

1.综述/简介

1.DenseBox

2.YOLO

3.CornerNet

4.ExtremeNet

5.FSAF

6.FCOS

7.FoveaBox

1.综述/简介

自从去年8月CornerNet开始，Anchor-Free的目标检测模型层出不穷，最近达到了井喷的状态，宣告着目标检测迈入了Anchor-Free时代。

其实Anchor-Free并不是一个新概念了，大火的YOLO算是目标检测领域最早的Anchor-Free模型，而最近的Anchor-Free模型如FASF、FCOS、FoveaBox都能看到DenseBox的影子。

下面主要讲一下有代表性的Anchor-Free模型(包括DenseBox、YOLO、CornerNet、ExtremeNet、FSAF、FCOS、FoveaBox)，分成3个部分来介绍(早期探索、基于关键点、密集预测)，具体细节就不展开了~

早期探索

1.DenseBox

<https://zhuanlan.zhihu.com/p/24350950>

两点贡献：

1. 证明单个FCN可以检测出遮挡严重、不同尺度的目标。
2. 通过多任务引入landmark localization，能进一步提升性能。

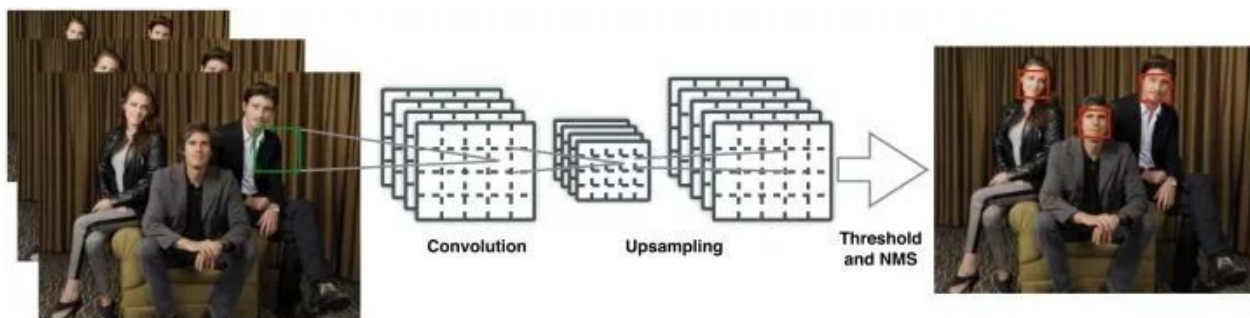


Figure 1: **The DenseBox Detection Pipeline.** 1) Image pyramid is fed to the network. 2) After several layers of convolution and pooling, upsampling feature map back and apply convolution layers to get final output. 3) Convert output feature map to bounding boxes, and apply non-maximum suppression to all bounding boxes over the threshold.

如图1所示，单个FCN同时产生多个预测bbox和置信分数的输出。测试时，整个系统将图片作为输入，输出5个通道的feature map。每个pixel的输出feature map得到5维的向量，包括一个置信分数和bbox边界到该pixel距离的4个值。最后输出feature map的每个pixel转化为带分数的bbox，然后经过NMS后处理。

Ground Truth Generation

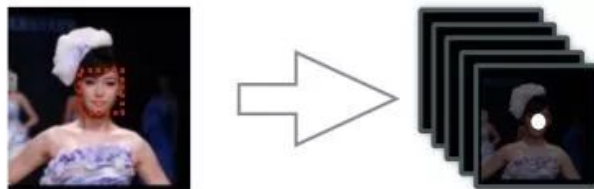
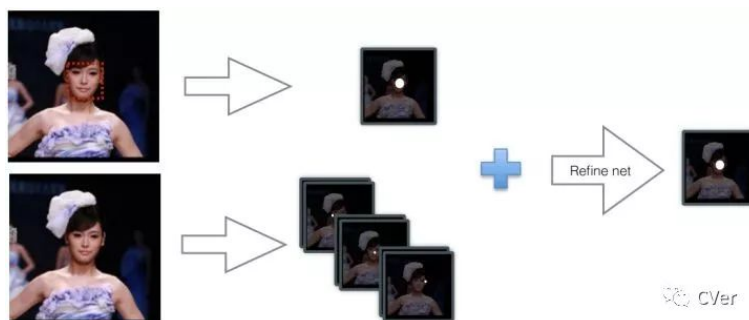


Figure 2: The Ground Truth Map in Training . The left image is the input patch, and the right one is its ground truth map.

