Image Processing (CO543)
Mini Project Final Report

PERERA S.S. (E/17/251)
SILVA H.S.C. (E/17/331)
WANNIGAMA S.B. (E/17/369)

May 6, 2022

Abstract

Automatic Number Plate Recognition (ANPR) is a software or embedded system that detects and recognizes license plates using image processing techniques. With the increasing number of vehicles on road in the past decade, the demand of having automated number plate recognition (ANPR) has increased. However, due to the varying characteristics of number plates in terms of colors, language, text positions, fonts, and images of number plates being under various conditions (poor quality, distortion, lighting conditions, weather, rotation, etc.), the ANPR has been a challenging task in image processing and computer vision. This domain has been covered in a lot of research literature because of the interesting and challenging nature of the problem. The main objective of this project is to develop a piece of software that can localize and recognize the license number and the province code, given an image consisting of Sri Lankan numberplate(s). We try to address this problem using a rule-based approach that utilizes some basic image processing techniques.

1 Introduction

Automatic number plate recognition systems (ANPR) have been around for a long time, but it wasn't until the late 1990s that they became a popular application because of the enormous growth in the number of vehicles on the road. The information retrieved from license plates is primarily utilized by law enforcement agencies for traffic monitoring, access control, parking, motorway road tolling, and border control, as well as for creating car records for parking systems, journey duration measurement, and other purposes.

The problem can be broken down into five main parts:

1. Image Acquisition and Preprocessing - Capturing of the image consisting of license plate(s) and preprocessing (normalizing the image by histogram equalization, denoising, etc.)
2. Plate localization - Detect and isolate the license plate from the rest of the image
3. Plate Enhancement / Restoration - Since the detected plate can be subjected to various types of distortions, it is important to restore the lost information of the plate image
4. Character Segmentation - Find the individual characters on the plate
5. Optical Character Recognition (OCR) - Recognition of the segmented characters

Since the images might be in poor conditions, before inputting the images into the system, they need to be pre-processed using various image processing techniques in order to recover the details needed for rule-based detection and localization. We've taken the nature of license plates as an advantage to detect and localize license plates among the other objects in the input image. Usually, the texts on the license plates are painted such that the foreground is darker than the background. The general format for Sri Lankan license plates is two English upper-case letters representing the province code followed by the license plate number.

The license number is two or three English upper-case letters followed by a 4-digit number. The English letters and the digits are sometimes separated by a dash ('-'). There're 9 provinces in Sri Lanka, and the province codes used are as follows:

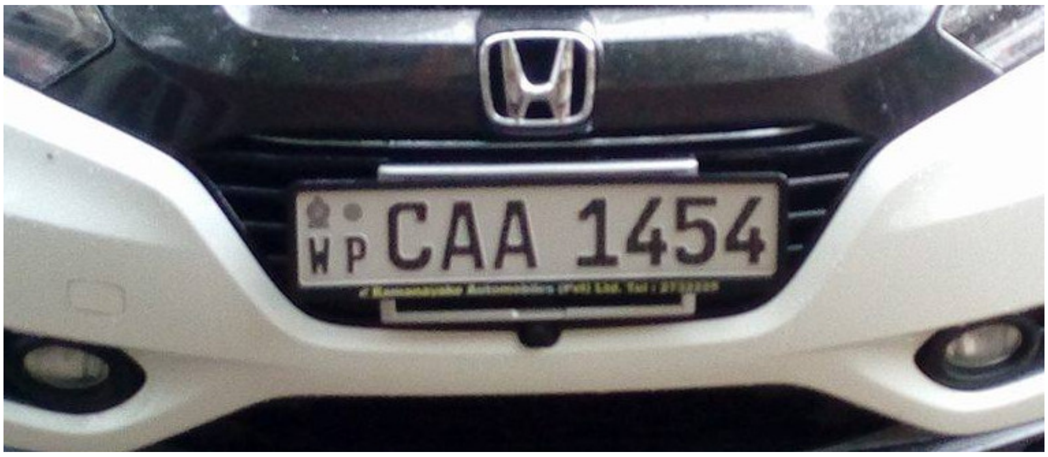


Figure 1: Example of a Sri Lankan license plate

1. CP - Central Province
2. EP - Eastern Province
3. NC - North Central Province
4. NE - North Eastern Province
5. NW - North Western Province
6. SB - Sabaragamuwa Province
7. SP - Southern Province
8. UP - Uva Province
9. WP - Western Province

Among the Sri Lankan license plates that are being used in the present day, two color combinations have been used: black text on a white background and black text on a yellow background. Thus, we have used morphological transformations along with thresholding in order to draw and find the corresponding contours that bound the license plates in the input image. There're two sizes of license plates used in Sri Lanka with aspect ratios being 5:1 and 3:2. However, our system only focuses on the license plates with an aspect ratio of 5:1 in order to eliminate the complexities that might arise during the detection. The detected license plates are then cropped and fed to the recognizer. Before applying OCR, the system attempts to restore the detected plate image if it has been subjected to distortions. If no number plate was detected or the recognizer couldn't find any characters, the system will output as there are no license plates in the input image.

2 Related Work

ANPR systems have been implemented in many countries such as Australia, Korea, and a few others, and the strict implementation of license plate standards in these countries has helped in the early development of ANPR systems. Most of the systems in use take into consideration the standard features such as the dimension of the plate, border for the plate, color and font of characters, etc. to localize the number plate easily and identify the license number of the vehicle.

