简介

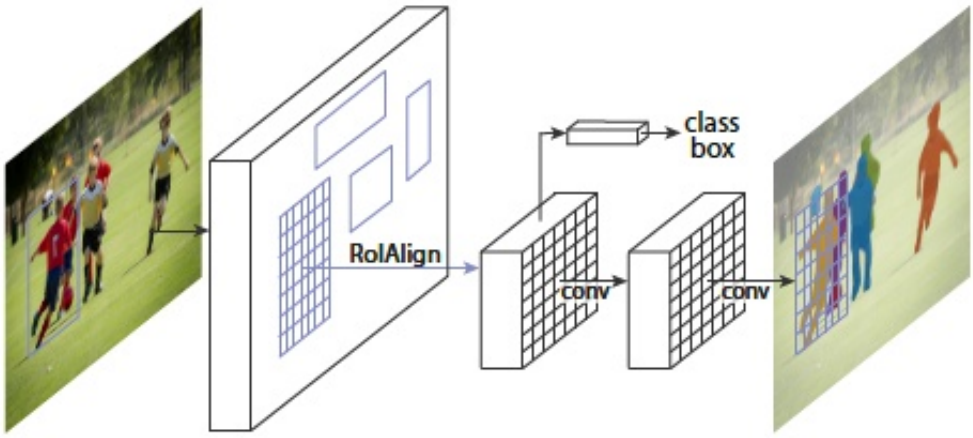
Mask R-CNN 原始论文地址《Mask R-CNN》，该论文发表于 2017年，在 Faster R-CNN 的基础之上修改而来，运行时可以达到 5 fps。而且 Mask R-CNN 可以非常容易的泛化到其他任务，如实例分割（Instance segmentation）、物体边缘检测（bounding-box object detection）、人体关键点检测（person keypoint detection）等，而且得到了非常好的成绩。作者给出了实现代码 Detectron。

如果熟悉Faster R-CNN 的话，Mask R-CNN 还是很好理解，Mask R-CNN 的理解可以参考何凯明的演讲 PPT

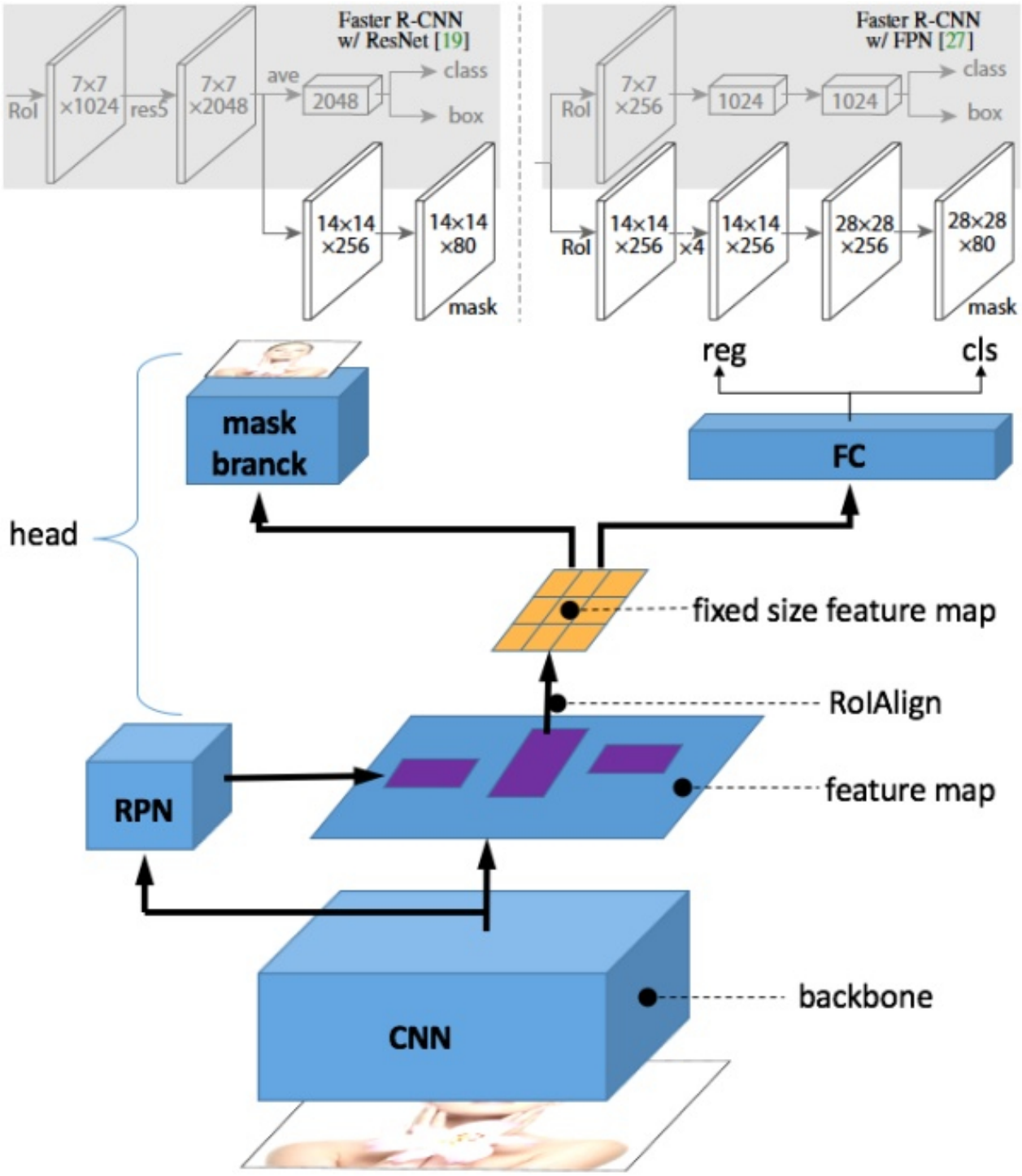
[参考]

- CSDN - Mask RCNN笔记
- CSDN - Mask-RCNN技术解析
- CSDN - 【目标检测】Mask RCNN算法详解
- slides - Mask R-CNN:A Perspective on Equivariance

Mask R-CNN 结构

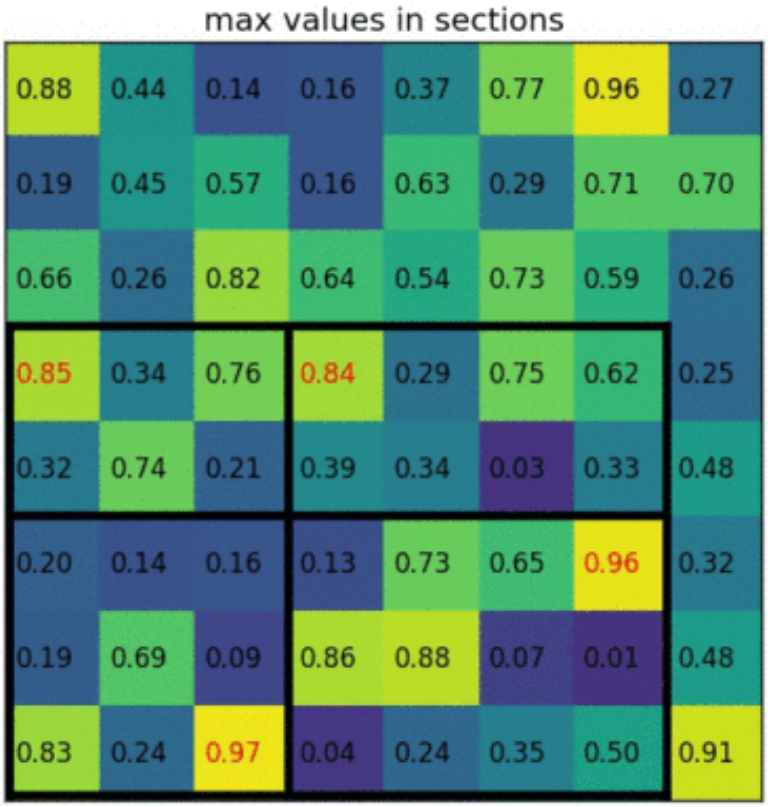


在原有的 Faster R-CNN 的基础上，通过为在每个 RoI 上添加预测分割遮罩（predicting segmentation masks）分支，此分支与原有的分类和 bbox reg 并行。mask 分支在每个RoI 上应用一个小的 FCN，逐像素的预测分割遮罩，因此构建合理的 mask 分支是获得好结果的关键。



RoiAlign

之所以提出RoIAlign，是因为之前的 RoIPooling 的操作会让实例分割有较大的重叠，而 RoIAlign 很好地解决了RoI Pooling操作中两次量化造成的区域不匹配(mis-alignment)的问题。RoIPooling 的操作过程如下：

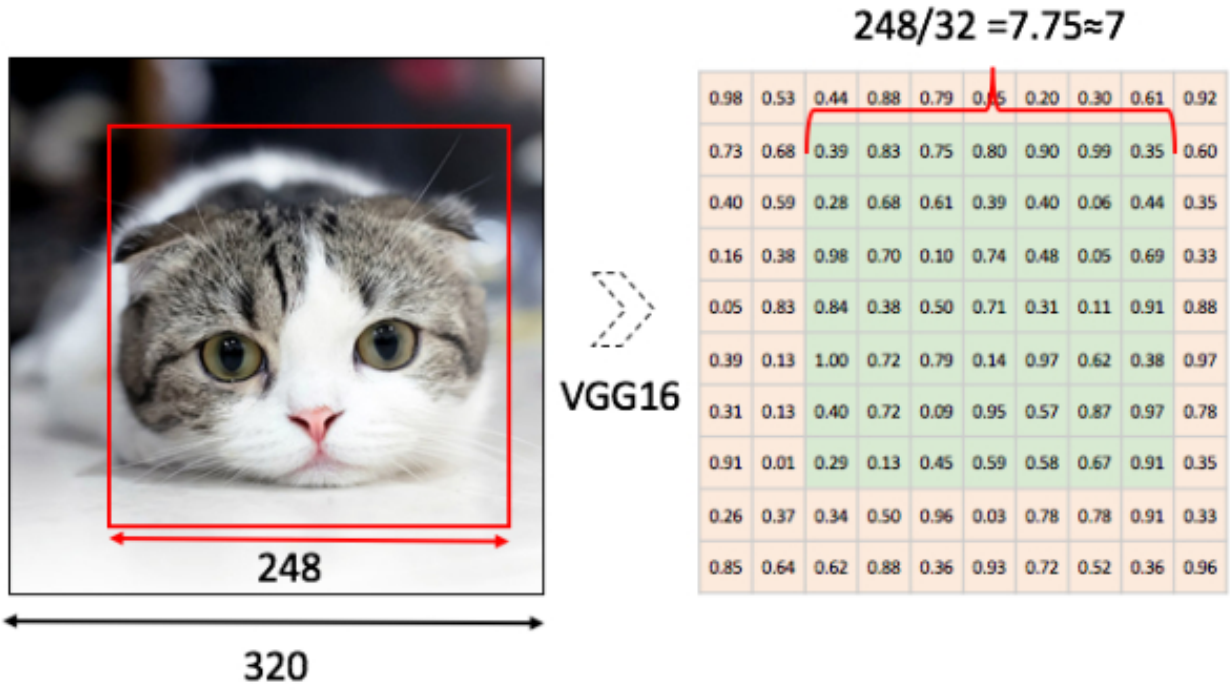


由于 Fast R-CNN 、Faster R-CNN 的候选框位置是通过 reg 方式得到，值大部分是浮点数，而要执行完成 RoIPooling 又需要得到固定尺寸的特征映射。因此 RoIPooling 存在两次量化的过程：

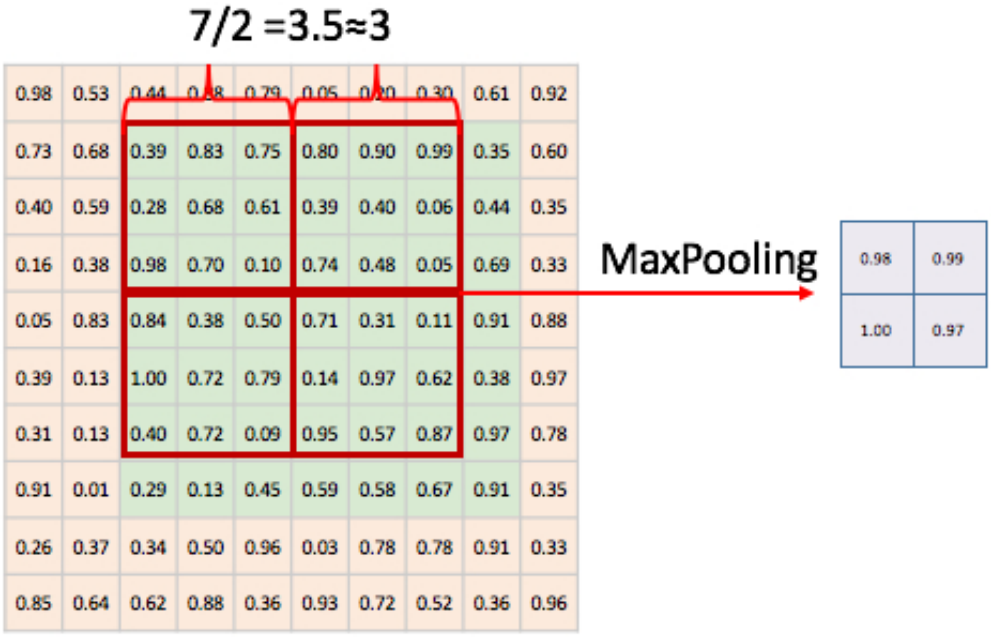
- 将候选框边界量化为整数点坐标值。
- 将量化后的边界区域平均分割成 k x k 个单元(bin),对每一个单元的边界进行量化（如上图）。

这样经过两次量化之后，此时得到的候选框和最开始的reg 得出的位置就存在了一定的偏差，而这个偏差会很大程度上影响分割的精度，就是论文中提到的 misalignment，misalignment 不会影响分类，但是对于预测 mask 却有着很大的影响。

以下图为例，输入是一张 320×320的图片，其中的目标候选框的大小为248×248，那么经过 Fast R-CNN VGG16累积步长 32 的处理之后，320大小图片刚好可以整除32 得到320/32 = 10 ，248 的输入候选框得到的大小为 248/32=7.75，带有小数，RoIPooling 会直接把他量化到 7：



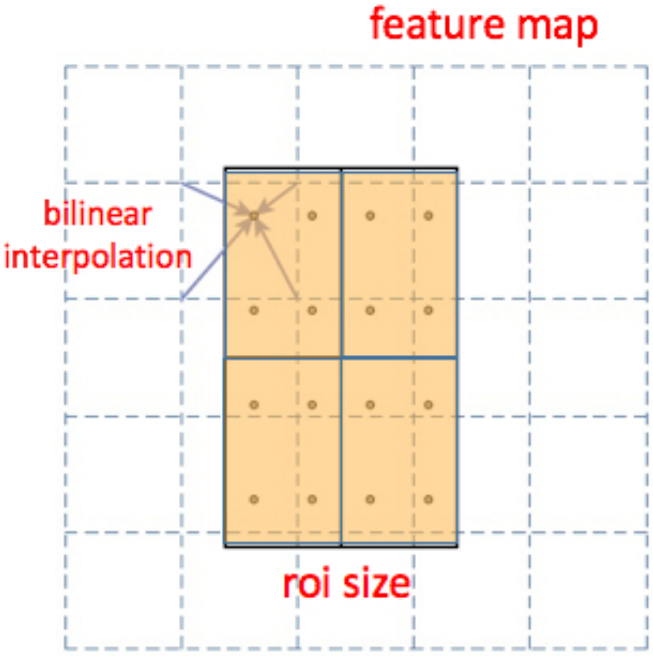
通过 RoIPooling 后我们想要得到大小为2×2的输出特征图，这就需要在 7×7 的特征图上划分出 2×2 矩形区域，每个矩形区域的边长为 7/2 = 3.5 ，含有小数，于是 RoIPooling 再次将其直接量化到 3 ，如下图：



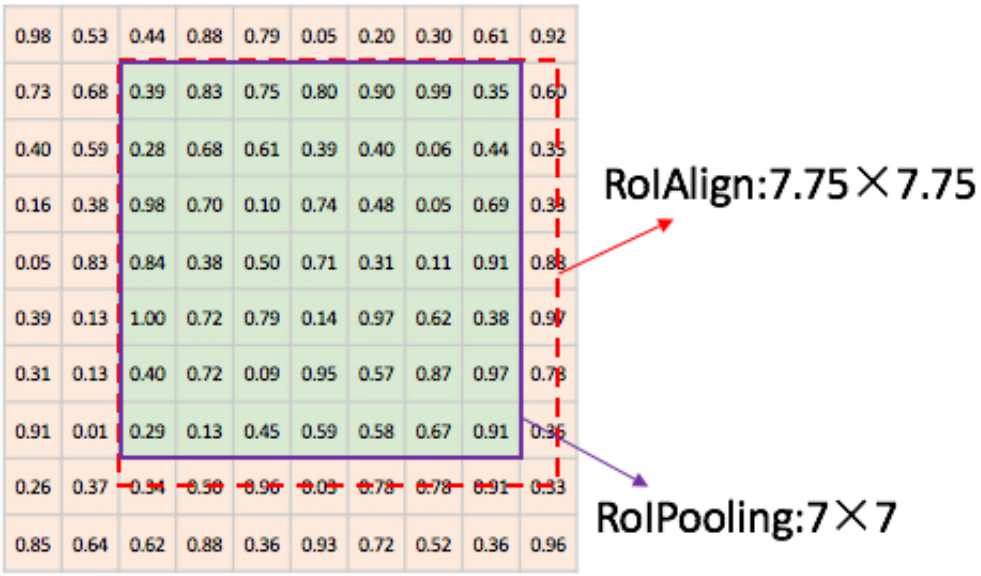
然而在此层上的一点像素的偏差在原始的图像上就会有更大的偏差，如上 0.5 的偏差在原始图像上就存在 0.5 * 32 = 16 像素的偏差，这样大的偏差在实例分割（Instance segmentation）中是非常致命的缺陷。为了要解决这个问题作者提出了 RoIAlign。

RoIAlign 是通过取消量化操作，使用双线性内插值（bilinear interpolation ）方法来获得坐标为浮点数的像素点上的数值，将整个特征聚集过程转化为一个连续的操作。 操作的流程如下：

- 遍历每一个候选区域，保持浮点数边界不做量化
- 将候选区域分成 k×k 个区域，每个边界的单元不做量化
- 在每个计算单元中固定四个坐标位置，使用双线性内插的方法计算四个位置的值，然后进行最大池化操作

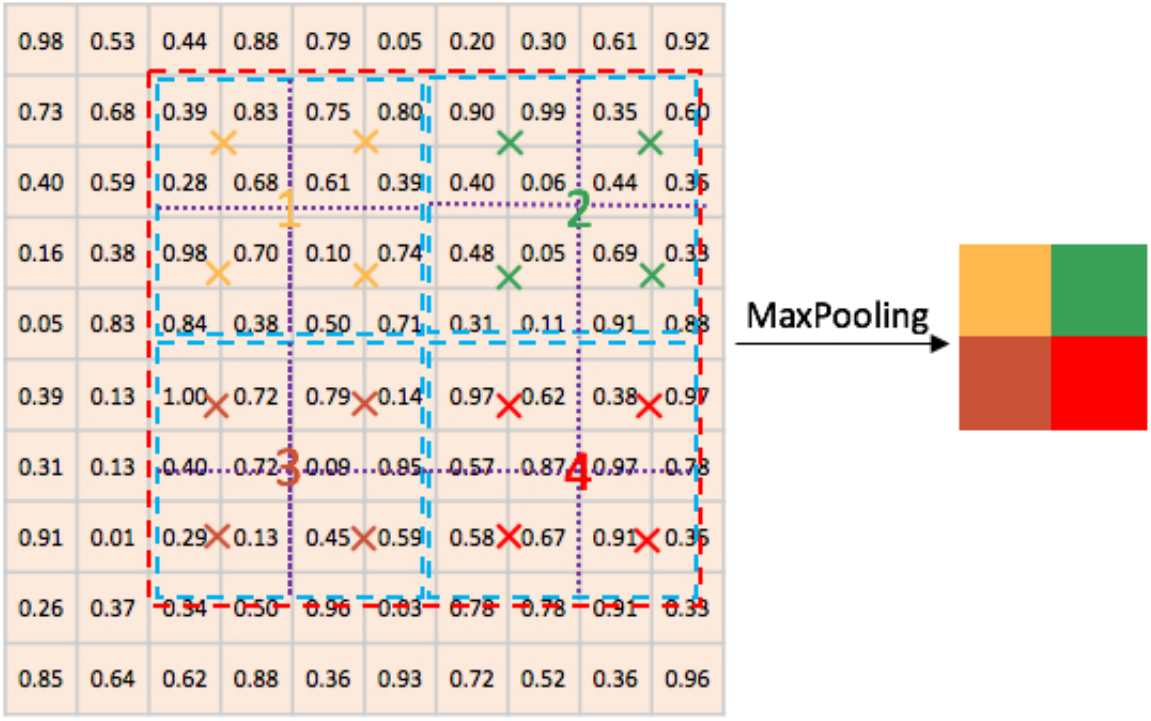


还以 RoIPooling 中的例子为例，这时原图中大小为 248×248 大小的 region proposals经过处理之后，映射到特征图中的大小为 248/32 = 7.75，即大小为 7.75×7.75，这时候并没有像 RoIPooling 中那样直接取整，而是保留了浮点数。



假设输出还是 2×2 大小的特征图，那么就将在 feature map 上的 7.75×7.75 的 region proposal 上划分 4 个区域，每个区域的大小为 7.75/2=3.875。

假定采样点是 4（下图中每个区域内 4 个 X），就表示在每个 3.875×3.875的小区域内平均划分4份，每一份取中心点位置的像素值，中心点值通过双线性内插值的方式进行计算，下图中的每个 X 。这样在每个区域内就得到了四个值，然后再对每个区域内的值进行MaxPooling 操作，每个区域产生一个值，最终输出一个大小为 2×2 特征图。如下图：



在实验中作者也发现采样点设置成 4 与设置成 1 在性能上相差无几。

RoIAlign 对于精度有巨大的影响，使用它会让Mask 的精度提升10%~50%。在使用选择上，对于大的目标检测时，两种方案是差不多的，而如果含有较多的小目标检测，使用 RoIAlign 会获得更好的精确度。通过在 VOC2007 与在 COCO 上的比较久很明显，VOC2007上的效果提升并不明显，在 COCO 上提升就很明显，这就是因为 COCO 数据集上还有很多较小的目标，小目标受到 misalignment 影响更大，RoIAlign 能更好的解决 misalignment 的问题。

[参考]

- github - RoI pooling in TensorFlow

- 个站 - 详解 ROI Align 的基本原理和实现细节
- cnblogs - RoIPooling、RoIAlign笔记

训练

与 gt-box 的 IoU 大于 0.5 的 RoI 认为是正样本，否则是负样本。mask 的损失是依据正样本 RoI 来计算。

使用 image-centric 方式训练，输入图片被重新缩放到最短边为 800 像素。每个批次每个 GPU 2 张图片，每张图片采集 N 个 RoIs，正负样本的比例为 1:3。对于 C4（ResNet 第 4 阶段）类型网络 N 为 64，FPN 网络则为 512。

RPN 网络中的 anchor 设置为 5 种尺度、3种比例。为了方便剪除（ablation）实验，RPN 网络单独训练，不予 Mask R-CNN 共享特征。因 RPN 与 Mask R-CNN 使用的是相同的主网络，因此他们是可以共享的。

在推断是，Proposal 的数量为 300 对于 C4 网络，对于 FPN 网络则是 1000。之后在这些 proposal 上运行 box 预测分值，之后在使用 NMS。mask 分支则应用在分值最高的 100 个检测 box 上，在每个 RoI 上预测 K 个 mask，但是我们只使用第 k 个 mask，这里的 k 是分类分支预测的类别。

实验

剪除实验

网络结构

net-depth-features	AP	AP_{50}	AP_{75}
ResNet-50-C4	30.3	51.2	31.5
ResNet-101-C4	32.7	54.2	34.3
ResNet-50-FPN	33.6	55.2	35.3
ResNet-101-FPN	35.4	57.3	37.5
ResNeXt-101-FPN	**36.7**	**59.5**	**38.9**

可以看到越深的网络得到的结果越好， ResNet-101 好于 ResNet-50；更好的设计也会使结果更好，ResNeXt 与 FPN 好于 ResNet 和 C4。但并不是所有的结构都能从更深的、更先进的网络中收益。

多项与独立Mask

	AP	AP_{50}	AP_{75}
softmax	24.8	44.1	25.1
sigmoid	**30.3**	**51.2**	**31.5**
	+5.5	*+7.1*	*+6.4*

解耦每个类二元遮罩（sigmoid）比多项 mask （softmax）获得更好的结果。

类相关与类无关 mask（Class-Specific vs. Class-Agnostic Masks）

得到的结果基本相同

RoiAlign

	align?	bilinear?	agg.	AP	AP_{50}	AP_{75}
RoIPool [12]			max	26.9	48.8	26.4
RoIWarp [10]		✓	max	27.2	49.2	27.1
		✓	ave	27.1	48.9	27.1
RoIAlign	✓	✓	max	**30.2**	**51.0**	**31.8**
	✓	✓	ave	**30.3**	**51.2**	**31.5**

实验使用 ResNet-50-C4作为基础网络，累积 stride为16。

	AP	AP_{50}	AP_{75}	AP^{bb}	AP^{bb}_{50}	AP^{bb}_{75}
RoIPool	23.6	46.5	21.6	28.2	52.7	26.9
RoIAlign	**30.9**	**51.8**	**32.1**	**34.0**	**55.3**	**36.4**
	+7.3	*+ 5.3*	*+10.5*	*+5.8*	*+2.6*	*+9.5*

上面是使用ResNet-50-C5的结果，累积 stride 为32。从上面也可以看到，使用 stride-32 C5特征（30.9 AP）的 RoIAlign 好于 stride-16 C4特征（30.3 AP）的精确度。可以看到 RoIAlign 解决了长久存在的大步长特征检测和分割的挑战。

Mask 分支

	mask branch	AP	AP_{50}	AP_{75}
MLP	fc: $1024 \rightarrow 1024 \rightarrow 80 \cdot 28^2$	31.5	53.7	32.8
MLP	fc: $1024 \rightarrow 1024 \rightarrow 1024 \rightarrow 80 \cdot 28^2$	31.5	54.0	32.6
FCN	conv: $256 \rightarrow 256 \rightarrow 256 \rightarrow 256 \rightarrow 256 \rightarrow 80$	**33.6**	**55.2**	**35.3**

