

# 物体检测-Mask-Rcnn实战

## ✓ 课程安排

✎ 通俗讲解知识点，项目实战驱动

✎ 当下主流解决框架，一站式搞定检测与识别任务

✎ 环境配置：选一款IDE即可，基于MASK-RCNN开源项目

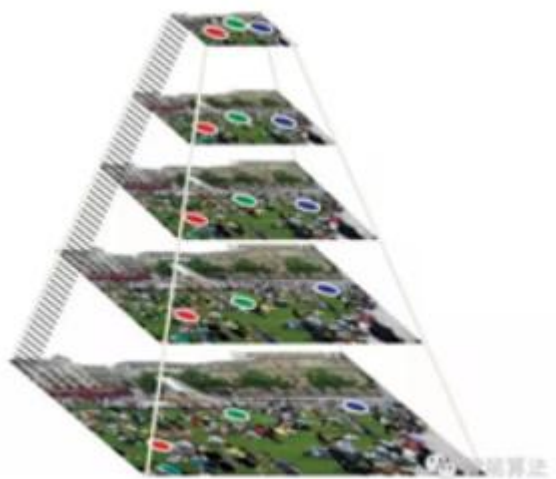
✎ 提供所有数据与代码，追随热点持续更新

# Mask-Rcnn

## ✓ FPN层的作用

✎ R-CNN中仅使用最后一层特征图进行特征提取，但是顶层特征中忽略了一些小物体的一些信息使得检测效果并不好，这该如何改进呢？

