

Object Detection Models

Two-Stage:

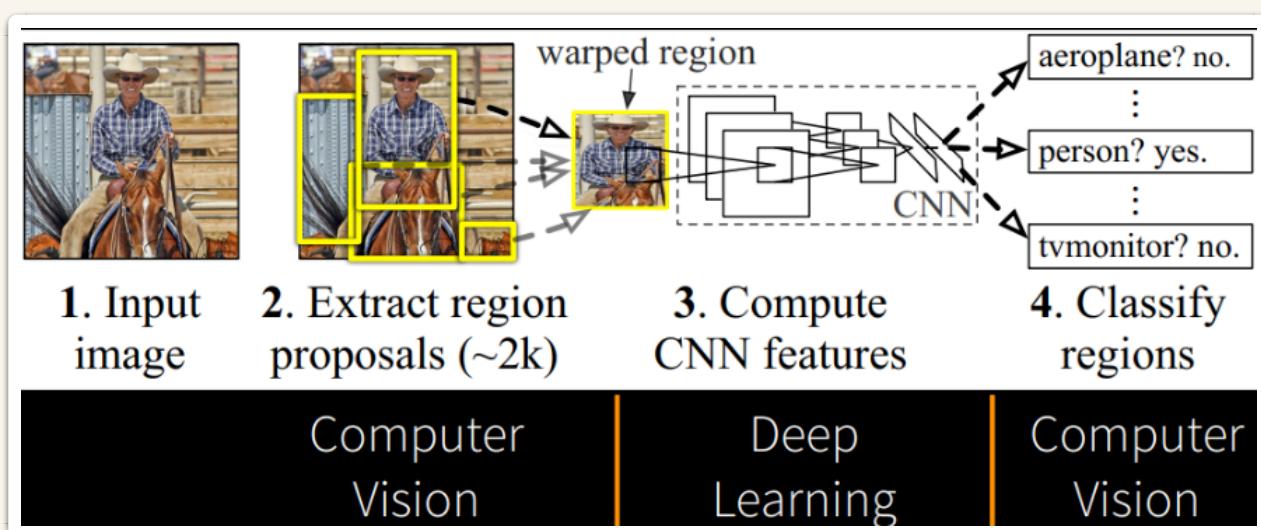
1. R-CNN

模型特点：

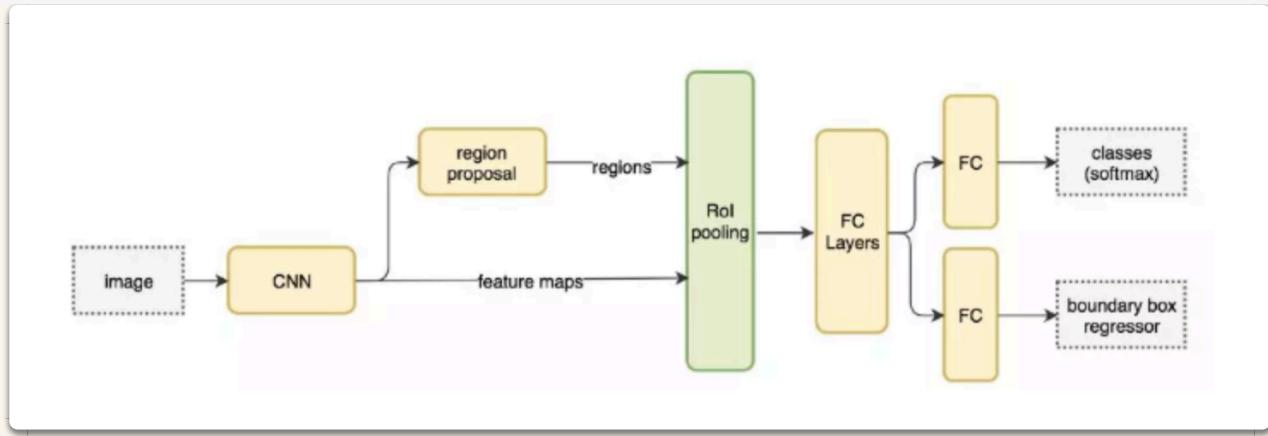
1. 使用CNN队Region Proposals 计算feature vectors。从经验驱动特征 (HOG, SIFT) 到数据驱动特征 (CNN feature Map)，提高特征对样本的表示能力。

R-CNN Pipeline

1. pre-train neural network
2. 重新训练全连接层。
3. 提取proposals并计算CNN特征。利用selective search算法提取所有 proposals，调整大小 (wrap)，满足CNN输入，然后将feature map保存本地磁盘
4. 训练SVM。利用feature map训练SVM来对目标和背景进行分类（每个类一个二分类SVM）
5. 边界框回归。训练将输出校正因子的线性回归分类器



