

CS373 License Plate Detection

The challenging images

An extreme challenge that I have faced during this assignment is that not all the number plate in the different images has the same size. Therefore, I have used the thresholding method to get a higher contrast between the number plate and the outer space. Then I used the compute vertical edge Sobel absolute method before the thresholding method to rid of the horizontal white lines. Afterwards, I used dilation and erosion to magnify and shrink the letters and numbers in the number plate area. Finally, we can detect the number plate by finding the outer margin of the number plate and detecting the numbers and letters for further operation.

The most complex picture during the whole processing is img2, every letter on the number plate is far from each other. This will cause a problem in which we will need to do more dilation operations to magnify that all letters and numbers are connected. We can connect the number plate to the other place of the car if we do too many dilation operations. I have to magnify the numbers and letters on the number plate as large as possible and make sure they are not connected to any other part of the car. This will increase the workload of determining the number of dilation operations done during the whole process. Finally, I find the number of dilation and erosion operations are 17 and 9 suitable for most cases.

Read the license plate letters and numbers - OCR

I have used two different types of methods to read the letters and numbers from the license plate, which is using an API from [Mindee](#) and [Easyocr](#).

Mindee

Mindee is an online API to read the letter and number from an image. By using this API, we have to use a computer with an internet connection.

v1.1 - Deployed



100% Trained on 20 data

Started at 12:12:16 pm June 8th 2022 - Ended at 12:17:52 pm June 8th 2022

This model is trained based on 20 images of number plates. The result of these six images is shown on the right, which offers sufficiently readability of these number plates with higher accuracy. In the model training, I have removed these local area codes from the reading and only show the letters and numbers.

```
['3028', 'BYS']  
['ABC123']  
['OTO', 'BLOG']  
['EWW', 'DVID']  
['786', 'POJ']  
['4898', 'GXY']
```

Easyocr

Easyocr is one of the Python third party libraries. Therefore we can use easyocr offline without connecting to the internet. We can see the result shown on the right, the readability accuracy is lower than the mindee.

```
['3028 BYS'], 'ALA']  
['ABCI23']  
['TEXASD', 'OTO BLOG']  
['EWH DVID']  
['HE786 POJ']  
['L4898 GXY']
```

These two implementations will use the image that I have cut from the connected origin from the image. This will increase the readability as we only focus on the number plate area and cut off the area out of the number plate. AS shown below.



There is an example of detecting the letters and numbers from "numberplate2.png". The results of these two implementations are shown below. Using these two methods will increase the readability of these images.

Final image of detection



```
read image width=960, height=540

Starting read the numbers and letters from the image with API...

The letters and numbers in the number plate is: ABC123

Starting read the numbers and letters from the image with Easyocr...

CUDA not available - defaulting to CPU. Note: This module is much faster with a GPU.
The letters and numbers in the number plate is: ABCI23
```