Selected Topics in Visual Recognition using Deep Learning

HW2 - Street View House Numbers detection

陳永承 0712534

Project: https://github.com/axde954e6/NYCU_VRDL/tree/main/HW2

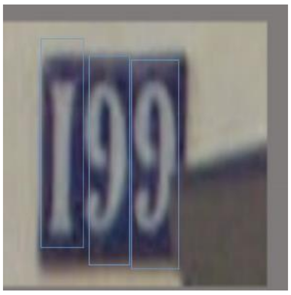
Introduction

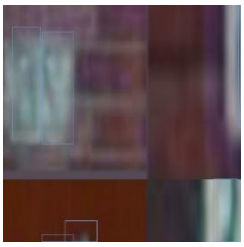
The challenge is to detect Street View House Numbers. Using data provided by TA, which include about 30000 training images and 10000 testing images. In this work, we need to use object detection model to detect the object's location and classify which number of this object.

Methodology

Data Pre-processing

It is because the label format is .mat file, I use MATLAB to generate label to txt file. In this homework, I use yolo v4 as my object detection model. Besides, the backbone of yolo v4 is darknet, which will automatically data augmentation such like random crop and random flip.





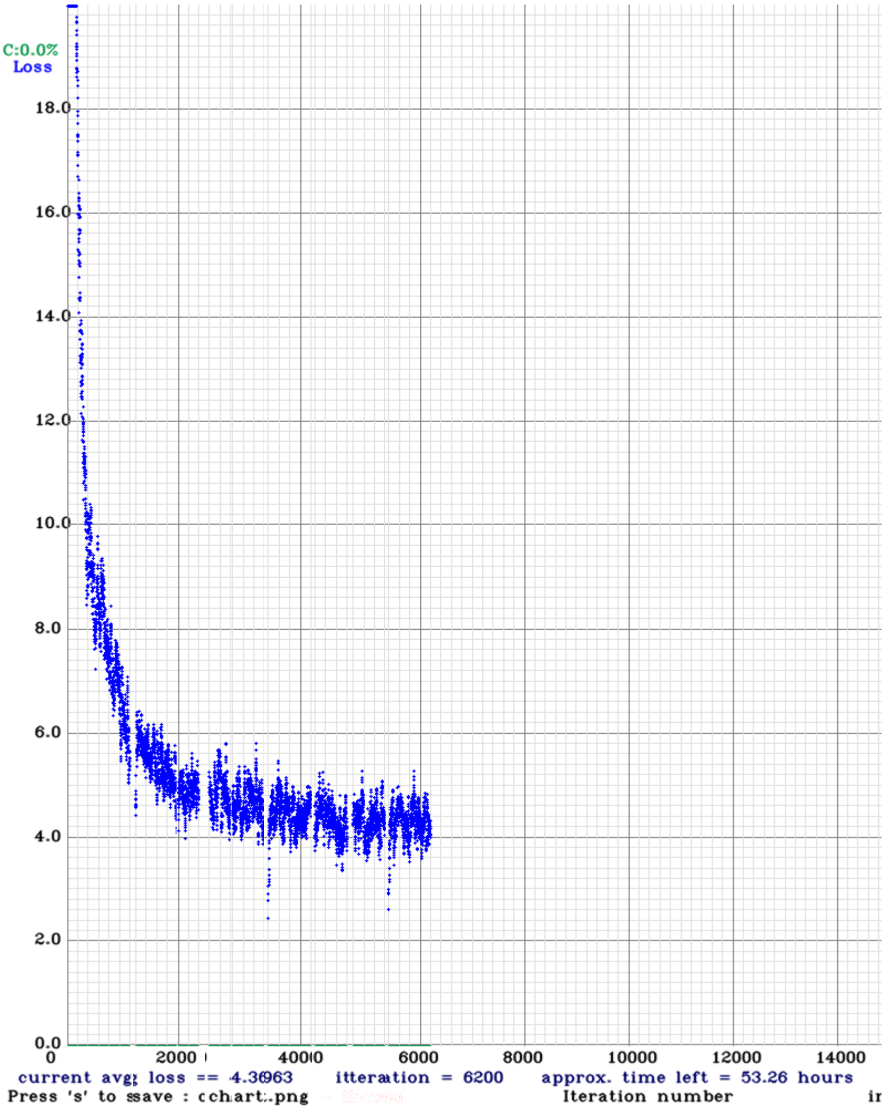
Before training model, I split all training data to training data and validation data at ratio 9:1.

