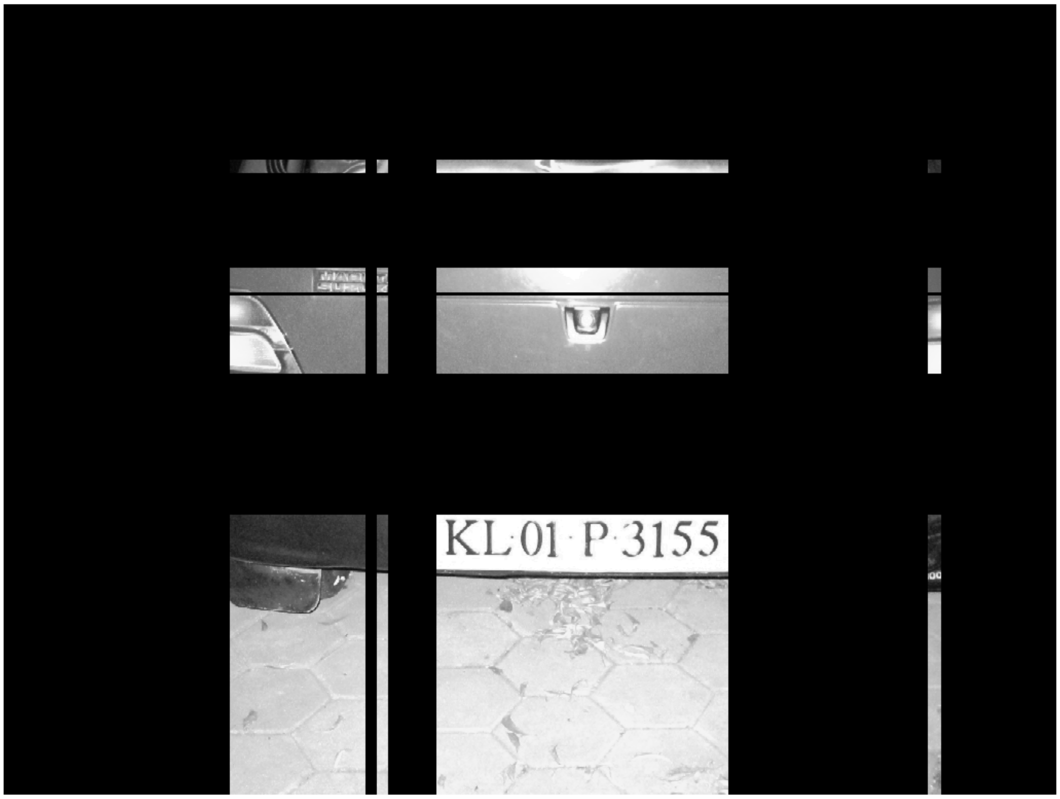
Histogram after Filtering



Vertical Edge Processing Histogram

Histogram after passing through Low Pass Filter

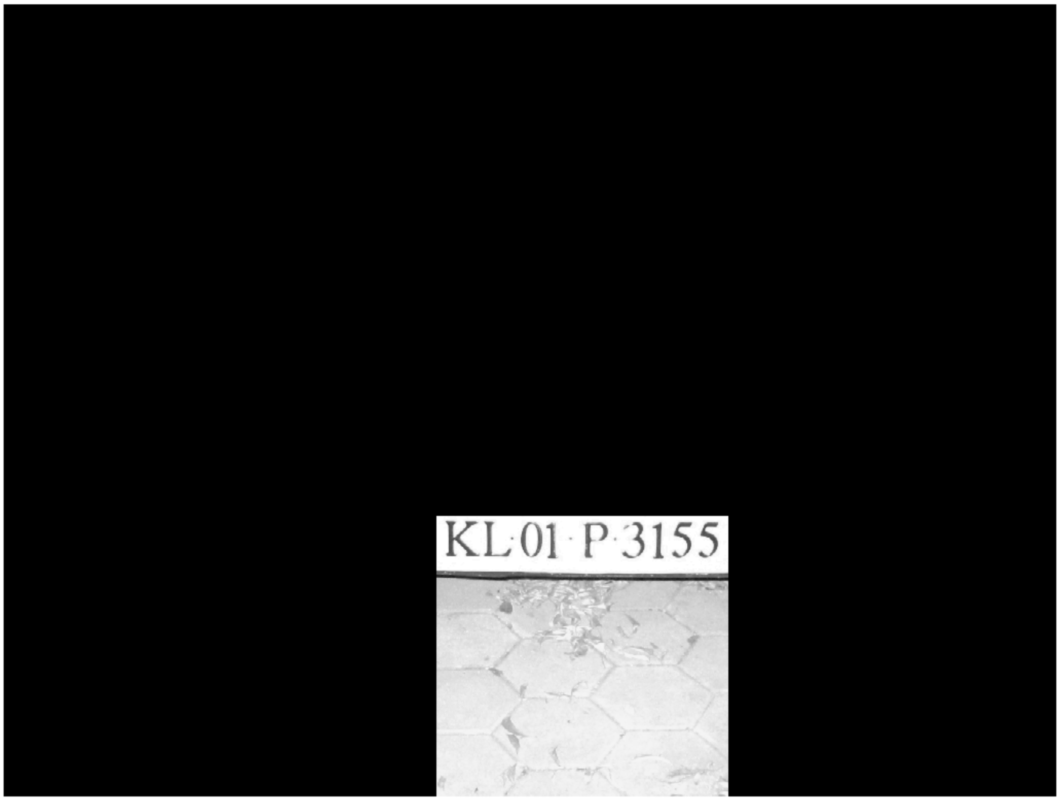
Histogram after Filtering



Regions after passing through digital threshold filter

7. **Segmentation**

The region which has the maximum average histogram value is considered as region containing license plate. Therefore, we again draw rowwise and columnwise histogram of the blocks and take the block which has maximum average histogram value.



Number Plate region after segmentation

8. Recognition of license number

After we have correctly extracted the segment containing the license plate, the task remaining is to read out the number written on the plate. This can be done using optical character recognition. The main challenge faced here is that there can be different fonts and ways in which the number is written on plate. Also, the orientation of plate, like if it is tilted slightly, then OCR fails at several cases.

The outputs of two OCR algorithms are as follows:

1. **KL~O1&P’3I55** (MATLAB inbuilt OCR function)
2. **BKL6016P13I155** (Our implementation of OCR)

The correct license number is : **KL01P3155**
We can see that none of them are correct but they are pretty close to the correct answer.

Limitations

This project has many limitations. The accuracy of the algorithm completely depends upon image, how well the number plate is exposed, noise in image, background of image, and many others like how well trained the OCR algorithm is trained for different kinds of fonts. Even after segmenting the license plate correctly, the optical character recognition algorithm fails to determine what is written.

The other limitation to this is if there is some other text nearby license plate or on the image like some sticker on the car, then the segmentation algorithm might confuse between license plate and sticker and end up taking sticker block as license plate and give the wrong output.

Conclusion

After implementing the project, we came upon the conclusion that if we have a set angle of camera and well defined view to capture the license plate of vehicle, we can correctly recognise the license number. This algorithm is not perfect and fails as many points but it can be made robust for some particular type of input images. Also, as this moment, we do not have much robust and accurate OCR mechanism to determine natural language written on segmented region of image, which is where this algorithm fails. If we see this algorithm just to extract the region containing license plate, it works fine in majority of cases, but fails miserably when it comes to recognise the characters.

References

1. http://files.spogel.com/abstracts/p-1628--number_place_recognition.pdf
2. https://in.mathworks.com/help/vision/examples/recognize-text-using-optical-character-recognition-ocr.html
3. https://in.mathworks.com/help/vision/examples/automatically-detect-and-recognize-text-in-natural-images.html