Assignment 2 - Report

ENEL808 – COMPUTER VISION

MIN CHOI 1132188

INTRODUCTION

The purpose of this assignment was to detect the number plate of a car from a short video of around 10 - 20 seconds, draw a rectangular box around the number plate and display the number above it, then output the resulting image in to video file. The video file I have used has two cars. I was able to correctly detect the number plates but not for every frame. For majority of the video the number plate was prominent part of the frame and was shown up close. But near the end of the video it showed the number plate at a further distance. I was not able to make a program that would detect both the large and the small number plate, so I have focused on detecting the large number plate as that was majority of the video.

METHODOLOGY

The main method I used to detect the characters from the number plate was 'Optical Character Recognition' or 'OCR' in short. Which is available in MATLAB through the computer vision toolbox add-on. MATLAB uses the Tesseract OCR via the 'ocr' function. Tesseract uses deep learning and pattern recognition.

To make the OCR function work more accurately and reliably I had to use several image processing techniques to improve the quality of the image I have obtained from the source video. The first image processing technique I have used was 'rgb2gray' function. This function converts the RGB colour image to greyscale image. This is done by eliminating the hue and saturation information and only retaining the luminance. MATLAB performs a weighted sum of the RGB components to convert to greyscale. The reason for having different weight for each colour is because the human eye perceives the intensity of the colours differently even when they have the same value digitally. The scaling MATLAB uses for each colour are as follows. Red = 0.2989, green = 0.5870 and blue = 0.1140.

Next, I have used the 'imsharpen' function to sharpen up the image. It uses unsharp masking technique. Unsharp masking takes a blurred negative image to create a mask, then combines the original image and the mask to produce a sharper image.

Then I used the 'bwconncomp' find connected components and erase components that are above and below certain size to isolate just the number plate as best as I can. Bwconncomp detects the connectivity of components in a binary image then saves the size of each component among other parameters but size of the component is all I required.

I have also used 'imclose' and 'imopen' functions. They are a morphological operation. Closing performs a dilation followed by erosion and opening performs erosion followed by dilation. In simple terms dilation makes the pixels bigger and erosion makes them smaller.