One such approach was implemented by *S.Ozbay and E.Ercelebi* [3] using a black pixel projection-based image segmentation scheme to recognize Turkish number plates in the binary domain. A smearing technique was used to localize the number plate and, horizontal and vertical runs of the binarized image were taken which was followed by segmentation of the plate from the rest of the image based on a particular threshold number of pixels. A similar algorithm was used to segment the component characters from the plate after the image was filtered and dilated, while cross-correlation coefficient techniques have been used to classify the text.

Classical Hough transform is another method used in ANPR and *Tran Duc Duan, Tran Le Hong Du, Tran Vinh Phuoc, and Nguyen Viet Hoang* [2] have tried a contour-based algorithm in association with the

Hough transform to bring down the computational overhead. The contour algorithm they've used narrows down the sample points on which the Hough transform is to be applied. A projection-based segmentation scheme is used for the component letters and an HMM-based recognition algorithm is suggested for the character recognition part. A major problem with the projection-based method they've used is that sometimes it leads to under segmentation. A threshold value of pixels needs to be assigned for segmentation. The Hough transformation alone would require a very high memory for processing because of the large number of computational processes involved.

An Optical Character Recognition (OCR) technique was used in [4] which is a widely used technology that translates scanned images of printed text into machine-encoded text. The literature proposed a feed-forward neural network-based OCR algorithm where two non-overlapping real character image data sets are used for training and testing the proposed neural network.

Each system proposed for vehicle identification and number plate recognition in the literature survey has its own pros and cons.

3 Our Approach

The system is focused to do two main tasks: recognizing the license plate number and recognizing the province code. Both of these together uniquely identify the vehicle. The following block diagram summarizes the main steps of the system:

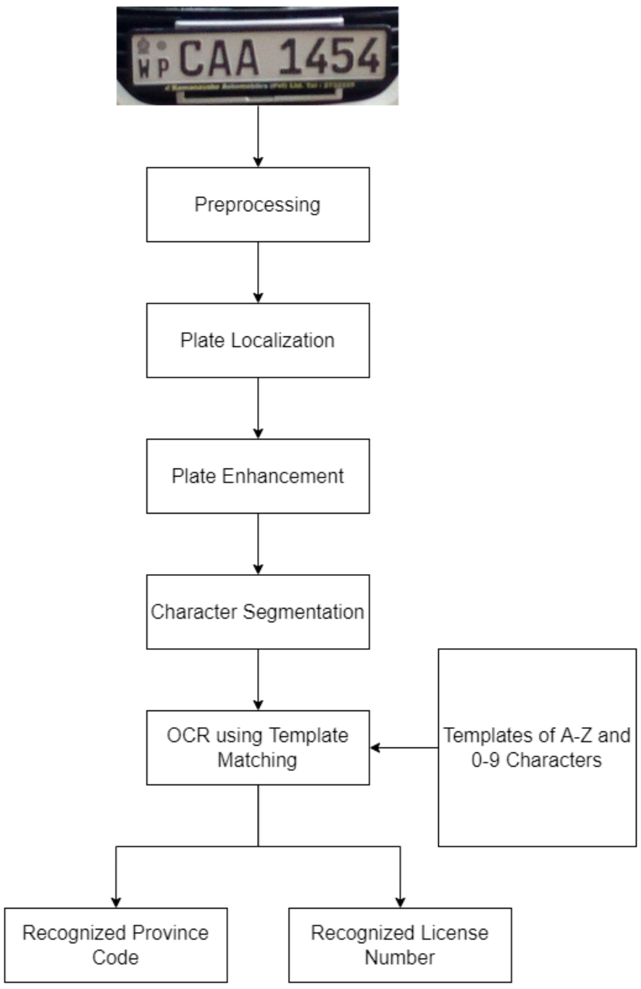


Figure 2: Block Diagram of the ANPR System

After going through the existing literature and understanding the methodologies they have used, we came up with an approach with 5 main steps, as was briefly explained in the introduction.

3.1 Image Acquisition and Preprocessing

Our approach was to recognize number plates and read characters from any sort of image which would contain a number plate. This means that images of various sizes, contrasts, degradation levels, exposure, and clarity could be used as input images. Therefore it was important to pre-process the image as the first step.

A number of pre-processing steps were performed on an input image. They were performed in the following order:

(I) Converting to grayscale.
It is possible that the input image might be of different color scales. For the uniformity of the input images, to reduce the dimensionality, and to reduce the model complexity we first convert all input images to grayscale using the *cvtColor()* function in the OpenCV library. This is an important step because some of the algorithms which would be used in later stages become a lot less complex when implemented on grayscale images.

We use the RGB2GRAY inbuilt algorithm for the conversion. The formula used is the same as for CCIR 601:

$$Y = 0.299R + 0.587G + 0.114B$$

(II) Resizing.
Resizing the image was crucial to ensure that our system will work on same size images, which would affect the accuracy with which the next step in the pipeline will be performed, which is plate localization. The resize function in the OpenCV library was used for this. This library can implement a number of interpolation methods out of which we used **bilinear interpolation** since it gave the resized image which was closest to the original during both increasing and decreasing the size.

(III) Histogram equalization.
The contrast of the input images can vary depending on the lighting conditions during which the photo was taken, interferences that could happen during transferring of the image, etc. hence histogram equalization is important to increase the contrast of the image which would in most cases increase the clarity and recognizability of features of the image.

