



Weekly Work Report

Zhang, Liqiang

VISION@OUC

July 29, 2018

1 Research problem

Converting the dataset of contest to VOC2007 format and training the dataset of contest in keras-yolo3.

2 Research approach

3 Research progress

Now, the environment for keras-yolo has been assembled in the server. And the xml files of contest dataset has been generated from the true value table by me.

4 Progress in this week

4.1 Generating the txt files for yolo3 [2]

Before generating these txt files, I should modify the file named voc_annotation.py. Because we have three classes in contest dataset, I should change the classes in voc_annotation.py to seacucumber, seaurchin and scallop. OK, I will get three txt files called 2007_test.txt, 2007_train.txt and 2007_val.txt, which will be used in the process of train

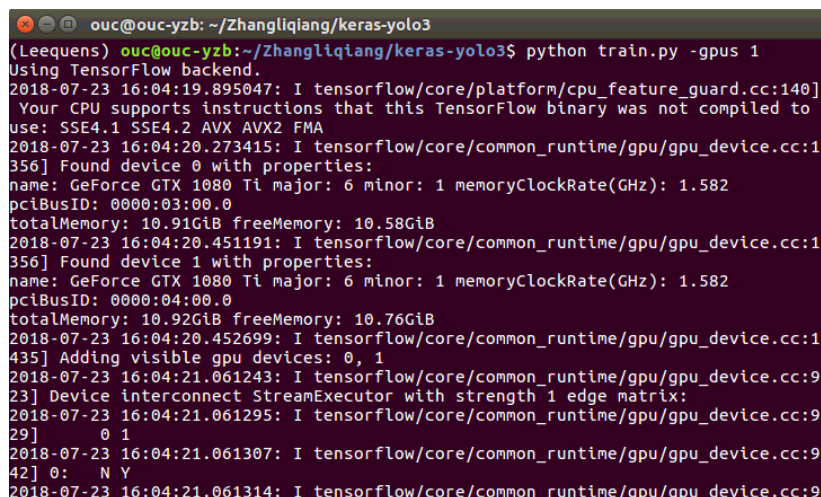
Next, I replaced the train.py file with online tutorials, and run the following code.

```
1 source activate py36 # Geting into the environment
2 python train.py -gpus 1 # Using the GPU 1.
```

Though I use the command -gpus 1, the CPU is still used in the training model. I go online to find the relevant information, someone says it will work if I code to force GPU in front of train.py. But it will generate a new error that I can't modify the value.

I query this error online and find that there is very little information available. I have no way to go but rebuild a new environment, it takes me 10 minutes to build a new environment named Leequens on conda and try to run the code:

```
1 source activate Leequens # Geting into the
2 python train.py -gpus 1 # Using the GPU 1.
```



```
ouc@ouc-yzb: ~/Zhangliqiang/keras-yolo3
(Leequens) ouc@ouc-yzb:~/Zhangliqiang/keras-yolo3$ python train.py -gpus 1
Using TensorFlow backend.
2018-07-23 16:04:19.895047: I tensorflow/core/platform/cpu_feature_guard.cc:140]
Your CPU supports instructions that this TensorFlow binary was not compiled to
use: SSE4.1 SSE4.2 AVX AVX2 FMA
2018-07-23 16:04:20.273415: I tensorflow/core/common_runtime/gpu/gpu_device.cc:1
356] Found device 0 with properties:
name: GeForce GTX 1080 Ti major: 6 minor: 1 memoryClockRate(GHz): 1.582
pciBusID: 0000:03:00.0
totalMemory: 10.91GiB freeMemory: 10.58GiB
2018-07-23 16:04:20.451191: I tensorflow/core/common_runtime/gpu/gpu_device.cc:1
356] Found device 1 with properties:
name: GeForce GTX 1080 Ti major: 6 minor: 1 memoryClockRate(GHz): 1.582
pciBusID: 0000:04:00.0
totalMemory: 10.92GiB freeMemory: 10.76GiB
2018-07-23 16:04:20.452699: I tensorflow/core/common_runtime/gpu/gpu_device.cc:1
435] Adding visible gpu devices: 0, 1
2018-07-23 16:04:21.061243: I tensorflow/core/common_runtime/gpu/gpu_device.cc:9
23] Device interconnect StreamExecutor with strength 1 edge matrix:
2018-07-23 16:04:21.061295: I tensorflow/core/common_runtime/gpu/gpu_device.cc:9
29]    0 1
2018-07-23 16:04:21.061307: I tensorflow/core/common_runtime/gpu/gpu_device.cc:9
42]    0:  N Y
2018-07-23 16:04:21.061314: I tensorflow/core/common_runtime/gpu/gpu_device.cc:9
```

Figure 1: The demo.

