YOLO系列

端到端的单阶段的检测网络

YOLOV1

思想

将图像分割成7*7的网格，在网格中生成box进行目标检测

网络架构

使用GoogleNet..,添加卷积和全连接构成整体网络，

输入：448*448*3的图像

输出：7*7*30的张量

- 7*7
  - 对应原始图像7*7个网格的输出结果
- 30
  - bbox的坐标
  - bbox的置信度
  - 分类结果：与数据集有关

网络训练

1.目标值设计

- 每一幅图像是一个7*7*30的张量
- 1.20个类别
  - GT中心点落入的gridcell对应的类别设置为1，其他为0
- 2.置信度
  - 与GTIOU较大的设置为1，其他为0
- 3.坐标
  - 置信度较大的boundingbox目标是GT，其他是0

2.损失函数（重点）

- 1.分类
  - 只有负责检测目标的gridcell才进行损失计算
  - $\sum_{i=0}^{S^2} \mathbb{1}_i^{obj} \sum_{c \in classes} (p_i(c) - \hat{p}_i(c))^2$ 对象分类误差
- 2.边框
  - 只有负责检测目标的boundingbox才进行损失计算
  - $\lambda_{coord} \sum_{i=0}^{S^2} \sum_{j=0}^{B} \mathbb{1}_{ij}^{obj} \left[(x_i - \hat{x}_i)^2 + (y_i - \hat{y}_i)^2 \right]$
  - $+ \lambda_{coord} \sum_{i=0}^{S^2} \sum_{j=0}^{B} \mathbb{1}_{ij}^{obj} \left[\left(\sqrt{w_i} - \sqrt{\hat{w}_i} \right)^2 + \left(\sqrt{h_i} - \sqrt{\hat{h}_i} \right)^2 \right]$
- 3.置信度
  - boundingbox的置信度都参与损失计算
  - $+ \sum_{i=0}^{S^2} \sum_{j=0}^{B} \mathbb{1}_{ij}^{obj} \left(C_i - \hat{C}_i \right)^2$ 置信度误差(边框内有对象)
  - $+ \lambda_{noobj} \sum_{i=0}^{S^2} \sum_{j=0}^{B} \mathbb{1}_{ij}^{noobj} \left(C_i - \hat{C}_i \right)^2$ 置信度误差(边框内无对象)

3.训练

使用ImageNet数据集对前20层卷积网络进行预训练，PASCAL VOC数据集训练整个网络

预测

1。将图片resize成448x448的大小，送入到yolo网络中，输出一个 7x7x30 的张量

2.在采用NMS（Non-maximal suppression，非极大值抑制）算法选出最有可能是目标的结果

总结

优点

1.速度快，可以达到实时处理

2.训练和预测可以端到端的进行，非常简便

缺点

1.准确率会大打折扣

2.对于小目标和靠的很近的目标检测效果并不好

YOLOV2

更准

1.BN　　在每一卷积层后都加了BN

2.使用更高分辨率的图像来训练分类网络（backBone）

3.anchorbox:每一网格中预设5个anchor

4.对训练集中标注的边框进行聚类分析,得到最常见的anchor的尺寸

5.边框的预测

1.宽高：　　anchor的尺度修正

2.中心点坐标　　网格坐标

3.置信度　　sigmoid

6.细粒度特征融合　　passthrough

7.多分辨率的图像进行网络训练

更快

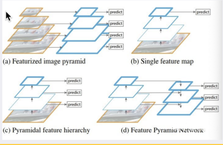
使用了Darknet-19网络，只有卷积和池化

更多

使用imageNet数据集进行辅助训练，使网络可以分辨更多类别的数据

YOLOV3

相对于其他网络速度较快



多尺度特征

1.图像金字塔

2.直接在最深层的特征图上进行检测

3.在不同层的特征图上直接进行检测

4.FPN:浅层特征融合深层特征后再进行检测（yoloV3）

网络架构

1.backbone:darknet53没有池化和全连接层

2.neck:FPN网络进行特征融合

3.head:输出3个尺度的结果

先验框

在coco数据集中聚类的结果

将不同尺度anchor分配在不同尺度的特征图上，用来检测不同大小的目标

logistics使用

使用多标签分类任务

网络输出（重点）

1.3个不同尺度的结果

2.每个尺度有3个anchor

3.每个anchor有分类（80）+回归（4）+置信度（1）=85个值

YOLOV4[了解]