Image Processing (CO543)
Mini Project Mid Progress Report

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

March 25, 2022

1 Introduction

With the increasing number of vehicles on road in the past decade, the demand of having automated number plate recognition (ANPR) has been increased. However, due to the varying characteristics of number plates in terms of colours, 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 lot of research literature because of the interesting and challenging nature of the problem. The ANPR task can be broken down into three main parts: Detection and localization of number plates, recognition of number plates and recognition of characters on number plates (OCR). We will be trying to build an ANPR system consisting of the above three parts using a Deep Learning approach.

As mentioned earlier, 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 will take the nature of license plates as an advantage to detect and localize license plates among the other objects in the input image. The texts on the license plates are painted such that the foreground is darker than the background. 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 will be using 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. We will use those aspect ratios in order to find the correct contour that belongs to the license plate. The detected license plates will then be cropped and fed to the recognizer. The recognizer will recognize the characters (A-Z and 0-9) one by one and output the corresponding text. 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 Problem Statement

The goal of an ANPR system is to automatically detect and recognize the characters on license plates of any given image. Since we're based entirely on a rule-based approach, we won't need to use very large datasets since there's no training involved. However, for testing and rule development purposes, we'll be using several frames extracted from this and this videos. We will be anotating each extracted frame with the corresponding license plate text in order to evaluate the performance of our system.

3 Technical Approach

3.1 Image Enhancement and Restoration

Image restoration is going to play an important role in the whole process of ANPR since the images the model will be getting as inputs are going to come along with many forms of degradation.

Degradation comes in many forms such as **motion blur, noise, and camera misfocus**. In cases like motion blur, it is possible to come up with a very good estimate of the actual blurring function and ”undo” the blur to restore the original image. In cases where the image is corrupted by noise, the best we may hope to do is to compensate for the degradation it caused. In this project, in part of image restoration, we will introduce and implement several of the methods used in the image processing world to restore images.

As the first approach, we have looked into restoring an image using the method of thresholding,

$$g(n_1, n_2) = f(n_1, n_2) \circledast b(n_1, n_2)$$

where f is the original image, b is some kind of a low pass filter and g is our blurred image. So to get back the original image, we would just have to convolve our blurred function with some kind of a high pass filter.

$$f(n_1, n_2) = g(n_1, n_2) \circledast h(n_1, n_2)$$

The problem here would be to find $h(n1, n2)$. This can be relatively easily done by getting the fourier transform of $b(n1, n2)$ and getting the inverse of it. This would give the fourier transform of $h(n1, n2)$. So far we’ve been working on implementing this methodology to restore a degraded image before feeding it to the next step in the pipeline of ANPR.

As an initial step before moving on to the above-mentioned method, we have tried several spatial domain filtering techniques in order to get an idea about the nature of degradation present in the given test images. We have tried several techniques in this domain but haven’t yet been able to get a complete understanding about the nature of the noise and the degradation present in the test images. Out of several spatial domain filtering techniques we tried out, following are a few that gave considerable results. (We have chosen one test image for demonstration).

Histogram Equalization:

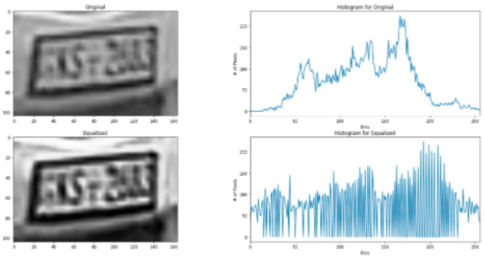
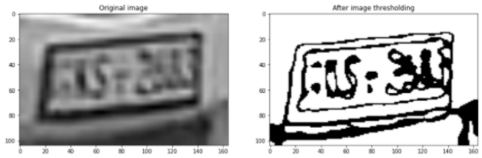
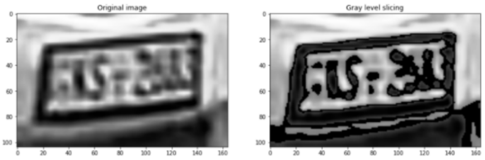


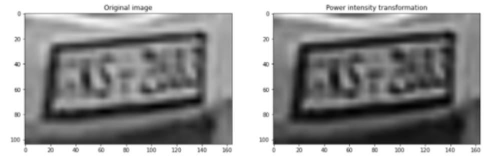
Image thresholding:



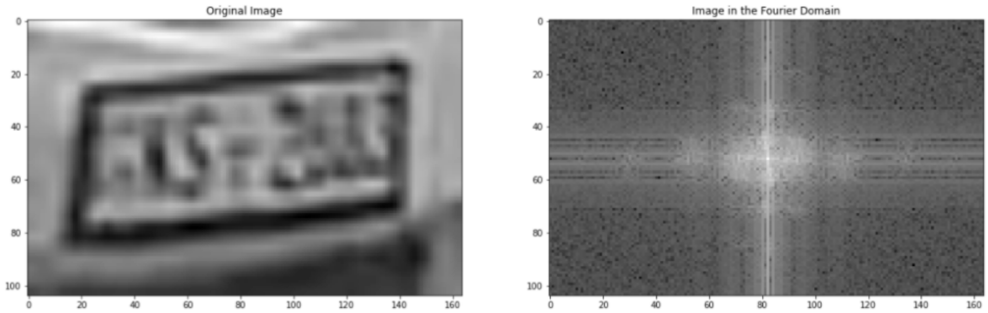
Gray level slicing:



Gamma (Power) transformation:



We have done a basic analysis of the fourier domain representations of the test images, but there’s room for improvement.



So as the next step we expect to do a deeper analysis in the fourier domain, and model the degradation in each image, and use the above mentioned thresholding technique to model the inverse of the degradation and apply the inverse filters on the images to get enhanced images. We also hope to explore the possibility of using Weiner Filtering, Wavelet Restoration, and Blind Deconvolution to improve the restoration phase of ANPR pipeline.

3.2 License Plate Detection and Localization

After the Image restoration process, the next step of the ANPR process is to find the license plate candidate contours in the Image. following are the steps that were followed,

- **Revealing the dark characters of the license plate.**

  Since a license plate contains dark characters (letters, digits, and symbols) in a lighter background we can reveal the characters using morphological operations. In order to enhance and reveal the darker characters, we will be using blackhat morphological transformation with experimentally determined structuring elements (Figure 1).

- **Finding the regions in the image that are light and may contain the license plate.**

  For this operation, as the first step, we are performing a morphological close operation to fill the small dark holes in the image (Figure 2).
  As the next step we perform a binary threshold on our image using Otsu's method to reveal the light regions in the image that may contain license plate characters. Otsu's method generates a threshold using the histogram of the image. (Figure 3)

- **Finding the edges in the image and emphasize the boundaries of the characters in the license plate.**

  For finding the edges we compute the Scharr gradient magnitude representation in x and y directions and perform an addition (Figure 4).

- **Smooth to group the regions that may contain boundaries to license plate characters.**

  We apply a Gaussian blur to the gradient magnitude image. Again we apply a closing operation and another binary threshold using Otsu's method to reveal the white region where the license plate characters are located (Figure 5).

- Using the light region image as a mask for a bitwise-AND between the thresholded result and the light regions of the image to reveal the license plate candidates.

- Find contours in the above image and sort them by their size in descending order, keeping only the largest ones. For this we are using cv2.findContours() function.

- Drawing the contours to analyze the Image.

- **As the final step, Detecting the license plate.**

  For this step we determine the bounding rectangle for each contour. Then, we calculate the aspect ratio for each contour. Normally, the aspect ratio would be around 4 -5 regions. Therefore, we will take the contour between that region. Finally, we will extract that rectangular contour region from the Original Image. (Figure 6)

3.3 License Plate 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 will try a rule-based recognition method. We will find a font that's similar to the font used in Sri Lankan number plates and separate out each character. The idea is to use those as the ground truth for recognizing the characters in license plates. In order to do this, we will crop out all the characters in a license plate by finding the connected components. Then, each cropped character will be compared with all the 34 ground truth characters. We will perform the comparison by computing the cross-correlation between the character images. The candidate with the highest score will be taken as the recognized character.