第一个通道ground truth map的正标签区域由半径为r的圆填充，圆的中心点位于bbox的中点。而剩下的4个通道由bbox的2个角点决定。

Refine with Landmark Localization



在FCN 结构中添加少量层能够实现 landmark localization，然后通过融合 landmark heatmaps和score map可以进一步提升检测结果。

2.YOLO

YOLO将目标检测作为一个空间分离的边界框和相关的类概率的回归问题。可以直接从整张图片预测出边界框和分类分数。

三个优点：

1. 速度快
2. 通过整幅图进行推理得到预测结果
3. 能学到目标的一般特征

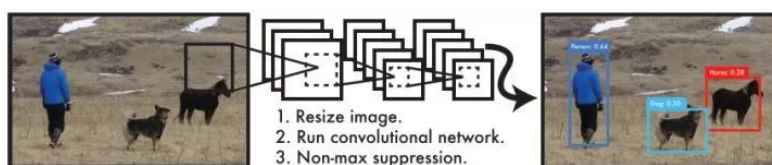


Figure 1: The YOLO Detection System. Processing images with YOLO is simple and straightforward. Our system (1) resizes the input image to 448×448 , (2) runs a single convolutional network on the image, and (3) thresholds the resulting detections by the model's confidence.

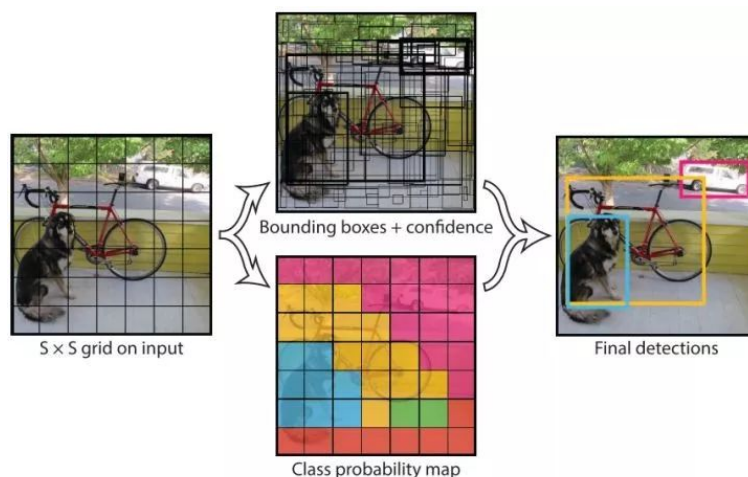


Figure 2: The Model. Our system models detection as a regression problem. It divides the image into an $S \times S$ grid and for each grid cell predicts B bounding boxes, confidence for those boxes, and C class probabilities. These predictions are encoded as an $S \times S \times (B * 5 + C)$ tensor.

YOLO将输入图片分成 $S \times S$ 个网格，如果某个目标的中心点落到其中一个格点，那么该格点就负责该目标的检测。每个格点预测出 B 个bbox和每个bbox的置信分数。

定义置信度为：

$$\text{Pr}(\text{Object}) * \text{IOU}_{\text{pred}}^{\text{truth}}$$

如果有目标落中心在格子里 $\text{Pr}(\text{Object})=1$ ；否则 $\text{Pr}(\text{Object})=0$ 。第二项是预测的 bounding box和实际的ground truth之间的IOU。所以，每个bounding box都包含了5个预测量： $(x, y, w, h, \text{confidence})$ ，其中 (x, y) 代表预测box相对于格子的中心， (w, h) 为预测box相对于图片的width和height比例，confidence就是上述置信度。

每个格点也预测 C 个类概率

$$\text{Pr}(\text{Class}_i | \text{Object})$$

测试的时候，将类概率和置信分数相乘，得到类置信分数

$$\text{Pr}(\text{Class}_i | \text{Object}) * \text{Pr}(\text{Object}) * \text{IOU}_{\text{pred}}^{\text{truth}} = \text{Pr}(\text{Class}_i) * \text{IOU}_{\text{pred}}^{\text{truth}}$$

举个例子，在Pascal VOC数据集上评估YOLO，使用 $S=7$ ， $B=2$ ，VOC有20类，所以 $C=20$ ，那么最终的预测结果是 $7 \times 7 \times 30$ 的向量。