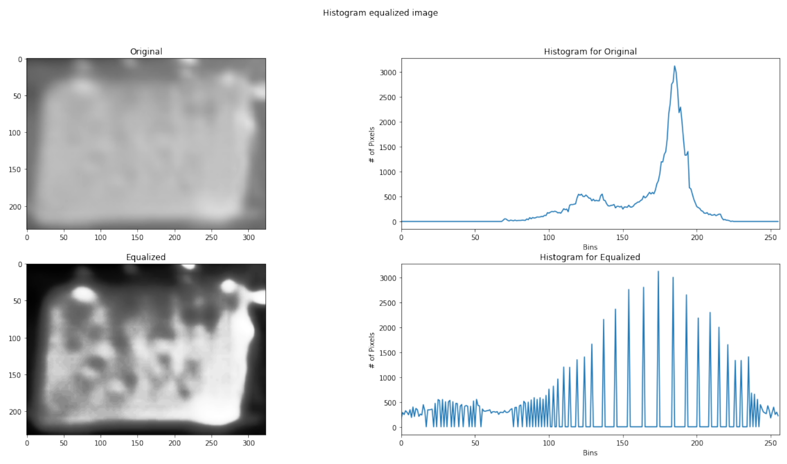
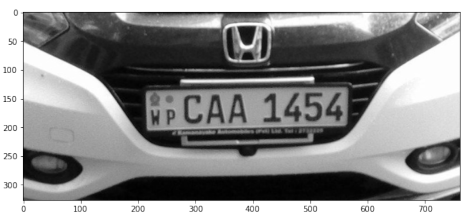
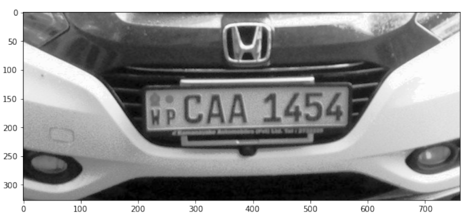


Figure 3: Histogram Equalization on Test Images

(IV) Bilateral Filtering.
Reducing noise and smoothing the image is definitely going to help the detection to happen more accurately. The bilateral filter is a non-local filter which is used to reduce noise and smooth the image. The reason for using this filter is that it reduces the noise while preserving the edges of the image.



(a) Grayscale Converted Image



(b) Histogram Equalized Image

Figure 4: Histogram Equalization

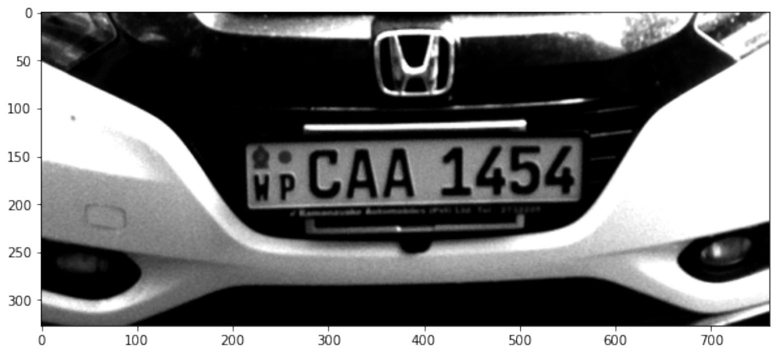
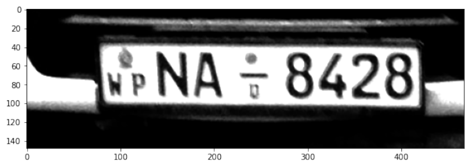


Figure 5: Pre-processed Image

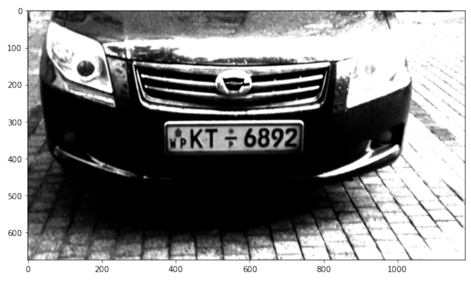
(V) Contrast Stretching.
Contrast stretching is an image enhancement technique which is used to improve the contrast of the image.

(VI) Power Law Transformation.

Since applying the above pre-processing steps on an image would take some time, we decided to use two pre-processed images on different lighting conditions as references to match histograms with. An **overall brightness value** was calculated on the image and it was matched with one of the reference images based on the overall brightness value.



(a) Reference Image 1



(b) Reference Image 2

Figure 6: Reference Images for Histogram Matching

The overall brightness values was computed using the following formulae,

For an RGB image,

$$f(I) = \frac{1}{3\sqrt{3}} \Sigma_c \left[\Sigma_{i,j} abs(I_{c,i,j})^2\right]^{\frac{1}{2}} \text{ (Average of Frobenius norms)} \quad (1)$$

For a grayscale image,

$$f(I) = \frac{1}{N * M} \Sigma_{i,j} I_{i,j} \text{ (Average of pixel values)} \tag{2}$$

3.2 Plate Localization

The detection / localization of the license plate was the most important and challenging step in the pipeline. Since we're focusing on rule-based algorithms, it was important to ensure that the algorithm for plate detection is generalized enough to work on any image. The algorithm used can be subdivided into following steps:

3.2.1 Morphological Image Processing

Morphological image processing persues the goals of eliminating the imperfections in the image by accounting for the form and structure of the image. Morphological techniques probe an image with a small shape or template called a **structuring element**. The structuring element is positioned at all possible locations in the image and it is compared with the corresponding neighbourhood of pixels. Some operations test whether the element "fits" within the neighbourhood, while others test whether it "hits" or intersects the neighbourhood.

To enhance the license plate(s) in the image before it proceeds to the next step, Morphological operations were performed on the pre-processed image in the following order:

(I) Top-hat transformation.
The top-hat transformation (i.e. white top-hat) is a morphological operation which is used to enhance bright objects in dark surroundings in an image. The intention here was to make the license plate more visible, since usually a license plate has a birghter background colour, so that contours can be drawn more accurately. The top-hat operation does this by subtracting the opening from the original image. It is given by,

$$T_w(I) = I - I \circ b \tag{3}$$

where **I** is the input image, $\circ$ is the opening operation, and **b** is the structuring element.

