1. yolo的目标：

将对象检测问题转化为对空间分离边界框的回归与相关类别概率求解的问题

1. 与之前的目标检测算法的比较：

**DPM：**

1、滑动窗口选出region proposal；

2、用分类器分类每一个region proposal。

**RCNN：**

1. selective search选出region proposal，再采用非极大值抑制去重；
2. 用卷积神经网络（5层卷积加2层全连接层）对每一个region proposal做

特征提取；

3、训练21类（类别+背景）SVM，对每一个提取特征向量后的region proposal使用线性SVM对做二分类；

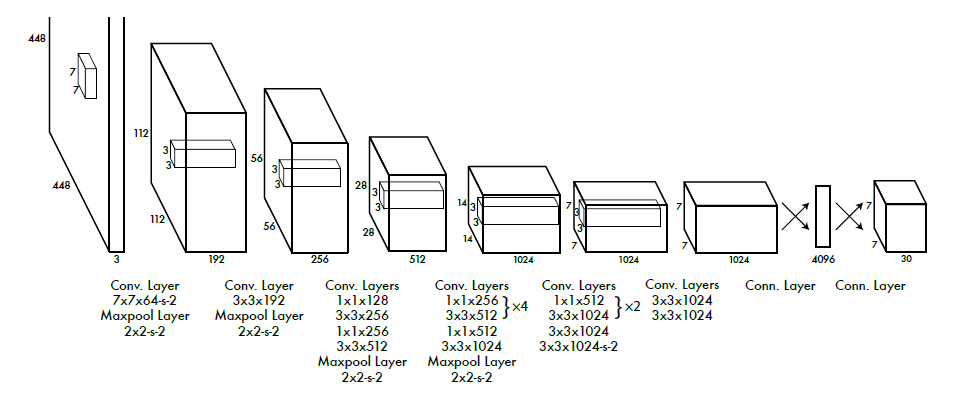
4、边框回归；

**YOLO：**

1. 将输入图片分割成SxS（7x7）的小方格（每一个小方格叫做grid cell）；
2. 每一个grid cell负责预测两个bounding box，以及bounding box中物体类别（20类）的概率。输出大小7x7x30的tensor。

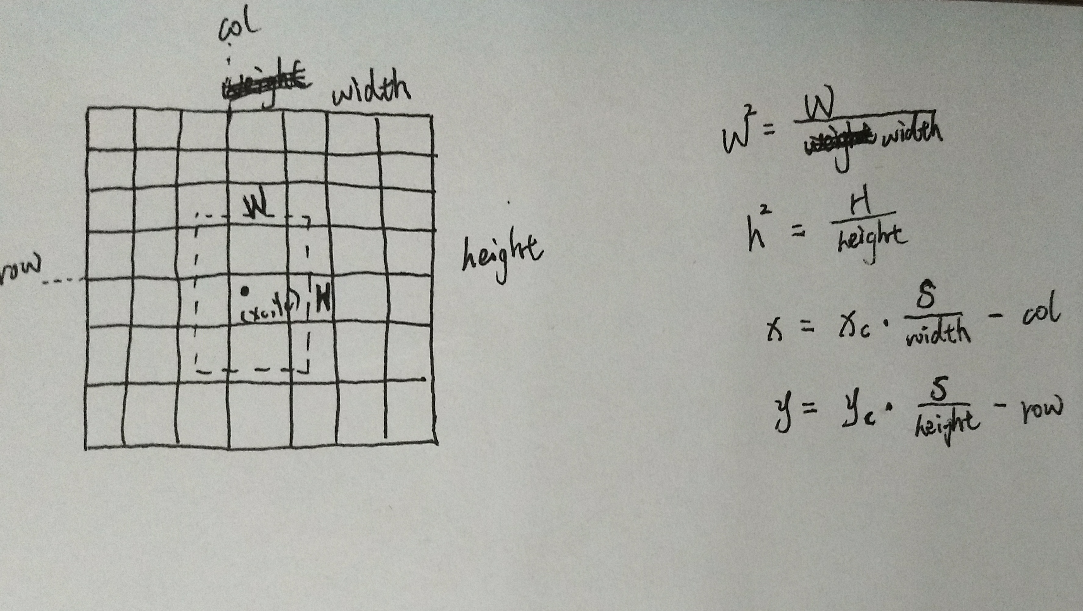
**注：包含该bounding box相对于grid cell的中心坐标x,y以及bounding box相对于图片的长宽w和h，以及该grid cell中包含物体的confidence score(Pr(Object)\*),若物体的中心点落在该grid cell中，该confidence score为真实边界框与预测边界框的IOU，否则为0。最终的score为Pr(Object)\*\*Pr(Classi/Object)**

1. 网络结构：

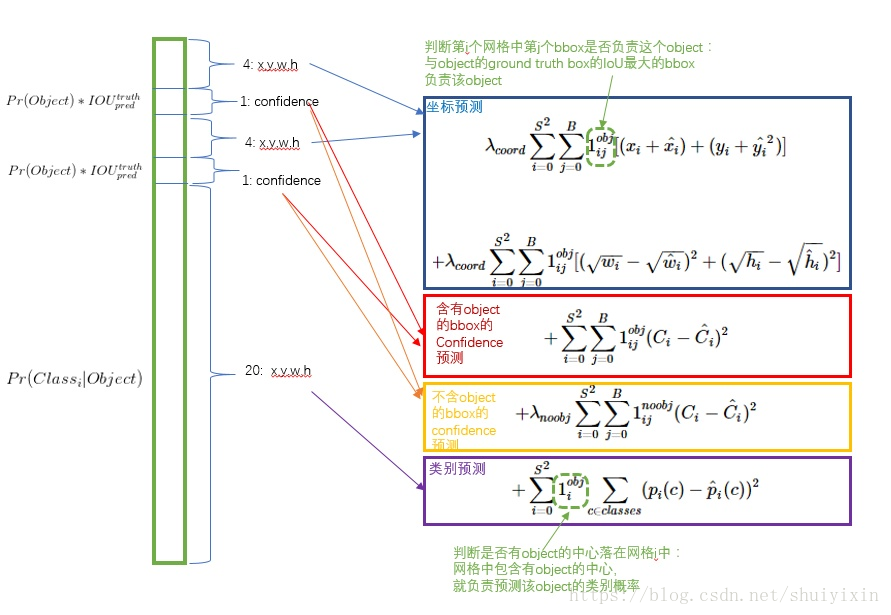


24个卷积层+2个全连接层

1. 细节部分：
2. 为了提供细粒度的视觉信息，将224x224的图像resize到448x448;
3. 对最后一层使用的激活函数是leky relu
4. 改进sum-square error（增加包含物体的grid cell损失权重，减小不包含物体的grid cell损失权重）；
5. 坐标x,y代表bounding box的中心与grid cell边界的相对值，w,h相对于整个图像的weight和height



1. 损失函数的设计：



1. YOLO的局限性：
2. 对小物体与邻近物体的检测不太好（因为每一个grid cell只预测两个bounding box，加上非极大值抑制很容易将群集物体当成一个物体，比如说鸟群）；
3. 当同一类物体出现不常见的长宽比和其他情况时，泛化能力比较弱；
4. 由于损失函数的原因，定位误差是影响检测效果的主要原因。

参考文献：

<https://blog.csdn.net/shuiyixin/article/details/82533849> //YOLO论文详解

<https://github.com/hizhangp/yolo_tensorflow/> //YOLO论文复现