# Anchor-free的目标检测方法调研及论文阅读笔记

刘继垚

西安电子科技大学，智能科学与技术系

18200100176，人工智能学院

《计算智能导论》读书报告

## 摘要

本文梳理了Anchor free目标检测方法的发展脉络，首先考虑早期探索性的DenseBox和YOLO等算法，之后总结了从2018年以来比较经典的Anchor free方法。我们将Anchor Free方法大致分为CornerNet、CenterNet、ExtremeNet这一类基于Keypoint的算法和和FCOS、FoveaBox这一类密集预测的Anchor Free方法，对比总结并分析了不同算法的特点。

关键词:Anchor free，目标检测

## Abstract

In this paper, the development of Anchor Free object detection method is summarized. The early exploratory algorithms such as Densebox and YOLO are considered at first, and then the classic Anchor Free method since 2018 is summarized.We divided the Anchor Free method into two categories, one is based on Keypoint, such as CornerNet, CenterNet and ExtremeNet, another is based on intensive prediction, such as FCOS and Foveabox. Meanwhile, we compared and summarized the characteristics of these different algorithms.

Key words：Anchor Free, Object detection

# 1.引言

在深度学习时代，物体检测通常都被建模成对一些候选区域进行分类和回归的问题。Anchor的本质是候选框，在设计了不同尺度和比例的候选框后，DNN学习入了将这些候选框进行分类和回归，即：是否包含object和包含什么类别的目标物体，对于positive的anchor会学习将其回归到正确的位置。在One-stage检测器中，这些候选区域都是通过滑窗的方式产生anchor。

但是基于Anchor的目标检测设计思路可能存在一些问题：

- 正负样本不均衡：一般在特征图上所有点均匀采样anchor，但是大部分地方都是没有物体的背景区域，导致负样本数量众多，导致预测偏向负类，降低模型泛化能力。
- 对于feature map上的每一个像素点去定义很多anchor boxes，导致其anchor boxes数量大。
- Anchor boxes的使用引入和很多的超参数和设计选择，包括多少个box、大小和高宽比，增大了网络调优的难度和计算量。
- anchor的设置具有主观性，使得极端尺度的物体难以被检测。

基于上述原因，很多人做出了改进，提出了Anchor-free方法。本文研究了几种主流的Anchor 方法，大致分为两类：

a. 直接预测边框：根据网络特征直接预测物体出现的边框，即上、下、左、右四个值，如： YOLOv1、利用分割思想解决检测的FCOS以及改善了边框回归方式的Foveabox算法。

b. 关键点的思想：使用边框的角点或者中心点进行物体检测，这类算法通常是受人体姿态的关键点估计启发，典型有DenseBox、 CornerNet、ExtremeNet 及 CenterNet 等。

# 2.DenseBox(2015)[1]

DenseBox是一个不需要产生候选框且可以进行端到端训练的基于FCN的检测网络，主要工作如下：

- 设计了一个基于FCN的目标检测网络，能够检测不同尺度下重遮挡的物体，非常精确、有效。
- 通过多任务学习融入landmark（关键点），极大地提高了检测的准确率。