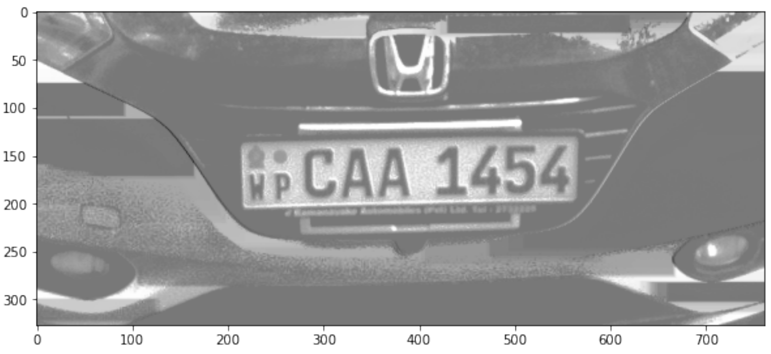
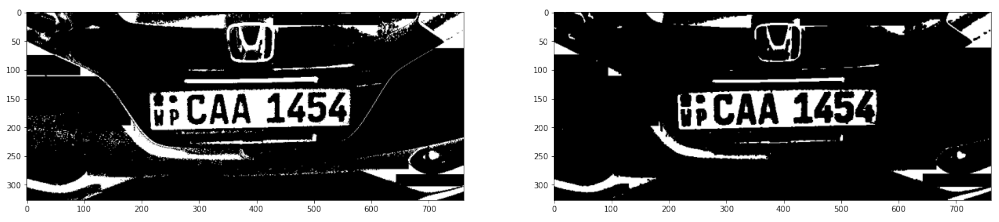


Figure 7: Top-Hat Transformation

(II) Erosion preceded by binary thresholding.
Before applying the erosion operation, the image resulted from the above step was thresholded in order to obtain a binary image so that the bright objects can be considered as foreground and the others as background. The thresholding algorithm that was used is the Otsu's Thresholding. The erosion operation is used to erode away the boundaries of regions of foreground pixels (i.e. white pixels in binary images). The idea behind using this operation is to remove the background noise that blocks the visibility of the license plate.

(III) Closing transformation.
Closing transformation is used to fill (close) the gaps (i.e. to remove false negatives) inside the foreground objects. It is basically dilation followed by erosion. This will make sure that the black texts in the license



(a) Thresholded Image (b) Eroded Image

Figure 8: Thresholding followed by Erosion

plate are filled in with white pixels so that the plate can be completely isolated as a mask. Closing operation is given by,

$$I \bullet b = (I \oplus b) \ominus b \tag{4}$$

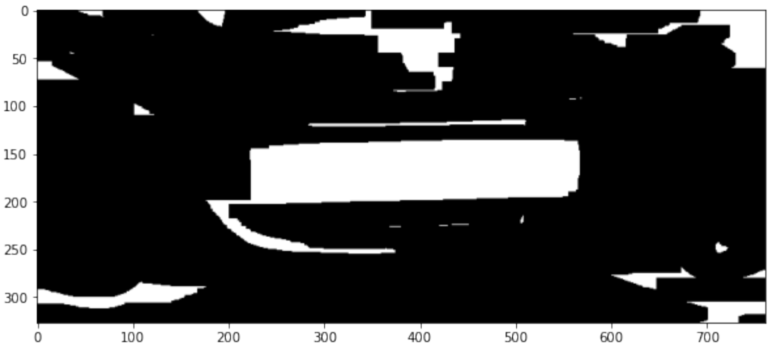


Figure 9: Closing Transformation

(IV) Opening transformation.
The opening of an image I by a structuring element b (denoted $I \circ b$) is an erosion followed by a dilation:

$$I \circ b = (I \ominus b) \oplus b \tag{5}$$

Opening is so called because it can open up a gap between objects connected by a thin bridge of pixels. Any regions that have survived the erosion are restored to their original size by the dilation. This operation is also used to remove the noise from the foreground but in this case, the noise removed are the false positives. Even though there's a small amount of false positives still present in the image, it won't be a

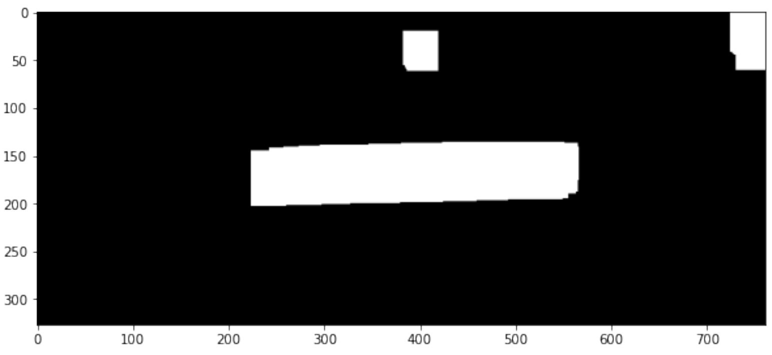


Figure 10: Opening Transformation

huge problem since we always expect to find the largest contour that matches the standard aspect ratios of a license plate.

(V) Dilation.
To remove the effects of the erosion performed in the second step, a dilation is applied so that the foreground objects are restored to their original size.