DenseBox和YOLO的区别：

1. DenseBox应用于人脸检测，相当于只有两类，而YOLO是通用检测，通常大于两类。

2. DenseBox是密集预测，对每个pixel进行预测，而YOLO先将图片进行网格化，对每个grid cell进行预测。

3. DenseBox的gt通过bbox中心圆形区域确定的，而YOLO的gt由bbox中心点落入的grid cell确定的。

基于关键点

3.CornerNet

两点贡献：

1. 通过检测bbox的一对角点来检测出目标。
2. 提出corner pooling，来更好的定位bbox的角点。

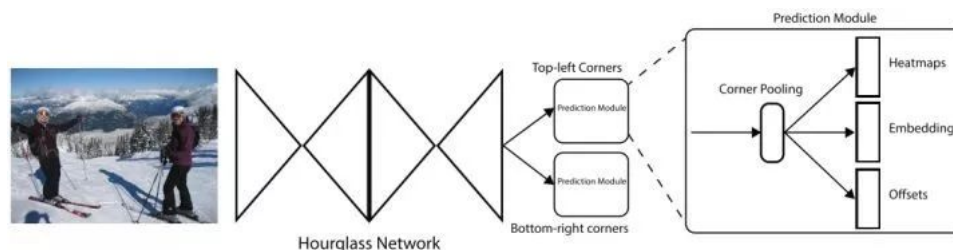


Fig. 4. Overview of CornerNet. The backbone network is followed by two prediction modules, one for the top-left corners and the other for the bottom-right corners. Using the predictions from both modules, we locate and group the corners.

对于每个角点来说，只有一个gt正例位置，其他都为负例位置。训练时，以正例位置为圆心，设置半径为r的范围内，减少负例位置的惩罚(采用二维高斯的形式)，如上图所示。

Grouping Corners

受到多人姿态估计论文的启发，基于角点embedding之间的距离来对角点进行分组。

Corner Pooling

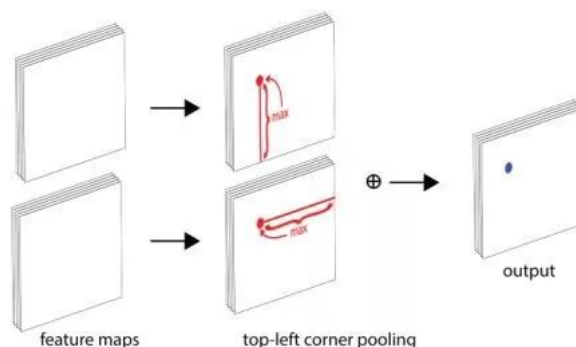


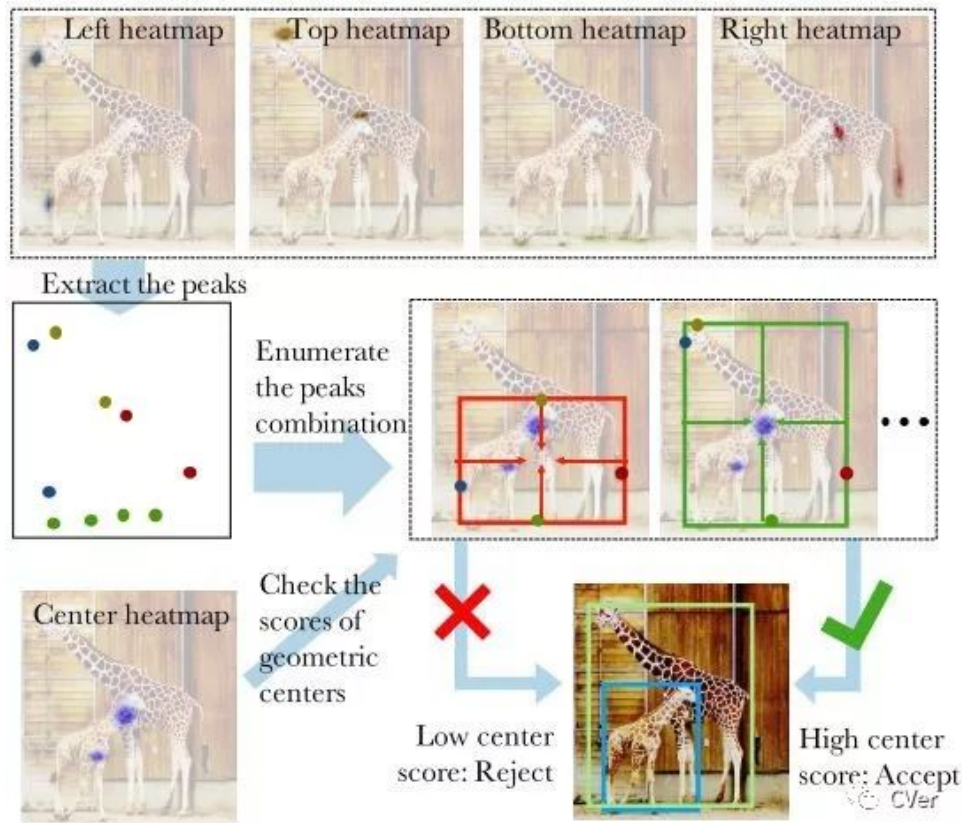
Fig. 3. Corner pooling: for each channel, we take the maximum values (red dots) in two directions (red lines), each from a separate feature map, and add the two maximums together (blue dot).