2. Faster R-CNN

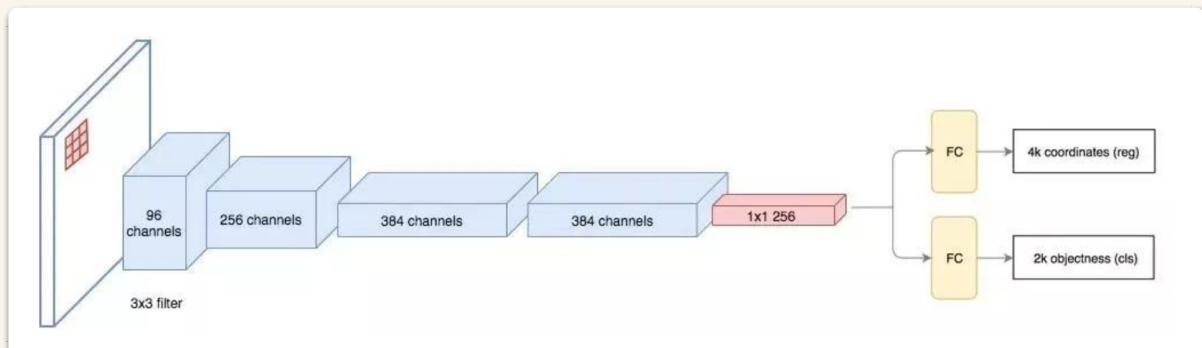


模型特点：

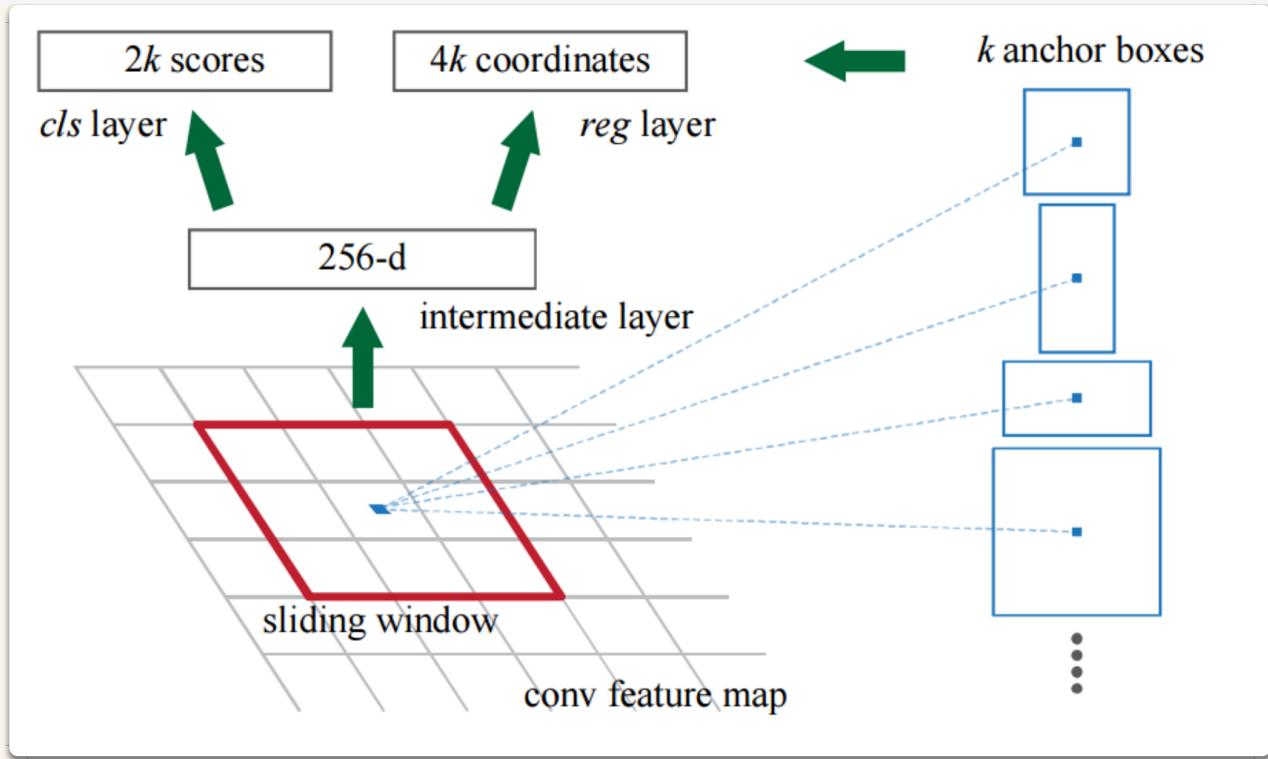
- 利用RPN代替最后一层的selective search

RPN：

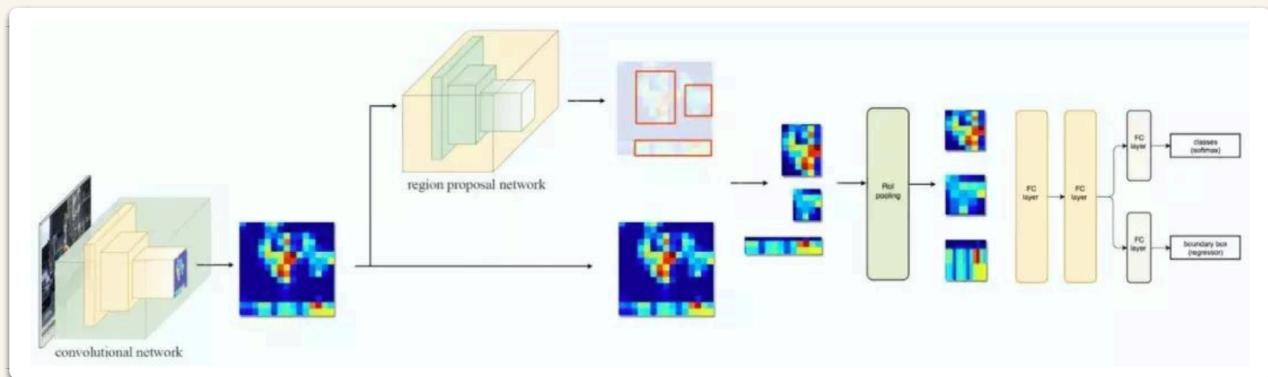
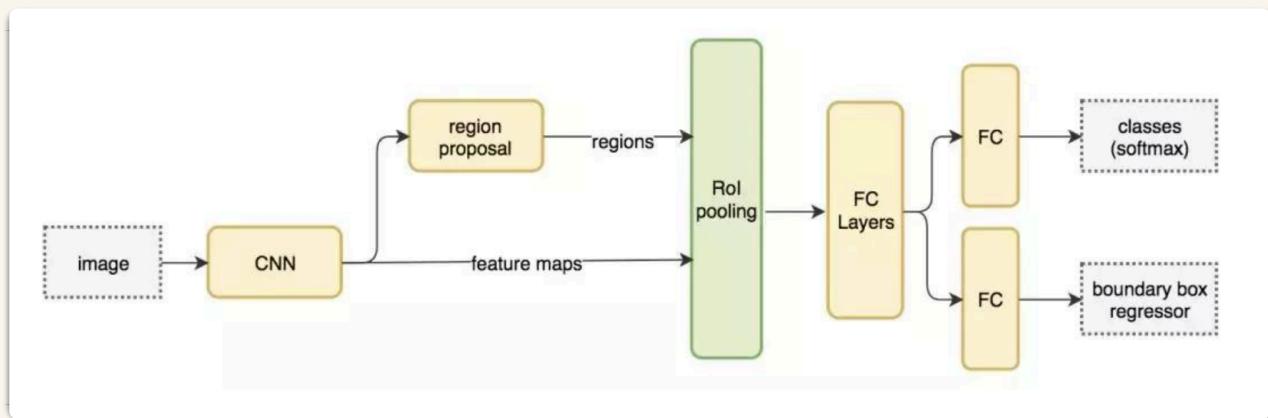
- 在最后一层Feature map上滑动一个 3×3 的卷积核，最后输出一个 $1 \times 1 \times 256$ 维的vector，送入两个FC进行bbox regression和obj分数(是否包含物体)。Obj分数通过生成的bbox和ground truth计算IOU得出。这两个objectiveness分数由两个分类器组成(带有目标的类别clf和不带有目标的类别clf)



- RPN对特征图每个位置做k次运算(k为Anchor Boxes数量)，然后输出 $4 \times k$ 个坐标和 $2 \times k$ 个得分



Faster-RCNN Pipeline:

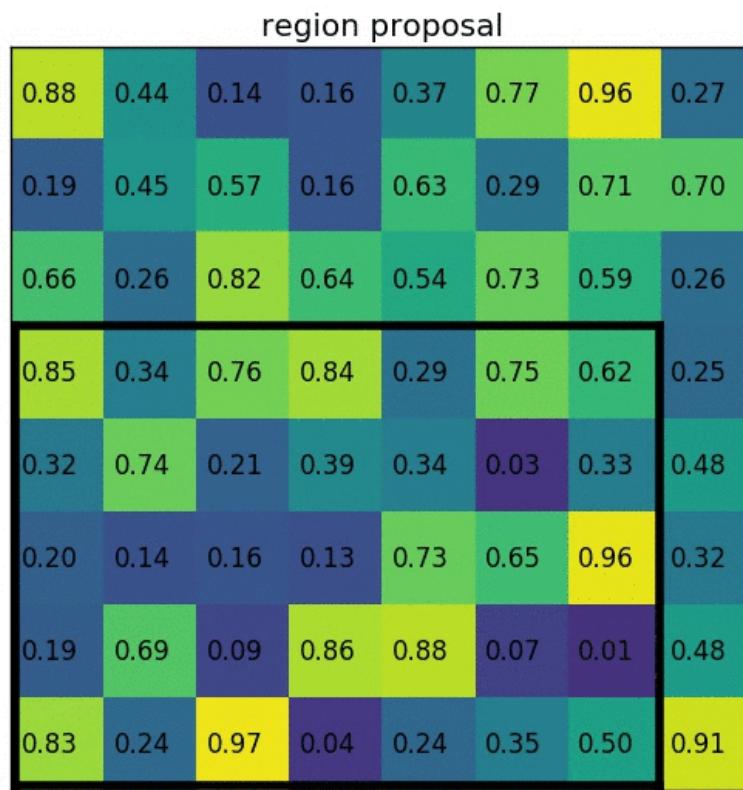


RPN存在问题:

1. 最小的Anchor尺度是128x128，而coco的小目标很多且尺度远小于这个。为了侦测到小目标，Faster–RCNN不得不放大输入image size，导致计算量增加，而同时大目标可能超过最大的anchor尺度（512x512）
2. 最大的anchor是512x512，而预测层感受野仅为228。一般来说，感受野一定要大于anchor大小
3. 小目标的anchor太少太稀疏，大目标的anchor太多太密集，造成计算冗余。需要忽略跨界边框才能使得模型收敛。

ROI Pooling:

- 输入：
 1. 特征图（feature map），指的是上面所示的特征图，在Fast RCNN中，它位于RoI Pooling之前，在Faster RCNN中，它是与RPN共享那个特征图，通常我们常常称之为“share_conv”；
 2. RoiS，其表示所有RoI的N*5的矩阵。其中N表示RoI的数量，第一列表示图像index，其余四列表示其余的左上角和右下角坐标。
- 具体操作：
 1. 根据输入image，将RoI映射到feature map对应位置
注：映射规则比较简单，就是把各个坐标除以“输入图片与feature map的大小的比值”，得到了feature map上的box坐标（一次量化）
 2. 将映射后的区域划分为相同大小的sections（sections数量与输出的维度相同，二次量化）
 - 不能整除时取整
 3. 对每个sections进行max pooling操作
- 输出：
 - 输出是batch个vector，其中batch的值等于RoI的个数，vector的大小为channel * w * h；RoI Pooling的过程就是将一个个大小不同的box矩形框，都映射成大小固定（w * h）的矩形框。
- 缺点：
 - 两次量化：第一次在确定RoI边界框，第二次在将边界框划分成输出维度个数的sections。

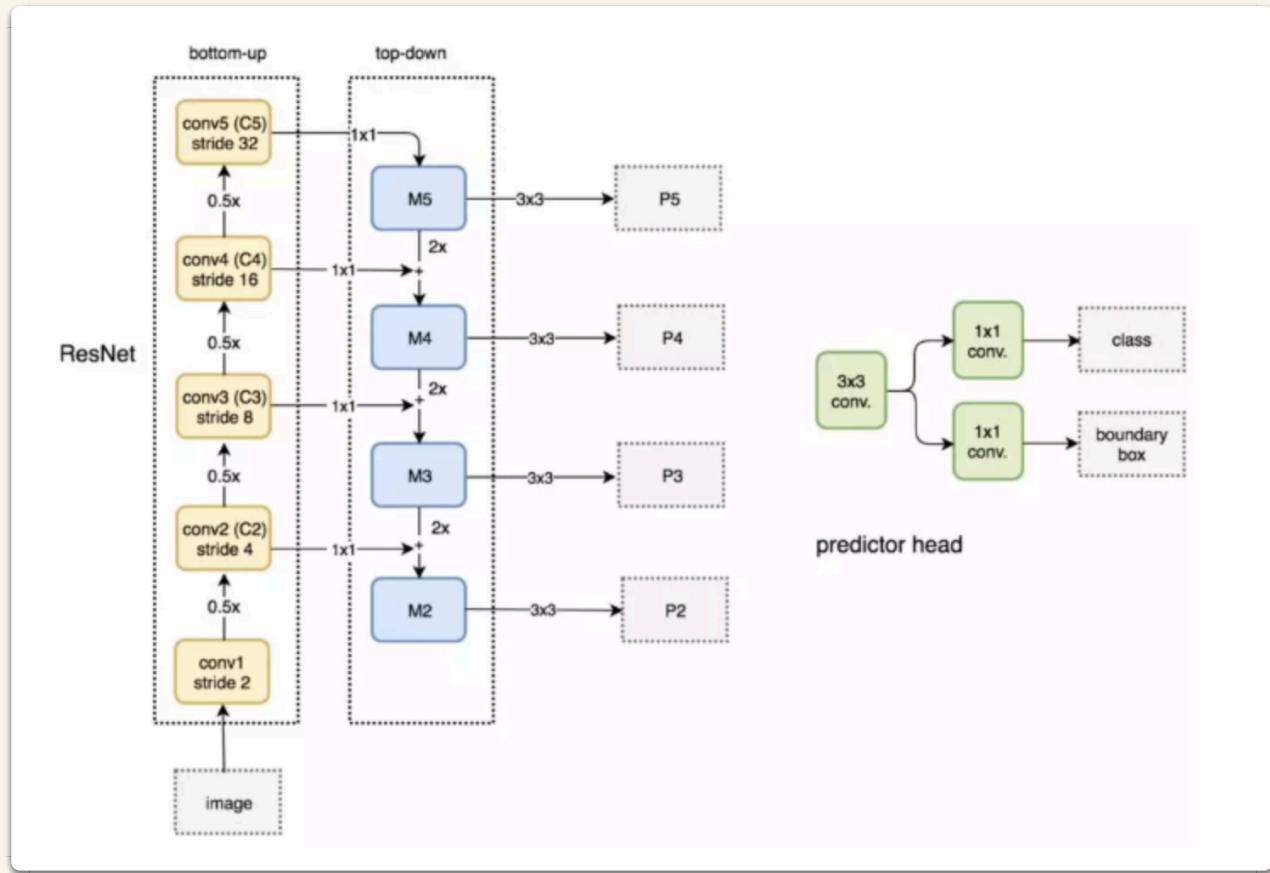


ROI Alignment:

- 输入：同ROI Pooling
- 具体操作：
 1. 根据输入image，将ROI映射到feature map对应位置。该步骤取消量化操作，使用双线性内插的方法获得坐标为浮点数的像素点上的图像数值,从而将整个特征聚集过程转化为一个连续的操作
 2. 将映射后的区域划分为相同大小的sections，继续使用双线性插值获取浮点数坐标位置的图像数值
 3. 在每个section中计算固定四个坐标位置，用双线性内插的方法计算出这四个位置的图像数值，然后进行最大池化操作。
- 输出：同ROI Pooling