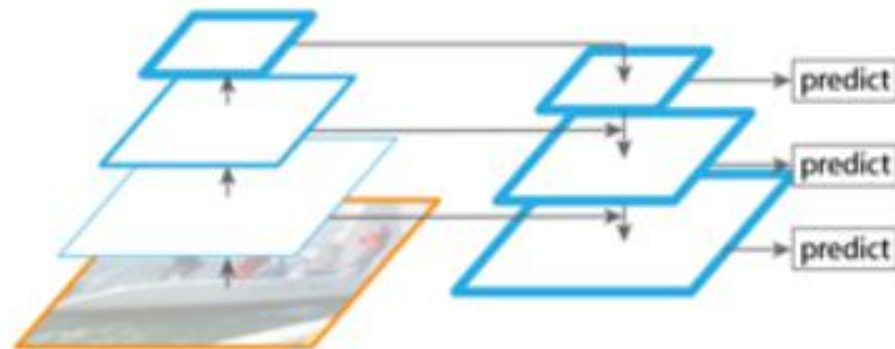
✎ 图像金字塔方法：



# Mask-Rcnn

✓ FPN层的作用

✎ FPN网络基本架构：



✎ 基本思想：将多个阶段特征图融合在一起，这就相当于既有了高层的语义特征，也有了低层的轮廓特征。

# Mask-Rcnn

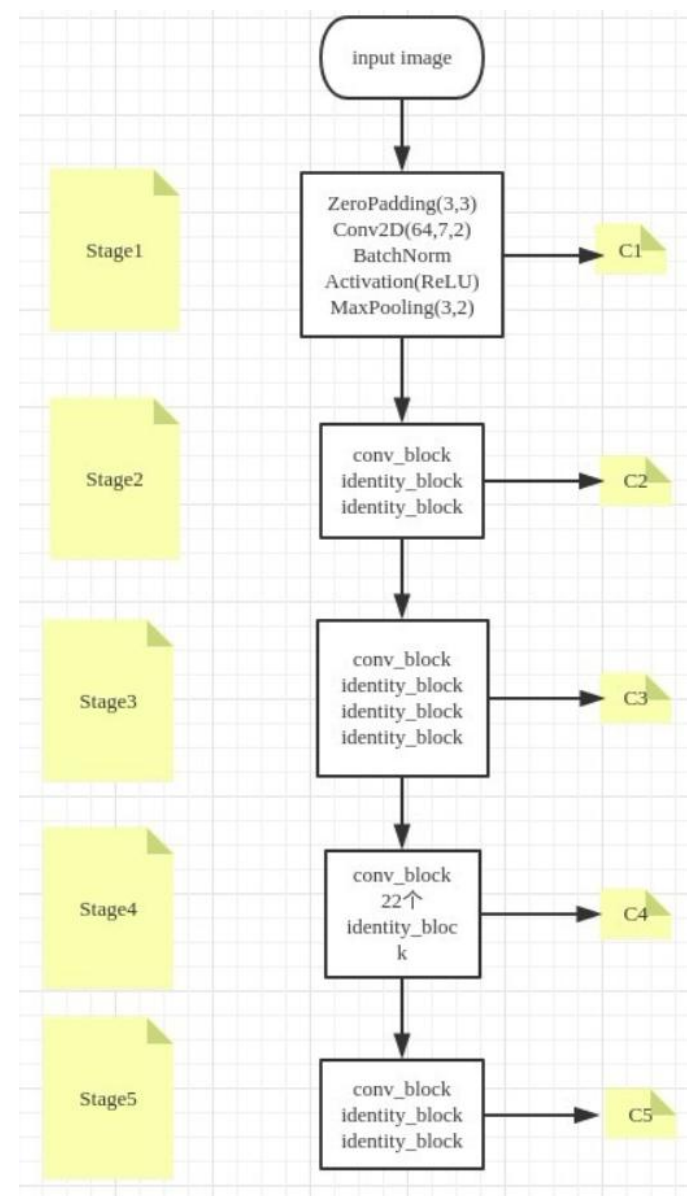
✓ FPN层的作用

✎ Resnet101的五个阶段：

✎ 不改变特征图大小的层归为一个阶段

✎ 每次抽取的特征都是每个阶段的最后一层的输出

✎ 问题：特征图大小会发生变化，如何融合呢？



