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*

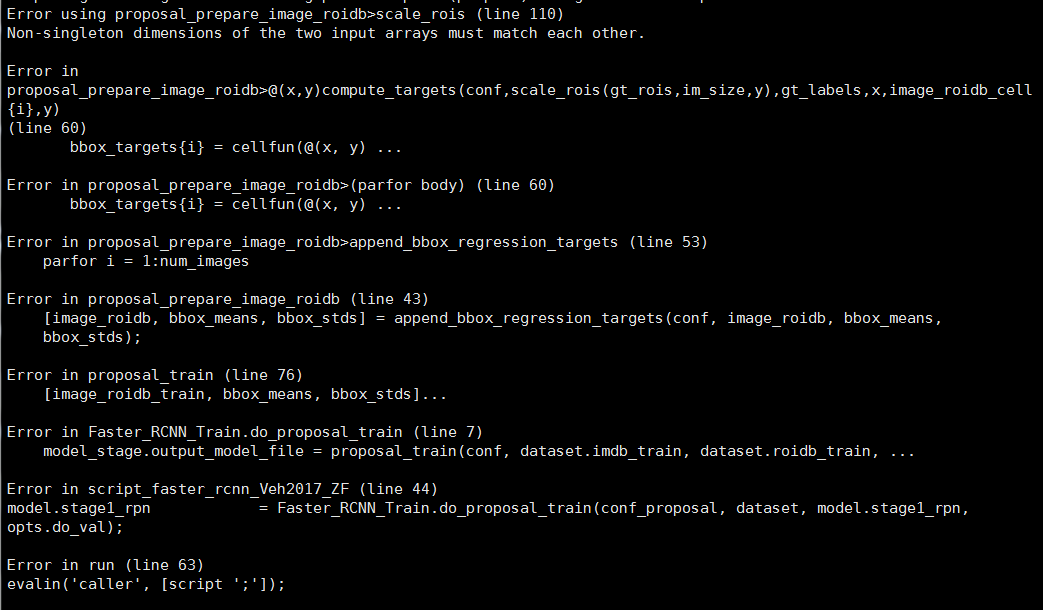
# 在加载数据库时进行程序更改

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;

# 运行方式

在faster\_rcnn\_2目录下输入命令’sh train.sh’开始训练网络，网络结果保存在./output/faster\_rcnn\_final/faster\_rcnn\_Veh2017\_ZF中。

# 检测结果

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

# 结果可视化

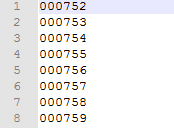
Final model存储在./output/faster\_rcnn\_final/中，更改Fast RCNN的网络结构定义，手动删除与RPN网络重合的部分，也就是从conv1~relu5。还要改最开始的input’data’部分，该输入维度，输入的应该是relu5输出后的维度。

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’。

在faster\_rcnn\_2目录下输入命令’sh test.sh 图片目录 图片名称列表文件.txt’。图片名称列表文件是一个记录所有待检测的图片名称（不带后缀）的txt文件，如下所示：



结果存放在两个文件夹中，分别是250\_testimages（前缀根据真实文件数量改变），存放目标检测的坐标结果和分类结果；test\_visualization，存放可视化结果。

观察结果，发现有如下几点问题：

1. Person漏检，是很明显的人，与车离得也不是很近，但是就是检测不出来。773（第一版检测出来了），784（第一版检测出来了），814（第一版检测出来了），823，826
2. 错检person，情况比第一版好很多。791，812，825，830
3. 漏检车，或区域定位不准确：818，819，826，827

（看到832了）