

For my image processing extension, I decided to implement something using tesseract that would be able to read the number plates.

I was already happy with the detection, after adding a ratio feature and tinkering with threshold as well as the amount of dilations and erosions. This is why I chose to instead implement a new feature.

I think that something like this would be a very useful addition to have along with the number plate finding. This is because the recognition helps with processing time, as once we have a number plate detected already, it is much easier to read off that small piece rather than the whole screen.

Reading number plates are important for many reasons, such as recording parking times or speeding and similar things.

This is why I chose to make something that reads plates, rather than further improving the detection algorithm.

Firstly, I made a function in the normal detection that can be called from the extension script, which will save the bound box found so that it can be used as the image to read from.

This saving was using the coordinates from the normal detection, and creating a numpy array from the rectangle of coordinates, giving just the number plate section of the image.

With this image, I implemented a tesseract OCR reading on, which would give the numbers and letters of the plate, however, after making this, I realised that this would not run out of the box, as tesseract is something that needs to be fully installed onto someone's computer.

This was a bit of a problem, and I now needed to look for other ways to use tesseract with the number plate images to return the letters and numbers.

I ended up finding an API that I could call to with certain data, including the image. This was very useful as I now had a way to run tesseract out-of-the-box, allowing me to once again process and return the number plate characters to the user.

Some problems I had encountered with using this API, is that it would only accept images that are less than 1MB in size, this was a bit of a challenge for some images, so I used pillow in python to reduce the quality of these images, however keep readable, so that the images become of a smaller file size. Unfortunately, in the event that an image is bigger than 1MB, it will still be unable to go through and this is a limitation of having to use the API, rather than using the local version of tesseract.

After putting together the number plate detection algorithm, exporting the plates, and reading with tesseract, overall I was very pleased with the results and although not perfect, I was happy to see some of the number plates being read accurately.

Instructions for running the extension will be on the next page.

To run extension:

Install requirements.txt

Uncomment lines of CS373LicensePlateDetection.py:

```
13: #import CS373Extension
```

```
382: #CS373Extension.SaveLicensePlateImage(px_array, bbox_min_x, bbox_min_y,  
bbox_max_x, bbox_max_y)
```

```
385: #CS373Extension.CompressImage()
```

```
388: #CS373Extension.PrintPlateFromAPI()
```

Output should be in the terminal :)

I hope you enjoy it!