在每个pixel位置，最大池化第一个feature map右方的所有特征向量，最大池第二个feature map下方的所有特征向量，然后将两个池化结果相加。

4.ExtremeNet

两个贡献：

1. 将关键点定义为极值点。
2. 根据几何结构对关键点进行分组。



作者使用了最佳的关键点估计框架，通过对每个目标类预测4个多峰值的heatmaps来寻找极值点。另外，作者使用每个类center heatmap来预测目标中心。仅通过基于几何的方法对极值点分组，如果4个极值点的几何中点在center map上对应的分数高于阈值，则这4个极值点分为一组。

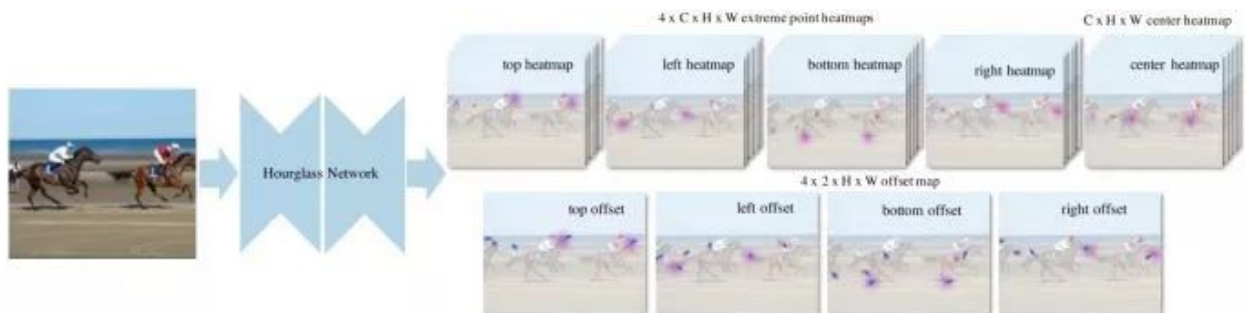


Figure 3: Illustration of our framework. Our network takes an image as input and produces four C -channel heatmaps, one C -channel heatmap, and four 2-channel category-agnostic offset map. The heatmaps are trained by weighted pixel-wise logistic regression, where the weight is used to reduce false-positive penalty near the ground truth location. And the C -channel heatmap is trained with Smooth L1 loss applied at ground truth peak locations.


offset的预测是类别无关的，而极值点的预测是类别相关的。center map没有offset预测。网络的输出是 $5 \times C$ heatmaps和 4×2 offset maps， C 是类别数。

Algorithm 1: Center Grouping

Input : Center and Extrempoint heatmaps of an image for one category: $\hat{Y}^{(c)}, \hat{Y}^{(t)}, \hat{Y}^{(l)}, \hat{Y}^{(b)}, \hat{Y}^{(r)} \in (0, 1)^{H \times W}$
Center and peak selection thresholds: τ_c and τ_p

