# Prior\_box VS gt\_box

## 1 获取待匹配列表

1.1 db文件列表保存

由于样本转db文件时会shuffle样本的顺序，因此我们需要在样本转db文件时保存shuffle后的列表顺序。添加的代码段在caffe的convert\_annoset.cpp文件中，由BOX\_LIST宏包围。批处理脚本CovertDataAnn.bat调用convert\_annoset.exe运行结束后会在生成的db文件所在目录下生成train\_lmdb\_list.txt和val\_lmdb\_list.txt两个样本列表文件。列表保存格式为：  
<image\_path> <xml\_path>

1.2 prior box与ground truth box匹配结果保存

caffe中添加的代码分别在multibox\_loss\_layer.cpp、sgd\_solver.cpp、caffe.cpp中，由BOX\_LIST宏包围。其中在multibox\_loss\_layer.cpp中可以设置保存所有匹配结果还是保存IOU<0.5的匹配结果：  
const char\* model = "all"; // 选择只统计IOU<0.5的bbox还是统计全部匹配：small / all

在获取匹配列表前，需计算所有样本完成一次迭代时solver.prototxt中训练迭代（max\_iter）需要几次（max\_iter = round\_up（num\_of\_train\_samples / batch\_size））。同时需要将train.prototxt中所有的样本图像处理和增广处理去掉。

批处理脚本squeezeNet\_train.bat调用caffe.exe训练结束后会在程序运行所在目录<caffe-master\_\windows\caffe>下生成check.txt列表文件。列表保存格式为：  
<***status***> <**prior\_with\_gt\_IOU**> <**prior\_box\_index**> <**prior\_box\_cordinates**>

<**batch\_index**> <**index\_of\_one\_batch**> <**gt\_box\_index**>

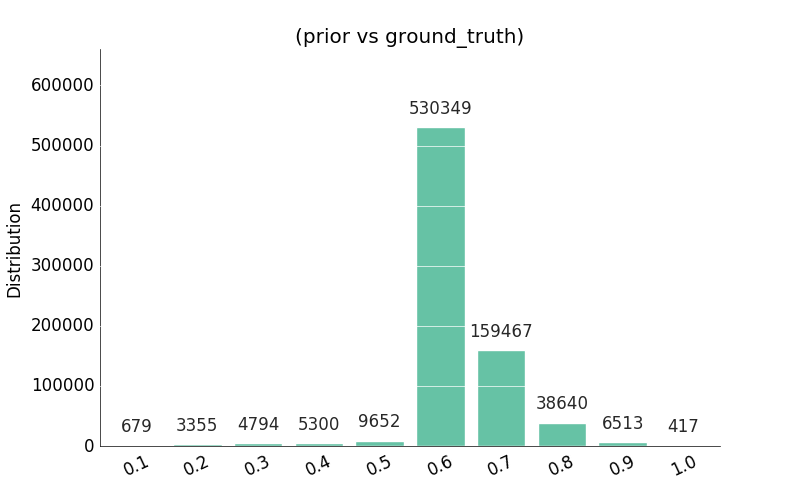
由于计算结果上向上取整，所以列表末尾可能有几个是迭代两次的样本，可对比生成的列表开头删除重复的匹配。

这里简单描述下列表保存过程：训练一开始会检查check.txt是否存在，如果存在删除。对于每个batch，将这个batch中每个样本的匹配情况写入临时文件temp.txt，然后结合batch信息将匹配情况以追加模式重写入check.txt并删除临时文件temp.txt。

## 2 匹配结果合并

2.1 统计绘制匹配结果直方图

将上面训练生成的check.txt复制到db文件所在目录，全部匹配列表重命名为IOU\_ALL.txt，IOU<0.5的匹配列表重命名为IOU\_small.txt。接下来绘制匹配结果直方图。绘制代码为 < Pedestrian>文件夹下priorVSgt\_bar.py。从直方图中我们可以观察出各个IOU区间段box匹配情况。



2.2 匹配合并

为了将匹配结果（IOU\_ALL.txt）与样本列表（train\_lmdb\_list.txt）对应起来，我们需要将二者进行合并。合并代码为IOU \_image\_List.py，代码将合并所有的匹配box。程序运行结束后会生成IOU\_ALL\_image\_List.txt列表。

也可以稍微修改代码（代码中添加**if** float(datas[1]) < 0.5:条件）获得IOU<0.5的box 合并列表IOU\_ small \_image\_List.txt。

合并后列表格式为：  
<**image\_name**> <**xml\_name**> <**prior\_boxes\_total**>

<<**prior\_box1\_index**> <**box1\_IOU**> <**box1\_coordinates**> <**gt\_box\_index**> ...>

获得列表后，我们可以直接显示列表中的box匹配情况：



其中绿色框是gt\_box，红色是各个匹配框。

## 3检查训练中对于prior box能否很好的回归，绘制Recall直方图

3.1 检测结果保存

caffe中添加的代码分别在detection\_output\_layer\_gpu.cpp、ssd\_detect.cpp中，由BOX\_LIST宏包围。批处理脚本Test\_ssd.bat调用test\_ssd.exe检测结束后会在<caffe-master\_\windows\test\_SSD>下生成result.txt列表文件。

列表格式为：

<**img\_name**> <<**prior\_box1\_index**> <**prior\_box1\_coordinates**> <**confidence**>**...**>

这里简单描述下列表该保存过程：程序开头会检查result.txt是否存在，若存在删除。对于IOU\_ALL\_image\_List.txt中每个样本，生成临时文件temp.txt以保存该样本中所有prior box index。检测代码中原本的非极大值抑制过程会筛选检测结果，这里直接将检测结果索引替换成临时文件temp.txt中prior box index并删除temp.txt。最后以追加模式将检测结果存入result.txt。

得到result.txt后，将其拷贝到db文件所在文件夹，重命名为result\_ALL\_image\_List.txt。

3.2 绘制prior box回归直方图

IOU\_change\_curve.py中统计了各个匹配IOU区间段里prior boxes的回归情况，并绘制直方图。程序将IOU\_ALL\_image\_List.txt列表中的prior boxes、gt box信息和result\_ALL\_image\_List.txt列表中的regression prior boxes读取出来计算IOU变化保存成.mat文件。随后绘制IOU变化直方图。

