**学生工作日报**

|  |  |  |  |
| --- | --- | --- | --- |
| **姓 名** | **段文静** | **日 期** | **2017.1.5** |
| **工作内容描述 达成情况 预期与解决方案等** | | | |
| 1. **今天的工作内容** 2. **完成faster rcnn的训练，得到final模型，将250幅图片送入final模型进行目标检测，保存可视化结果。** 3. **今天的工作总结**   **去掉’van’类别后，另外三种类别的检测结果几乎没变，只有’person’的结果稍有提升，最后mAP上手到83.9331%。具体如下：**   |  |  | | --- | --- | | Class | AP(%) | | Bus | 100 | | Car | 90.8603 | | Person | 60.9390 | | mAP | 83.9331 |   **将250张图片送入final model进行可视化，得到结果，还未分析。**   1. **存在问题及想法**   **无** | | | |
| **明天的工作计划:**  **分析目标检测结构，结合毕设要求，考虑下一步的动作。** | | | |

**附件**

去除货车类别的车辆检测

*第二步想法：用faster rcnn模型，依然采用自己的1000张图片数据集，更改程序，去掉van类别，再训练一遍网络，观察训练结果*

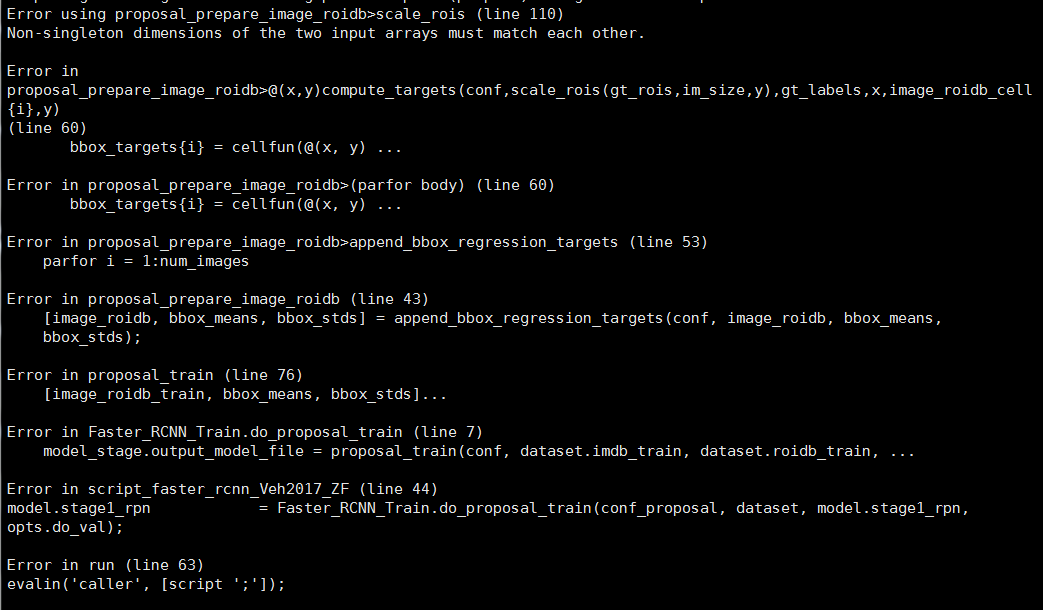
# 在加载数据库时进行程序更改

更改位置在./experiments/script\_faster\_rcnn\_Veh2017\_ZF.m中加载dataset处。设计文件如下：

1. VOCinit.m: ./datasets/VOCdevkit2007/VOCcode/VOCinit.m，将VOCopts.poses改为三类{‘bus’, ‘car’, person};
2. roidb\_from\_veh.m: ./imdb/roidb\_from\_veh.m，该函数将annotations文件与程序中定义的类别进行匹配，具体匹配函数调用处位于130行和131行，函数名称为attach\_proposals()，函数定义位于144行开头。它首先判断annotations文件中是否包含目标，若包含目标，就根据实际需求加载符合要求的目标的ground truth的类别和boxes参数，因此要在符合要求的部分进行更改。更改位置从167行开始到191行结束。思想是首先找到’van’类别对应的下标位置，并在valid\_objects中将对应位置标注为0；更改完valid objects之后还要判断图片目标是否为空，若无空，就按照图片无object处理，及gt\_boxes和gt\_classes均为空，具体可参照’if isfield(voc\_rec, ‘objects’)’的else来写。

# 在RPN的训练部分进行程序更改

在第一步训练rpn时遇到下面的问题：



函数调用链条为:

do\_proposal\_train(7) → proposal\_train(76) → proposal\_prepare\_image\_roidb(43) →

append\_bbox\_regression\_tragets(60) → compute\_targets()

其中compute\_targets()在传递参数时，还调用了scale\_rois()这个函数，当gt\_rois为空，也就是没有目标时，在其第三条语句计算scaled\_rois = …会出错。而gt\_rois为空是去掉’van’类别引发的后果，因此应该去掉那些仅有货车一种目标的图片，经过筛选之后，只有编号为150.jpg的图片是这种情况，输入trainval数据集，在trainval.txt中将150.jpg去掉再重新运行程序就可以了。

# 在Fast RCNN的训练部分进行程序更改

由于将类别’van’去掉了，所以现在只有三个类别，要将fast rcnn的网络结构定义进行更改。最后两个并列子层的输入，分类输入4个值（三个类别一个背景）；边框回归部分输入16个值（每个类别对应4个边框参数）。另外train\_val.prototxt部分的开头处的loss function部分输出个数也要改成16。涉及到的更改文件如下：