Output: Bounding box with score

// Convert heatmaps into coordinates of keypoints.
// $\mathcal{T}, \mathcal{L}, \mathcal{B}, \mathcal{R}$ are sets of points.
 $\mathcal{T} \leftarrow \text{ExtractPeak}(\hat{Y}^{(t)}, \tau_p)$
 $\mathcal{L} \leftarrow \text{ExtractPeak}(\hat{Y}^{(l)}, \tau_p)$
 $\mathcal{B} \leftarrow \text{ExtractPeak}(\hat{Y}^{(b)}, \tau_p)$
 $\mathcal{R} \leftarrow \text{ExtractPeak}(\hat{Y}^{(r)}, \tau_p)$
for $t \in \mathcal{T}, l \in \mathcal{L}, b \in \mathcal{B}, r \in \mathcal{R}$ **do**
 // If the bounding box is valid
 if $t_y \leq l_y, r_y \leq b_y$ **and** $l_x \leq t_x, b_x \leq r_x$ **then**
 // compute geometry center
 $c_x \leftarrow (l_x + r_x)/2$
 $c_y \leftarrow (t_y + b_y)/2$
 // If the center is detected
 if $\hat{Y}_{c_x, c_y}^{(c)} \geq \tau_c$ **then**
 Add Bounding box (l_x, t_y, r_x, b_y) with score
 $(\hat{Y}_{t_x, t_y}^{(t)} + \hat{Y}_{l_x, l_y}^{(l)} + \hat{Y}_{b_x, b_y}^{(b)} + \hat{Y}_{r_x, r_y}^{(r)} + \hat{Y}_{c_x, c_y}^{(c)})/5.$
 end
 end
end

 CVPR

分组算法的输入是每个类的5个heatmaps，一个center heatmap和4个extreme heatmaps，通过检测所有的峰值来提取出5个heatmaps的关键点。给出4个极值点，计算几何中心，如果几何中心在center map上对应高响应，那么这4个极值点为有效检测。作者使用暴力枚举的方式来得到所有有效的4个关键点。

CornerNet和ExtremeNet的区别：

1. CornerNet通过预测角点来检测目标的，而ExtremeNet通过预测极值点和中心点来检测目标的。
2. CornerNet通过角点embedding之间的距离来判断是否为同一组关键点，而ExtremeNet通过暴力枚举极值点、经过中心点判断4个极值点是否为一组。

密集预测

5.FSAF

Motivation

让每个实例选择最好的特征层来优化网络，因此不需要anchor来限制特征的选择。

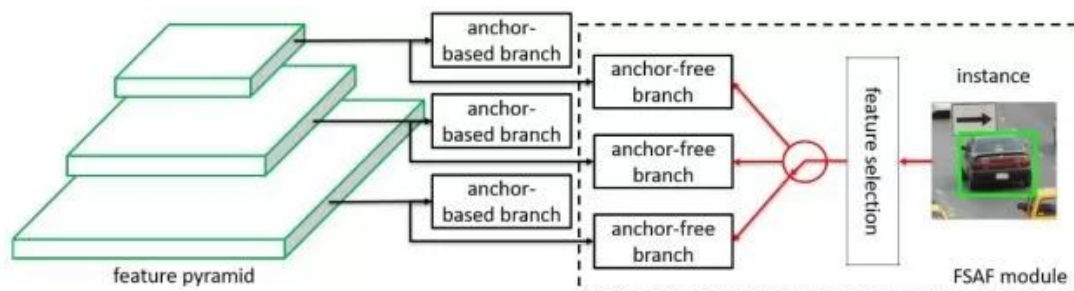


Figure 3: Overview of our FSAF module plugged into conventional anchor-based detection methods. During training, each instance is assigned to a pyramid level via feature selection for setting up supervision signals.

一个anchor-free的分支在每个特征金字塔层构建，独立于anchor-based的分支。和anchor-based分支相似，anchor-free分支由分类子网络和回归子网络。一个实例能够被安排到任意层的anchor-free分支。训练期间，基于实例的信息而不是实例box的尺寸来动态地为每个实例选择最合适的特征层。选择的特征层学会检测安排的实例。推理阶段，FSAF模块和anchor-based分支独立或者联合运行。

