**Redmon\_You\_Only\_Look\_CVPR\_2016\_paper**

阅读笔记:

YOLO的核心思想就是

yolo解决提图像检测问题中 慢的问题

将图像变为回归问题,从图像输入 只过了一个神经网络,就得到boundingbox一级每个boundingbox的概率

利用整张图作为网络的输入，直接在输出层回归boundingbox的位置和bounding box所属的类别。

YOLO的实现方法：

将一幅图像分成S×S个网格，若某个object的中心落在这个网格中，则该网格就负责预测这个object。每个网格要预测B个bounding box，以及C个类别概率Pr，C是网络分类总数，由训练时决定，在YOLO中每个格子只有一个C类别，即相当于忽略了B个boundingboxes，每个格子只判断一次类别，这样做非常简单粗暴。每个bounding box除了要回归自身的位置之外，还要附带预测一个confidence值，这个confidence值代表了所预测的box中含有object的置信度和这个box预测的多准两重信息，其值计算公式为：

其中若有object落在一个网格中，第一项取1，否则取0，第二项是预测的bounding box和实际的ground truth之间的IOU值。

每个bondingbox要预测(x,y,w,h)和confidence共5个值，每个网格还要预测一个类别信息，记为C类，则S×S个网格，每个网格要预测B个bounding box，还要预测C个categories，输出就是S×S×(5\*B+C)的一个tensor（class是针对每个网格的，confidence信息是针对每个bounding box的）。

在测试的时候，每个网格预测的class信息和bounding box预测的confidence信息相乘，就得到每个bounding box的class-specific confidence score。得到每个box的class-specific confidence score之后，设置阈值，滤掉得分低的boxes，对保留的boxes进行NMS处理，就得到最终的检测结果。

YOLO的实现细节：

在实现中，最主要的就是怎么设计损失函数，作者简单粗暴的全部采用了sum-squared eror loss来处理这件事。更重视8维的坐标预测，给这些损失前面赋予更大的loss weight；对没有object的box的confidence loss，赋予小的loss weight；有object的box的confidence loss和类别的loss的loss weight正常取1。

为了缓和对不同大小的box预测中，相比于大box预测偏一点，小box预测偏一点是无法接受的，作者将box的width和height取平方根代替原本的height和width。

一个网格预测多个box，希望每个box predictor专门负责预测某个object，具体就是看当前预测的box与ground truth box中哪个IOU大，就负责哪一个。

YOLO的缺点：

YOLO对相互靠的很近的物体，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，并且只属于一类；对测试图像中，同一类物体中出现新的不常见的长宽比和其他情况时，泛化能力偏弱；由于损失函数的问题，定位误差是影响检测效果的主要原因，尤其是大小物体的处理上，还有待加强。

**Scalable Recognition with a Vocabulary Tree\_nister\_stewenius\_cvpr2006**

阅读笔记: