# DeepTest: Automated testing of deep-neural-network-driven autonomous cars

**评估模型对现有测试数据的准确性**

**1.数据集文件**

### 从 HMB\_3.bag 生成 hmb3 数据集



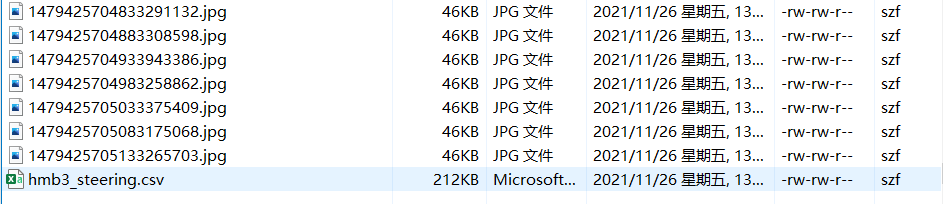
mkdir hmb3

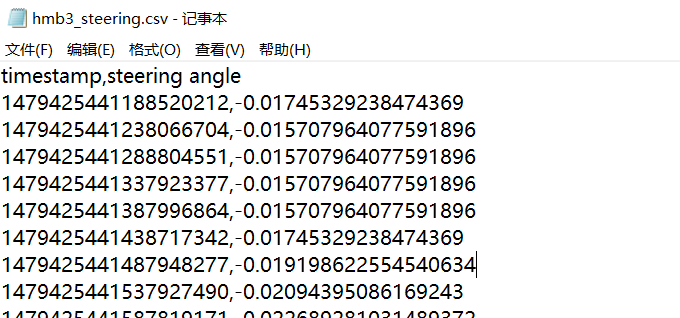
python generate\_hmb3.py --bagfile HMB\_3.bag

生成测试数据集文件

包含每个时间戳的汽车照片，以及转向角。

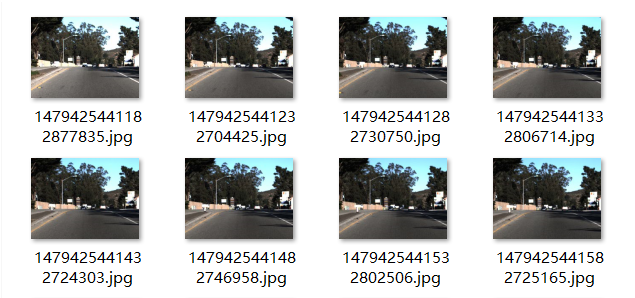




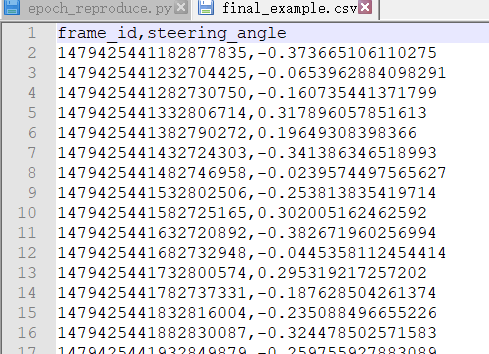


转向角定义为车辆航向(垂直线)与方向盘轴(通常为前轮)航向之间的旋转程度。负值表示左转，正值表示左转，使用的Udacity自动驾驶挑战数据集的最大转向角度为+/- 25°[15]。然后转向角度被缩放1/25，因此预测应该落在-1到1之间。

最终的[challenge2 的](https://hub.fastgit.org/udacity/self-driving-car/tree/master/datasets/CH2)最终测试数据照片Ch2\_001/center/