Feature Selective Anchor-Free Module

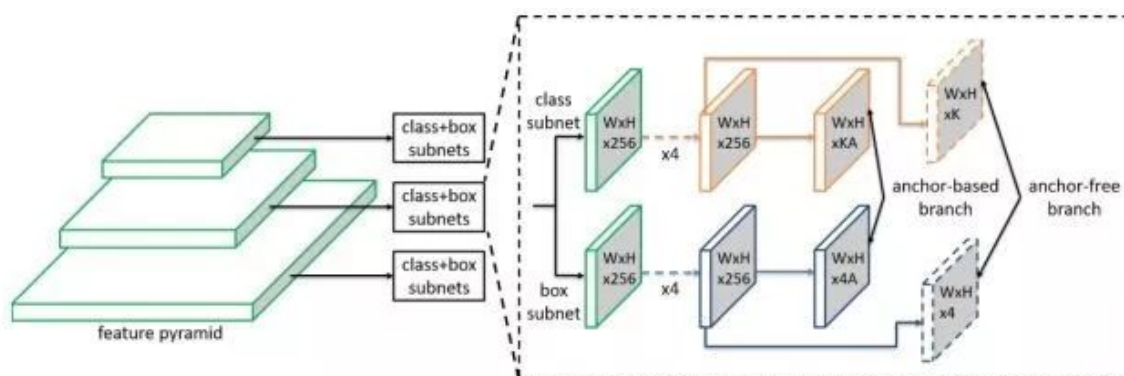


Figure 4: Network architecture of RetinaNet with our FSAF module. The FSAF module only introduces two additional conv layers (dashed feature maps) per pyramid level, keeping the architecture fully convolutional.

在RetinaNet的基础上，FSAF模块引入了2个额外的卷积层，这两个卷积层各自负责anchor-free分支的分类和回归预测。具体的，在分类子网络中，feature map后面跟着K个3x3的卷积层和sigmoid，在回归子网络中，feature map后面跟着4个3x3的卷积层和ReLU。

Ground-truth

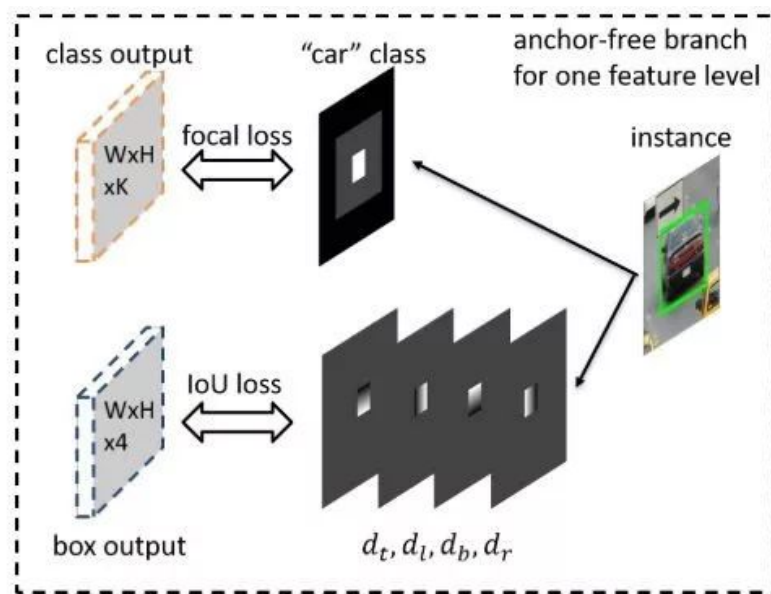


Figure 5: Supervision signals for an instance in one feature level of the anchor-free branches. We use focal loss for classification and IoU loss for box regression.

白色为有效区域，灰色为忽略区域，黑色为负样本区域。

Online Feature Selection

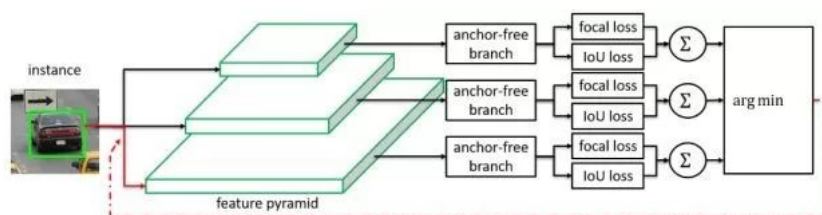


Figure 6: Online feature selection mechanism. Each instance is passing through all levels of anchor-free branches to compute the averaged classification (focal) loss and regression (IoU) loss over effective regions. Then the level with minimal summation of two losses is selected to set up the supervision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

实例输入到特征金字塔的所有层，然后求得所有anchor-free分支focal loss和IoU loss的和，选择loss和最小的特征层来学习实例。训练时，特征根据安排的实例进行更新。推理时，不需要进行特征更新，因为最合适的特征金字塔层自然地输出高置信分数。