3. FPN (Feature Pyramid Network)

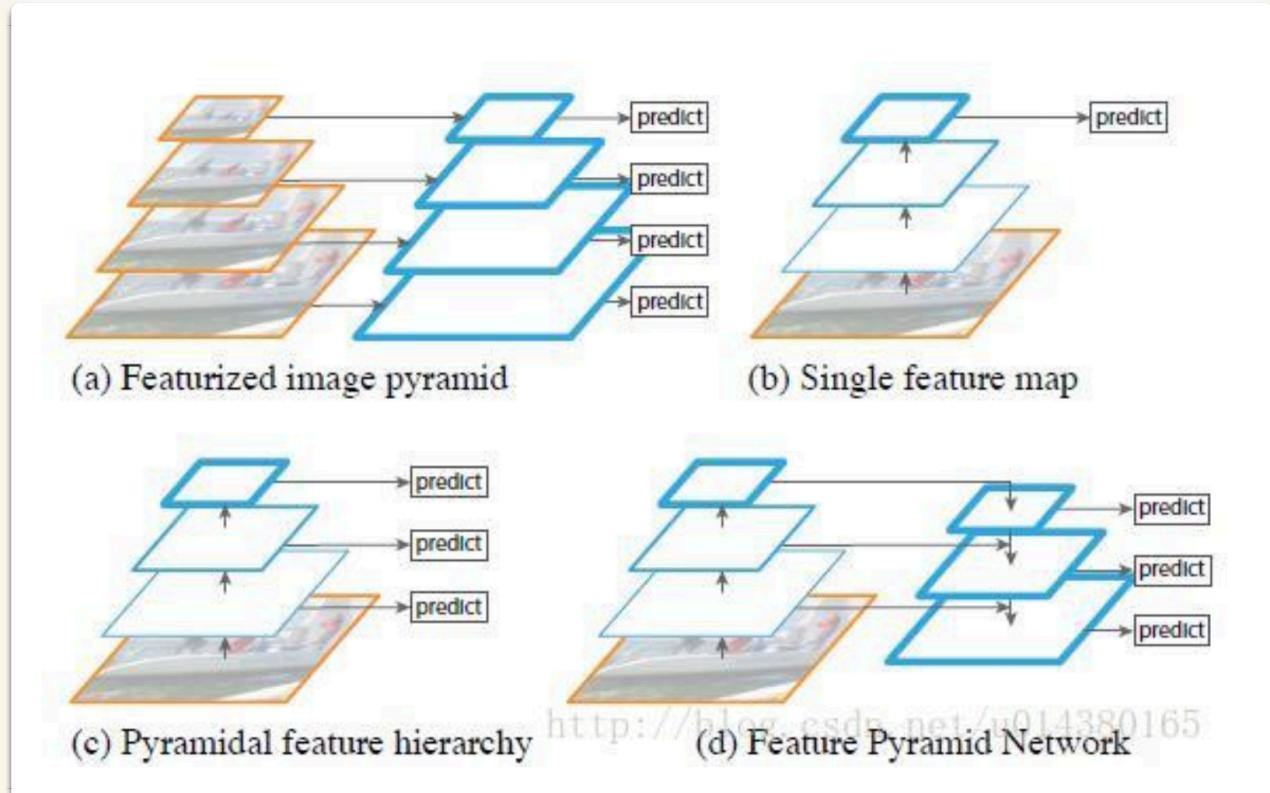
模型特点：



低层的特征语义信息比较少，但是目标位置准确；高层的特征语义信息比较丰富，但是目标位置比较粗略。另外虽然也有些算法采用多尺度特征融合的方式，但是一般是采用融合后的特征做预测，而本文不一样的地方在于预测是在不同特征层独立进行的。

1. Bottom-up pathway and Top-down pathway: Top-down 是上采样的一个过程
2. FPN for RPN: 将single-scale feature map替换成FPN(multi-scale feature)，代替原来只在最后一层C5滑动卷积核

四种类似结构：



1. 图像金字塔，将图像做成不同scale并提取其对应scale的特征
2. 类似SPP net, Fast/Faster RCNN, 采用最后一层特征
3. 类似SSD，从不同特征层抽取做独立预测，不增加额外计算量
4. FPN，通过上采样和低层特征融合，每层独立预测。相同大小feature map归为同一个stage

模型结构：

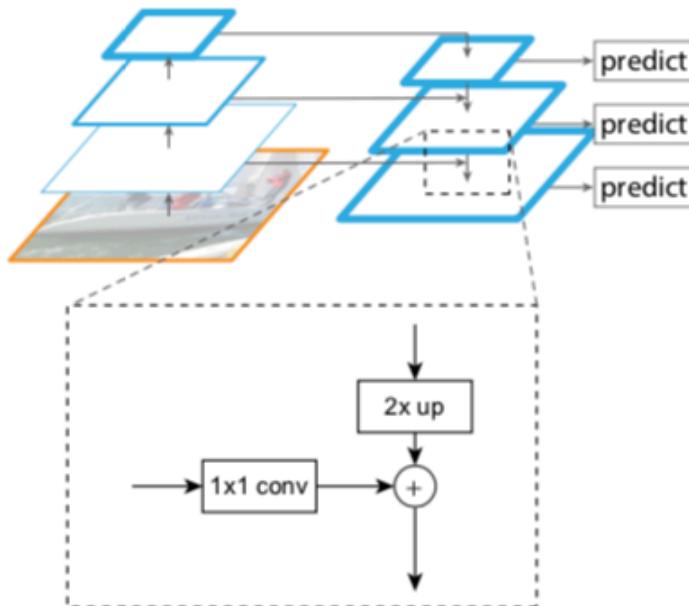
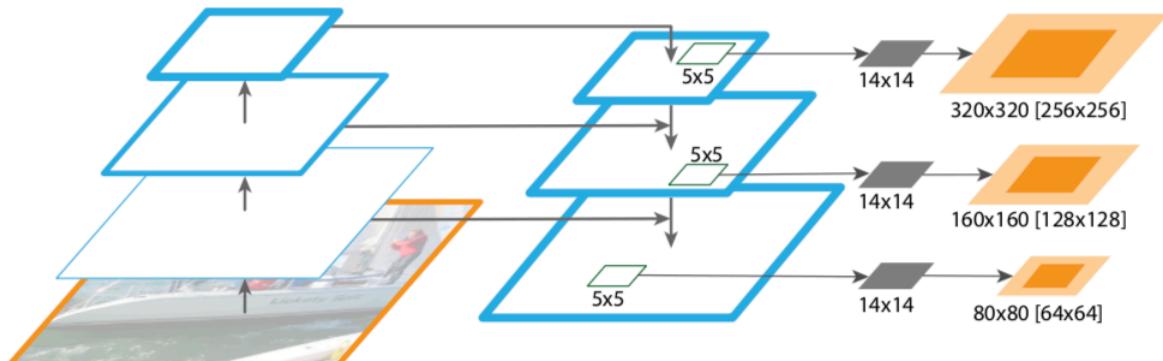


Figure 3. A building block illustrating the lateral connection and the top-down pathway, merged by addition.



输出：

- 将不同层级的融合feature map输出到RPN里面作进一步ROI提取，得到的anchors数量显著提升，AR也显著提升。

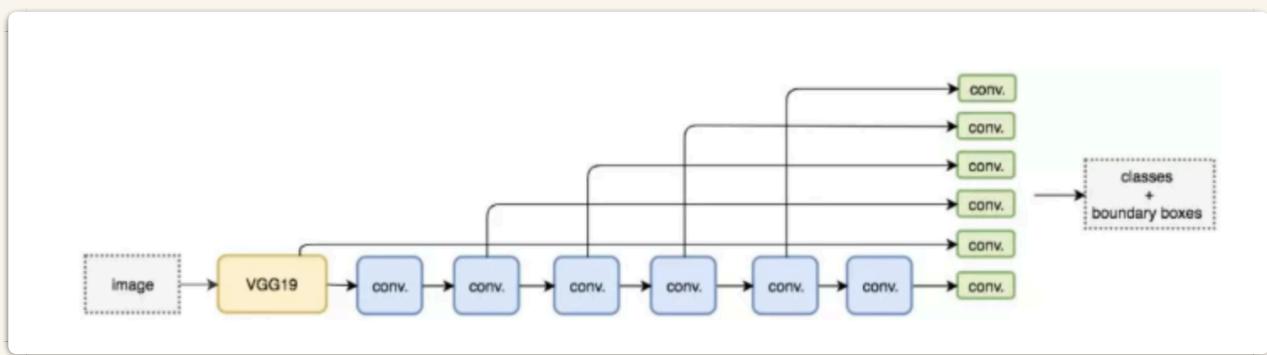
RPN	feature	# anchors	lateral?	top-down?	AR ¹⁰⁰	AR ^{1k}	AR ^{1k} _s	AR ^{1k} _m	AR ^{1k} _l
(a) baseline on conv4	C_4	47k			36.1	48.3	32.0	58.7	62.2
(b) baseline on conv5	C_5	12k			36.3	44.9	25.3	55.5	64.2
(c) FPN	$\{P_k\}$	200k	✓	✓	44.0	56.3	44.9	63.4	66.2
<i>Ablation experiments follow:</i>									
(d) bottom-up pyramid	$\{P_k\}$	200k	✓		37.4	49.5	30.5	59.9	68.0
(e) top-down pyramid, w/o lateral	$\{P_k\}$	200k		✓	34.5	46.1	26.5	57.4	64.7
(f) only finest level	P_2	750k	✓	✓	38.4	51.3	35.1	59.7	67.6

4. RefineDet

One-Stage:

1. SSD (Single Shot Multibox Detector)

模型特点：



1. Default Boxes:

1. 与Fast-RCNN中Anchor相似，不同的是SSD在多个特征层上面取Default Boxes。
2. Default Boxes中包含location (绝对坐标)，对每个类别的confidence包括背景(假设c个类，这里区别于YOLO，YOLO的类不包含背景)。所以每个bbox需要预测 $4+c$ 个值
3. Default Boxes长宽比一般为{1,2,3,1/2,1/3}中选取
4. 输入为图片 300x300，在conv4_3, conv7, conv8_2, conv9_2, conv10_2, conv11_2分别提取4,6,6,6,4,4个default boxes。由于以上特征图的大小分别是38x38, 19x19, 10x10, 5x5, 3x3, 1x1，所以一共得到 38x38x4(小anchor)+19x19x6+10x10x6+5x5x6(中anchor)+3x3x4+1x1x4(大anchor)=8732个default box. 对一张300x300的图片输入网络将会针对这8732个default box预测8732个边界框。

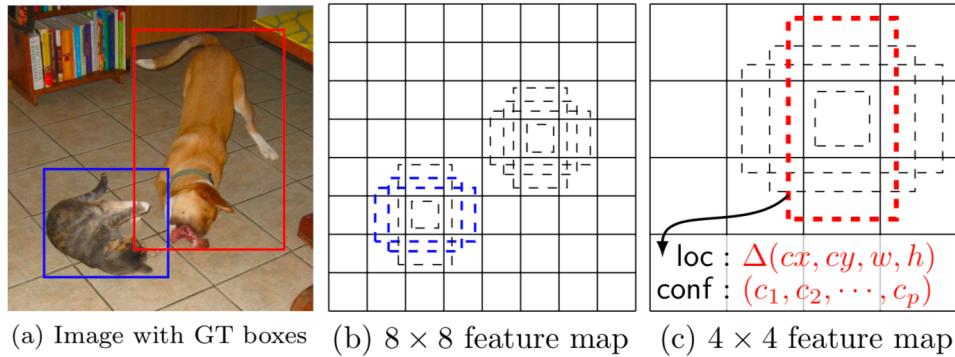


Fig. 1: SSD framework. (a) SSD only needs an input image and ground truth boxes for each object during training. In a convolutional fashion, we evaluate a small set (e.g. 4) of default boxes of different aspect ratios at each location in several feature maps with different scales (e.g. 8×8 and 4×4 in (b) and (c)). For each default box, we predict both the shape offsets and the confidences for all object categories ((c_1, c_2, \dots, c_p)). At training time, we first match these default boxes to the ground truth boxes. For example, we have matched two default boxes with the cat and one with the dog, which are treated as positives and the rest as negatives. The model loss is a weighted sum between localization loss (e.g. Smooth L1 [6]) and confidence loss (e.g. Softmax).