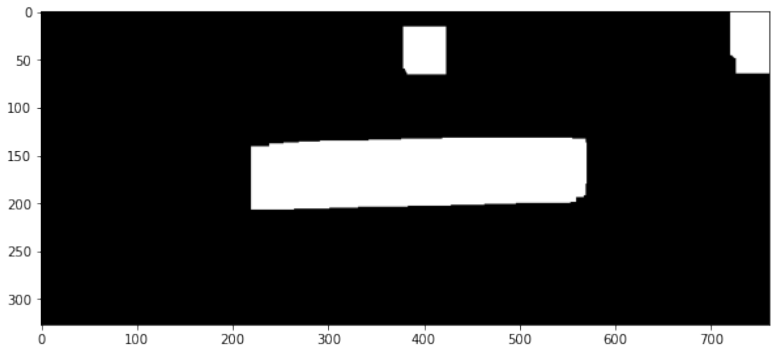


Figure 11: Final processed image after morphological operations

After we're satisfied with the image obtained by the morphological operations, we proceed to the next step.

3.2.2 Drawing Contours and Choosing the Best Contour

Drawing contours and the morphological processing that is mentioned above are not a trivial tasks to perform using rule-based techniques. Therefore, we experimented with several parameters used in the morphological processing (i.e. structuring element sizes, thresholds, number of iterations the operations are performed, etc.). The number of iterations that a particular morphological operation is performed was different for different images. Therefore, we had to iterate over some of them and brute force to find the best contour. In the burte-force search, we tried two ways of computing the contours:

1. Draw the contours directly using the morphologically transformed binary image.
2. Draw the contours by first finding the edges of the image (using the Canny edge detector) and then finding the contours of the edges.

If one didn't succeed during the search, the other way was tried. After computing the contours, the list of all contours were sorted in descending order by the area of the contours. Then, the top 10 contours were chosen and each of them was approximated by a polygon. The approximated contours are then checked for their number of vertices to be 4 (since we expect the license plate to have 4 vertices). If the number of vertices is 4, the contour is checked for its aspect ratio. If the error between the aspect ratio and the standard aspect ratio is less than a certain threshold, the contour is considered as a valid license plate contour.

After a contour is chosen, the binary image is cropped to the contour. Then, an evaluation score is computed as follows:

Let M_I be the ideal mask (i.e. the mask with all white pixels with the same size as the contour).
Let $g(I_1, I_2)$ be a function that computes the similarity between two images I_1 and I_2.
Let $s(I_1, I_2)$ be a function that computes the structural similarity between two images I_1 and I_2.
Let r be the white pixels ratio inside the contour of the binary image.
Let w and h be the width and height of the contour.

Then, the evaluation score is given by,

$$\text{score} = k_1 g(I, M_I) - k_2 |\frac{w}{h} - 5| - k_3 r - k_4 s(I, M_I) \tag{6}$$

The idea behind this weighted score is to minimize the score returned by $g(I_1, I_2)$ while maximizing the area of the image. k_1 to k_4 are constants and can be tuned to get the best results. The contour with the minimum score was selected as the best contour.

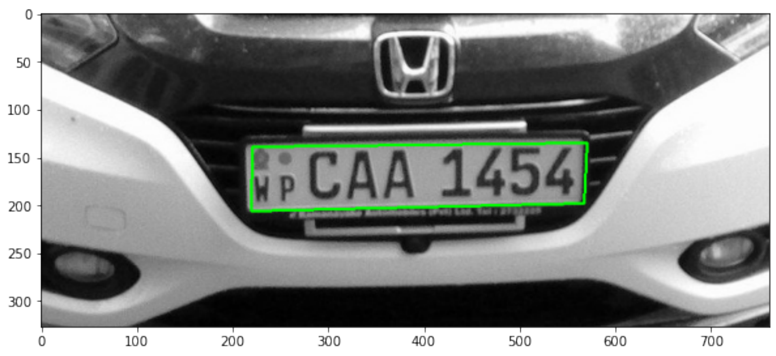


Figure 12: The best contour drawn around the license plate

3.3 Plate Enhancement and Restoration

Since the detected plate can be subjected to various types of distortions, it is important to restore the lost information of the plate image. As this step was very challenging because the group-wise test images provided to us had very low resolution, and was highly distorted up to a level where the numbers could not even be recognized by the naked eye.

A considerable number of spatial and frequency domain processing techniques were tried out on the provided sample images. A few of them which gave satisfactory results are briefly described below.

(I) Power Law (Gamma) Transformations.
Even though this is a very basic spatial domain transformation, this is one of the few filtering techniques which improved the identifiability of the given sample images by a satisfactory amount.

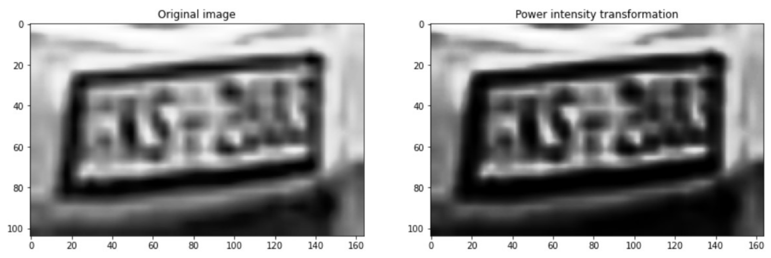


Figure 13: Pre-processed image before and after applying the gamma power transform with tuned hyper-parameters

(II) Gray Level Slicing.
This is equivalent to bandpass filtering in the frequency domain. In the spatial domain, this is implemented by manipulating a group of intensity levels in an image up to a specific range by diminishing rest or by leaving them alone.