OK, it works by GPU as Fig. ???. When I finished my meal and returned to the lab, I find that the loss has not been reduced, but it has increased. I looked at train.py and find that learning rate is 0.1 which

is so big, it is modified to 0.001 by me because 0.001 is a classical value. I hope that loss will perform well this time.

In the morning on Tuesday, I find that the decline of loss is not obvious, after 78 epoch times, loss is about 85 which is not the result I want. Normally, the value of loss should be in decline. But I really can't figure out what's wrong, I have to wait and see whether the value of loss will be accepted in the later training.

In the afternoon, the senior Zhangshaoyong plan to check our dataset and find that the ratio of training set and test set of our dataset is not suitable. We should make the ratio of training set to test machine 10 to 1. So I rearranged the data set according to the requirements.

4.2 Arranging the dataset

I rearranged the catalog of the data set according to the requirements. In the folder of JPEGImages, I divide it into two sub folders, train and test respectively. In train folder, there have five sub folders which correspond to five segments of video. And there is one folder contain one segments of video. It's the same as JPEGImages, the folder of Annotations still have two sub folders called train and test as Fig. 2. The name of image should be renamed from 000000.jpg to xxxxxx.jpg in every sub folder.



Figure 2: The structure of data set.

Next, I must think how to generate the files named train.txt, test.txt and val.txt. It seems so hard because there are many same name in five sub folders and the content of train just contain the name of images without path in the VOC dataset. I must add the path to new train.txt so I it is a little complex. Because if I use the new train.txt, I must modify the file voc_label.py which can convert the train.txt to 2007_train.txt. OK, I can only do it step by step. It is easy to generate train files before, and just input the names by the code. But now I must add the path. it could not be work only by myself, so I go to Internet search for some information. People with ability are everywhere. I have found a way to create a path file with only one code.

```
1 ls -R /home/ouc/Zhangliqiang/datasets/Image/G0024172/*.jpg > train1.txt
2 ls -R /home/ouc/Zhangliqiang/datasets/Image/G0024173/*.jpg > train2.txt
3 ls -R /home/ouc/Zhangliqiang/datasets/Image/G0024174/*.jpg > train3.txt
4 ls -R /home/ouc/Zhangliqiang/datasets/Image/YDXJ0003/*.jpg > train4.txt
5 ls -R /home/ouc/Zhangliqiang/datasets/Image/YDXJ0013/*.jpg > train5.txt
6 ls -R /home/ouc/Zhangliqiang/datasets/Image/YDXJ0012/*.jpg > test.txt
```

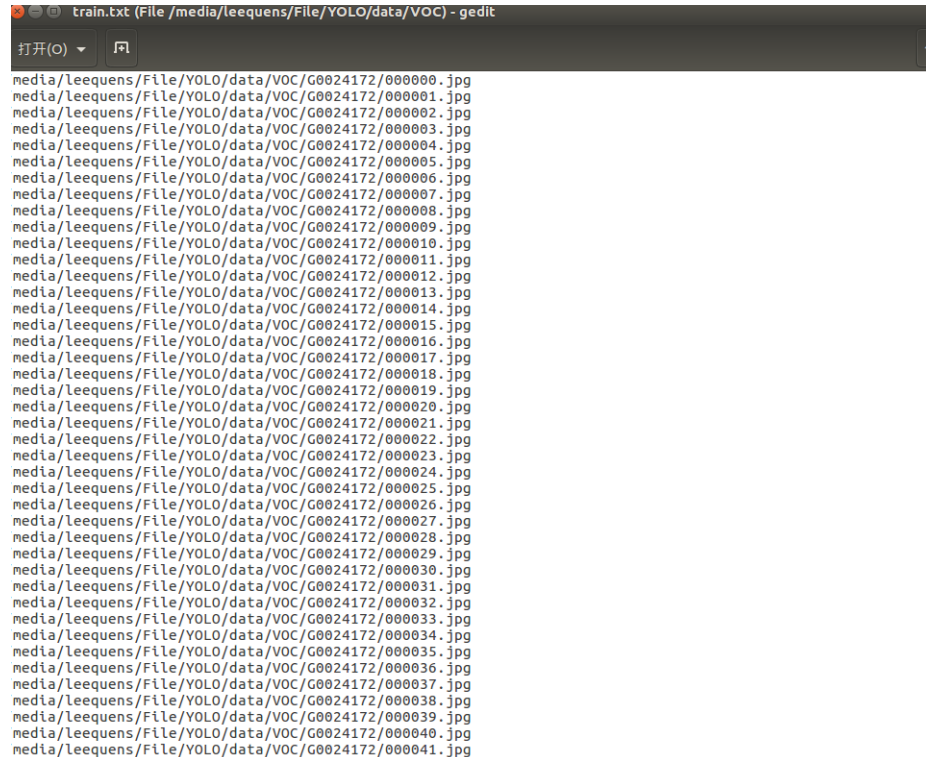
Using the code above, I get five train files and one test files. Now what I am facing is how to concentrate these five files into one document, which is a easy job, I just need run the following code:

```

1 cat train1.txt >>train.txt
2 cat train2.txt >>train.txt
3 cat train3.txt >>train.txt
4 cat train4.txt >>train.txt
5 cat train5.txt >>train.txt

