

# 646 Pattern Recognition & Classification Project

License Plate Number Recognition

With Convolutional neural network

Yufeng Liu

## Abstract:

Nowadays, automobiles almost become basic things for every family in their daily life. Thus, find an efficient way to recognize and classify every automobile is very important, especially in a public place like: garage, highway, etc.

My project is designed to recognize the license plate number of the automobiles' license plates.

## Approaches:

My project's goal is to identify the license plate number. In order to identify the license plate number, it's important to localization the place of licence plate and crop that region. My project only includes the OCR part, thus i only used licence plate picture to train my model.

My training model is mainly built base on the keras library and followed the algorithm of Convolutional neural network and the database from openalpr <https://github.com/openalpr/benchmarks/tree/master/endtoend/>. I trained and tested my model using US part. So it might be more accurate predict the number with us license plate.

```

Loss: 18.1975161
Iter. 79 (of 300000)
    Loss: 20.007284
Iter. 80 (of 300000)
    Loss: 19.621071
Iter. 81 (of 300000)
    Loss: 21.892460
Iter. 82 (of 300000)
    Loss: 18.662472
Iter. 83 (of 300000)
    Loss: 19.363300
Iter. 84 (of 300000)
    Loss: 26.360073
Iter. 85 (of 300000)
    Loss: 20.287767

```

Fig.1 Training Process

After that, I have used my trained model to check the number in the preload test file. In order to load the trained model, I have used darknet libraries.

```

yufeng@yufeng-Lenovo-IdeaPad-S340-15APITouch: ~/project/licensePlateRec
文件(F) 编辑(E) 查看(V) 搜索(S) 终端(T) 帮助(H)
 2 conv    64  3 x 3 / 1  120 x 40 x 32 -> 120 x 40 x 64  0.177 BFLOPs
 3 max      2 x 2 / 2  120 x 40 x 64 ->  60 x 20 x 64
 4 conv   128  3 x 3 / 1   60 x 20 x 64 ->  60 x 20 x 128  0.177 BFLOPs
 5 conv    64  1 x 1 / 1   60 x 20 x 128 ->  60 x 20 x 64  0.020 BFLOPs
 6 conv   128  3 x 3 / 1   60 x 20 x 64 ->  60 x 20 x 128  0.177 BFLOPs
 7 max      2 x 2 / 2   60 x 20 x 128 ->  30 x 10 x 128
 8 conv   256  3 x 3 / 1   30 x 10 x 128 ->  30 x 10 x 256  0.177 BFLOPs
 9 conv   128  1 x 1 / 1   30 x 10 x 256 ->  30 x 10 x 128  0.020 BFLOPs
10 conv   256  3 x 3 / 1   30 x 10 x 128 ->  30 x 10 x 256  0.177 BFLOPs
11 conv   512  3 x 3 / 1   30 x 10 x 256 ->  30 x 10 x 512  0.708 BFLOPs
12 conv   256  3 x 3 / 1   30 x 10 x 512 ->  30 x 10 x 256  0.708 BFLOPs
13 conv   512  3 x 3 / 1   30 x 10 x 256 ->  30 x 10 x 512  0.708 BFLOPs
14 conv    80  1 x 1 / 1   30 x 10 x 512 ->  30 x 10 x 80  0.025 BFLOPs
15 detection
mask scale: Using default '1.000000'
Loading weights from data/ocr/ocr-net.weights...Done!
Performing OCR...
  Scanning test/output/10F759_test.png
    LP: 10F759
  Scanning test/output/ABC1234_test.png
    LP: ABC234
  Scanning test/output/AZM9590_test.png
    LP: AZM9590
yufeng@yufeng-Lenovo-IdeaPad-S340-15APITouch:~/project/licensePlateRec$

```

Fig.2 OCR Detector

The photos I used to test are the front and close view of the license plate which are from google images:



Fig3. Test image example

It will be inaccurate with the picture with the side or unclear view of the license plates.

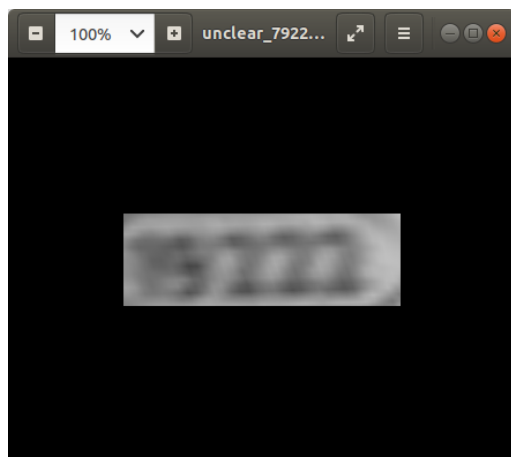


Fig4.Unclear License Plate

```
yufeng@yufeng-Lenovo-IdeaPad-S340-15APITouch: ~/project/licensePlateRec
文件(F) 编辑(E) 查看(V) 搜索(S) 终端(T) 帮助(H)
6 conv 128 3 x 3 / 1 60 x 20 x 64 -> 60 x 20 x 128 0.177 BFLOPs
7 max 2 x 2 / 2 60 x 20 x 128 -> 30 x 10 x 128
8 conv 256 3 x 3 / 1 30 x 10 x 128 -> 30 x 10 x 256 0.177 BFLOPs
9 conv 128 1 x 1 / 1 30 x 10 x 256 -> 30 x 10 x 128 0.020 BFLOPs
10 conv 256 3 x 3 / 1 30 x 10 x 128 -> 30 x 10 x 256 0.177 BFLOPs
11 conv 512 3 x 3 / 1 30 x 10 x 256 -> 30 x 10 x 512 0.708 BFLOPs
12 conv 256 3 x 3 / 1 30 x 10 x 512 -> 30 x 10 x 256 0.708 BFLOPs
13 conv 512 3 x 3 / 1 30 x 10 x 256 -> 30 x 10 x 512 0.708 BFLOPs
14 conv 80 1 x 1 / 1 30 x 10 x 512 -> 30 x 10 x 80 0.025 BFLOPs
15 detection
mask scale: Using default '1.000000'
Loading weights from data/ocr/ocr-net.weights...Done!
Performing OCR...
  Scanning test/output/10F759_test.png
    LP: 10F759
  Scanning test/output/ABC1234_test.png
    LP: ABC1234
  Scanning test/output/AZM9590_test.png
    LP: AZM9590
  Scanning test/output/far_MU1EPSE_test.png
    No characters found
  Scanning test/output/unclear_79222_test.png
    LP: 77
yufeng@yufeng-Lenovo-IdeaPad-S340-15APITouch:~/project/licensePlateRec$
```

Fig5. Results of the unclear and side view of the license plate

The code has been uploaded to <https://github.com/Chappelliu/646project>

## Reference:

1.Silva S.M., Jung C.R. (2018) License Plate Detection and Recognition in Unconstrained Scenarios. In: Ferrari V., Hebert M., Sminchisescu C., Weiss Y. (eds) Computer Vision – ECCV 2018. ECCV 2018. Lecture Notes in Computer Science, vol 11216. Springer, Cham

2.CRNN Recognition <https://github.com/qjadud1994/CRNN-Keras>

3.Text Detector [https://github.com/qjadud1994/Text\\_Detector](https://github.com/qjadud1994/Text_Detector)