Model Architecture

I use yolo v4 as our Street View House Numbers detection model. To fit our mission, I modify the configuration of default model like the filter in convolution layer before yolo layer and batch size and so on.

Summary

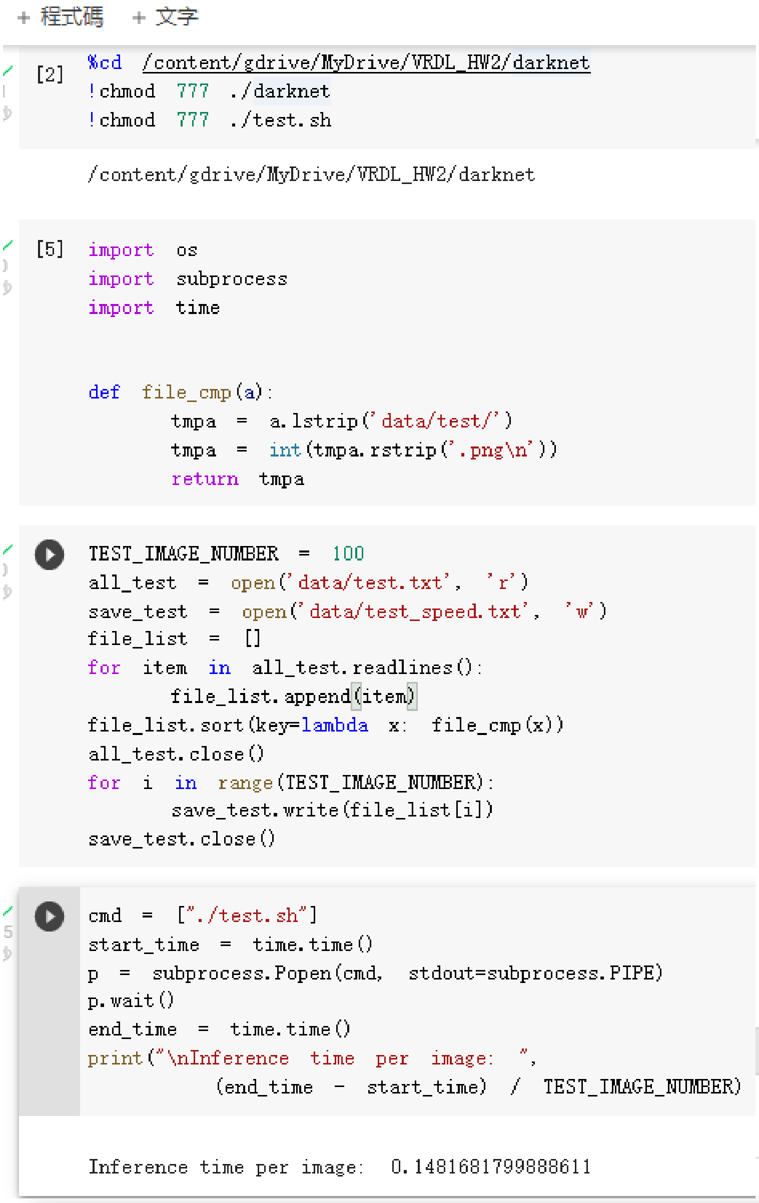
In order to evaluate model, I split training dataset to training data and validation data at ratio 9:1. Below figures are loss to each iteration. I run model on colab, which only can train model at most 6 hour each time and training about 600 iterations, so it is difficult to draw map figure and below figure is collaged.



In this work, it will get about 0.3 map score in yolo v4. I think it will much better if I can train more epochs.