以及标签



Ch2\_001/center/ 为汽车录制的视频中的每一帧，文件名为帧id

Steering\_angle:转向角

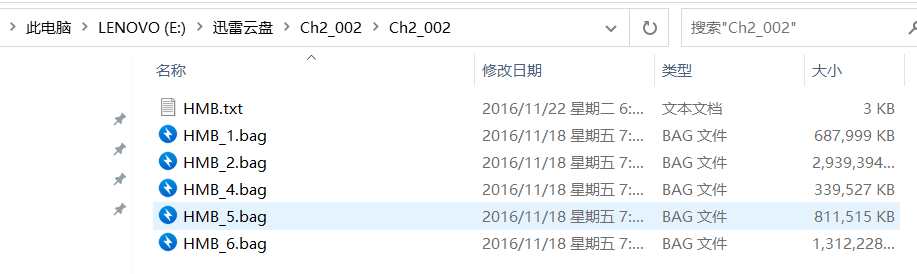
/Dataset/  
├── hmb3/  
        └── hmb3\_steering.csv  
        └── images  
└── Ch2\_001/  
        └── center/  
                 └── images  
        └── CH2\_final\_evaluation.csv

**2代码目录**

Rambo model由三个CNN组成，其输出使用最后一层合并。其中两个CNN的灵感来自NVIDIA的自动驾驶汽车架构，第三个CNN是基于逗号。Ai的转向模型是，与其他模型不同的是，蓝博采用的是三幅连续图像之间的差异作为输入。该模型使用Keras和Theano框架。

Chauffeur model包括一个提取图像特征的CNN模型和一个预测转向角的LSTM模型。CNN模型的输入是一幅图像，而LSTM模型的输入是CNN模型从之前的100幅连续图像中提取的100个特征的拼接。Chauffeur使用Keras和Tensorflow框架。

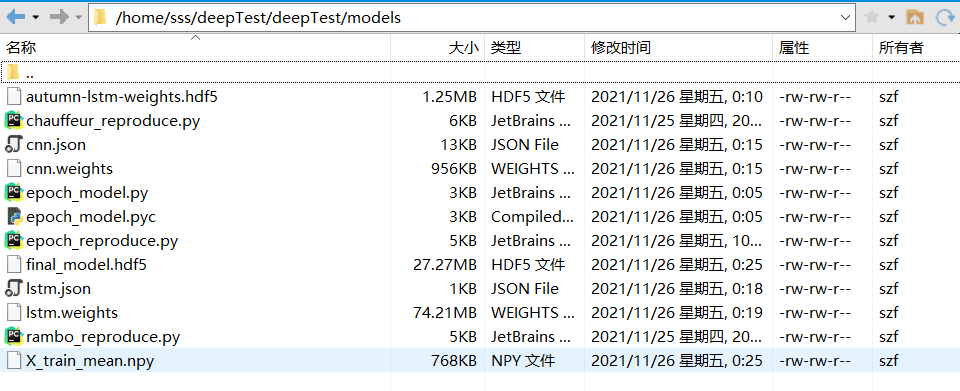
Epoch model只使用一个CNN。由于Epoch的预训练模型不是公开可用的，代码使用作者提供的说明来训练模型,使用来自Udacity自动驾驶挑战的CH2\_002数据集（如下图所示）来训练epoch模型。Epoch与Chauffeur类似，使用Keras和Tensorflow框架。



2.1 [models/](https://hub.fastgit.org/ARiSE-Lab/deepTest/blob/master/models)目录 **评估模型对现有测试数据的准确性**

Rambo、Chauffeur 和 Epoch 模型重现 Udacity 自动驾驶汽车 Challenge2结果。

三种车型的测试代码以及需要的权重文件如下图所示。



（1）

cd models/

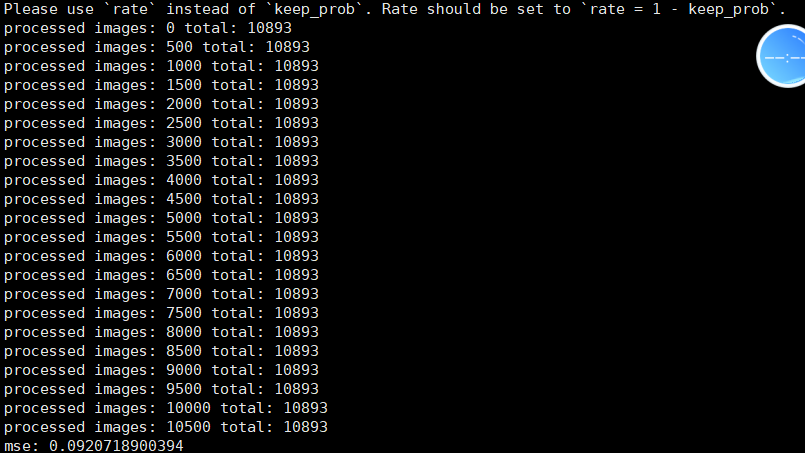
python epoch\_reproduce.py --dataset DATA\_PATH/Dataset/

该模型缺少weights\_HMB\_2.hdf5模型权重文件，预训练模型不是公开可用的，可以使用来自Udacity自动驾驶挑战的CH2\_002数据集来训练epoch模型，因为设备性能有限，故此处不再测试。

(2) chauffeur test

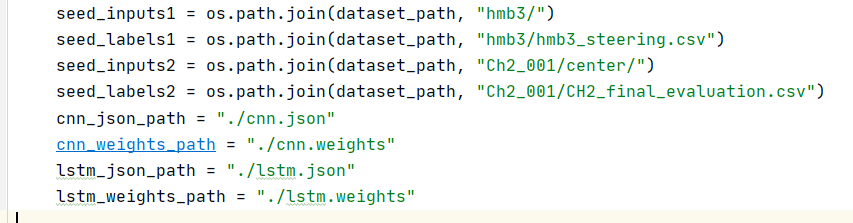
python chauffeur\_reproduce.py --dataset DATA\_PATH/Dataset/

运行视频见文件/model test/ chauffeur test.mp4



说明：

权重数据集文件：



seed\_image = cv2.imread(os.path.join(seed\_inputs2, f)) #读取测试照片  
yhat = model(seed\_image)#模型进行label预测  
yhats.append(yhat)#存储预测label  
labels.append(truth[f])#真实label

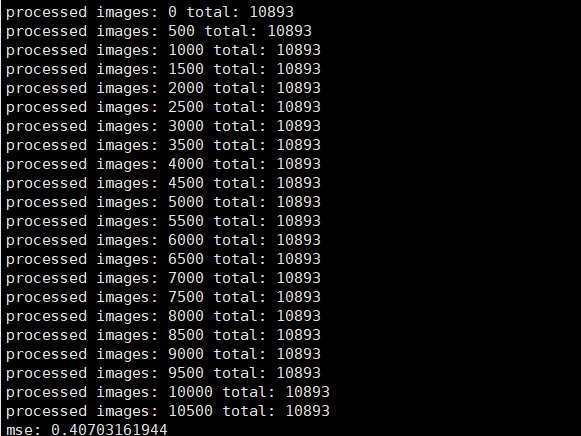
mse = calc\_rmse(yhats, labels) #计算MSE



(3) Rambo test

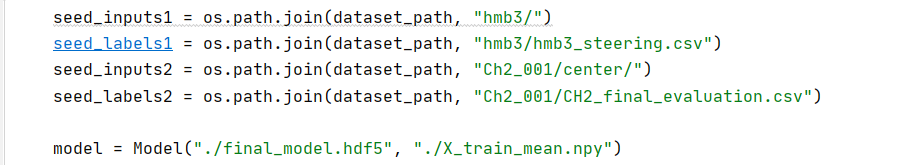
python rambo\_reproduce.py --dataset DATA\_PATH/Dataset/