## 2.1 网络结构

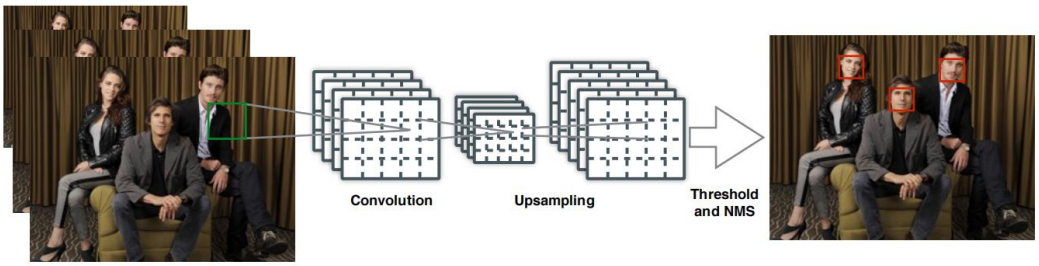



图1 DenseBox 网络结构

网络的整体流程如图1，单一的卷积网络同时输出多个不同的预测框和类别置信度，最后进行非极大抑制NMS过滤+阈值过滤，得到最终检测框。

## 2.2 Ground truth generation（人工标注生成）

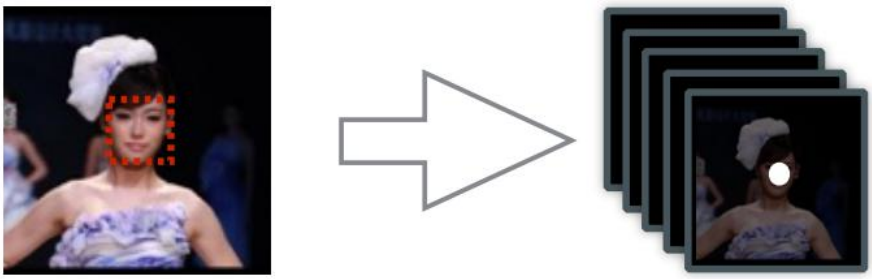


图2 The Ground Truth Map in Training

样本的标签并没有用IOU阈值来判定是正样本这种方式，作者认为把整张图放进网络意义不大，因为背景太多，而且增大了计算量。因此将图片crop，让人脸处于image的正中心，并且有足够的背景信息。作者构建的GT map是60*60*5的5维map，和网络输出的feature map类似。在 60 * 60 * 5的GT m

ap的第一层上，位于中心点附近rc半径内的都是正样本，rc为bounding box大小的0.3倍，即圆内的像素标签都标注为1，后面的四个channel为四个坐标，距离bounding box的左上和右下坐标的距离。

## 3. Yolov1(2015)[2]

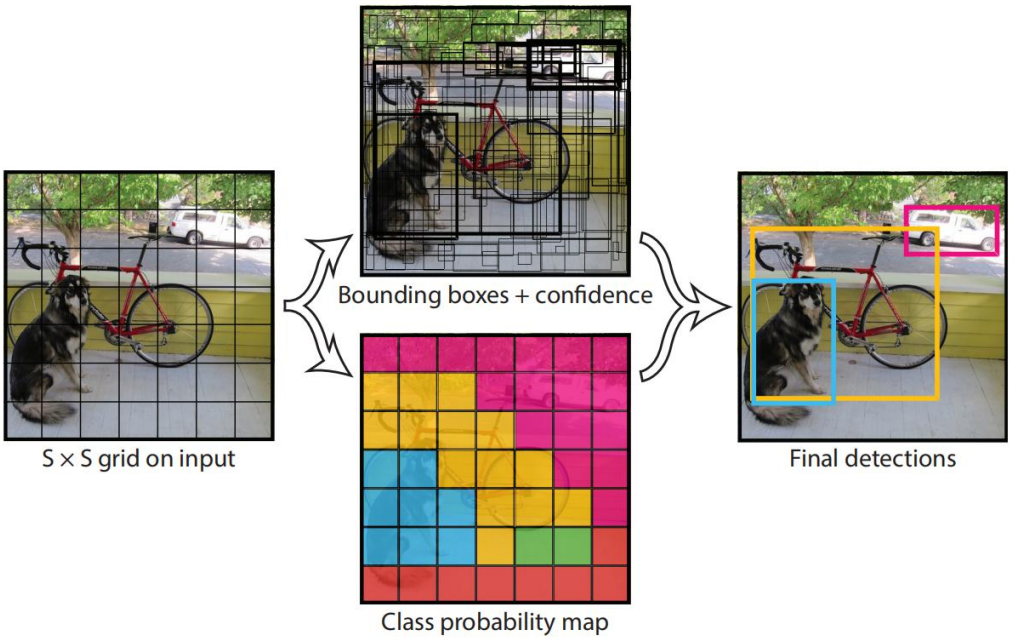



图3 YOLO模型

### 3.1 模型介绍

YOLO是一个单CNN模型的同时可以预测多个bounding box以及其分类概率，直接对全图进行训练。首先，YOLO会把输入图片分为SxS个网格，如果目标的中心落在了网格内，那么这个网格负责检测这个目标。接着我们定义其置信度