模型结构：

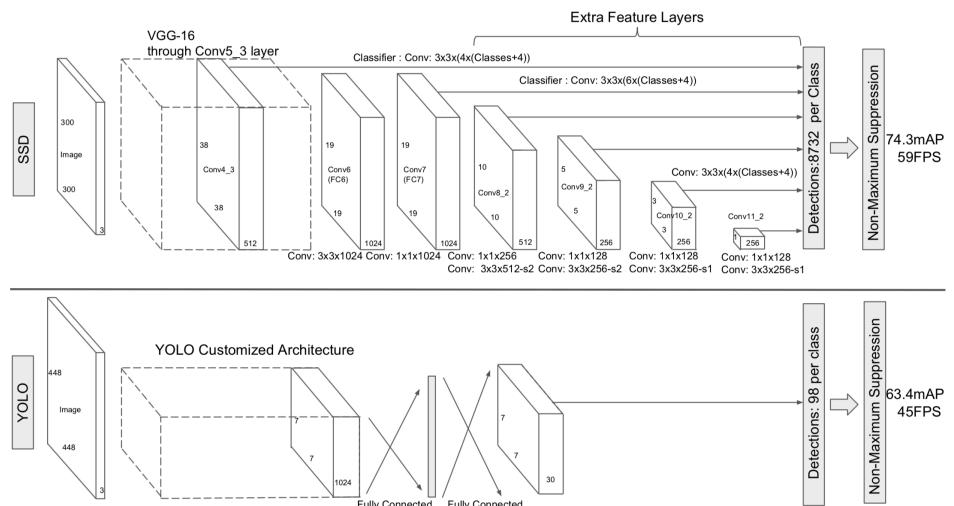


Fig. 2: A comparison between two single shot detection models: SSD and YOLO [5]. Our SSD model adds several feature layers to the end of a base network, which predict the offsets to default boxes of different scales and aspect ratios and their associated confidences. SSD with a 300×300 input size significantly outperforms its 448×448 YOLO counterpart in accuracy on VOC2007 test while also improving the speed.

Hard Negative Mining:

- 因为正负样本差异巨大，我们选择将负样本按照confident score排序，选取置

信度最小的default box来train，正负比达到1:3。能够有效提升收敛的效率

Loss Function:

$$L(x, c, l, g) = \frac{1}{N} (L_{conf}(x, c) + \alpha L_{loc}(x, l, g))$$

- N为bounding box数量。前一项为Bounding box与目标的confidence，后一项为相应的回归位置
- 回归采用**L1-smooth Loss**, confident loss是典型的softmax loss

Conclusion:

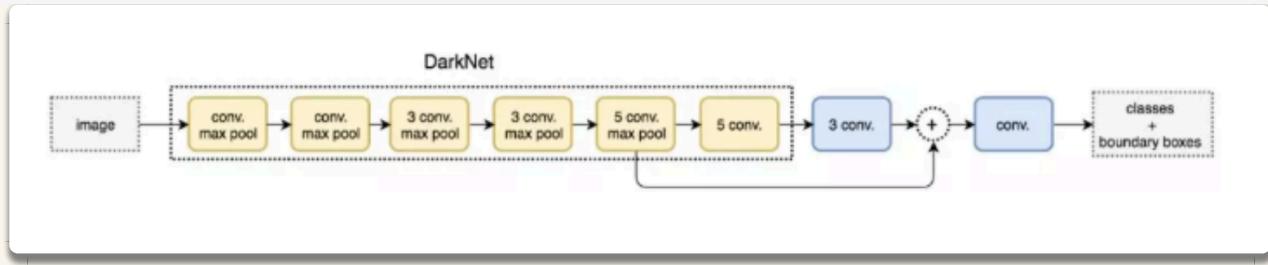
- One trick of detecting small object is **Random Crop** operation which can be thought as a "zoom in" operation.
- Multi-scale cone bounding box outputs attached to multiple feature maps at the top of the network
- **SSD300 (input size 300x300). SSD512 (input size 512x512)**
- Future work: using **RNN** to detect and track objects in video simultaneously
- 小尺度anchor多且密集，大尺度anchor少且稀疏，输入图像无需放大去侦测小目标，计算速度更快，不忽略跨界anchor训练效果更好。

关于Anchor的问题：

1. anchor没有设置感受野对齐，对大目标IOU变化不大，对小目标IOU变化剧烈，尤其是感受野不够大时候，anchor有可能偏移出感受野区域，影响性能。
2. anchor必须人工给定，尺度和比例必须覆盖相应的任务可能出现的所有目标。

2. DSSD

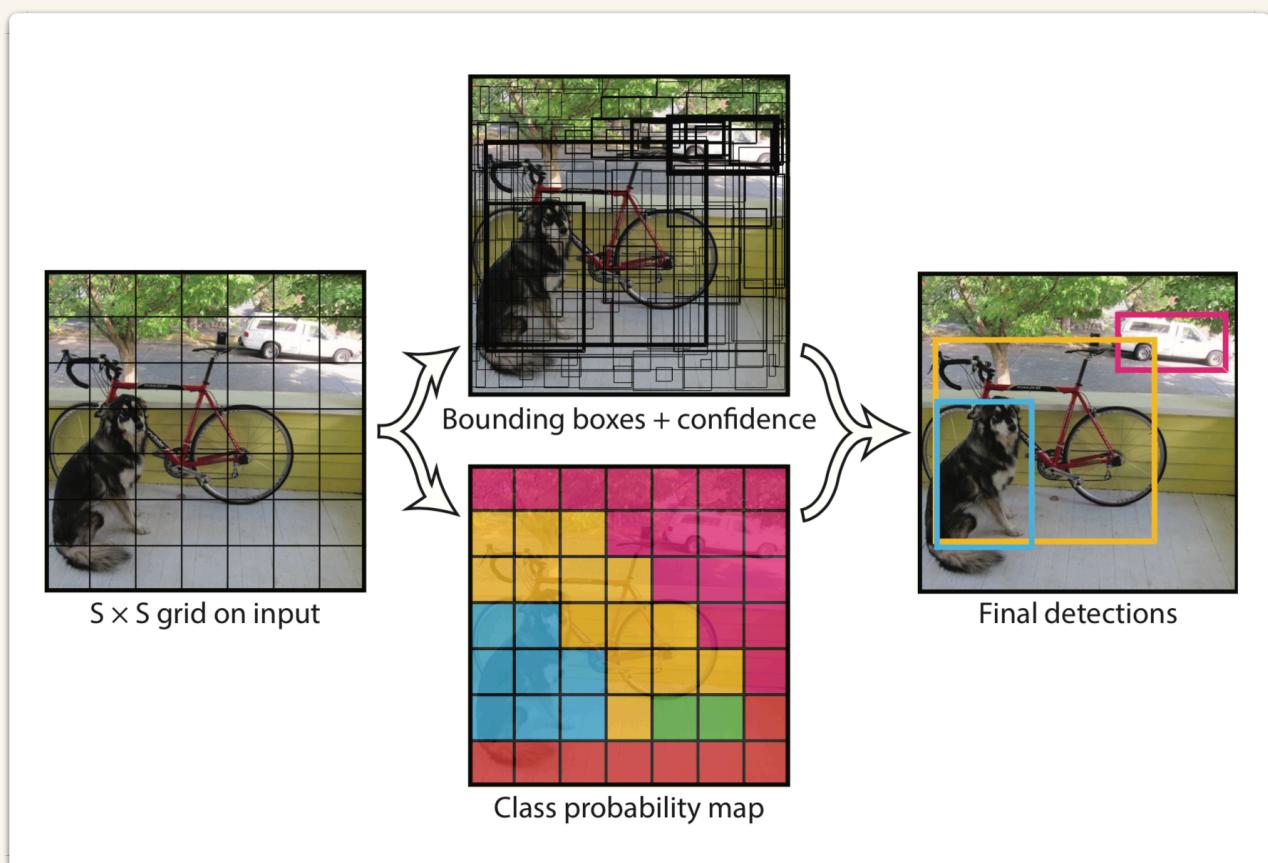
3. YOLO



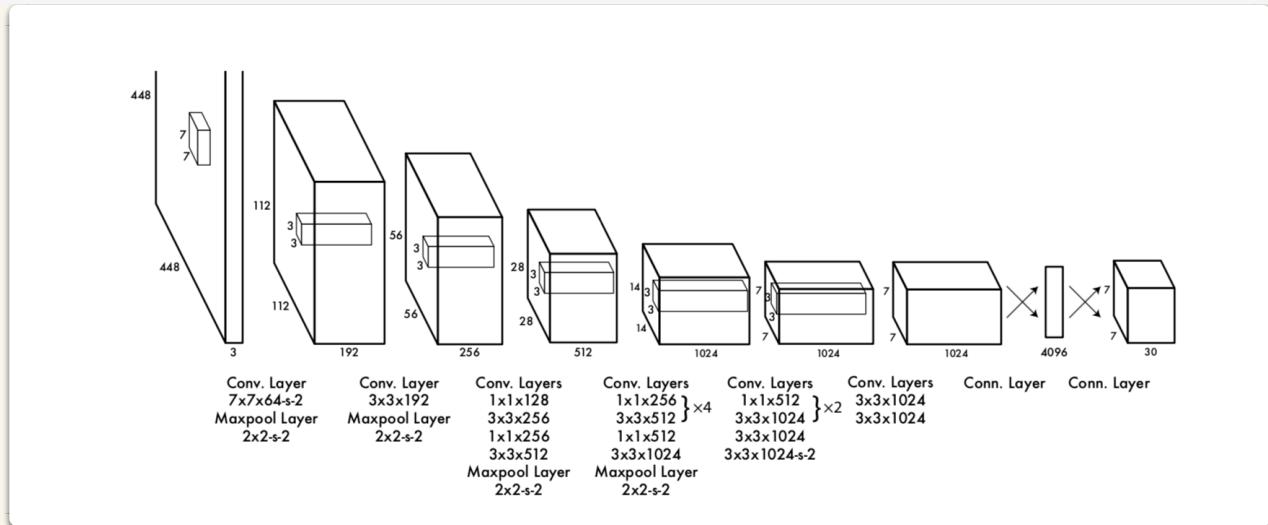
v1

Bounding boxes Design:

- x, y, w, h (相对坐标；相对于所属边框的x, y; 相对于原始图像的w, h比例)
And $confidence$, $confidence = Pr(Object) * IOU_{pred}^{truth}$, $Pr(Object)$ 为0或1代表存在object or not
- 若某个GT的中心点落入某个grid cell中，则这个格子负责预测该物体。每个格子输出B个bboxes信息，以及C个class中属于某个类别的概率
- 总共预测 $S \times S \times (B * 5 + C)$ tensor, S为长宽grid数量, B为每个grid预测 bounding boxes数量, C为class的总数



Network Design:



- 24 Layers Convolutional layers followed by 2 fc layers
- training method: pretrain conv on imagenet dataset

Loss:

- 使用均方差MSE，即网络输出 $S \times S \times (B * 5 + C)$ 维向量和GT对应的 $S \times S \times (B * 5 + C)$ 均方差
- Coord为 x, y, w, h 相对于cell的偏置，IOU为MSE

$$Loss = \sum_{i=0}^{S^2} CoordError + IOUError + ClassError$$

Limitation:

- 每个grid cell只能预测一个class，对临近物体或者小物体的侦测有影响。若两个中心点落入同一个格子，则选取IOU大的一个物体作为预测输出。
- 对长宽比异常的物体侦测有难度 (unusual ratios and configurations)
- 大小bounding boxes的errors贡献一致，可是小的error对小的bounding box的IOU影响巨大，会最终导致定位错误

V2

改进:

	YOLO								YOLOv2
batch norm?	✓	✓	✓	✓	✓	✓	✓	✓	✓
hi-res classifier?		✓	✓	✓	✓	✓	✓	✓	✓
convolutional?		✓	✓	✓	✓	✓	✓	✓	✓
anchor boxes?		✓	✓						
new network?			✓	✓	✓	✓	✓	✓	✓
dimension priors?				✓	✓	✓	✓	✓	✓
location prediction?					✓	✓	✓	✓	✓
passthrough?						✓	✓	✓	✓
multi-scale?							✓	✓	✓
hi-res detector?								✓	
VOC2007 mAP	63.4	65.8	69.5	69.2	69.6	74.4	75.4	76.8	78.6

Convolution with anchor box:

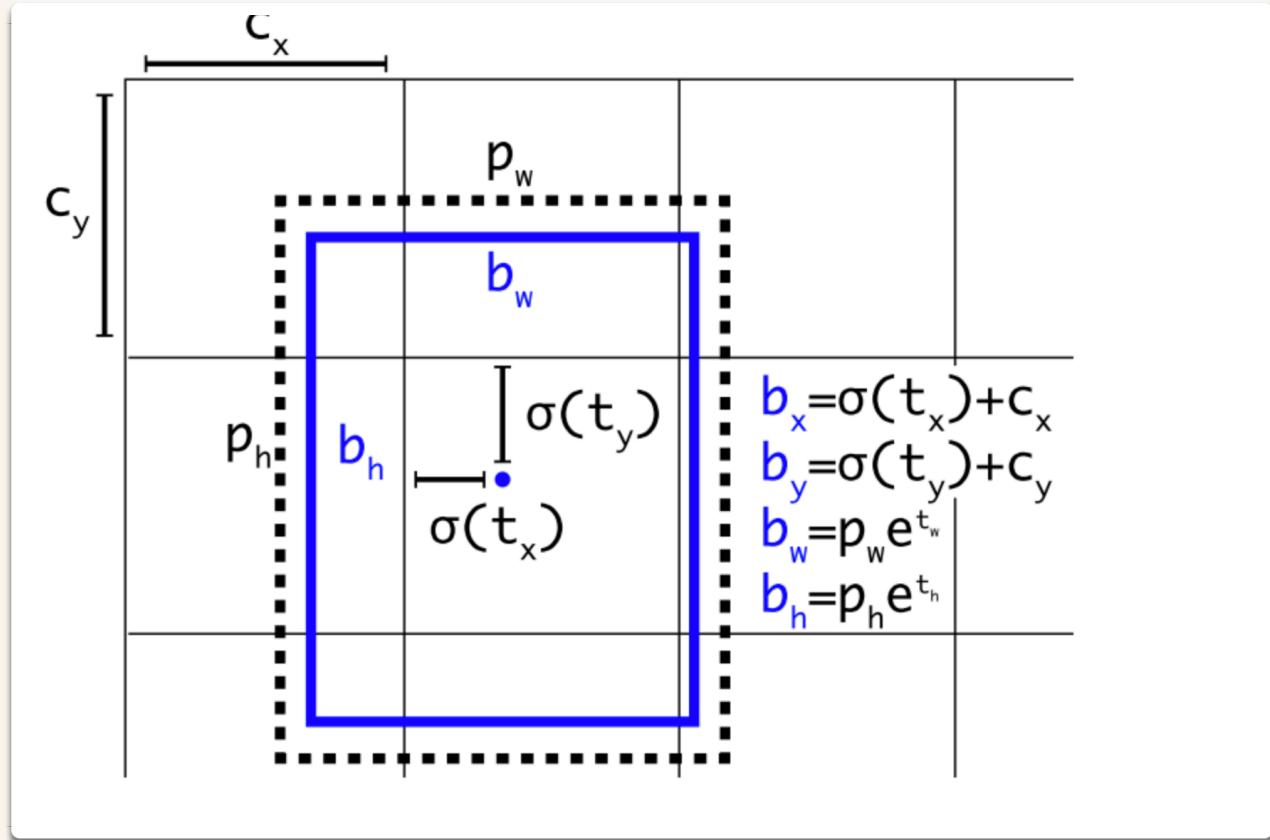
- 摒弃FC layer, FC导致空间信息丢失
- 使用avg代替Flatten, 舍弃部分池化层以换取更高的分辨率, 输入分辨率为416, 下采样总步长32, 最终特征图为 13×13

Dimension Clustering:

- 舍弃原本的人工筛选anchor box尺度方法
- 对训练集做kmeans聚类获取5个聚类中心, 作为anchor box的设置。

Direct Location Prediction:

- 使用相对坐标, 不同于faster RCNN。该坐标具有网格进行约束, 模型的更稳定, 减少训练时间。



Fine grained Feature(细粒度feature提取):

1. YOLOv2提取Darknet-19最后一个max pool层的输入，得到26x26x512的特征图。
2. 经过1x1x64的卷积以降低特征图的维度，得到26x26x64的特征图，然后经过pass through层的处理变成13x13x256的特征图（抽取原特征图每个2x2的局部区域组成新的channel，即原特征图大小降低4倍，channel增加4倍）
3. 再与13x13x1024大小的特征图连接，变成13x13x1280的特征图，最后在这些特征图上做预测。使用Fine-Grained Features，YOLOv2的性能提升了1%.

V3

改进：

1. 新网络结构：DarkNet-53（更深的网络，残差模型），
2. 融合FPN（多尺度预测），在最后三层是用FPN得出的feature maps作为特征层
3. 用逻辑回归替代softmax作为分类器，因为softmax并没有对结果有提升

模型结构：

Type	Filters	Size	Output
Convolutional	32	3×3	256×256
Convolutional	64	$3 \times 3 / 2$	128×128
1×	Convolutional	32	1×1
	Convolutional	64	3×3
	Residual		128×128
2×	Convolutional	128	$3 \times 3 / 2$
	Convolutional	64	1×1
	Convolutional	128	3×3
8×	Residual		64×64
	Convolutional	256	$3 \times 3 / 2$
	Convolutional	128	1×1
8×	Convolutional	256	3×3
	Residual		32×32
	Convolutional	512	$3 \times 3 / 2$
8×	Convolutional	256	1×1
	Convolutional	512	3×3
	Residual		16×16
4×	Convolutional	1024	$3 \times 3 / 2$
	Convolutional	512	1×1
	Convolutional	1024	3×3
	Residual		8×8
	Avgpool		Global
	Connected		1000
	Softmax		

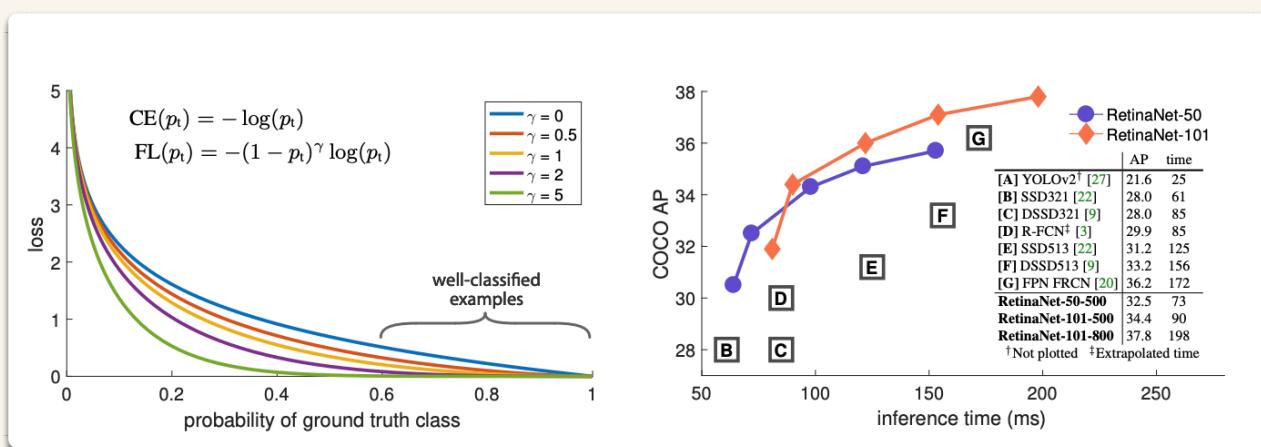
FPN多尺度提取：

- 在最后3个stage中提取feature map，大小分别为 32×32 , 16×16 , 8×8

模型效果：

	backbone	AP	AP ₅₀	AP ₇₅	AP _S	AP _M	AP _L
<i>Two-stage methods</i>							
Faster R-CNN+++ [5]	ResNet-101-C4	34.9	55.7	37.4	15.6	38.7	50.9
Faster R-CNN w FPN [8]	ResNet-101-FPN	36.2	59.1	39.0	18.2	39.0	48.2
Faster R-CNN by G-RMI [6]	Inception-ResNet-v2 [21]	34.7	55.5	36.7	13.5	38.1	52.0
Faster R-CNN w TDM [20]	Inception-ResNet-v2-TDM	36.8	57.7	39.2	16.2	39.8	52.1
<i>One-stage methods</i>							
YOLOv2 [15]	DarkNet-19 [15]	21.6	44.0	19.2	5.0	22.4	35.5
SSD513 [11, 3]	ResNet-101-SSD	31.2	50.4	33.3	10.2	34.5	49.8
DSSD513 [3]	ResNet-101-DSSD	33.2	53.3	35.2	13.0	35.4	51.1
RetinaNet [9]	ResNet-101-FPN	39.1	59.1	42.3	21.8	42.7	50.2
RetinaNet [9]	ResNeXt-101-FPN	40.8	61.1	44.1	24.1	44.2	51.2
YOLOv3 608 × 608	Darknet-53	33.0	57.9	34.4	18.3	35.4	41.9

4. RetinaNet



问题所在：

1. Single Stage Detector之所以识别率低是因为class imbalance。负样本数量远远大于正样本。
2. Negative sample数量过多导致贡献的Loss淹没了positive的Loss，即分类器将它全部分为负样本的准确率虽然高，但是召回率低。
3. 大多数训练样本为**easy negative**，非常容易被区分的背景类，单个样本loss非常小，反向计算时梯度非常小，对收敛作用非常小。我们需要的是hard positive/negative的大loss来促进收敛。
4. OHEM：对loss排序，筛选最大的loss来进行训练。保证每次训练都是hard example

四类example: hard positive、hard negative、easy positive、easy negative

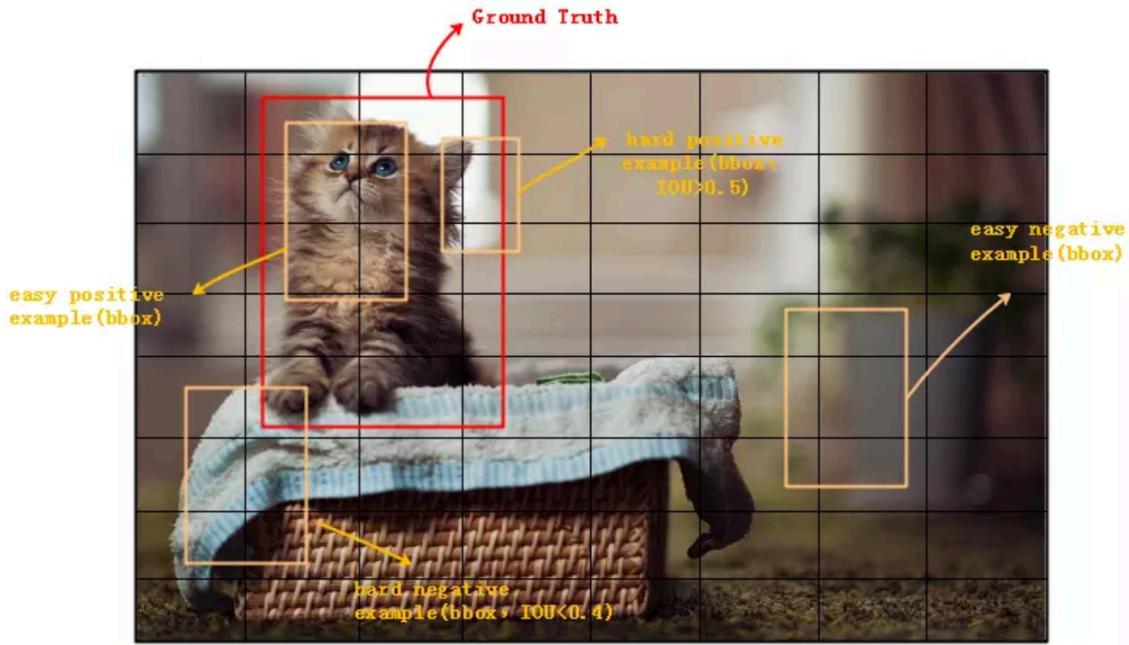


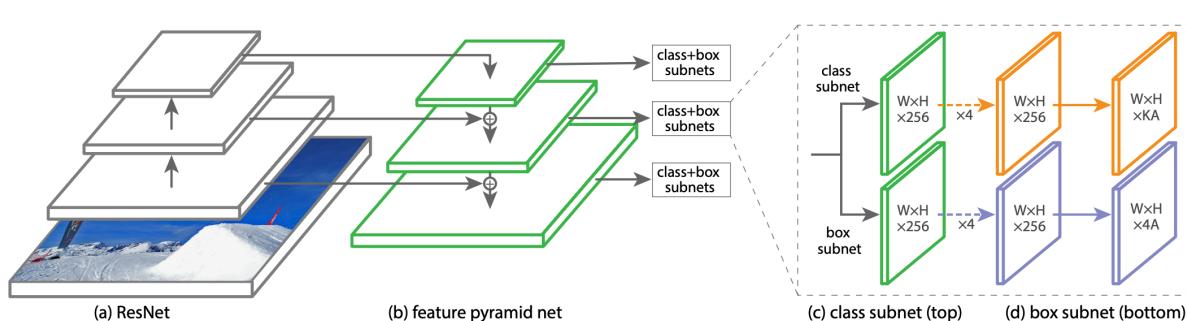
图1：4类example

Focal Loss:

$$FL(p_t) = -\alpha_t(1 - p_t)^\gamma \log(p_t)$$

- 无论是前景类还是背景类， p_t 越大，权重 $(1 - p_t)^\gamma$ 就越小。也就是说easy example可以通过权重进行抑制；
- α_t 用于调节positive和negative的比例，前景类别使用 α_t 时，对应的背景类别使用 $1 - \alpha_t$ ；
- γ 和 α_t 的最优值是相互影响的，所以在评估准确度时需要把两者组合起来调节。作者在论文中给出 $\gamma = 2$ $\alpha_t = 0.25$ 时，ResNet-101+FPN作为backbone的结构有最优的性能。

RetinaNet结构：



- ResNet+FPN

5. RFBNet

模型特点：

1. 模拟人类视觉的感受野加强网络的特征提取能力，RFB利用空洞卷积模拟复现人眼pRF尺寸和偏心率之间的关系。卷积核大小和膨胀率与pRF尺寸和偏心率有着正比例关系。
2. 借助了split–transform–merge的思想，在Inception的基础上加入了dilated conv，增大了感受野
3. 整体基于SSD进行改进，速度快而且精度高（VGG的精度能与two–stage ResNet 精度相提并论）

Receptive Field Blocks：

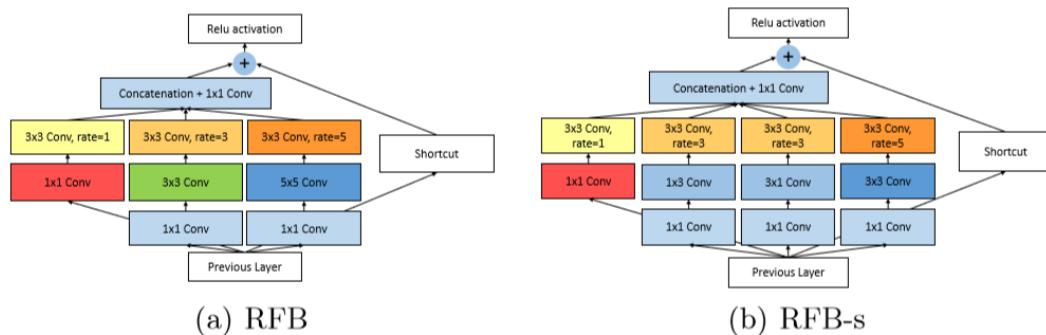


Fig. 4. The architectures of RFB and RFB-s. RFB-s is employed to mimic smaller pRFs in shallow human retinotopic maps, using more branches with smaller kernels. Following [32], we use two layers of 3×3 conv replacing 5×5 to reduce parameters, which is not shown for better visualization.

- 左：在第一层卷积层后面增加dilated conv，最后concat之后过一个1x1 Conv
- 右：使用1x3 和 3x1的kernel替代 3x3 conv，减少参数。使用两个3x3 conv替代5x5 conv，图中没有显示出来是为了更好的visualization

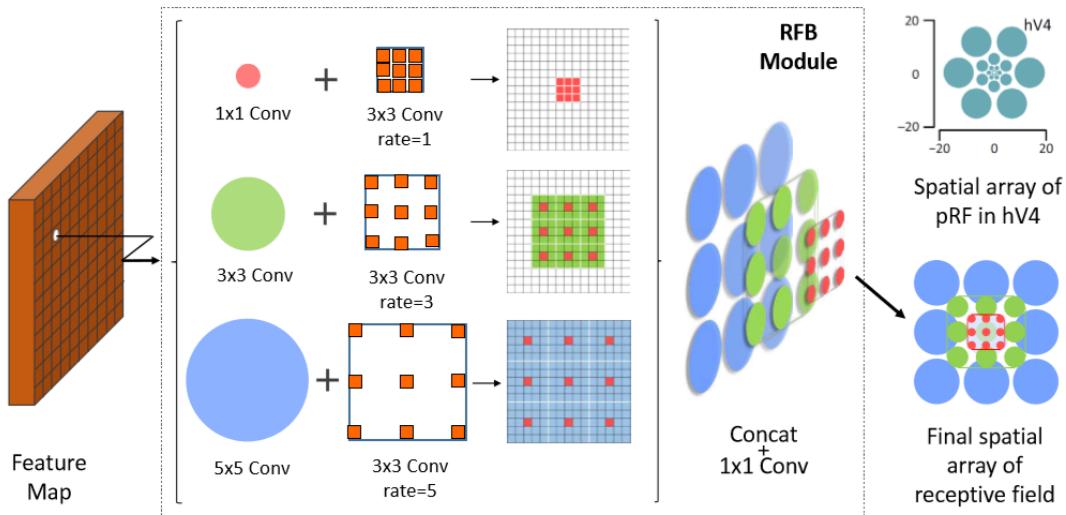
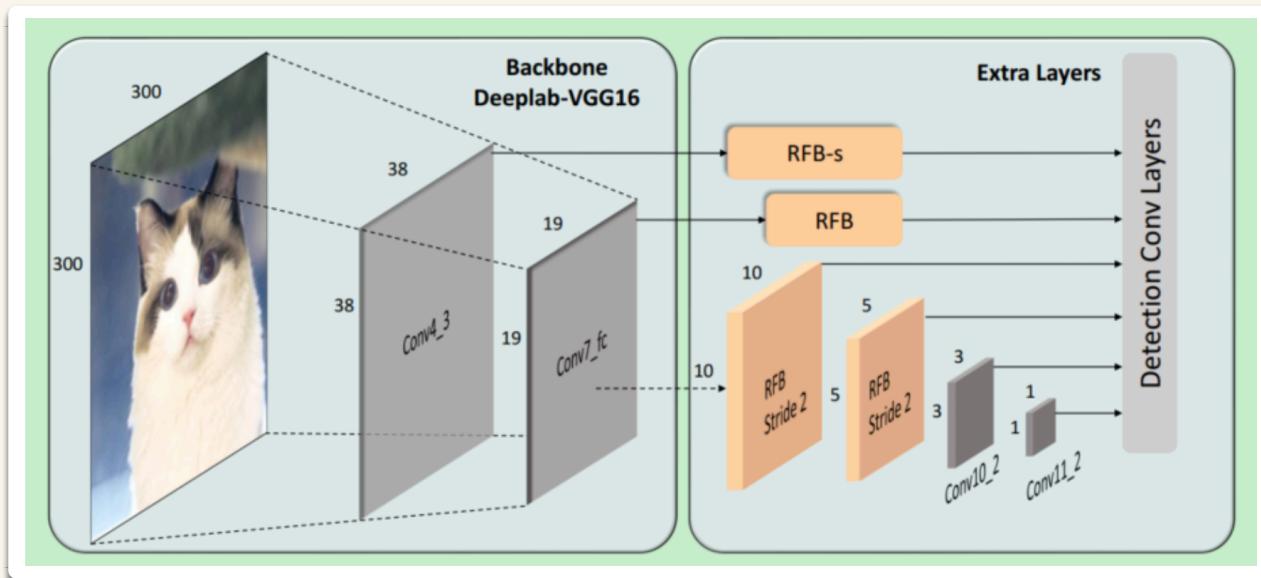


Fig. 2. Construction of the RFB module by combining multiple branches with different kernels and dilated convolution layers. Multiple kernels are analogous to the pRFs of varying sizes, while dilated convolution layers assign each branch with an individual eccentricity to simulate the ratio between the size and eccentricity of the pRF. With a concatenation and 1×1 conv in all the branches, the final spatial array of RF is produced, which is similar to that in human visual systems, as depicted in Fig. 1.

模型：



1. 将SSD主干网络中conv8, conv9替代为RFB
2. 将conv4_3, conv7_fc后分别接入RFB-s和RFB结构

模型对比：

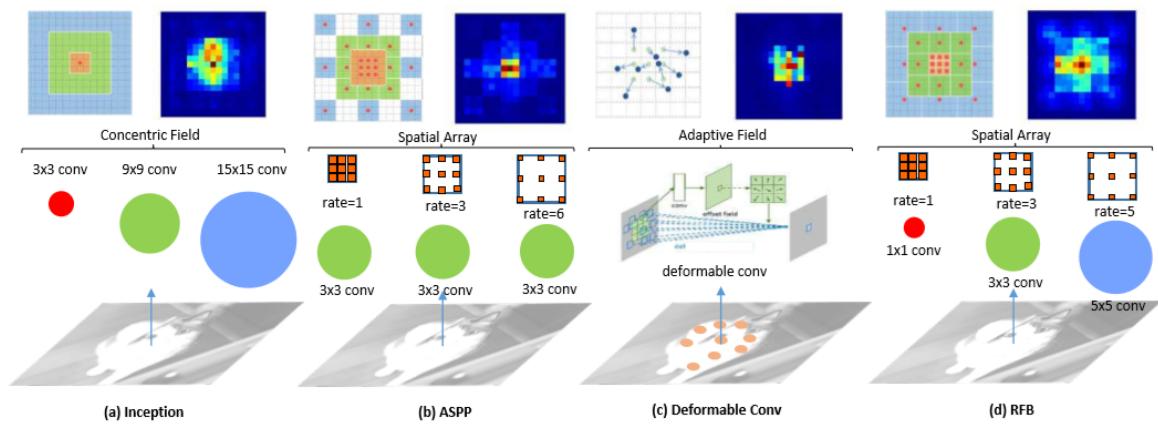


Fig. 3. Four typical structures of Spatial RFs. (a) shows the kernels of multiple sizes in Inception. (b) demonstrates the daisy-like pooling configuration in ASPP. (c) adopts deformable conv to produce an adaptive RF according to object characteristics. (d) illustrates the mechanism of RFB. The color map of each structure is the effective RF derived from one correspondent layer in the trained model, depicted by the same gradient back-propagation method in [23]. In (a) and (b), we adjust the RF sizes in original Inception and ASPP for fair comparison.

- Inception: 单纯的用不同size的conv 并将Feature map concate。感受野从中心发散，没有偏心率的体现
- ASPP: 将不同rate的dilated conv叠加，感受夜中心发散，可是感受范围小。
pRF尺寸过小
- Deformable: 使用偏置项使conv能够解决空间一致性的
object特征
- RFB: 使用dilated卷积，平衡了a, b两者的优点。

6. Region based Fully Convolutional Network(R-FCN)

模型结构：

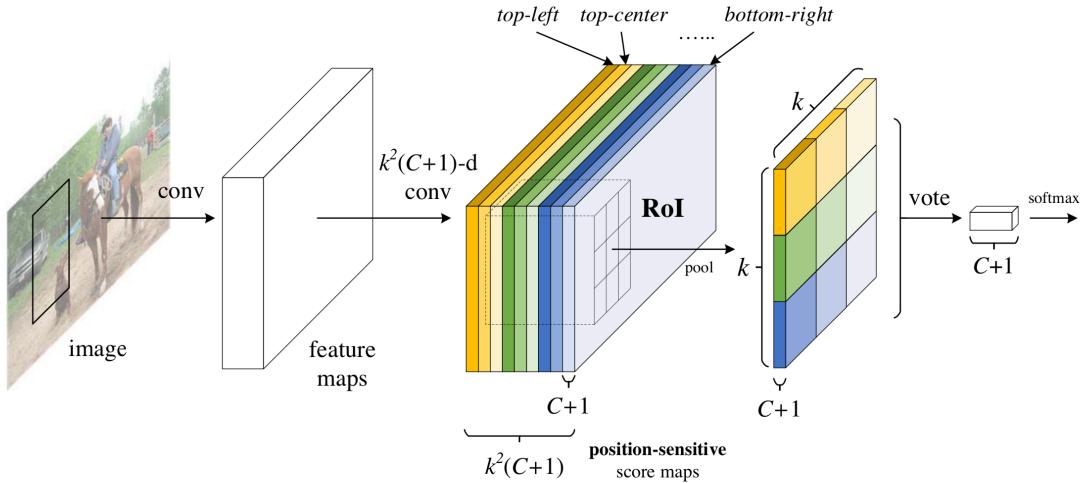


Figure 1: Key idea of **R-FCN** for object detection. In this illustration, there are $k \times k = 3 \times 3$ position-sensitive score maps generated by a fully convolutional network. For each of the $k \times k$ bins in an ROI, pooling is only performed on one of the k^2 maps (marked by different colors).

模型特点：

- 在最后一层的feature map上做卷积运算，channel为 $k^2(C + 1)$ 。与之平行的还有一个RPN分支生成ROIs

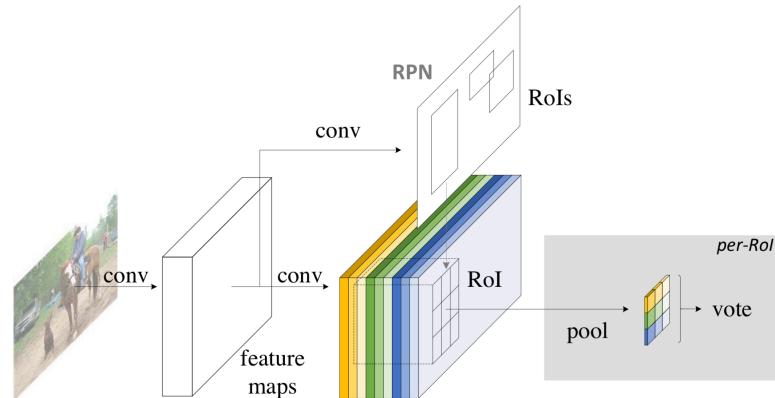


Figure 2: Overall architecture of R-FCN. A Region Proposal Network (RPN) [18] proposes candidate ROIs, which are then applied on the score maps. All learnable weight layers are convolutional and are computed on the entire image; the per-RoI computational cost is negligible.

- k^2 为每一个class的position score map，一共 $C+1$ 个类别包括背景
- 之后通过position sensitive ROI Pooling，为每一个ROI生成相对应Score

Position-sensitive score maps & Position-sensitive ROI pooling:

- 将ROI分隔成 $k \times k$ bins，如果ROI大小为 $w \times h$ ，则每个bins约为 $\frac{w}{k} \times \frac{h}{k}$
- Pooling对于第(I, J)个bin的操作，c为第c-th个class：

$$r_c(i, j | \theta) = \sum_{(x, y) \in bin(i, j)} z_{i, j, c}(x + x_0, y + y_0 | \theta) / n$$

$z_{i, j, c}$ 是 $k^2(C + 1)$ 其中的一个通道, n 是一个 bin 里 pixels 数量

- 作用于 entire images, 区别于之前的 ROI 只作用在 ROI 区域。

7. Pooling Pyramid Network (SSD改进)

模型结构:

- shared tower: 输入为 19×19 , 10×10 , 5×5 , 3×3 , 2×2 , and 1×1 的 feature map
通过 $1 \times 1 \times 512$ 卷积核
- 训练:
 - L1-smooth for box regression

$$Smooth \quad L_1 = \begin{cases} 0.5x^2, & |x| < 1 \\ |x| - 0.5, & x < -1 \text{ or } x > 1 \end{cases}$$

$$Smooth \quad L'_1 = \begin{cases} x, & |x| < 1 \\ -1, & x < -1 \\ 1, & x > 1 \end{cases}$$

- Focal Loss $\alpha = 0.25$, $\gamma = 2$

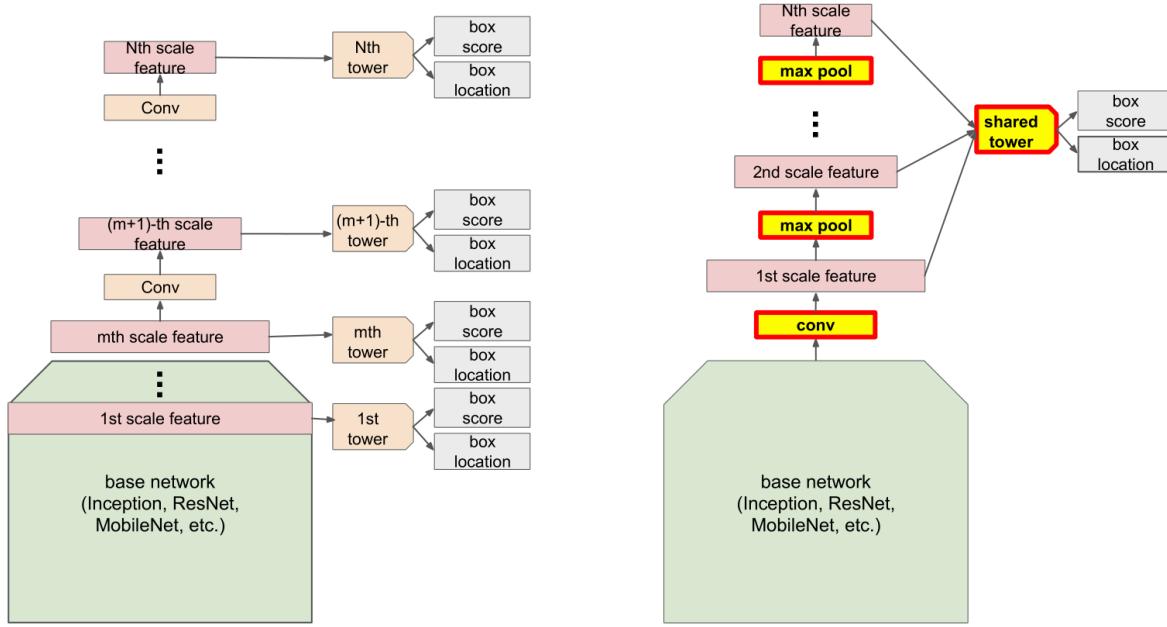


Figure 1. Architecture comparison between the Pooling Pyramid Network (PPN) and vanilla SSD. Left: vanilla SSD, Right: PPN. Note that the changes in PPN are highlighted: (1) using max pool to build the feature pyramid, (2) using shared convolutional predictors for box classification and regression.