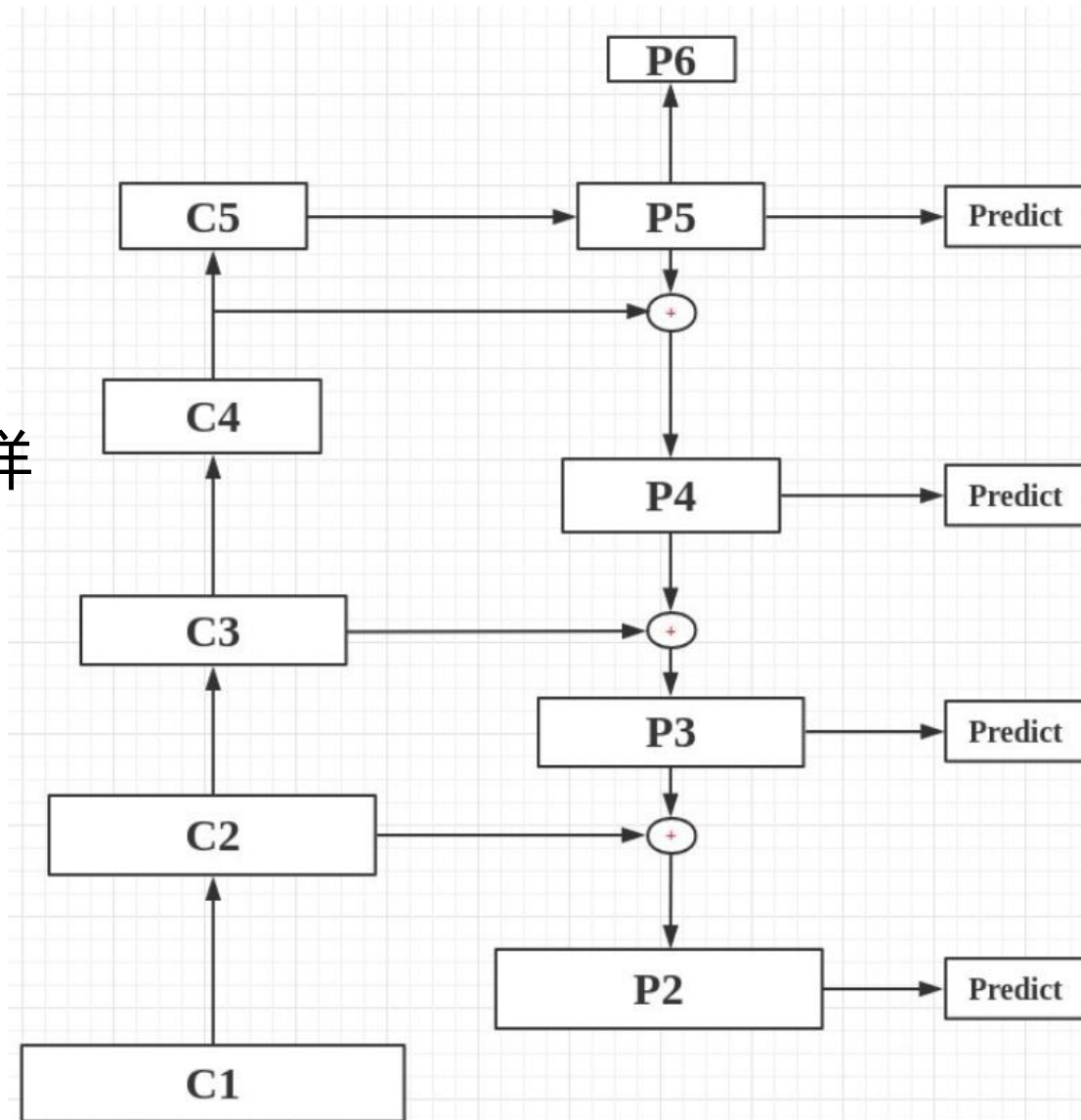
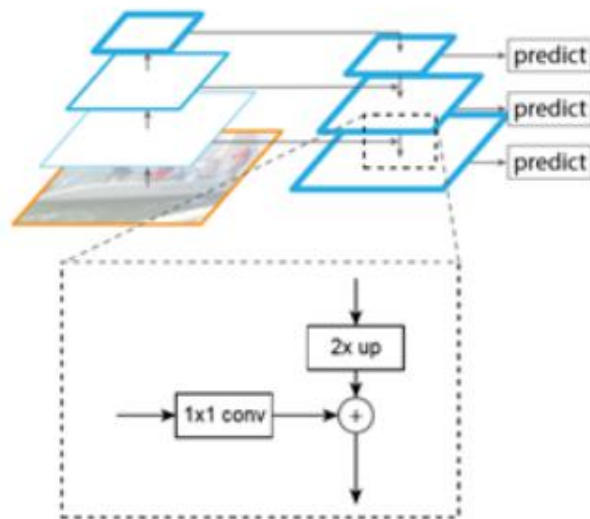
# Mask-Rcnn

✓ FPN层的作用

✎ C1-C5的特征图尺寸是不同的

✎ 使用1\*1卷积先得到P5，然后上采样

✎ 基本操作：





# Mask-Rcnn

✓ generate\_pyramid\_anchors

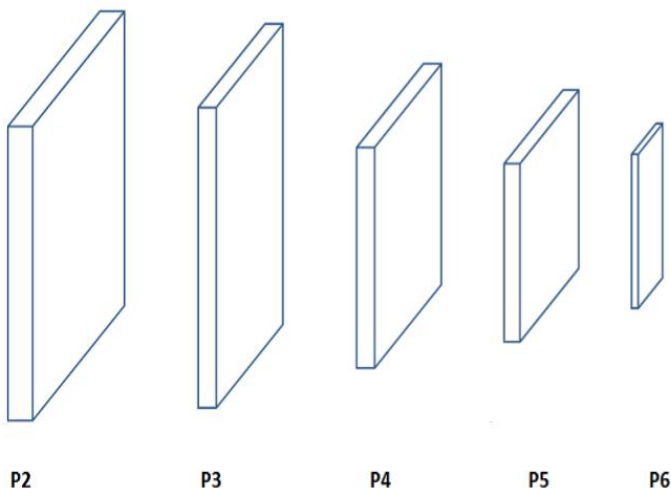
✎ 根据特征图生成所有候选框 (anchors)