1. ./models/fast\_rcnn\_prototxts/ZF/train\_val.prototxt;
2. ./models/fast\_rcnn\_prototxts/ZF/test.prototxt;
3. ./models/fast\_rcnn\_prototxts/ZF\_fc6/train\_val.prototxt;
4. ./models/fast\_rcnn\_prototxts/ZF\_fc6/test.prototxt;

# 检测结果

|  |  |
| --- | --- |
| Class | AP(%) |
| Bus | 100 |
| Car | 90.8603 |
| Person | 60.9390 |
| mAP | 83.9331 |

# 结果可视化

Final model存储在./output/faster\_rcnn\_final/中，更改Fast RCNN的网络结构定义，手动删除与RPN网络重合的部分，也就是从conv1~relu5。还要改最开始的input’data’部分，该输入维度，输入的应该是relu5输出后的维度。

在./output下创建两个文件夹，250\_testimages(据具体的测试图片数量进行名称更改)，存放测试图片的目标检测结果；test\_visualization，存放可视化结果。

改./experiments/script\_faster\_rcnn\_1\_demo.m，将目标检测结果存放目录改为’home/dwj/faster\_rcnn\_2/output/250\_testimages’。

改./utils/showboxes.m，将可视化结果目录改为’/home/dwj/faster\_rcnn\_2/output/test\_visualization’。

运行script\_faster\_rcnn\_1\_demo.m得到可视化结果test\_visualization。

|  |  |  |  |
| --- | --- | --- | --- |
| **姓 名** | **段文静** | **日 期** | **2017.1.4** |
| **工作内容描述 达成情况 预期与解决方案等** | | | |
| 1. **今天的工作内容** 2. **完成faster rcnn训练函数的更改，在去掉’van’类别后重新训练，目前网络训练到第二步，即fast-rcnn的第一次训练。** 3. **今天的工作总结**   **今天更改了faster rcnn的几个位置后成功运行faster rcnn，进行去’van’的训练。更改位置如下：**   1. **加载数据库时进行程序更改：涉及到类别的设置，以及规范每张图片region的ground truth时要先将所有包含’van’的目标去掉。** 2. **在RPN的训练部分进行程序更改：在训练过程中，发现如果一张图片什么目标都没有，会出现数据溢出的计算问题，这是因为个别图片中只包含’van’这一种类别的目标，目标过滤掉之后图片为空，从而引发问题。我将数据库筛选了一遍，去掉了所有只包含货车的图片，重新训练。** 3. **在fast rcnn训练部分进行程序更改：将fast rcnn的网络结构定义进行更改，指最后的分类概率输出和边框回归输出，分别改为4和16.** 4. **存在问题及想法**   **无** | | | |
| **明天的工作计划:**  **继续训练faster rcnn，争取得到训练结果。** | | | |

|  |  |  |  |
| --- | --- | --- | --- |
| **姓 名** | **段文静** | **日 期** | **2017.1.3** |
| **工作内容描述 达成情况 预期与解决方案等** | | | |
| 1. **今天的工作内容** 2. **做test数据集目标检测结果的可视化，可视化结果见《可视化结果程序》，看完结果后发现几个问题：** 3. 测试结果中，不止一次将白色树干的某个部位认为是人，且分类概率高于60%。图片标号：753，775（虽然人在树干后，但几乎被完全挡住，看不出人类特征，所以认为是把树干当成了人，与753是同一树干，几乎在同一位置），781，787，790，812，813，817，822，825，826，827，838，847，850，851，852，855，856，860，861，862，869，870，871，873，882；   将其他位置错检为person，754，874（将垃圾桶视为人，垃圾桶是站立式内凹型垃圾桶，从轮廓来看确实像人，分类概率高于80%），834（车后屁股加车轮），888（垃圾车）；   1. Person类别在region proposal时几乎会将人和座骑一起框出，符合要求。 2. 在辨认van类别时，将van视为car，或将car视为van(也有可能是我自己标注的时候就不是很能区分这些)。图片标号：775，831，832； 3. 当人与车很接近时，有时检测不出人。图片标号：823，826，831，835，915，918，960，998； 4. 当人与背景颜色较接近时，检测不出人。图片编号：895，896，921；   但是在某个场景下，人和背景颜色差别较大时，仍无法检测出人，906，913，927，962，964，996，997；  把几个人框在一起，916，961，980，981，982，983   1. 极个别情况，连车都没有检测完全。图片标号：827，842 2. **更改程序，将’van’去除，重新训练网络。更改位置是加载数据库的过程中，但是遇到了一个问题，程序在region proposal的部分会检查annotations文件中标注的类别和程序中设立的类别是否一直，因此要在检查一致前将包含’van’的那些标注去掉，在调试过程中仍有问题，正在找解决办法。** 3. **今天的工作总结**   **今天首先做了结果可视化，并观察可视化结果，总结出了几点问题。我认为，货车和人这两个类别的准确率较低，一方面是在人工标注的过程中，我也不能很好地区分货车和汽车地区别，可能在标注中将某些货车标注成了汽车，导致训练出的网络在检测时将货车认成了汽车，另外训练集中某些目标与人的特征很相近，导致在测试时许多误检的情况。后边再想解决办法。**   1. **存在问题及想法**   **无** | | | |
| **明天的工作计划:**  **去掉van训练网络。** | | | |