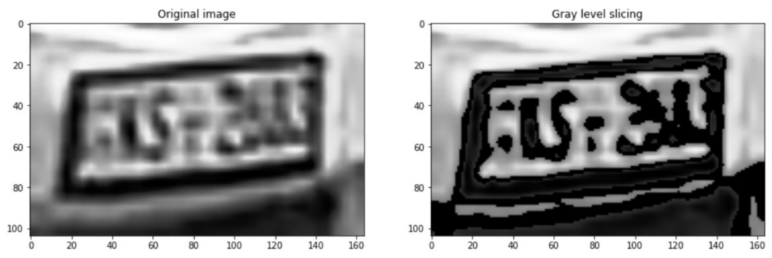


Figure 14: Pre-processed image before and after applying the gray level slicing technique

(III) Sobel Filter.
This is a gradient-based high pass filtering method that looks for strong changes in the first derivative of an image. It does so by taking the difference between pixel intensities in the form of a second order derivative. After experimenting with hyper parameters it was found out that a sobel filter with kernel size 3 works best

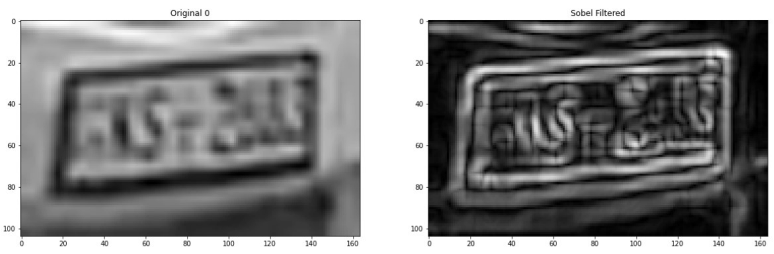


Figure 15: Pre-processed image before and after applying the Sobel filter

for the given sample images. The kernel for the x and y axes for this filter can be computer as follows,

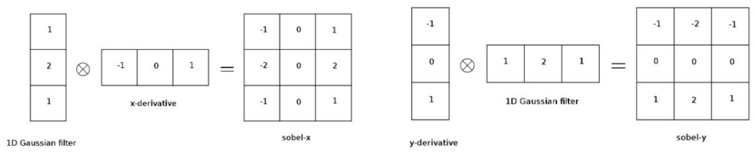


Figure 16: Computation of the 3x3 Sobel Filter Kernel

(IV) Canny Edge Detection.
We have used the implementation of this algorithm in the OpenCV library and it is a multi-stage algorithm that includes stages for,

(a) Noise reduction.
(b) Finding the intensity gradient of the image.
Image is filtered using the Sobel filter in the horizontal and vertical directions to get the first derivative and then these values are used to calculate the edge gradient and the direction for each pixel as follows,

$$Edge\_Gradient\ (G) = \sqrt{G_x^2 + G_y^2} \tag{7}$$

$$Angle\ (\theta) = \tan^{-1}\left(\frac{G_y}{G_x}\right) \tag{8}$$

(c) Non-maximum Suppression.
(d) Hysteresis Thresholding.

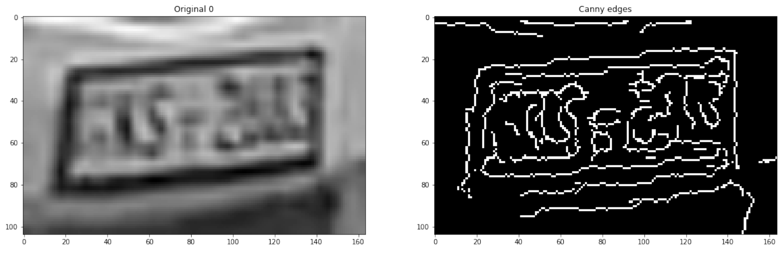


Figure 17: Before and After applying Canny Edge Detection

(V) Gabor Filters.

A set of Gabor filters with different frequencies and orientations may be helpful for extracting useful features from an image. Gabor filters with different frequencies and with orientations in different directions have been used to localize and extract text-only regions from complex document images since the text is rich in high-frequency components, whereas pictures are relatively smooth in nature.

The filter has a real and an imaginary component representing orthogonal directions. We are using the real part of this to formulate the kernel for the bank of filters.

Complex:

$$g(x,y;\lambda,\theta,\psi,\sigma,\gamma) = exp\left(-\frac{x^{'2}+\gamma^2 y^{'2}}{2\sigma^2}\right) exp\left(i\left(2\pi\frac{x^{'}}{\lambda}+\psi\right)\right) \tag{9}$$

Real:

$$g(x,y;\lambda,\theta,\psi,\sigma,\gamma) = exp\left(-\frac{x^{'2}+\gamma^2 y^{'2}}{2\sigma^2}\right) cos\left(2\pi\frac{x^{'}}{\lambda}+\psi\right) \tag{10}$$

Imaginary:

$$g(x,y;\lambda,\theta,\psi,\sigma,\gamma) = exp\left(-\frac{x^{'2}+\gamma^2 y^{'2}}{2\sigma^2}\right) sin\left(2\pi\frac{x^{'}}{\lambda}+\psi\right) \tag{11}$$

Out of all the spatial domain filtering techniques, according to our observation this technique gave the most promising results. After experimenting we found out that using a filter bank with 30 Gabor filters gives the optimal results.