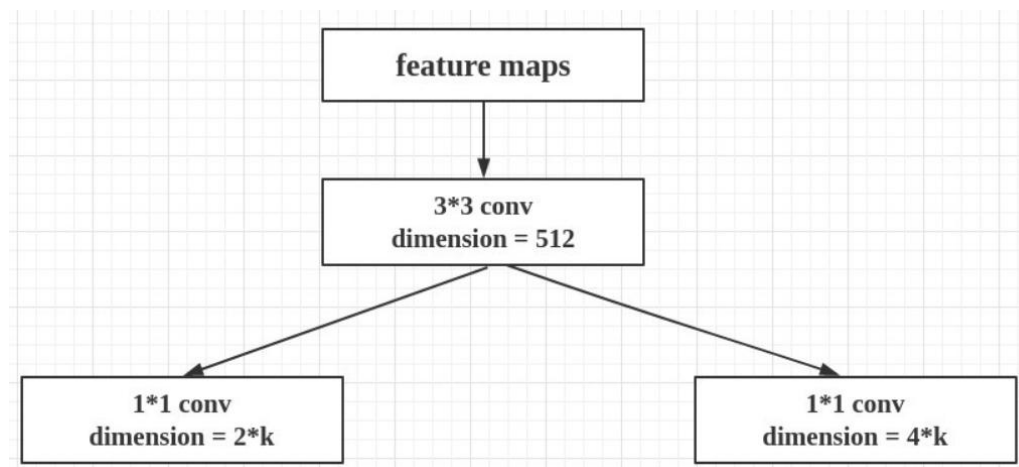
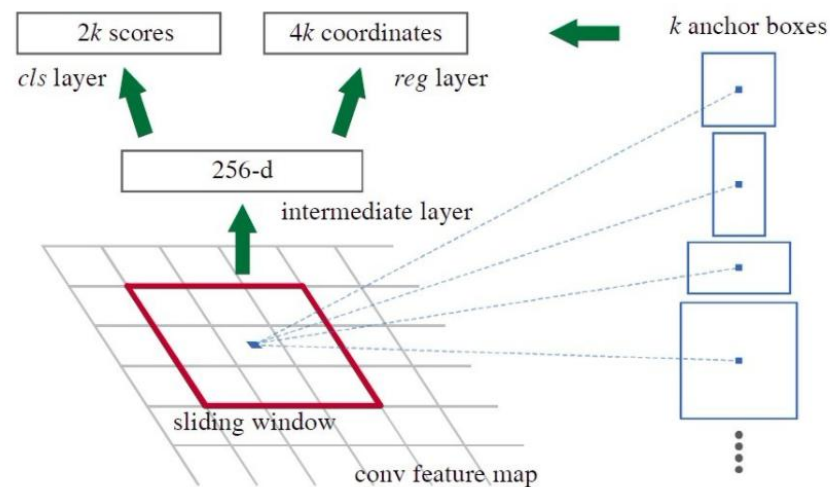
# Mask-Rcnn

✓ RPN层的作用

✎ 计算得到每一个候选框的3个指标:



✎ 代码中都是用1\*1卷积来完成:



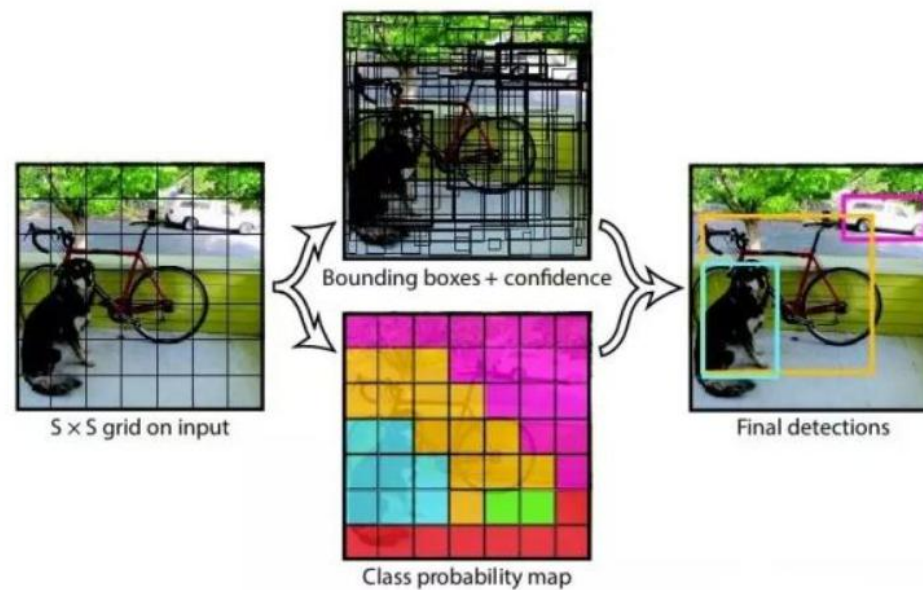
# Mask-Rcnn

✓ ProposalLayer层的作用

✎ 对20W+候选框进行过滤，先按照前景得分排序

✎ 取6000个得分高的，把之前得到的每个框回归值都利用上

✎ NMS再过滤：





# Mask-Rcnn

## ✓ DetectionTargetLayer层的作用

- ✎ 1.之前得到了2000个ROI，可能有pad进来的（0充数的）这些去掉
- ✎ 2.有的数据集一个框会包括多个物体，这样情况剔除掉
- ✎ 3.判断正负样本，基于ROI和GT，通过IOU与默认阈值0.5判断
- ✎ 4.设置负样本数量是正样本的3倍，总数默认400个

# Mask-Rcnn

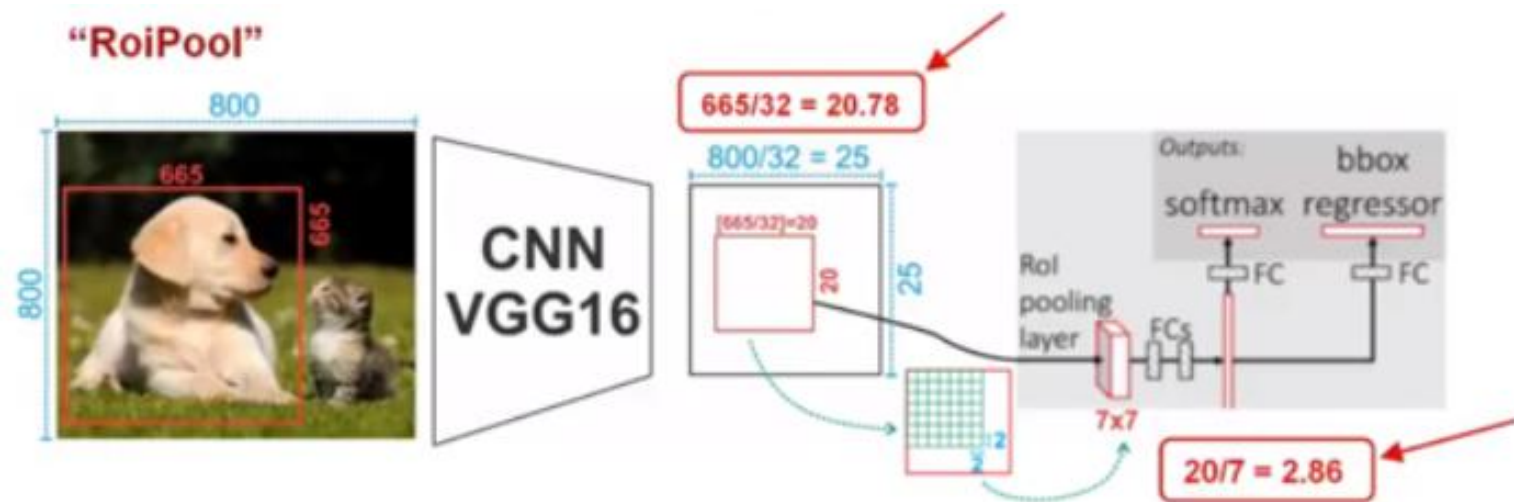
## ✓ DetectionTargetLayer层的作用

- ✎ 5.每一个正样本 (ROI) , 需要得到其类别, 用IOU最大的那个GT
- ✎ 6.每一个正样本 (ROI) , 需要得到其与GT-BOX的偏移量
- ✎ 7.每一个正样本 (ROI) , 需要得到其最接近的GT-BOX对应的MASK
- ✎ 8.返回所有结果, 其中负样本偏移量和MASK都用0填充