可以看到 FCN 比多层感知机（MLLP）加上 FC 得到了更好的结果。

bounding box 检测结果

	backbone	AP^{bb}	AP^{bb}_{50}	AP^{bb}_{75}	AP^{bb}_{S}	AP^{bb}_{M}	AP^{bb}_{L}
Faster R-CNN+++ [19]	ResNet-101-C4	34.9	55.7	37.4	15.6	38.7	50.9
Faster R-CNN w FPN [27]	ResNet-101-FPN	36.2	59.1	39.0	18.2	39.0	48.2
Faster R-CNN by G-RMI [21]	Inception-ResNet-v2 [41]	34.7	55.5	36.7	13.5	38.1	52.0
Faster R-CNN w TDM [39]	Inception-ResNet-v2-TDM	36.8	57.7	39.2	16.2	39.8	**52.1**
Faster R-CNN, RoIAlign	ResNet-101-FPN	37.3	59.6	40.3	19.8	40.2	48.8
Mask R-CNN	ResNet-101-FPN	38.2	60.3	41.7	20.1	41.1	50.2
Mask R-CNN	ResNeXt-101-FPN	**39.8**	**62.3**	**43.4**	**22.1**	**43.2**	51.2

训练的 Mask R-CNN ，只关注输出的类别和 box，mask 输出忽略。Faster R-CNN/ RoIAlign 表示训练一个没有 mask 分支的 Mask R-CNN。

在人体姿势关键点检测中的应用

	AP^{kp}	AP^{kp}_{50}	AP^{kp}_{75}	AP^{kp}_{M}	AP^{kp}_{L}
CMU-Pose+++ [6]	61.8	84.9	67.5	57.1	68.2
G-RMI [32]†	62.4	84.0	68.5	**59.1**	68.1
Mask R-CNN, keypoint-only	62.7	87.0	68.4	57.4	71.1
Mask R-CNN, keypoint & mask	**63.1**	**87.3**	**68.7**	57.8	**71.4**

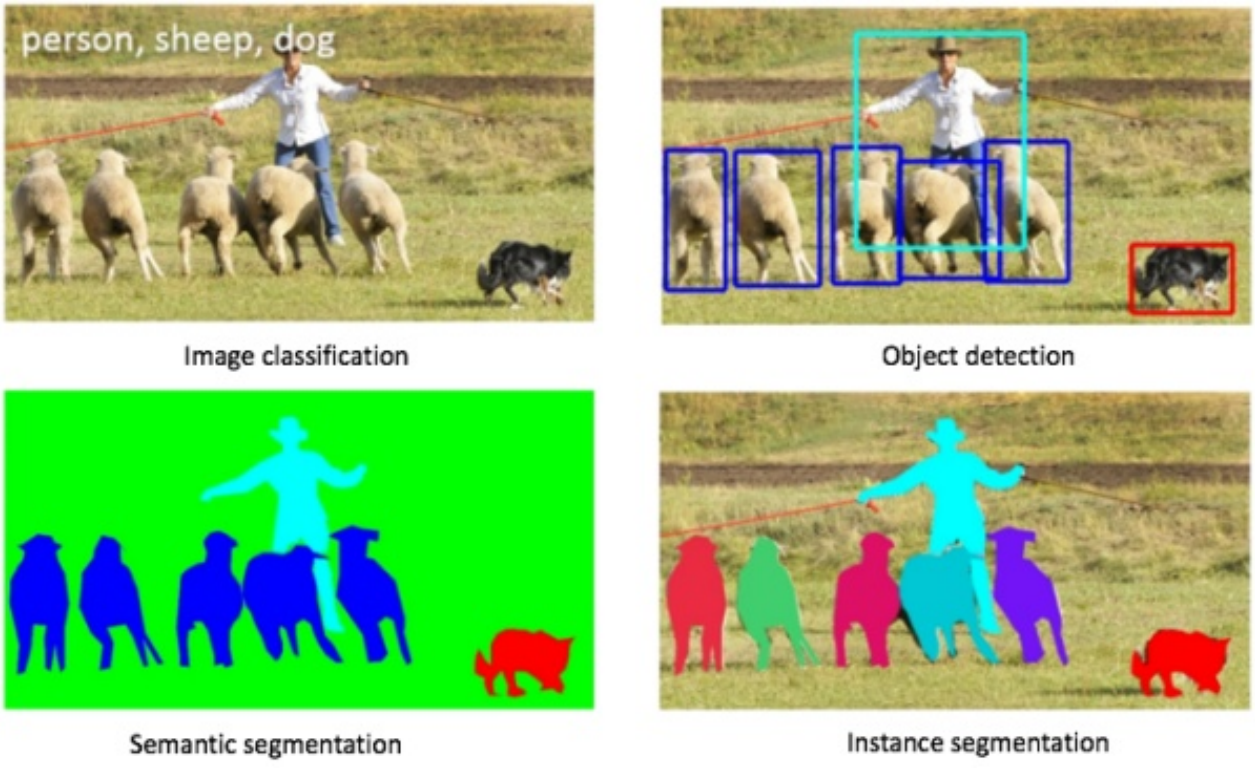
	AP^{bb}_{person}	AP^{mask}_{person}	AP^{kp}
Faster R-CNN	52.5	-	-
Mask R-CNN, mask-only	**53.6**	**45.8**	-
Mask R-CNN, keypoint-only	50.7	-	64.2
Mask R-CNN, keypoint & mask	52.0	45.1	**64.7**

	AP^{kp}	AP^{kp}_{50}	AP^{kp}_{75}	AP^{kp}_{M}	AP^{kp}_{L}
RoIPool	59.8	86.2	66.7	55.1	67.4
RoIAlign	**64.2**	**86.6**	**69.7**	**58.7**	**73.0**

问题

图像分类、目标检测、语义分割、实例分割

在图像研究领域经常遇到图像分类、目标检测、语义分割、实例分割，那么他们究竟指的什么呢？看了下面这张图就会明了：

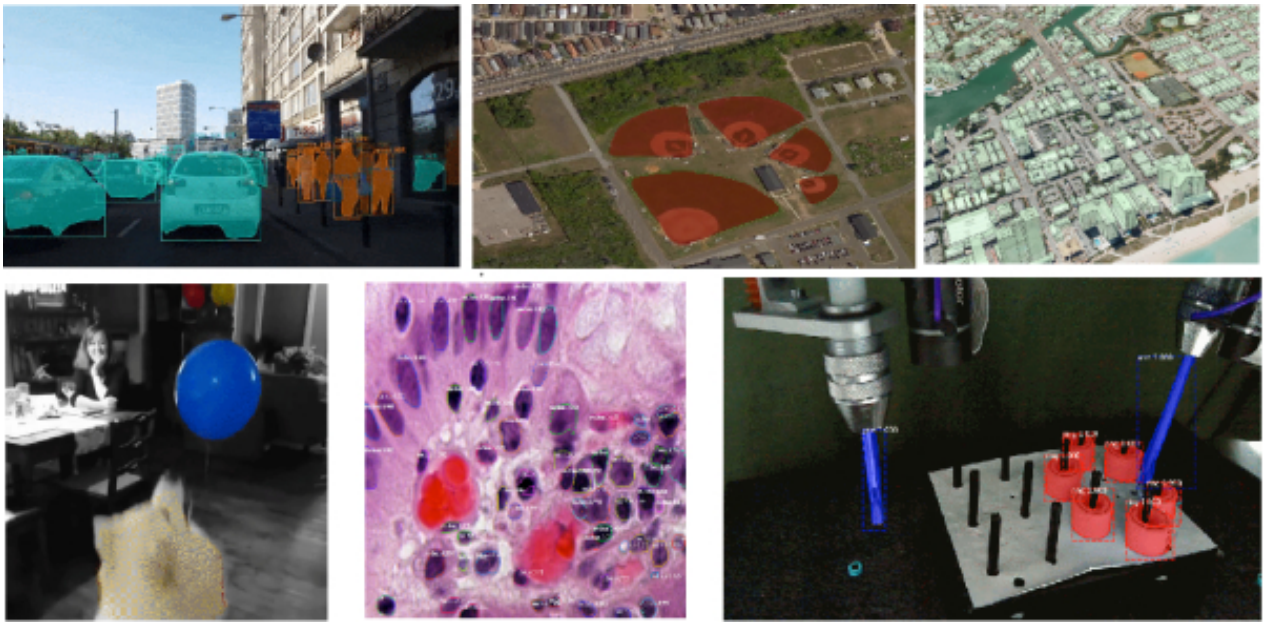


[参考]

- 知乎 - 图像识别中，目标分割、目标识别、目标检测和目标跟踪这几个方面区别是什么？
- CSDN - 图像分类，物体检测，语义分割，实例分割等概念

应用

在 github 上有一个使用 TensorFlow 与keras 实现的 mask r-cnn，里面给出了在实例分割、OSM（OpenStreetMap）、splash of color、细胞核、机器人检测、3D 构建、细胞自动跟踪等方面的应用：



代码地址：matterport/Mask_RCNN

【参考汇总】

- CSDN - Mask RCNN笔记
- CSDN - Mask-RCNN技术解析
- CSDN - 【目标检测】Mask RCNN算法详解
- slides - Mask R-CNN:A Perspective on Equivariance
- 知乎 - 图像识别中，目标分割、目标识别、目标检测和目标跟踪这几个方面区别是什么？
- CSDN - 图像分类，物体检测，语义分割，实例分割等概念
- github - RoI pooling in TensorFlow
- 个站 - 详解 ROI Align 的基本原理和实现细节
- cnblogs - RoIPooling、RoIAlign笔记