EXPERIMENT

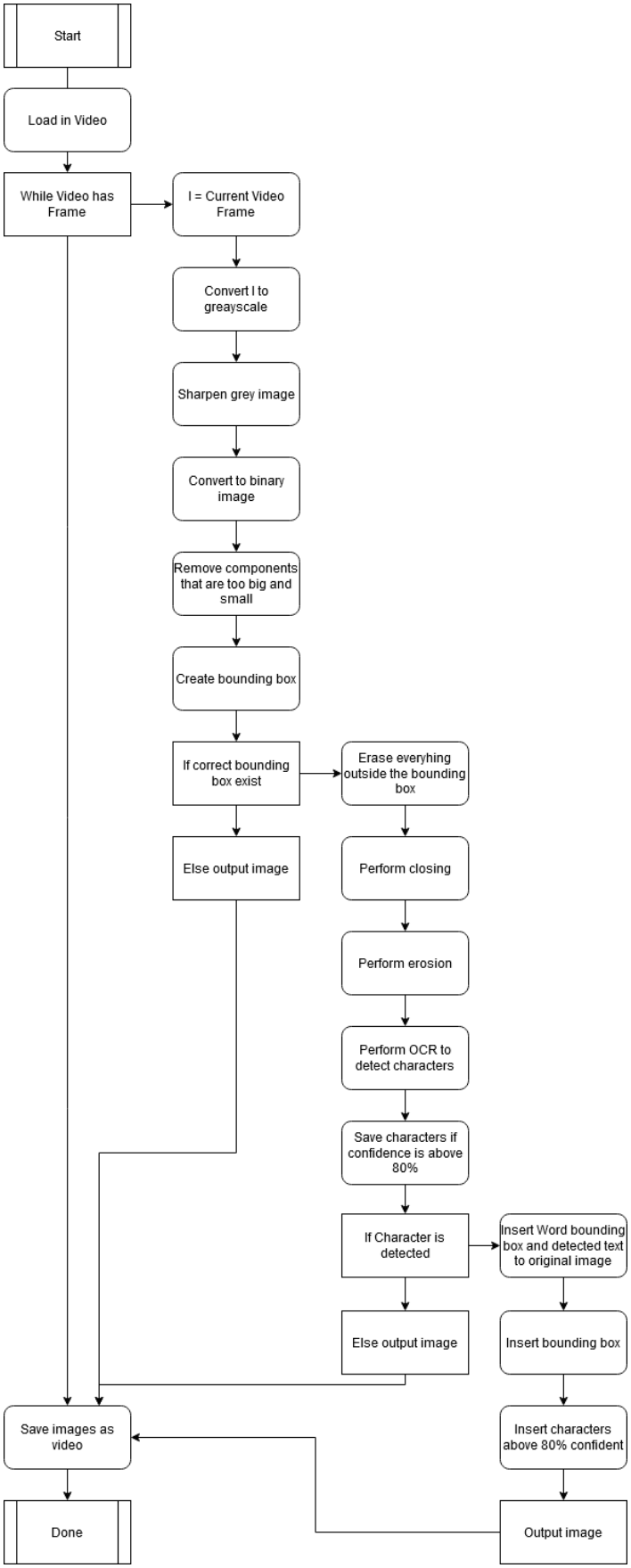

Figure 1 - Process Diagram

Figure 1 above shows the process diagram of my program. The process is as follows. First it reads the video file then converts the frames into an image. Then the image is converted to greyscale. Next it is sharpened and converted to binary image. After connected components that are too large or too small are removed from image. Then using blob analysis, a bounding box is created. Everything outside the bounding box are removed. A closing then an erosion is performed. Next OCR is used to detect characters from the image. Then a bounding box and detected text is added around the detected characters in the original image. Finally, the bounding box and large text of characters that have above 80% confidence is added and final image is saved as a video sequence.

RESULTS

Here are some of the resulting images from my program.

Figure 2 - Processed Binary Image of Car 1

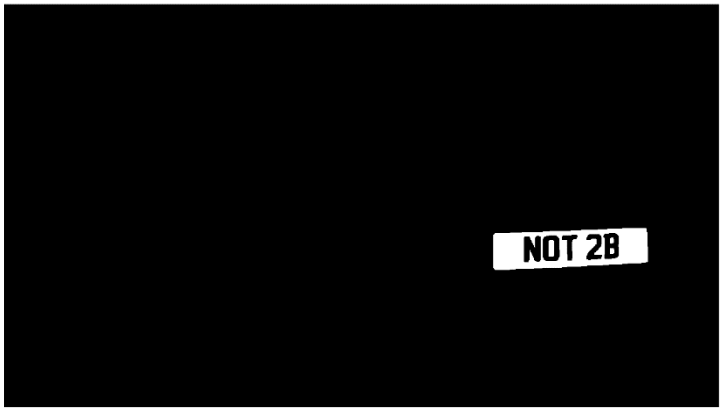

Figure 3 - Processed Binary Image of Car 2

Figure 2 shows the processed binary image of first car and figure 3 is the second car. This is the result after all the image processing have been applied.

Figure 4 - Successful Frame

Figure 5 - Unsuccessful Frame

Figure 4 shows the frame when the number plate has successfully been read. You can see that a white box with the correct detected text is displayed. Figure 5 shows an unsuccessful frame which does not have this box.

CONCLUSION

I was not able to successfully detect correct number plate every frame. But because I have implemented a function to check the confidence level of the OCR, I was able to only display the white box with mostly correct reading. If you only check the result in the white box then it is almost always correct. The output video is 1 minute long but the input video is only about 20 seconds. This is because I had to slow down the frame rate of the output video because unless I slowed it down the correctly detected white box disappears too quickly.

REFERENCE

https://au.mathworks.com/help/matlab/ref/rgb2gray.html

https://au.mathworks.com/help/images/ref/imsharpen.html

https://au.mathworks.com/help/images/ref/bwconncomp.html

https://au.mathworks.com/help/images/ref/imclose.html

https://au.mathworks.com/help/images/ref/imopen.html

https://au.mathworks.com/help/images/ref/imerode.html

https://au.mathworks.com/help/images/ref/imdilate.html

https://github.com/tesseract-ocr/tesseract/blob/master/README.md

Klette, R. (2014). *Concise computer vision: an introduction into theory and algorithms*. Springer.

https://www.scantips.com/simple6.html