运行视频见文件/model test/ Rambo test.Mp4

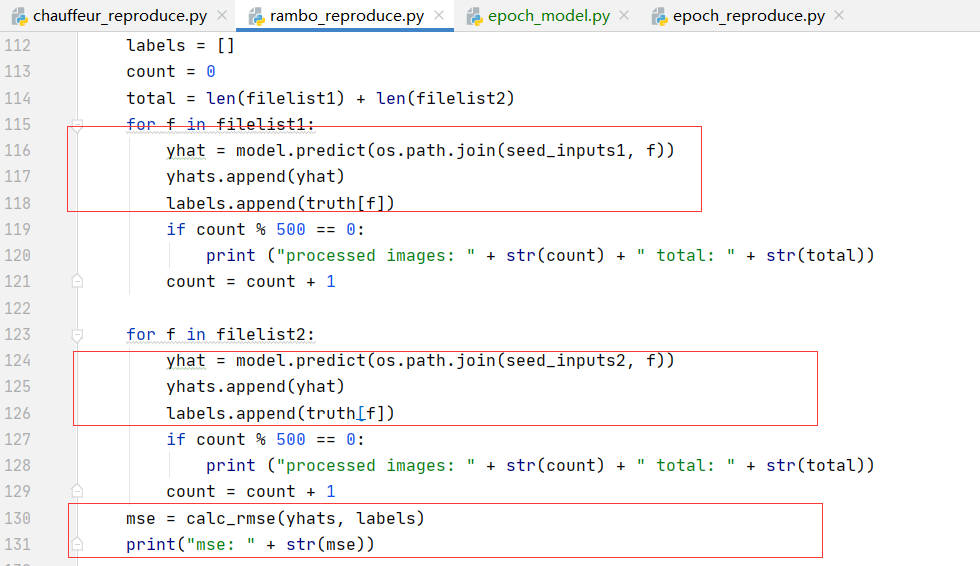


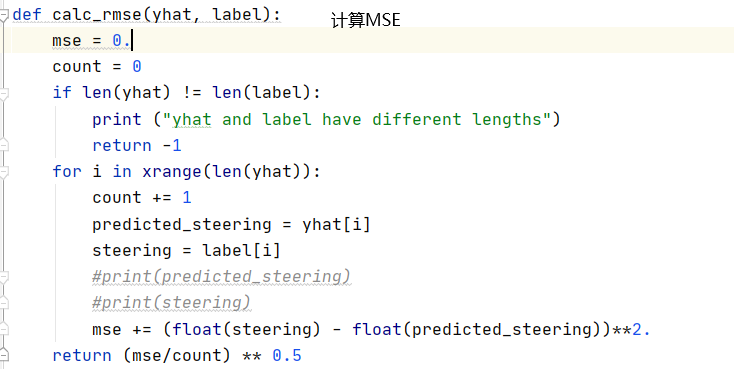
说明：

加载权重以及数据集文件：



for f in filelist1:#加载image  
 yhat = model.predict(os.path.join(seed\_inputs1, f))#模型预测  
 yhats.append(yhat)#预测标签  
 labels.append(truth[f])#真实标签  
 if count % 500 == 0:  
 print ("processed images: " + str(count) + " total: " + str(total))  
 count = count + 1





2.2 目录/testgen 生成合成图像并计算累积神经元覆盖率

作用：生成合成图像，计算累积覆盖率并记录预测输出。

[ncoverage.py](https://hub.fastgit.org/ARiSE-Lab/deepTest/blob/master/testgen/ncoverage.py)：定义用于计算神经元覆盖率的 NCoverage 类