6.FCOS

逐像素回归

四个优点：

1. 将检测和其他使用FCN的任务统一起来，容易重用这些任务的思想。
2. proposal free和anchor free，减少了超参的设计。
3. 不使用trick，达到了单阶段检测的最佳性能。

4. 经过小的修改，可以立即拓展到其他视觉任务上。

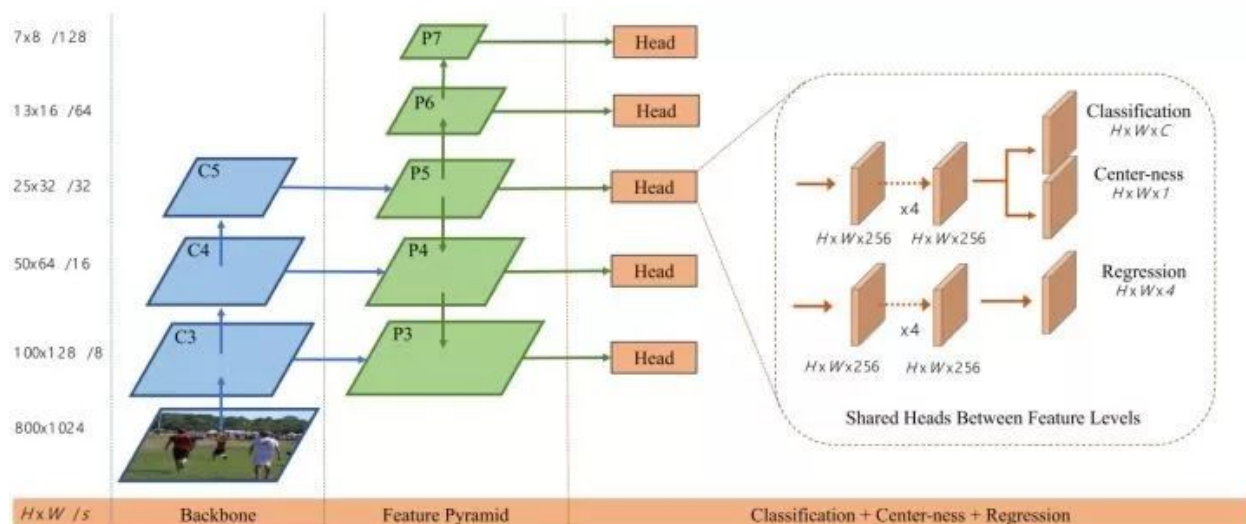


Figure 2 – The network architecture of FCOS, where C3, C4, and C5 denote the feature maps of the backbone network and P3 to P7 are the feature levels used for the final prediction. $H \times W$ is the height and width of feature maps. ‘/s’ ($s = 8, 16, \dots, 128$) is the down-sampling ratio of the level of feature maps to the input image. As an example, all the numbers are computed with a 800×1024 input.

和语义分割相同，检测器直接将位置作为训练样本而不是anchor。具体的，如果某个位置落入了任何gt中，那么该位置就被认为是正样本，并且类别为该gt的类别。基于anchor的检测器，根据不同尺寸安排anchor到不同的特征层，而FCOS直接限制边界框回归的范围(即每个feature map负责一定尺度的回归框)。

Center-ness

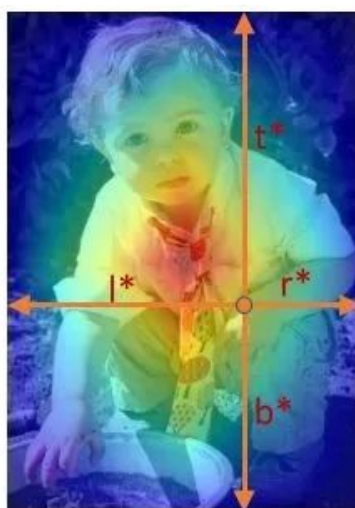


Figure 3 – Center-ness. Red, blue, and other colors denote 1, 0 and the values between them, respectively. Center-ness is computed by Eq. (3) and decays from 1 to 0 as the location deviates from the center of the object. When testing, the center-ness predicted by the network is multiplied with the classification score thus can down-weight the low-quality bounding boxes predicted by a location far from the center of an object.