```

The content of new train.txt as Fig.3, which contain all the path information of every training image. Next, I need use voc.label.py to convert the train.txt to 2007_train.txt which can be used in YOLO training. But it will report errors if I run the python voc.label.py. It takes me two hours to modify it and search information, but I don't have any results.



```

media/leequens/File/YOLO/data/VOC/G0024172/000000.jpg
media/leequens/File/YOLO/data/VOC/G0024172/000001.jpg
media/leequens/File/YOLO/data/VOC/G0024172/000002.jpg
media/leequens/File/YOLO/data/VOC/G0024172/000003.jpg
media/leequens/File/YOLO/data/VOC/G0024172/000004.jpg
media/leequens/File/YOLO/data/VOC/G0024172/000005.jpg
media/leequens/File/YOLO/data/VOC/G0024172/000006.jpg
media/leequens/File/YOLO/data/VOC/G0024172/000007.jpg
media/leequens/File/YOLO/data/VOC/G0024172/000008.jpg
media/leequens/File/YOLO/data/VOC/G0024172/000009.jpg
media/leequens/File/YOLO/data/VOC/G0024172/000010.jpg
media/leequens/File/YOLO/data/VOC/G0024172/000011.jpg
media/leequens/File/YOLO/data/VOC/G0024172/000012.jpg
media/leequens/File/YOLO/data/VOC/G0024172/000013.jpg
media/leequens/File/YOLO/data/VOC/G0024172/000014.jpg
media/leequens/File/YOLO/data/VOC/G0024172/000015.jpg
media/leequens/File/YOLO/data/VOC/G0024172/000016.jpg
media/leequens/File/YOLO/data/VOC/G0024172/000017.jpg
media/leequens/File/YOLO/data/VOC/G0024172/000018.jpg
media/leequens/File/YOLO/data/VOC/G0024172/000019.jpg
media/leequens/File/YOLO/data/VOC/G0024172/000020.jpg
media/leequens/File/YOLO/data/VOC/G0024172/000021.jpg
media/leequens/File/YOLO/data/VOC/G0024172/000022.jpg
media/leequens/File/YOLO/data/VOC/G0024172/000023.jpg
media/leequens/File/YOLO/data/VOC/G0024172/000024.jpg
media/leequens/File/YOLO/data/VOC/G0024172/000025.jpg
media/leequens/File/YOLO/data/VOC/G0024172/000026.jpg
media/leequens/File/YOLO/data/VOC/G0024172/000027.jpg
media/leequens/File/YOLO/data/VOC/G0024172/000028.jpg
media/leequens/File/YOLO/data/VOC/G0024172/000029.jpg
media/leequens/File/YOLO/data/VOC/G0024172/000030.jpg
media/leequens/File/YOLO/data/VOC/G0024172/000031.jpg
media/leequens/File/YOLO/data/VOC/G0024172/000032.jpg
media/leequens/File/YOLO/data/VOC/G0024172/000033.jpg
media/leequens/File/YOLO/data/VOC/G0024172/000034.jpg
media/leequens/File/YOLO/data/VOC/G0024172/000035.jpg
media/leequens/File/YOLO/data/VOC/G0024172/000036.jpg
media/leequens/File/YOLO/data/VOC/G0024172/000037.jpg
media/leequens/File/YOLO/data/VOC/G0024172/000038.jpg
media/leequens/File/YOLO/data/VOC/G0024172/000039.jpg
media/leequens/File/YOLO/data/VOC/G0024172/000040.jpg
media/leequens/File/YOLO/data/VOC/G0024172/000041.jpg

```

Figure 3: The content of train.txt.

4.3 Beginning to train [1]

When I have any method to solve this problem I look over the old file named 2007_train.txt. I find it is similar to new train.txt generated by me. but 2007_train.txt have the coordinate information. I want to try to use train.txt replace it to train. If YOLO can begin to train, which shows that this method is effective.

I implemented this method and it begin to train successfully as Fig. 4, the value of loss also begin to decline. But I afraid that it will generate a module which can not test because my train.txt have not coordinate information. It still need three hours to complete this training when write this report, I hope that I will get a good result.