/testgen 权重以及测试文件



(1）epoch test

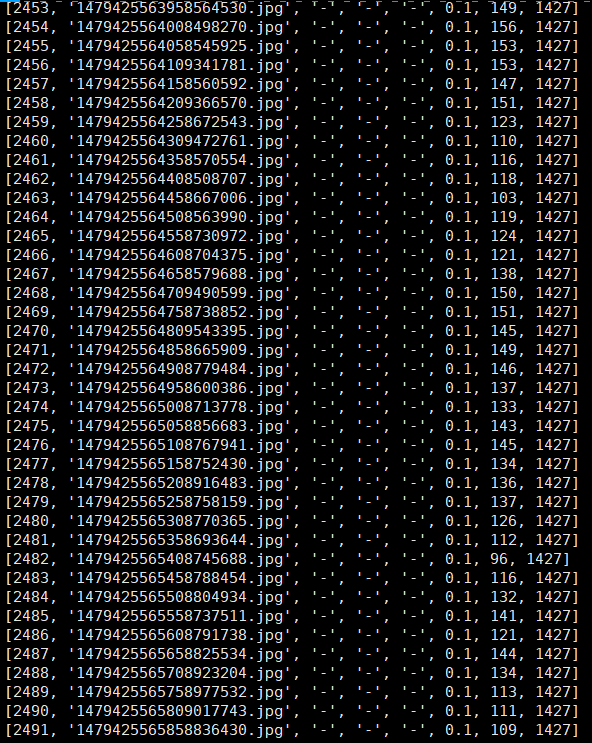
cd testgen/

./epoch\_testgen\_driver.sh 'DATA\_PATH/Dataset/'

该模型缺少weights\_HMB\_2.hdf5模型权重文件，预训练模型不是公开可用的，因为设备性能有限，故此处不再测试。

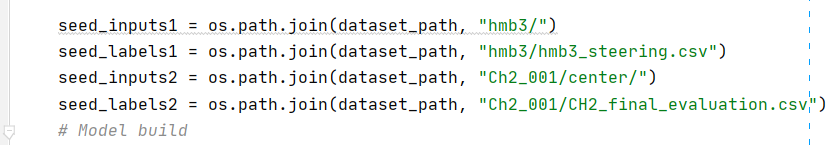
(2) chauffeur test

./chauffeur\_testgen\_driver.sh 'DATA\_PATH/Dataset/'