# Mask-Rcnn

✓ ROI Align层

✎ ROI Pooling有什么问题：



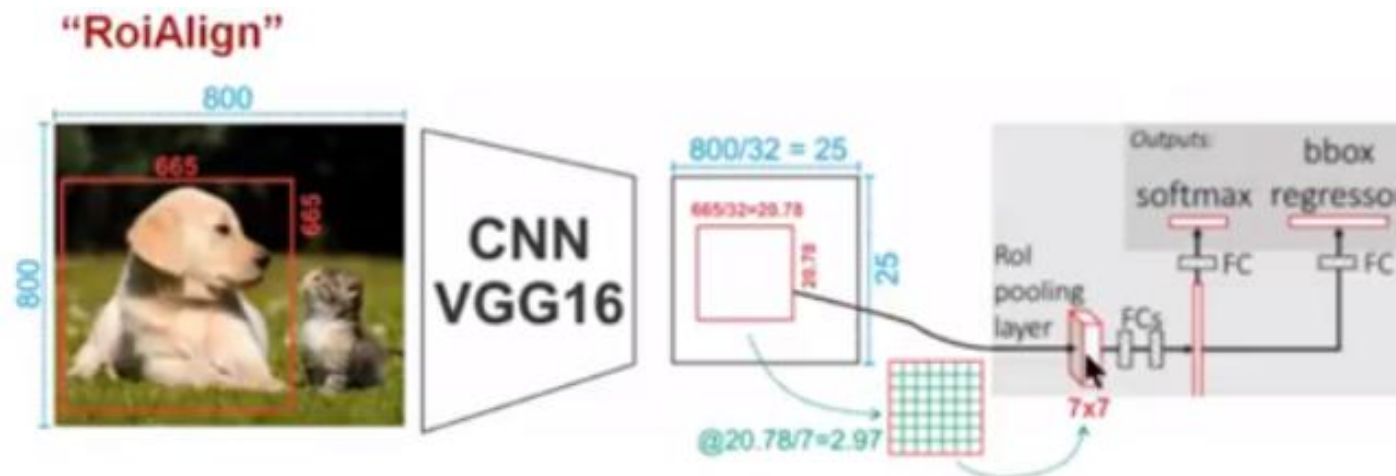
✎ 两次量化，使得特征图对应位置不准

✎ 虽然特征图上差的并不多，但是映射回原图就差的多了

# Mask-Rcnn

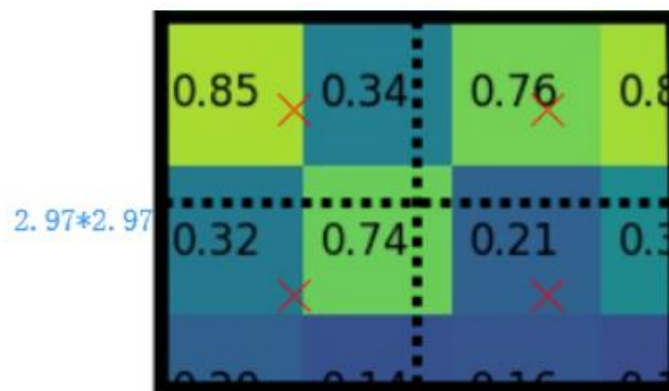
