

Exercise 5 Extension Report

Roy 马卓衡 1930026096

Question Restatement

The project extension focuses on improving the accuracy of Chinese car plate detection. And try a newer open-source solution for Chinese car plate recognition. Make a comparison between different methods.

Problem Design and Analysis

For accuracy improvement, I design a residual network model. The accuracy of new model improves slightly in test set. The model architecture is built on two residual blocks (Res1 & Res2), which architecture are shown in the following pictures:

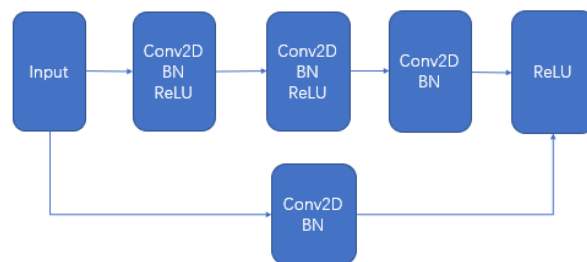


Figure 1: Res1 Architecture

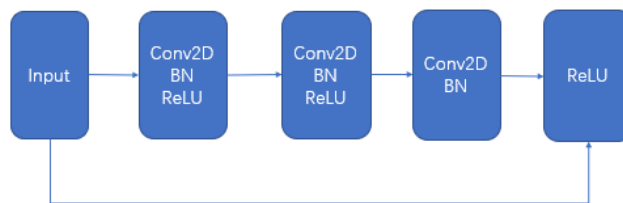


Figure 2: Res2 Architecture

Thus, the overall architecture of residual network is shown in the following picture:

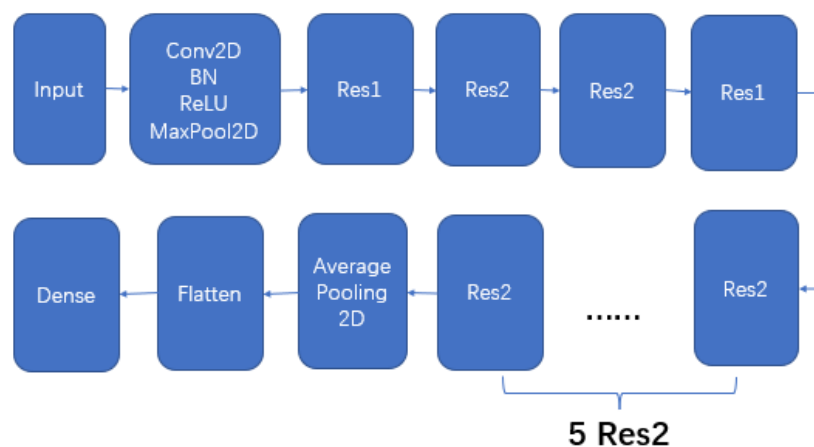


Figure 3: ResNet Architecture

For the open-source solution for Chinese car plate recognition, I apply a third-party package, *hyperlpr*. For the mis-detective picture in self-defined model, *hyperlpr* could return a correct result. However, there are still two out of five car plates that are still not be captured by *hyperlpr*.

In order to perform image enhancement, *cv2.GaussianBlur()* is applied to the retrieved characters. Image enhancing removes noises from original car plate images.

Result Presentation



Figure 4: ResNet Training & Validation Accuracy

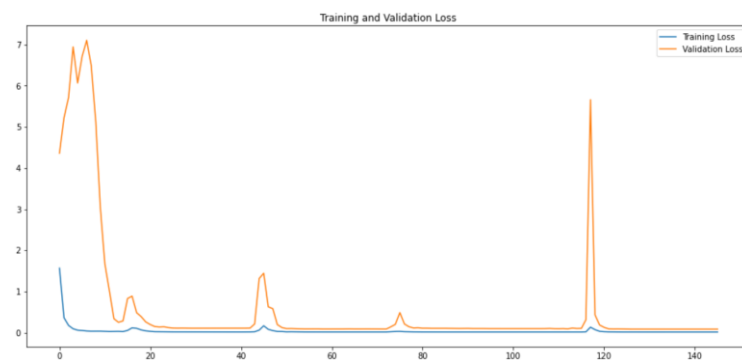


Figure 5: ResNet Training & Validation Loss

```
3231/3231 [=====]
Test loss: 0.08479992873688383
Test accuracy: 0.9845249
```

Figure 6: ResNet Test Accuracy Screenshot

CNN Model	ResNet Model
0.9777	0.9845

Table 1: CNN & ResNet Model Accuracy Comparison

```

(1, 3, 79, 263)
0 0 248 78
[['京H99999', 0.9773347292627607, [0, 0, 248, 78]]]
(1, 3, 125, 402)
0 0 347 95
[['鲁Y44748', 0.9803385393960136, [0, 0, 347, 95]]]
(1, 3, 271, 749)
[]
(1, 3, 54, 179)
0 0 168 53
[['皖AA3610', 0.9772850956235614, [0, 0, 168, 53]]]
(1, 3, 128, 401)
0 0 350 93
[['粤B5PQ23', 0.982789899621691, [0, 0, 350, 93]]]
(1, 3, 157, 424)
0 0 362 104
[['粤新T9048', 0.961929202079773, [0, 0, 362, 104]]]

```

Figure 7 Hyperlpr result Screenshot

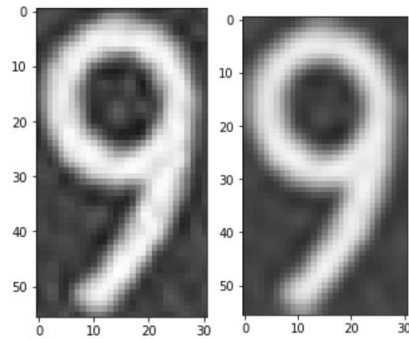


Figure 8: Enhanced images Comparison

Main Problem

Under the character segmentation detection workflow, the core to improve accuracy is to get the correct segments of each character separately. However, the car plate image varies due to environment light, rivets, and other factors, it is hard for algorithm to keep a high accuracy to capture characters, especially, sometimes there are more circle than characters. In order to address this issue, I try to drop frames when it is has a small height or width, or a small aera.

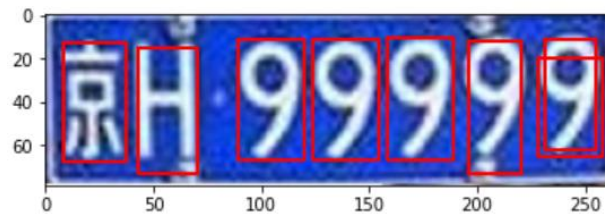


Figure 9: Before Circle Removal

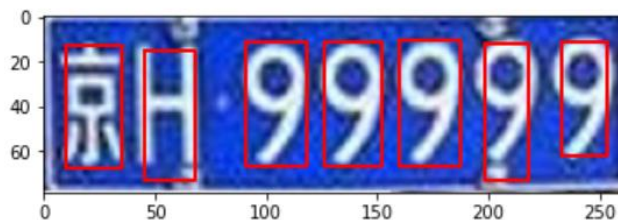


Figure 10: After the Circle Removal