

Vmmr 系统研发文档

Yong Gao
Yuhang District, Hangzhou
gaoyongwyl@163.com

2015/04/14

Abstract

对实验系统进行记录。包括实验框架、实验流程和注意事项等内容。

1 System work flow

The main steps to training and test vmmr dnn model are listed as the following:

1. 解压数据。多个压缩包的话直接向同一目录解压，免去之后的数据拷贝。
2. 生成训练和验证集的list，及验证集的GroundTruth(**TODO item**)。调用python脚本CreateLabelTrainTest.py。该脚本的使用说明如下：

Create make and makemodel label dict, and all , train and test label list

Positional arguments:

- SampleFolder: DataSet's data folder, Sample top folder.
e.g.: ~/Projects/Vmmr/Data/V1_AAUMColor/V1_AAUMColor

optional arguments:

- -h, -help: show this help message and exit
- -r TrainTestRatio, -TrainTestRatio: (Default=24) Train-TestRatio. Ratio between the num of train and test
- -sh IsRandShuffle, -RandomShuffle IsRandShuffle:(Default=1,yes)
Whether shuffle list randomly. Must be true for multi-round exp or test. 0 denote not

- -tgn TrainTestGroupNum, -TrainTestGroupNum TrainTestGroupNum: (Default=1)Train and test label list group number

Example:

CreateLabelTrainTest.py ~/Projects/Vmmr/Data/V1-AAuMColor/V1-AAuMColor

3. 生成测试集的list。调用CreateTestList.py，生成测试集的list。所有测试集在，数据集目录的Test文件夹下。生成的list文件放在Test文件夹下的_List目录中。 GroundTruth文件也会自动生成。 CreateTestList.py脚本的使用说明如下：

Create relpath file name list

positional arguments:

- SampleFolder: Sample top folder, in next level is make model folder. E.g.: *~/Projects/VehicleRecog/Data/V1-AAuMColor/Test*

optional arguments:

- -h, -help: show this help message and exit

Example:

CreateTestList.py ~/Projects/VehicleRecog/Data/V1-AAuMColor/Test

4. 生成训练数据LevelDB。调用convert_imagesetex.bin / exe，产生一个或多个区域的杨本数据LevelDB。 convert_imagesetex应用程序使用说明如下：

Convert a set of images to the leveldb format used as input for Caffe. Usage:

convert_imagesetex DataSetNameVer(eg. V0, PreprocTypeIDs(sep by":") PatchIDs(sep by ":") NewWidth AugmentNum(jAugTrain:AugTestj) FuncCode(TestOrTrain, 0:testonly,1:tainonly,2: both) IsNewLdb

positional arguments:

- DataSetNameVer:(eg. V0) data set name;
- PreprocTypeIDs:(sep by":") 0 is Color, 1 is gray, 2 is gray hist equal, 3 is color hist equal;
- PatchIDs:(sep by ":") -1 is vface, 4 is LeftHLamp, ... 10 is MidLineBot.
- NewWidth:Normalized vehicle face width before cropping;
- AugmentNum: (jAugTrain:AugTestj increase how many samples by transformation, such as adding noise, change color. 0 means not to augment.

- FuncCode: TestOrTrain(0:testonly,1:tainonly,2: both)
- IsNewLdb: 1 denote creating new db. 0 denote append data to an existing db.

Example:

convert_imagesetex.bin V1 0:1 -1:6 380 0:0 2 1

5. **计算样本的均值**。这里使用训练样本的leveldb作为caffe工具compute_image_mean的输入。为方便调用，DNNMakemodel_PreparedData.py对该工具进行了封装，其用法如下：

usage: DNNMakemodel_PreparedData.py [-h] [-ttlid TrainTestListID] DATASET_NAME PATCH_ID NewWidth **Example:**

./DNNMakemodel_PreparedData.py V1_AAuMColor 6 150

6. **数据准备检查**

- 在生成list，类别label等文件同时，CreateLabelTrainTest.py还会调用GenVMMTruthFromFileName.py生成所有文件的GroundTruth信息文件。注意检查该文件 是否已生成，否则测试时会因找不到该文件而报错，中止运行。
- 一定要检查生成的make-label和makemodel-label的字典中，make及makemodel是否按顺序排列，否则可能会影响性能。
- 检查convert_imageset 是否转换了预期数目的文件到LevelDB数据库中。可以通过LdbImgViewer.bin或者查看转换的Log文件。

7. **训练或者fine tune 网络**。老版本的train_net和finetune_net现在被统一的caffe命令集替代。

caffe train -solver=... [-snapshot=...]

caffe train -solver=... [-weights=...]

新的caffe命令集说明如下：

caffe.bin: command line brew

usage: *caffe jcommand jargs*

Commands:

- train: train or finetune a model
- test: score a model
- device_query: show GPU diagnostic information
- time: benchmark model execution time

Flags from tools/caffe.cpp:

- -gpu (Run in GPU mode on given device ID.) type: int32 default: -1

- -iterations (The number of iterations to run.) type: int32 default: 50
- -model (The model definition protocol buffer text file..) type: string default: ""
- -snapshot (Optional; the snapshot solver state to resume training.) type: string default: ""
- -solver (The solver definition protocol buffer text file.) type: string default: ""
- -weights (Optional; the pretrained weights to initialize finetuning. Cannot be set simultaneously with snapshot.) type: string default: ""

8. 在标准测试集上进行测试

9. **根据每个车型的样本数目生成数据子集** 将原始的数据集按每个车型的样本数条件，生成子集。比如：生成每个车型样本数大于等于8的子集；生成每个车型样本数小于8的子数据集。
通过调用*ShowClassNumHist.py*和*ExtractSubDataSet.py*两个脚本来完成。主要分三个步骤：

- 调用*ShowClassNumHist.py*生成类别-数量列表文件，该文件中按样本数量排序。
- 手工拷贝或删除，生成符合样本数量条件的类别-数量列表文件。
- 调用*ExtractSubDataSet.py*，以上一部生成的类别-数量文件作为参数之一，建立符合样本数量条件的新数据集。

2 Mistakes Usually Made

- 类别数目出错。比如：网络模型中的类别和实际类别不一致
- 数据转换不对或不完全。只有一部分数据转换进了LevelDB，即只有一部分数据参与训练，效果肯定不好！
- val-proto网络模型文件中的batch_size决定实际的内存分配和运算，影响速度。在应用层一次送进一张识别的时候，应该设为1。否则，有无用的计算产生！
-