

Assignment Extension

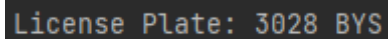
COMPSCI 373, Assignment: License Plate Detection

Cameron Nathan

1. Extension Implementation:

1.1 What does it do?

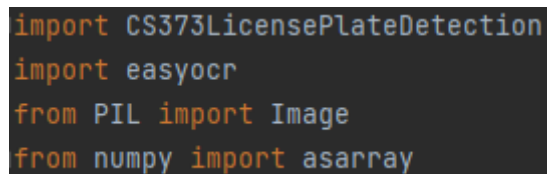
My extension takes a given image of a car with its license plate and builds on the functionality of CS373LicensePlateDetection. This is done by using the calculated bounding box for the related license plate to read the license plate numbers and letters and printing it as a string in the terminal.



```
License Plate: 3028 BYS
```

1.2 Implementation Steps

1. To read a given image and access the text on the license plate I needed to import multiple libraries.



```
import CS373LicensePlateDetection
import easyocr
from PIL import Image
from numpy import ndarray
```

- Pillow was used to read and crop the given image in python
 - easyocr was used to read the license plate text from the given image
 - numpy was used to facilitate use of easyocr
 - CS373LicensePlateDetection (and its own imports) was used to access the bounding box functionality
2. Used the Image import from Pillow to read the given image
 3. Used the functions “readRGBImageToSeparatePixelArrays” and “detectPlate” from the CS373LicensePlateDetection file to get the position of the license plate in the given image.
 4. Get the bounding box position from function “detectPlate” and put the min and max values of x and y into variables
 5. Use the calculated bounding box position with the Image import to crop the given image to just show the license plate
 6. Convert the cropped image to numpy array so that we can utilize easyocr
 7. Read the license plate numbers and letters using the easyocr import and print only the license plate text

2. Experiments:

For extension experimentation, I have tested each of the six license plate images given for the main task.

2.1 Findings:

- numberplate1.png
When using this image, I was able to successfully read the license plate.



License Plate: 3028 BYS

- numberplate2.png
When using this image, I was able to successfully read the license plate.



License Plate: ABC123

- numberplate3.png
When using this image, I did not read the license plate successfully.
Since the license plate has “TEXAS” on the top of the plate and the boundary box of the license plate includes this text, easyocr has treated it as the license plate.



License Plate: TEXAS

- numberplate4.png
When using this image, I did not read the license plate successfully.
The choice of text font and plate size has made certain letters and numbers hard to distinguish. This has led to easyocr treating ‘W’ as ‘H’ and ‘1’ as ‘I’.



License Plate: EHH DVID

- numberplate5.png
When using this image, I did not read the license plate successfully.
Ocr has tried to read the symbols in between 'H' and '7' on the license plate as a character, this has resulted in ocr treating these symbols as a '3'.



License Plate: H3786 P0J

- numberplate6.png
When using this image, I was able to successfully read the license plate.



License Plate: 4898 GXY

2.2 Strengths:

- This method was efficiently implemented as I could use the functions from my implementation of the main task, only needing an extra six lines to read, scan, and access the license plate's text.
- This method for reading license plates was a reliable way of reading plates with a simple design and quality image.

2.3 Limitations:

- The main problem with this method is that it can be hard to read the license plate depending on its design and the quality of images used. This can be shown through our results of the method reading the text surrounding our wanted text (numberplate3), misinterpreting characters (numberplate4), and treating other elements as the number plate's text (numberplate5). We could try to address this problem by including additional image filtering to try and remove some of the unwanted elements affecting the scanning of our wanted text.
- Another thing to consider is that this method of reading the license plates would require the user to manually input the file name of their image directly into the python code.