5 Modifying the code

Well... I get a bad result. This competition is for us to test sea cucumbers, sea urchins, scallops, but I use my module to have a test with test set. I find that my module can distinguish the starfish without

```
ouc@ouc-yzb: ~/Zhangliqiang/kerasyolo3/keras-yolo3
Load weights model_data/yolo_weights.h5.
Freeze the first 249 layers of total 252 layers.
Train on 17638 samples, val on 17 samples, with batch size 32.
Epoch 1/50
551/551 [=====] - 1926s 3s/step - loss: 152.1210 - val_loss: 16.0117
Epoch 2/50
551/551 [=====] - 1924s 3s/step - loss: 14.8595 - val_loss: 11.8532
Epoch 3/50
551/551 [=====] - 1928s 3s/step - loss: 11.8636 - val_loss: 10.6153
Epoch 4/50
551/551 [=====] - 1882s 3s/step - loss: 10.8846 - val_loss: 10.1918
Epoch 5/50
551/551 [=====] - 1914s 3s/step - loss: 10.4258 - val_loss: 10.0345
Epoch 6/50
551/551 [=====] - 1916s 3s/step - loss: 10.1728 - val_loss: 9.9012
Epoch 7/50
551/551 [=====] - 1919s 3s/step - loss: 10.0143 - val_loss: 9.8216
Epoch 8/50
551/551 [=====] - 1906s 3s/step - loss: 9.9135 - val_loss: 9.7662
Epoch 9/50
551/551 [=====] - 1910s 3s/step - loss: 9.8422 - val_loss: 9.7392
Epoch 10/50
551/551 [=====] - 1897s 3s/step - loss: 9.7959 - val_loss: 9.7217
Epoch 11/50
551/551 [=====] - 1916s 3s/step - loss: 9.7621 - val_loss: 9.7008
Epoch 12/50
551/551 [=====] - 1913s 3s/step - loss: 9.7377 - val_loss: 9.6937
Epoch 13/50
551/551 [=====] - 1916s 3s/step - loss: 9.7185 - val_loss: 9.6887
Epoch 14/50
551/551 [=====] - 1916s 3s/step - loss: 9.7055 - val_loss: 9.6767
Epoch 15/50
551/551 [=====] - 1921s 3s/step - loss: 9.6951 - val_loss: 9.6768
```

Figure 4: Begning to train.

any sea cucumbers, sea urchins and scallops. I am so crumble that I want to smash the computer. Calm down and think about it, it must have some problem in train.txt which have not coordinate information.

I can only sort out the data and make the image stored in one folder. Only in this way can I generate a train.txt which contain coordinte information. I afraid that I spend one day to train my module but it can not work, so I make the epoch be equal to 20.

When it finish I input a image to see if I could test it and it works! But it have a bad effect. I think the reason is that epoch is so small. So, I've added epoch to 200 and begin to train!

6 Testing

After 150 times of epoch, I find that the value of loss has been floating around 75, which is too big. I took out a temporary model to test the pictures with an anxious mood because I think the result will not good. Indeed, The result is so bad as Fig. 5 to Fig. 8.

I don't know where something went wrong, which makes the value of loss is so high and have such a poor detection effect. The next task is to solve this problem.

7 Plan

Objective: Solving the problem of high loss and completing mAP calculation.

Deadline: 2018.08.05.

2018.07.30—2018.08.02 Solving the problem of high loss.

2018.08.02—2018.08.05 Completing mAP calculation.

References

- [1] CSDN. https://blog.csdn.net/Patrick_Lxc/article/details/80615433. 3
- [2] YOLO. <https://pjreddie.com/darknet/yolo/>. 1

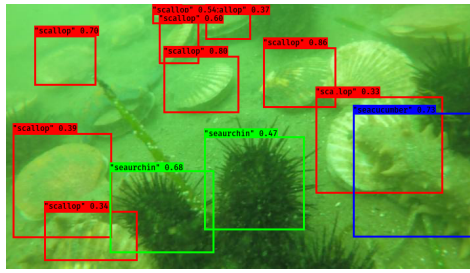


Figure 5: Sample.



Figure 6: Sample.

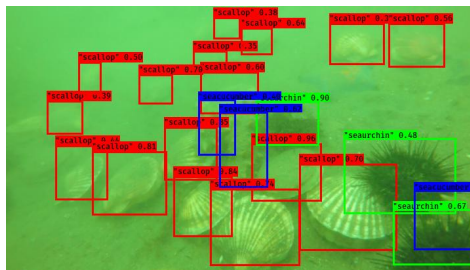


Figure 7: Sample.

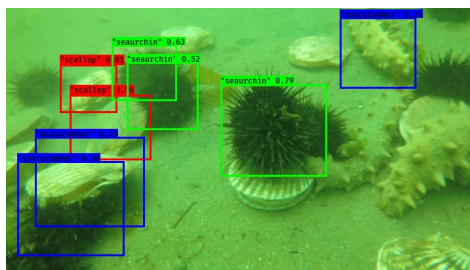


Figure 8: Sample.