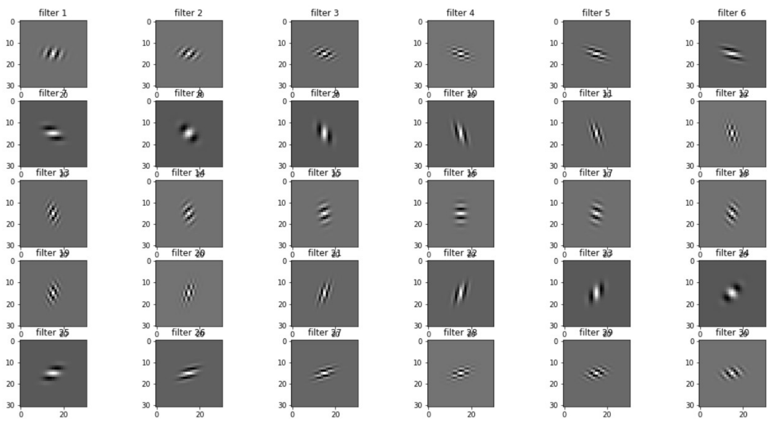


Figure 18: Filter bank with 30 Gabor filters

Applying the above filter bank on the same sample image gave the following results.

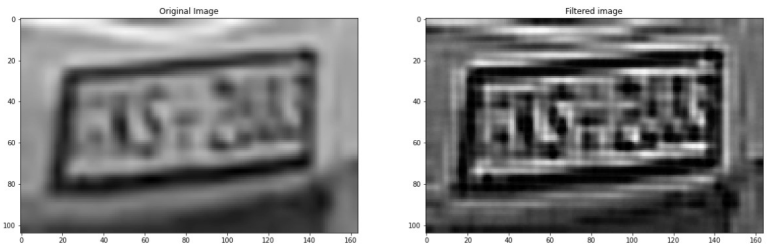


Figure 19: Resultant image after applying the Gabor filter bank

Even though this image does not have an acceptable level of clarity to be used in the next steps of the pipeline, this was the best result that we were able to obtain by enhancing the given sample images.

3.4 Character Segmentation

Before the actual character segmentation, we need to extract (make the characters pop from the surroundings) the characters from the image. The following steps are taken to extract the characters from the image. The preprocessed image was cropped using the contour coordinates that were obtained from a previous step. After the enhancements / restorations were done on it, a black-hat morphological operation was applied to enhance the black objects (the characters on the number plate are usually has a darker color) in a white surrounding. Then, a binary image was obtained by thresholding the image using the Otsu's method.

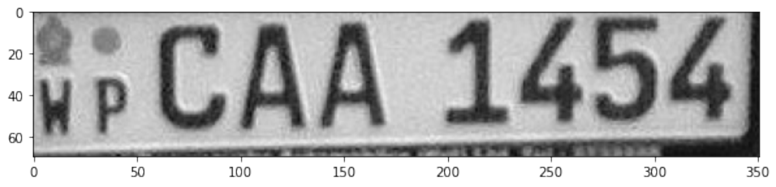
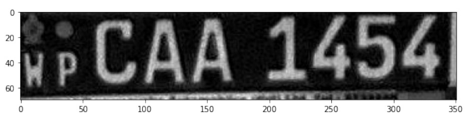
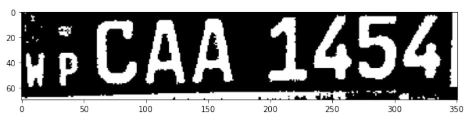


Figure 20: Cropped license plate image



(a) Black-hat operation



(b) Binary thresholded image

Before the proceeding to the character segmentation algorithm, an opening morphological operation was applied to remove the noise that affects the visibility of the characters on the image.

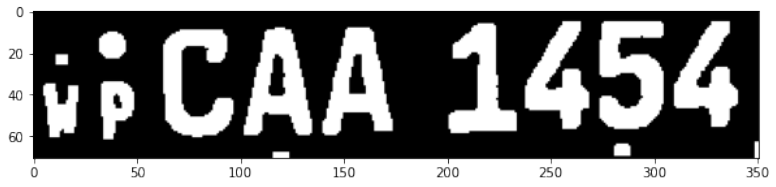
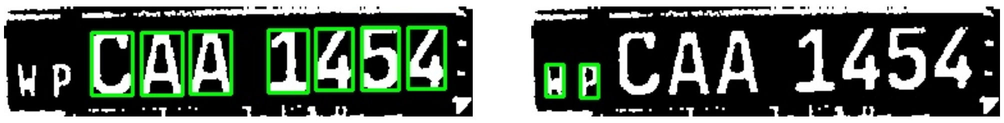


Figure 22: Opening morphological operation

Since the binary image now contains the characters as white regions, we use a connected components algorithm to extract the characters from the image. This was done using the OpenCV library function `connectedComponentsWithStats` which returns the following values,

1. The total number of connected components that were detected.
2. A mask that masks out the connected components.
3. Statistics of each connected component.
4. The centroids of each connected component.

To aid finding the connected components that corresponds to the characters (the province code and the license plate number), we pre-computed the average area, and the average height of all the connected components. We extracted the province code and the license plate number in two separate steps. The license plate number was extracted first and then the province code. To extract the license plate number, we iterated over all the connected components and identified a connected component as a character in the license plate number if the relative position of the component, the relative area of the component and the height of the component are satisfied with some conditions that are only applicable to the license plate number. While extracting the characters in the license plate number, we computed the average width and the height of those in order to help in finding the characters that corresponds to the province code. The province code was extracted in a similar way, also using the average width and height of the characters in the license plate number as measurements since the province code is usually around half the size of the license plate number.



(a) Segmented license plate number (b) Segmented province code

Figure 23: Connected component analysis

3.5 Character Recognition

Recognition of license plates falls into the domain of OCR (Optical Character Recognition). There're so many libraries out there that are implemented for performing OCR such as PyTesseract and Keras-OCR. However, since the license plates have only the upper case alphabetic characters and the digits that total into being only 34 characters, we implemented a rule-based recognition method inspired by [1]. We found a font that's similar to the font used in Sri Lankan number plates and segmented out each character into a data set.