✓ ROI Align层

✎ ROIAlign如何解决问题：



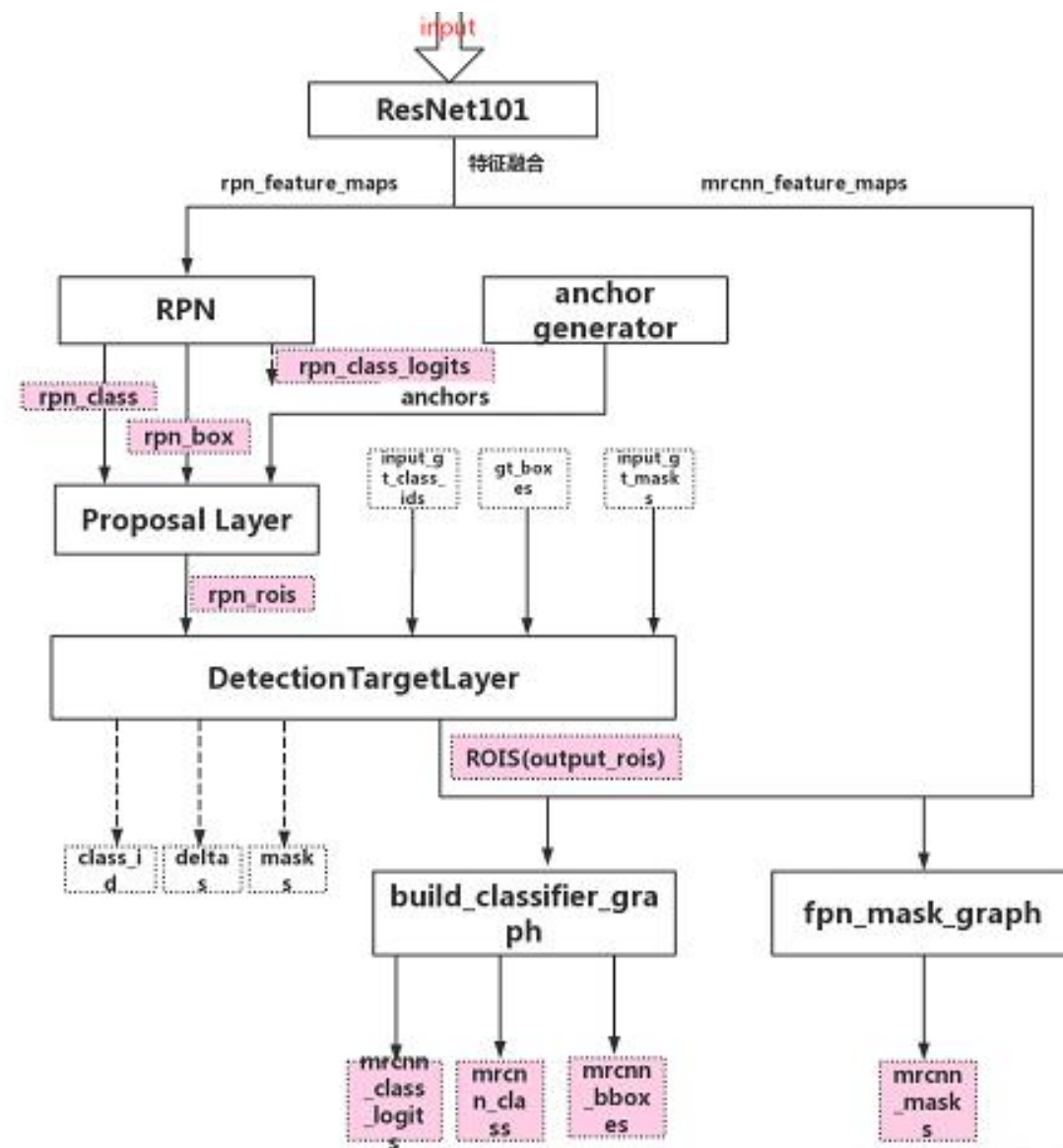
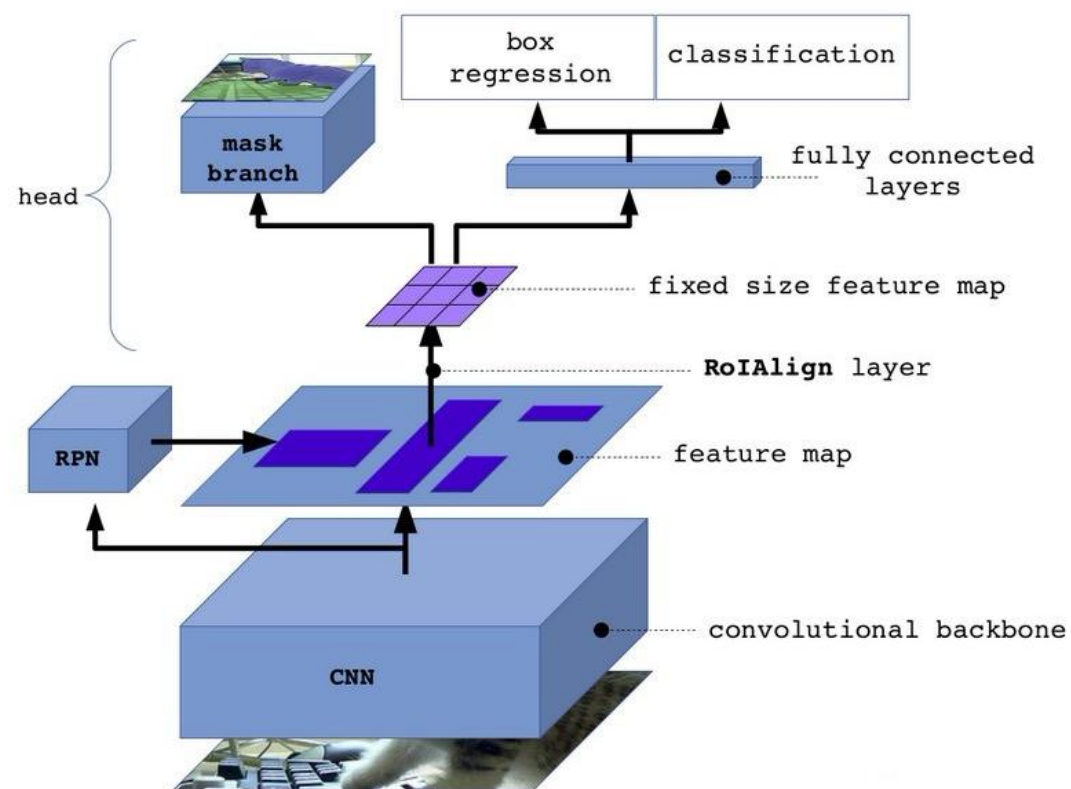
✎ 双线性插值方法解决获得浮点数坐标点上的特征值

