3.6 YOLO(You only look once)

3.6.1 YOLO

- GoogleNet + 4个卷积+2个全连接层
- 网络输出大小：7 x 7 x 30

3.6.1.2 流程理解

- 3.6.2 单元格(grid cell)
  - 7 * 7=49个像素值，理解成49个单元格
  - 1、每个单元格负责**预测一个物体类别，并且直接预测物体的概率值**
  - 2、每个单元格：两个(默认)bbox位置，两个置信度(confidence)
    - 一个bbox:xmin,ymin,xmax, ymax, confidence
    - 两个bbox: 4 +1 + 4 + 1 = 10个值
    - 30： 10个， 20个(20代表 20类的预测概率结果)
- **3.6.2.1 网格输出筛选**
  - **一个网格会预测两个Bbox，在训练时我们只有一个Bbox专门负责预测概率（一个Object 一个Bbox）**
  - 20个类别概率代表这个网格当中的一个bbox
  - **一个confidence score**
    - 如果grid cell里面没有object，confidence就是0
    - 如果有，则confidence score等于 预测的box和ground truth的IOU乘积
      - 两个bbox的4个值都与GT进行IoU计算，得到两个IoU值
  - YOLO框，概率值都直接由网络输出 7 x 7 x 30（认为给30个值赋了具体的定义）

3.6.4 训练

- 预测框对应的目标值标记
  - 三部分损失 bbox损失+confidence损失+classfication损失

缺点

- 准确率会打折扣
- YOLO对相互靠的很近的物体（挨在一起且中点都落在同一个格子上的情况），还有很小的群体检测效果不好，这是因为一个网格中只预测了两个框

3.7 SSD(Single Shot MultiBox Detector)

- SSD结合了YOLO中的回归思想和Faster-RCNN中的Anchor机制
- **不同尺度的特征特征图**上采用卷积核来预测一系列Default Bounding Boxes的类别、坐标偏移

- **不同尺度feature map所有特征点上使用PriorBox层(Detector&classifier)**

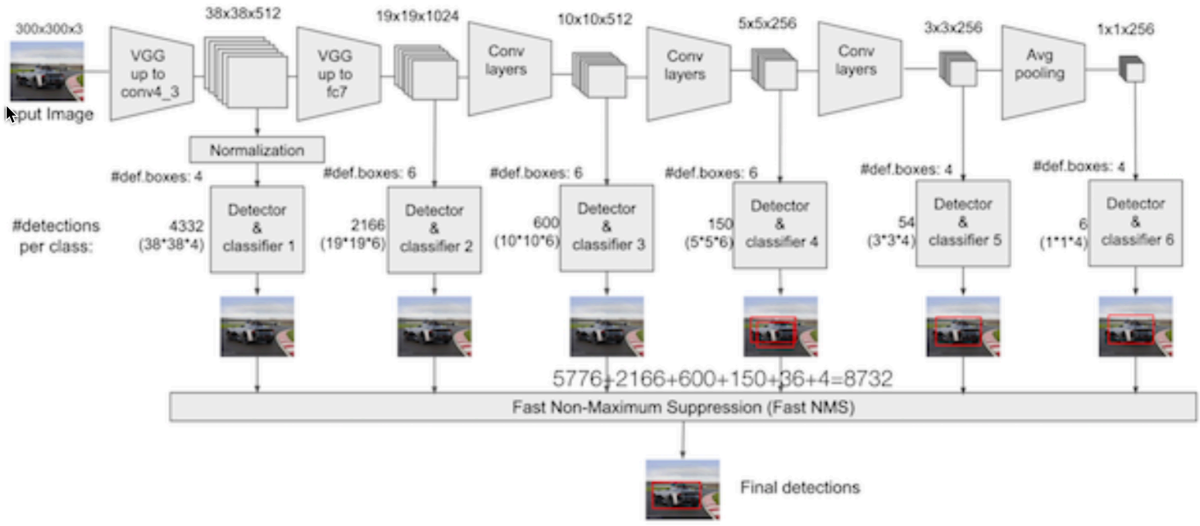
3.7.1.4 Detector & classifier

- 1.PriorBox层：生成default boxes，默认候选框
  - 候选框生成结果之后
  - 做微调，利用4个variance做回归调整候选框
- 2.Conv3 x 3:生成localization， 4个位置偏移
- 3.Conv3 x 3:confidence，21个类别置信度(要区分出背景)

3.7.2 训练与测试流程

- 训练
  - 1、样本标记：8732个候选框default boxes，得到正负样本
  - 正： 付=1：3
  - Softmax Loss(Faster R-CNN是log loss)，位置回归则是采用 Smooth L1 loss (与Faster R-CNN一样)
- **3.7.2.2 test流程**
  - 输入->输出->nms->输出

5.3 SSD网络接口介绍



- 步骤实现
  - 定义好类别数量以及输出
  - 模型预测流程
    - SSD300模型输入以及加载参数
      -
    - 读取多个本地路径测试图片，preprocess_input以及保存图像像素值（显示需要）
      -
    - 模型预测结果，得到7308个priorbox

        - 进行非最大抑制算法处理
    - **图片的检测结果显示**

2.1 目标检测数据集

2.1.2 pascal voc数据集介绍

2.1.3 XML

2.2 目标数据集标记

为什么要进行数据集标记呢?

1、提供给训练的数据样本，图片和目标真是实结果

2、特定的场景都会缺少标记图片

2.2.1 数据集标记工具介绍

2.2.1.1 介绍

2.2.2 商品数据集标记

首先在确定标记之前的需求，本项目以商品数据为例，需要明确的有

- 1、商品图片
- 2、需要被标记物体有哪些

2.2.2.2 标记

标记原则为图片中所出现的物体与我们确定的8个类别物体相匹配即可

5.1 项目训练结构介绍

- ckpt:分为预训练与微调模型
- datasets:放训练原始数据以及存储数据、读取数据代码以及模型priorbox
- servingmodel:模型部署使用的模型位置
- export_serving_model:导出TFserving指定模型类型
- train_ssd:训练模型代码逻辑

5.2.1 案例：xml读取本地文件存储到pkl

- ElementTree工具使用，解析xml结构
- 保存物体坐标结果以及类别

    - pickle工具导出
- 保存的信息：
    - clothes 183 92 261 234
- 1、找到路径对应的图片
- 2、对于每张图片，解析其中的多个物体
    - 位置：保存归一化的结果，/ width, / height
- 3、拼接目标值与onehot到数组中

```
[[0.25587467 0.225      0.56657963 0.55416667 1.         0.
  0.         0.         0.         0.         0.         0.        ]
 [0.21409922 0.54166667 0.50130548 0.97708333 0.         1.
  0.         0.         0.         0.         0.         0.        ]]
```

- 4、将数据保存到pkl文件当中
    - pickle.dump()