The idea is to use those as the groundtruth templates for recognizing the characters in license plates.

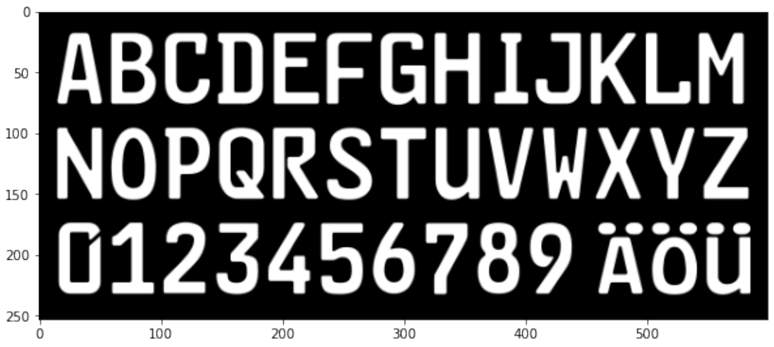


Figure 24: Font used in Sri Lankan number plates

We filtered out only the alphanumeric characters and saved them in a data set. They were labeled with their corresponding character. We used this data set to spawn a new data set that contains only the characters 'B', 'C', 'E', 'N', 'P', 'S', 'U', and 'W' to separately recognize the province code since the province codes are limited to only those characters. Since the Sri Lankan government logo is also present in the license plate, we also included it in the data set as an unknown character with the label 'UNK'.

In the character recognition process, the groundtruth templates are used to compare against the segmented candidates. The comparison is based on a pre-defined evaluation score that we came up with. The score is computed as follows,

- If the recognition is for the province code, the score is computed as a weighted sum of the template matching score (computed using the OpenCV functions `matchTemplate` and `minMaxLoc`), structural similarity score, and the difference between the white pixels ration between the two images. The similarity score, template matching score and the difference in white pixel ratio was intended to be minimized while maximizing the structural similarity score.
- Otherwise, the template matching is ignored and the rest is computed similar to the above.

During the character recognition for the license plate number, if two or three consecutive alphabetical characters were found, we immediately throw away the alphabetic characters in the data set temporarily and only keep the digits because of the format of the license plate number. This reduces the comparison time as well as the complexity of the character recognition increasing the likelihood of a correct recognition.

4 Experiments

As our ANPR system is a rule-based system, we had to do tons of experiments to find the best and general settings for the system. Starting from the very first step and going to the last step, we had to fine tune a lot of parameters. We found 18 different images that contains Sri Lankan number plates. Some of them were extracted from this and this videos that was given in a project discussion thread. We used those images to test and fine tune the parameters of the system. Out of the 18 images we tested, the system was able to detect and recognize the license plate number and the province code in some of them. For some images, the system was only able to recognize the license plate number or the province code. There were some images that the system detected the license plate but could not recognize the characters corrrectly. There were also few images that the system wasn't even able to detect a license plate.

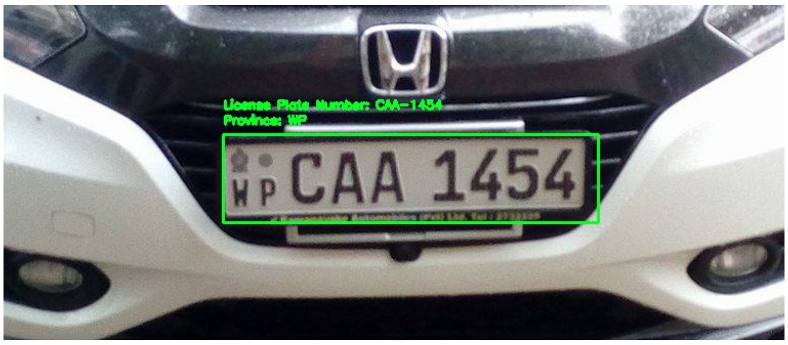


Figure 25: A correctly detected and recognized image

Correctly Recognized %	Correctly Localized %	Correct License Plate Number %	Correct Province Code %
33.3%	61.1%	33.3%	38.9%

Table 1: Final Results

5 Conclusion

In this report, we have presented an automatic number plate recognition system (ANPR) using a rule-based approach. The system was able to recognize the license plate number and the province code only in 33.3% of the images. Thus, the system is not performing well with the current configurations and needs to be improved.

There're many researches done on the topic of ANPR using different approaches such as machine learning and deep learning. Those researches have been able to achieve very high accuracies as well. However, we decided to focus on the rule-based approach because it deals with so many image processing techniques. Therefore, this project was a great opportunity to learn and experiment with different interesting techniques in the image processing domain.

References

[1] J. M. N. D. B. Jayasekara and W. G. C. W. Kumara. Text extraction for sri lankan number plates. In *2015 8th International Conference on Ubi-Media Computing (UMEDIA)*, pages 89–91, 2015.

[2] Brian Jin, Rayal Raj-Prasad, Brigid Angelini, and Jake Rabinowitz. *Vehicle License Plate and Color Recognition using Computer Vision*. 12 2016.

[3] Serkan Ozbay and Ergun Ercelebi. Automatic vehicle identification by plate recognition. *International Journal of Computer and Information Engineering*, 1(9):1418 – 1421, 2007.

[4] Xiaojun Zhai, Faycal Bensaali, and Reza Sotudeh. Ocr-based neural network for anpr. In *2012 IEEE International Conference on Imaging Systems and Techniques Proceedings*, pages 393–397, 2012.