为了剔除远离目标中心的低质量预测bbox，作者提出了添加center-ness分支，和分类分支并行。

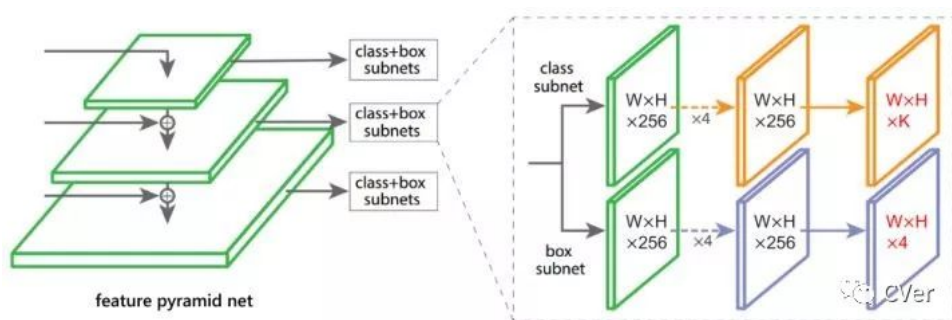
$$\text{centerness}^* = \sqrt{\frac{\min(l^*, r^*)}{\max(l^*, r^*)} \times \frac{\min(t^*, b^*)}{\max(t^*, b^*)}}$$

开根号使center-ness衰退缓慢。center-ness范围为0-1之间，通过BCE训练。测试时，最终分数由center-ness预测结果和分类分数乘积得到。

7.FoveaBox

Motivation

人类眼睛的中央凹：视野(物体)的中心具有最高的视觉敏锐度。FoveaBox联合预测对象中心区域可能存在的位置以及每个有效位置的边界框。由于特征金字塔的特征表示，不同尺度的目标可以从多个特征层中检测到。



FoveaBox添加了2个子网络，一个子网络预测分类，另一个子网络预测bbox。

Object Fovea

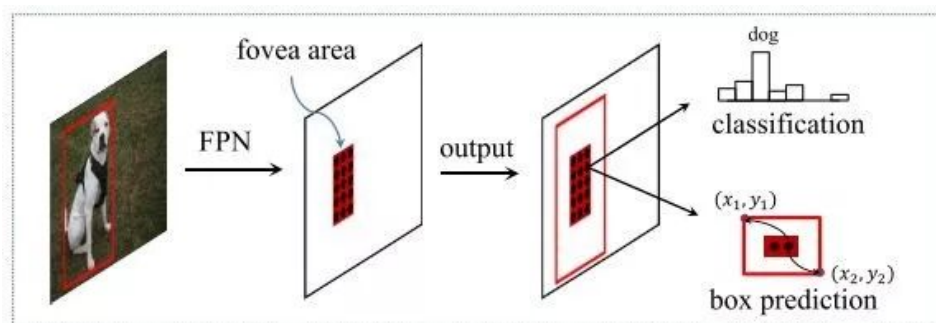


Figure 3. FoveaBox object detector. For each output spacial position that potentially presents an object, FoveaBox directly predicts the confidences for all target categories and the bounding box.

目标的中央凹如上图所示。目标中央凹只编码目标对象存在的概率。为了确定位置，模型要预测每个潜在实例的边界框。

FSAF、FCOS、FoveaBox的异同点：

1. 都利用FPN来进行多尺度目标检测。
2. 都将分类和回归解耦成2个子网络来处理。
3. 都是通过密集预测进行分类和回归的。
4. FSAF和FCOS的回归预测的是到4个边界的距离，而FoveaBox的回归预测的是一个坐标转换。
5. FSAF通过在线特征选择的方式，选择更加合适的特征来提升性能，FCOS通过center-ness分支剔除掉低质量bbox来提升性能，FoveaBox通过只预测目标中心区域来提升性能。

(DenseBox、YOLO) 和 (FSAF、FCOS、FoveaBox) 的异同点：

1. 都是通过密集预测进行分类和回归的。
2. (FSAF、FCOS、FoveaBox) 利用FPN进行多尺度目标检测，而 (DenseBox、YOLO) 只有单尺度目标检测。
3. (FSAF、FCOS、FoveaBox) 将分类和回归解耦成2个子网络来得到，而 (DenseBox、YOLO) 分类和定位统一得到。

总结：

1. 各种方法的关键在于gt如何定义
2. 主要是基于关键点检测的方法和密集预测的方法来做Anchor-Free
3. 本质上是将基于anchor转换成了基于point/region