✎ 对每个小区域 (2.97\*2.97) 平均分成四份，再取其4个中心点位置进行Pooling：



# Mask-Rcnn

✓ 整体框架:



# Mask-Rcnn

## ✓ DetectionTargetLayer层的作用

- ✎ 1.之前得到了2000个ROI，可能有pad进来的（0充数的）这些去掉
- ✎ 2.有的数据集一个框会包括多个物体，这样情况剔除掉
- ✎ 3.判断正负样本，基于ROI和GT，通过IOU与默认阈值0.5判断
- ✎ 4.设置负样本数量是正样本的3倍，总数默认400个



# Mask-Rcnn

## ✓ DetectionTargetLayer层的作用

- ✎ 1.之前得到了2000个ROI，可能有pad进来的（0充数的）这些去掉
- ✎ 2.有的数据集一个框会包括多个物体，这样情况剔除掉
- ✎ 3.判断正负样本，基于ROI和GT，通过IOU与默认阈值0.5判断
- ✎ 4.设置负样本数量是正样本的3倍，总数默认400个

# Mask-Rcnn

## ✓ FPN层的作用

上采样几乎都是采用内插值方法，即在原有图像像素的基础上在像素点之间采用合适的插值算法插入新的元素，从而扩大原图像的大小

不改变特征图大小的层归为一个阶段

因此每次抽取的特征都是每个阶段的最后一个层的输出