4 Preliminary Results

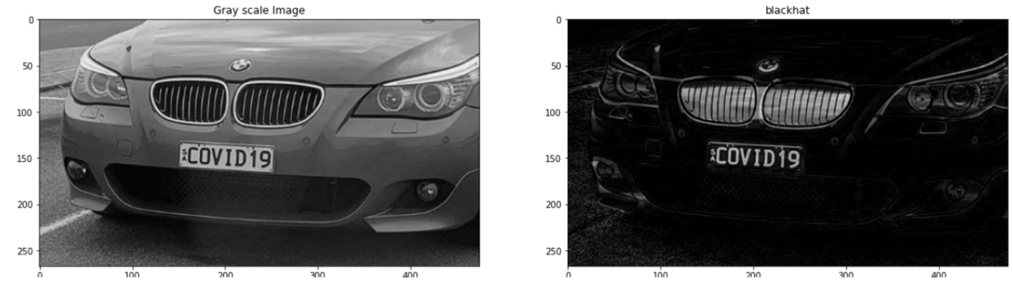


Figure 1: Application of blackhat transformation to enhance dark objects

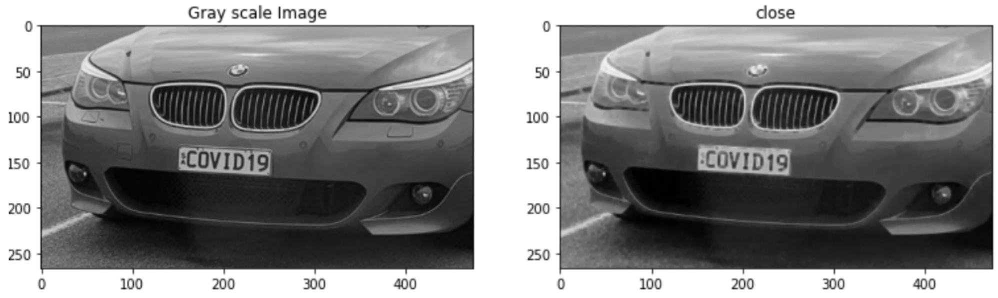


Figure 2: Application of closing morphological operation to fill foreground objects

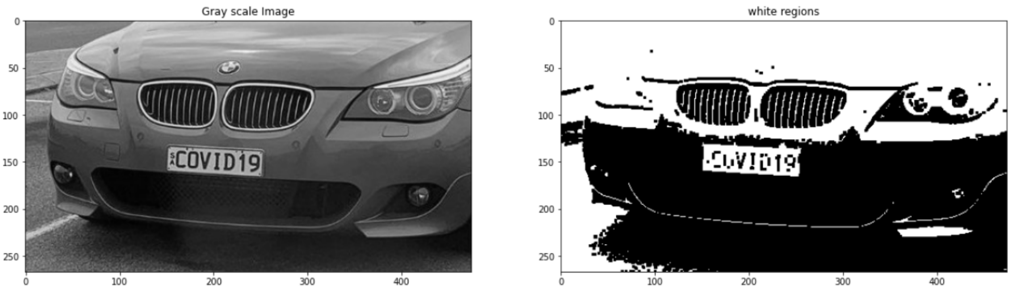


Figure 3: Application of binary thresholding to reveal light regions

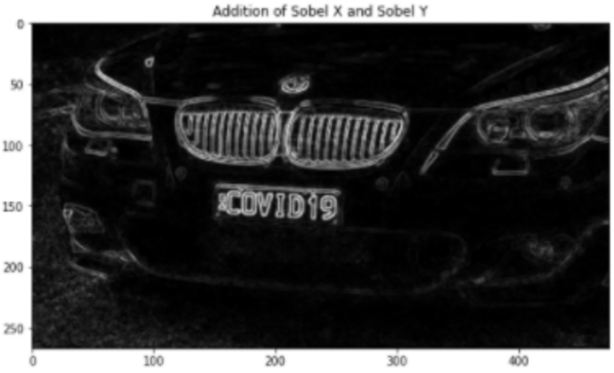


Figure 4: Sobel edge detection

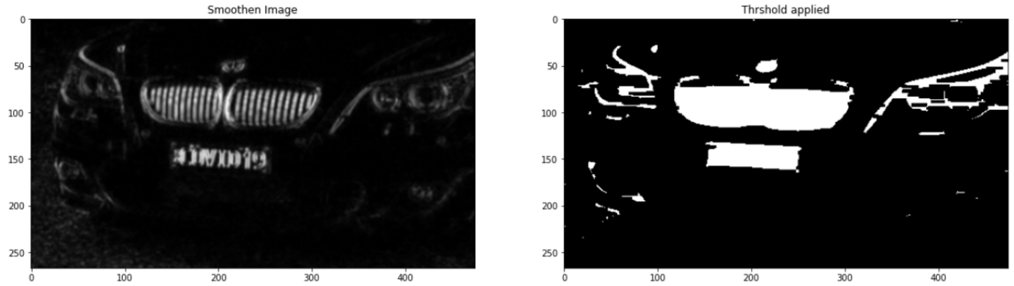


Figure 5: Finding the license plate boundaries

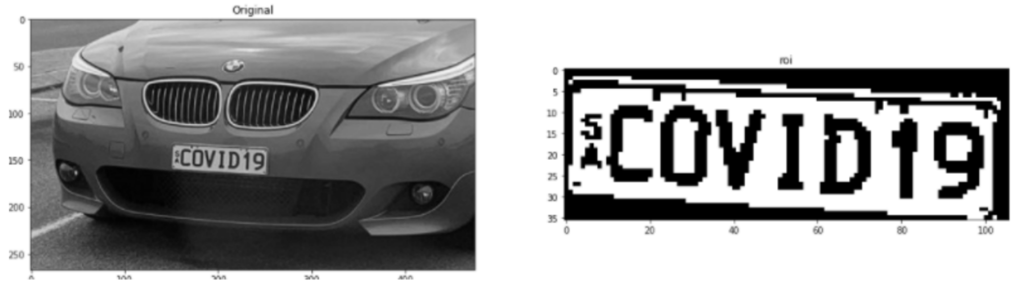


Figure 6: Final localization of the license plate