Benchmark your detection model

Inference time per image: 0.1481681799888611 sec



```
[2] %cd /content/gdrive/MyDrive/VRDL_HW2/darknet
    !chmod 777 ./darknet
    !chmod 777 ./test.sh
```

```
/content/gdrive/MyDrive/VRDL_HW2/darknet
```

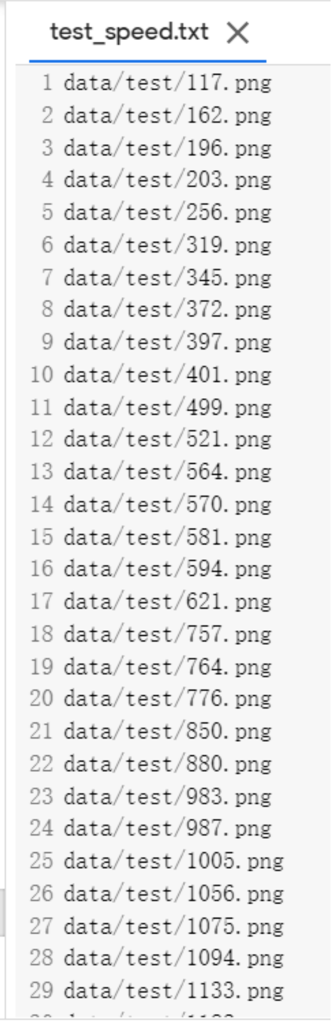
```
[5] import os
    import subprocess
    import time


    def file_cmp(a):
           tmpa = a.lstrip('data/test/')
           tmpa = int(tmpa.rstrip('.png\n'))
           return tmpa
```

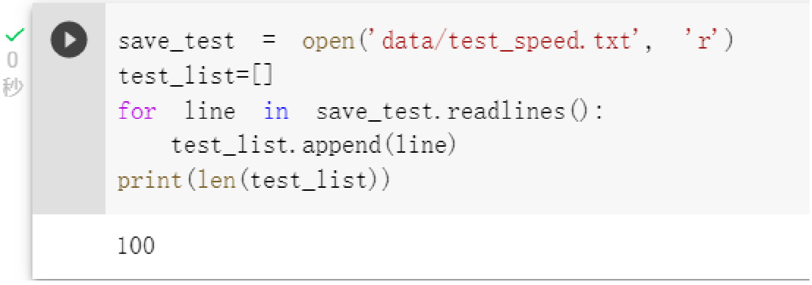
```
TEST_IMAGE_NUMBER = 100
all_test = open('data/test.txt', 'r')
save_test = open('data/test_speed.txt', 'w')
file_list = []
for item in all_test.readlines():
       file_list.append(item)
file_list.sort(key=lambda x: file_cmp(x))
all_test.close()
for i in range(TEST_IMAGE_NUMBER):
       save_test.write(file_list[i])
save_test.close()
```

```
cmd = ["./test.sh"]
start_time = time.time()
p = subprocess.Popen(cmd, stdout=subprocess.PIPE)
p.wait()
end_time = time.time()
print("\nInference time per image: ",
           (end_time - start_time) / TEST_IMAGE_NUMBER)
```

```
Inference time per image:  0.1481681799888611
```



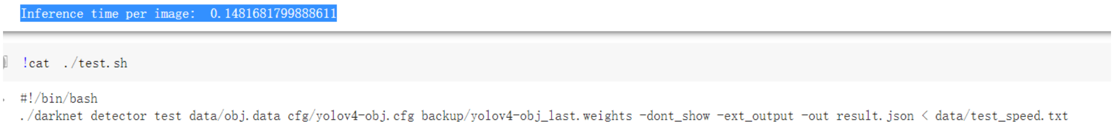
```
data/test/117.png
data/test/162.png
data/test/196.png
data/test/203.png
data/test/256.png
data/test/319.png
data/test/345.png
data/test/372.png
data/test/397.png
data/test/401.png
data/test/499.png
data/test/521.png
data/test/564.png
data/test/570.png
data/test/581.png
data/test/594.png
data/test/621.png
data/test/757.png
data/test/764.png
data/test/776.png
data/test/850.png
data/test/880.png
data/test/983.png
data/test/987.png
data/test/1005.png
data/test/1056.png
data/test/1075.png
data/test/1094.png
data/test/1133.png
```



```
save_test = open('data/test_speed.txt', 'r')
test_list=[]
for line in save_test.readlines():
    test_list.append(line)
print(len(test_list))
```

```
100
```

The number of images.



```
Inference time per image:  0.1481681799888611
```

```
!cat ./test.sh
```

```
#!/bin/bash
./darknet detector test data/obj.data cfg/yolov4-obj.cfg backup/yolov4-obj_last.weights -dont_show -ext_output -out result.json < data/test_speed.txt
```

test.sh

Reference

https://github.com/AlexeyAB/darknet