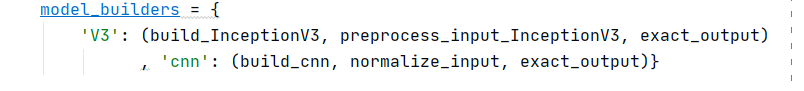


运行视频见文件/ testgen test/ chauffeur test.mp4

加载数据集：



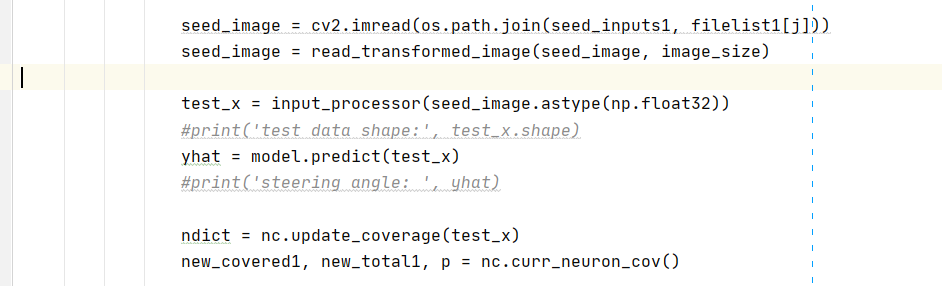
构建模型的参数

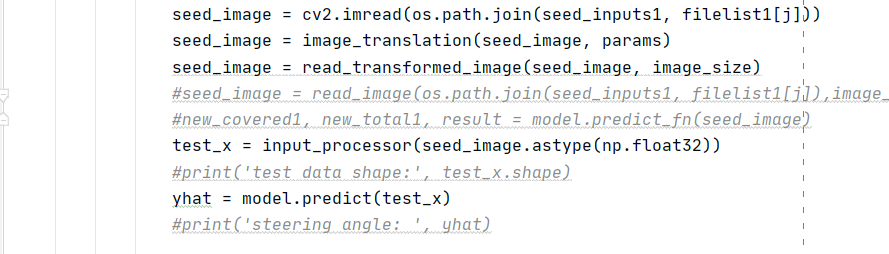


加载以及构建模型

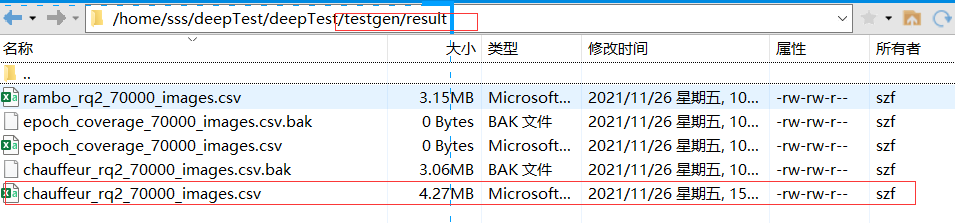


预测





生成的合成图像文件



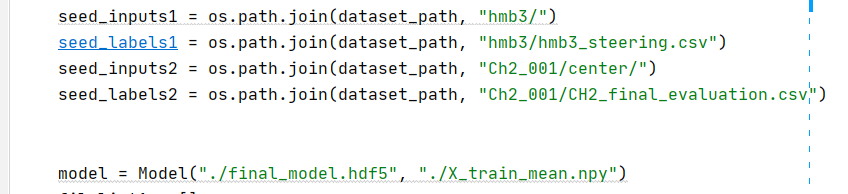
(3) rambo test

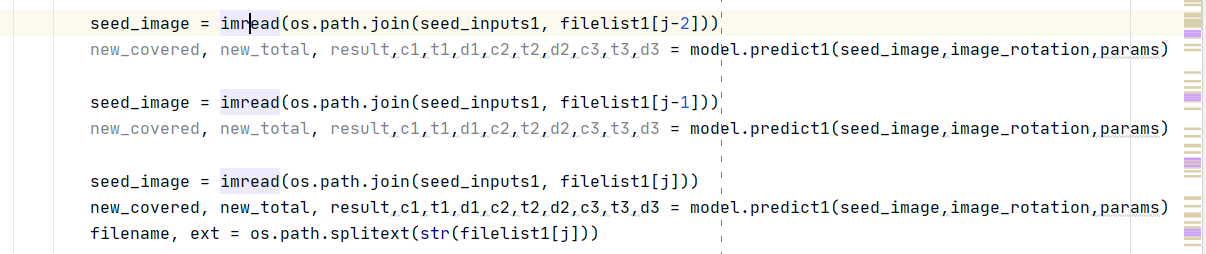
python rambo\_testgen\_coverage.py --dataset DATA\_PATH/Dataset/

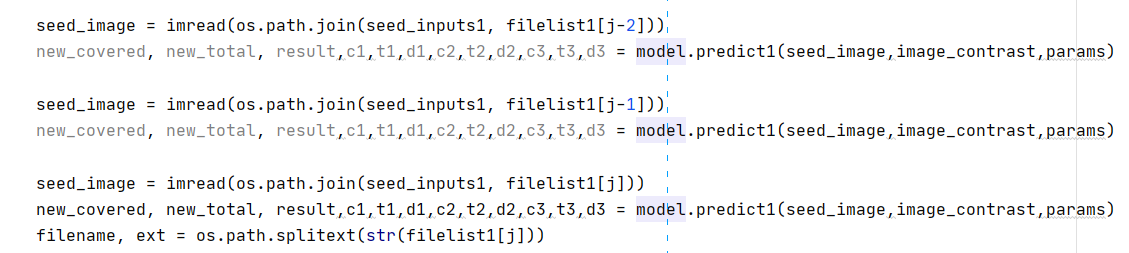
运行视频见文件/ testgen test/ rambo test.mp4



文件代码说明与2.2（2）类似，不再说明







### 2.3 目录/guided 通过引导搜索组合变换和合成图像

cd guided/

mkdir new/

rm -rf \*.pkl

python epoch\_guided.py --dataset DATA\_PATH/Dataset/



python chauffeur\_guided.py --dataset DATA\_PATH/Dataset/



python rambo\_guided.py --dataset DATA\_PATH/Dataset/

均缺少对应的CSV数据集文件，或者模型权重文件，故不再测试。