模型特点：

- 使用max pooling构建feature pyramid 代替conv特征提取。更快，共享 embedding space
- 将多个box detector减少至一个，避免了跨尺度的分类器的分数错误校准
- 与SSD效果相同，但是size小3倍

Model	mAP	inference FLOPs	number of parameters	GPU inference time
MobileNet SSD	20.8	2.48B	6.83M	27ms
MobileNet PPN	20.3	2.35B	2.18M	26ms

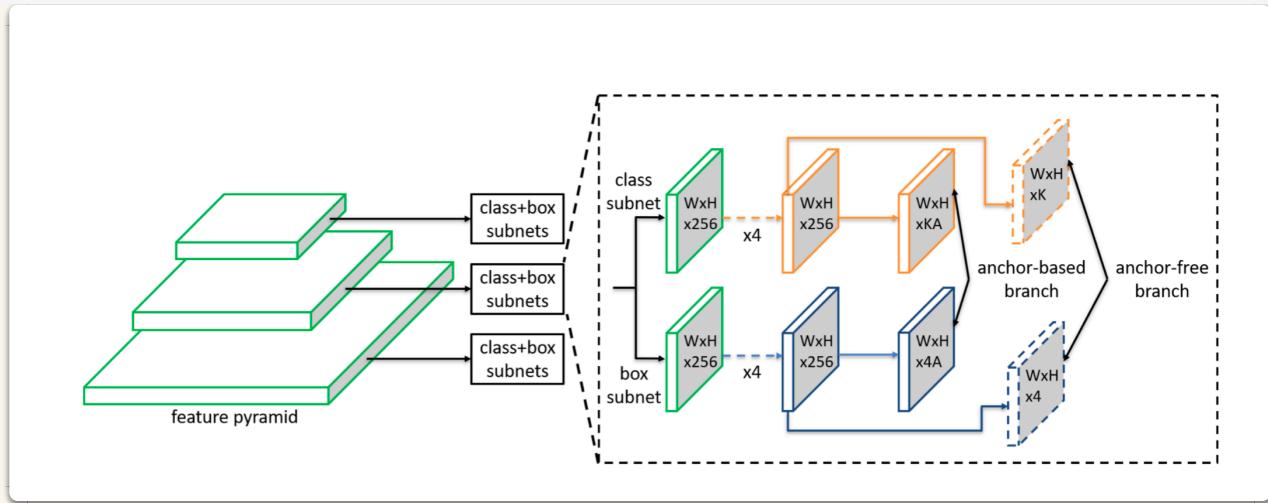
Table 1. COCO detection: MobileNet SSD vs MobileNet PPN

疑问：

1. shared tower怎么操作不同尺寸的feature map

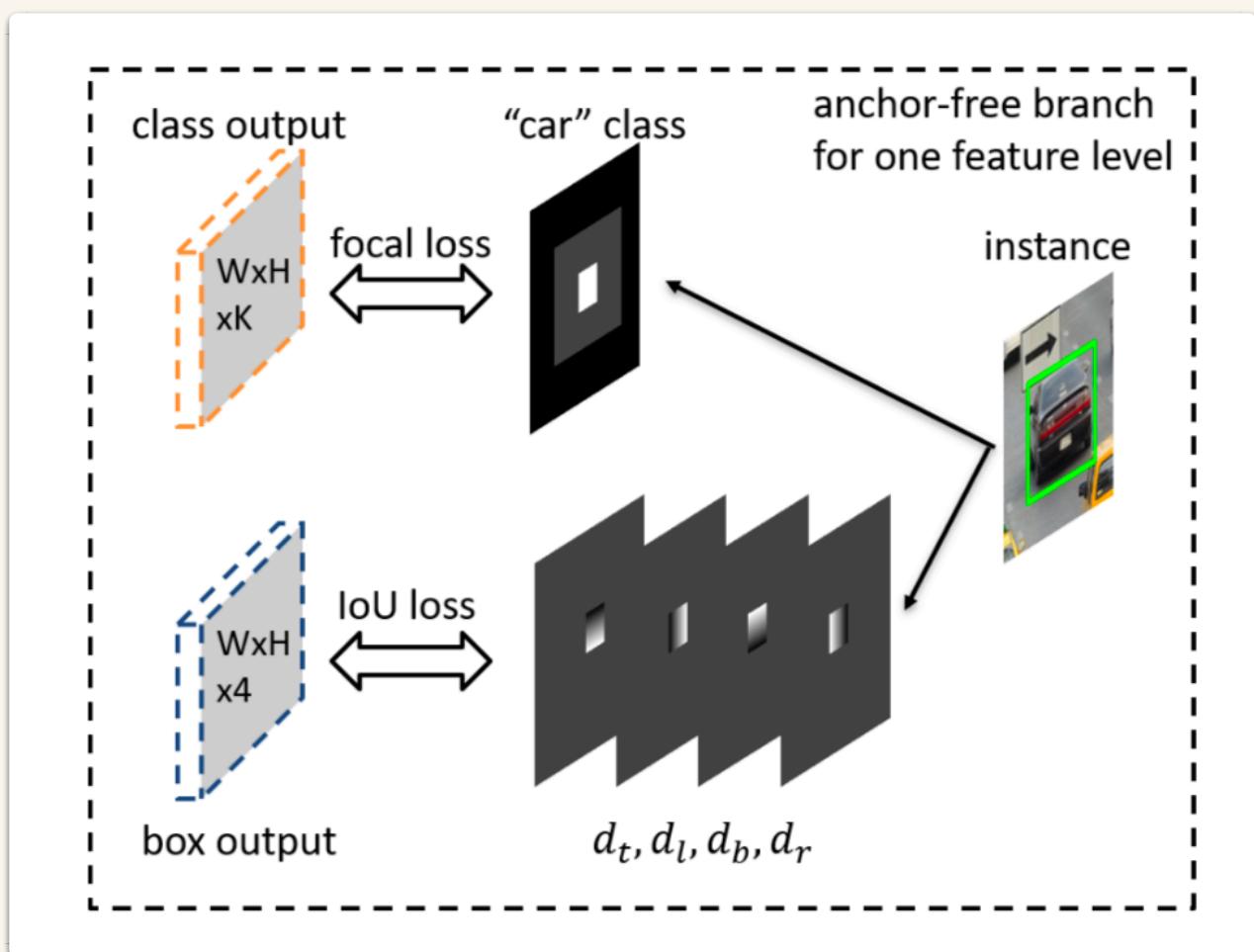
8. Feature Selective Anchor-Free Module for Single-Shot Object Detection

模型结构：



- 内含anchor-based和anchor-free模块 (A为anchor个数, K为class num)

监督训练信号 (supervision signal) :



1. ground truth box: k
2. ground truth box 坐标: $b = [x, y, w, h]$
3. ground truth box 在第 l 层上的投影: $b_p^l = [x_p^l, y_p^l, w_p^l, h_p^l]$
4. effective box: $b_e^l = [x_e^l, y_e^l, w_e^l, h_e^l]$, 他表示 b_p^l 的一部分, 缩放系数比例

$$\epsilon_e = 0.2$$

5. ignoring box: $b_i^l = [x_i^l, y_i^l, w_i^l, h_i^l]$, 他表示 b_p^l 的一部分, 缩放系数比例

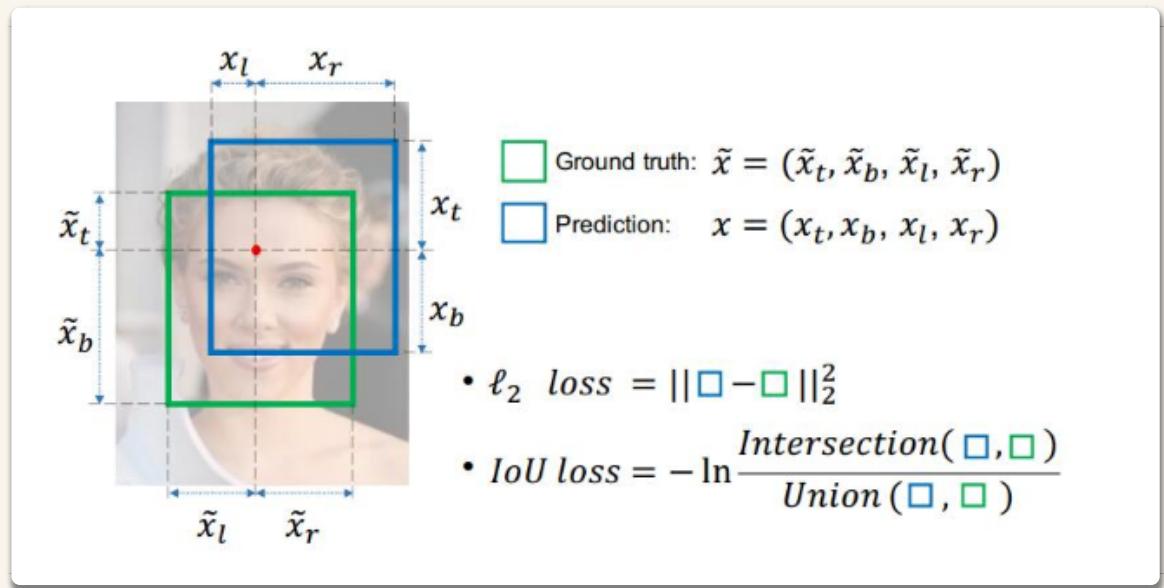
$$\epsilon_i = 0.5$$

- Classification Output

- effective box 表示positive区域, 如图白色部分所示。 $b_i^l - b_e^l$ 这个部分 ignoring 区域不参与分类任务, 如图灰色部分所示。剩余黑色部分为 negative。分类任务是对每个像素做分类, 考虑到正负样本不均衡, 采用 Focal Loss

- Box Regression Output

- 输出4个offset map。这里取的是像素 (i, j) 与 b_p^l 四个边框的距离。gt bbox 只影响了 b_e^l 区域, 所以这里 (i, j) 是该区域的所有像素。回归分支采用的是IOU Loss



- Online Feature Selection

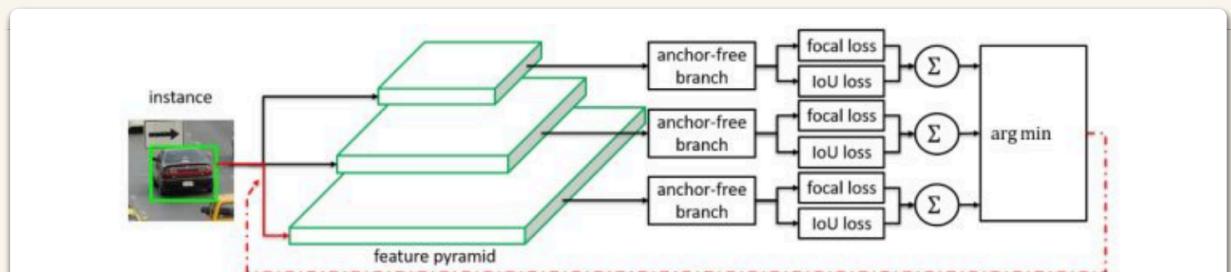


Figure 6: Online feature selection mechanism. Each instance is passing through all levels of anchor-free branches to compute the averaged classification (focal) loss and regression (IoU) loss over effective regions. Then the level with minimal summation of two losses is selected to set up the supervision signals for that instance.

- 实例输入到金字塔所有层, 求得IoU loss和focal loss的和
- 选取loss和最小的层来学习实例得到feature map

3. 训练时，特征根据安排的实例进行更新。
4. 推理时，不需要进行特征更新，因为最适合学习的金字塔层自然输出最高置信分数。