

实验环境

- CPU: Intel i7-6850K
- 内存: 32GB
- 显卡: Nvidia GTX-1080Ti
- 操作系统: Ubuntu 16.04
- Python版本: Python 3.5
- 库
 - numpy 1.14.3
 - tensorflow 1.8.0
 - keras 2.1.6
 - cv2 3.4.0
 - pandas 0.23.0
 - sklearn 0.19.1
 - pickle
 - time
 - random
 - matplotlib 2.2.2
 - PIL 5.1.0
 - multiprocessing
 - glob
 - os

额外支持材料

本项目中对数据集进行了增广，增广后的数据集太大，暂时无法上传。但是，增广数据集的程序在 jupyter notebook中已经给出，同时在报告中也给出了部分样例。

由于涉及实验较多，tensorboard的记录文件较大，暂时无法上传。但是在报告中给出了主要结果可视化的截图。

fine-tuning后的VGG模型文件也较大，暂时无法上传。

代码运行

对于基准神经网络，在原始数据集上，按照训练/测试集划分，训练集上一个epoch的运行时间约为18秒。

对于VGG-16迁移学习，在原始数据集上，按照训练/测试集划分，训练集上一个epoch的运行时间约为248秒。

对于VGG-16迁移学习，在增广的数据集上，按照训练/验证集划分，训练集上一个epoch的运行时间约为511秒。

代码运行情况在jupyter notebook中给出，这里给出部分截图。

```
import os
test_prediction = pd.DataFrame(columns=['img','c0','c1','c2','c3','c4','c5','c6','c7','c8','c9'])
while True:
    try:
        filename, value, prediction = sess.run([filename, y_predict, feed_dict={train_flag:False}])
        filename_value = [os.path.split(name)[1].decode('utf8') for name in filename_value]
        test_prediction.append(pd.DataFrame(np.concatenate((np.array(filename_value)), np
        columns=['img','c0','c1','c2','c3','c4','c5','c6','c7','c8','c9']
    except tf.errors.OutOfRangeError:
        break
    test_prediction.to_csv('./test_results' + save_str + '.csv', index = False)
    print("Fold {} test result saved, test time: {}".format(fold, time.time() - start_time))

# save model
saver = tf.train.Saver()
saver.save(sess, './model/' + save_str + '.ckpt')

*****
Fold: 1
*****
作为验证集的驾驶员分类:
['p039', 'p061', 'p081', 'p064', 'p075']
训练集大小: 36322
验证集大小: 3917
INFO:tensorflow:Restoring parameters from ./vgg_16.ckpt
Epoch: 0, Step: 568, Loss: 0.0214, acc: 0.9943
Epoch: 0, train time consumption: 511.50
Epoch: 0, Step: 568, valid Loss: 0.5231, valid acc: 0.8222
Epoch: 0, valid time consumption: 12.65
Epoch: 1, Step: 1136, Loss: 0.0036, acc: 0.9989
Epoch: 1, train time consumption: 511.49
Epoch: 1, Step: 1136, valid Loss: 0.5386, valid acc: 0.8473
Epoch: 1, valid time consumption: 12.62
Epoch: 2, Step: 1704, Loss: 0.0016, acc: 0.9995
Epoch: 2, train time consumption: 511.77
Epoch: 2, Step: 1704, valid Loss: 0.4219, valid acc: 0.8781
Epoch: 2, valid time consumption: 12.69

In [ ]: 1 average_list = [1,2,3,4,5]
2 for i, idx in zip(range(len(average_list)), average_list):
3     tmp = pd.read_csv('./test_results_vgg_aug_fine-tuning_lr_1e-5_kp_0.4_fold{}.csv'.format(idx))
4     if i == 0:
5         concat_csv = tmp
6     else:
7         concat_csv = pd.concat([concat_csv, tmp])
8
9 average_list = [1,2,3,4,5]
10 for i, idx in zip(range(len(average_list)), average_list):
11     tmp = pd.read_csv('./test_results_vgg_aug_fine-tuning_lr_1e-5_kp_0.3_fold{}.csv'.format(idx))
12     concat_csv = pd.concat([concat_csv, tmp])
13
```

另外，报告中也给出了部分运行结果在tensorboard中可视化的截图。例如