$$confidence = Pr(Object) * IOU_{pred}^{truth}$$

，因此每个bounding box包括了5个预测值：x，y，w，h和confidece。YOLO算法把检测当做一个回归问题来解决，加入把图片分成SxS个网格且每个网格预测B个bounding box以及对应的confidence以及最终的类别

预测C。那么最后一张图片的预测结果是一个 $S \times S \times (B*5+C)$ 的张量。

论文中B=2，S=7，C=20.

### 3.2 YOLO缺点分析

作者在论文中也谈到了YOLO的缺点，最主要的缺点就是识别小物体，如小鸟。以及边框的位置偏差很大，当识别物体数量很多时也不可以。

其主要原因为只使用最后一层作为预测，且只有49个网格进行预测，而每个网络的感受野又很大，所以导致小物体检测不到。

## 4. CornerNet（2018）[3]

抛弃Anchor之后，一个比较核心的问题是怎么描述Box，这方面CornerNet提出了一个比较有意思的思路，就是将Box转化为关键点描述，即用box左上角和右下角两个点去确定一个box，将问题转化为对top-left和bottom-right两个点的检测与匹配问题。

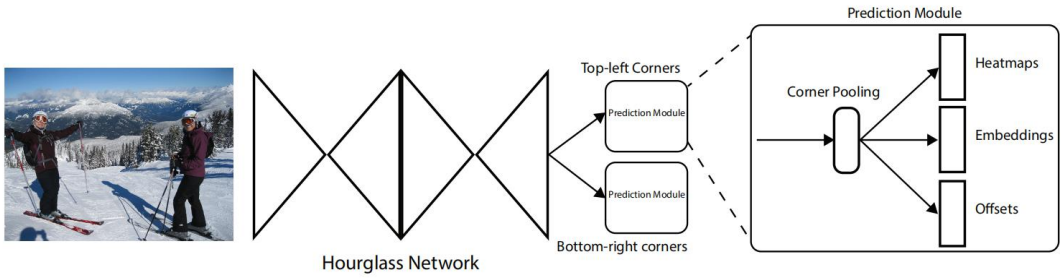



图4 CornterNet模型

1.基础网络也就是Backbone使用的是Hourglass网络，这也是从人体姿态估计处借用来的灵感。这是一个在keypoint检测中非常流行的骨干网络。

2. 关于输出，网络有两个分支，一个分支预测Top-left Corners，另一个预测Bottom-right corners 每个分支又有三个子分支,heatmaps输出的是feature map上每个点可能是Corners点的概率、预测哪些点最有可能是Corners点；embeddings输出的是每个位置属于某个object的概率，例如如果fea

ture map上两个点分别是0.9和1.1，它们可能都属于object1，如果是1.7和2.2，可能都属于object2，这个分支主要用于分别在heatmaps上提去出了可能的top-left和bottom-right对它们进行匹配;最后的offsets用于对点的位置进行修正，因为由于下采样率的存在，heatmaps上的(x,y)映射回原图都是(kx,ky)，k是下采样率，这就有一定的偏差，这个分支预测的是映射回(kx,ky)后，还要做怎么样的偏移，从而让位置更精准。

我们可以发现，实际上heatmaps和offset解决的是keypoint点在哪里的问题，这两个子分支在后面的两个网络中也有用到；而embeddings分支解决的是找到点后，怎么把属于同一个Object的点从候选里找出来的问题。

## 5. CenterNet（2019）[4]

CenterNet的Motivation其实很简单：CornerNet只能提供边缘信息，实际上对于obj来说，最有辨识度的信息应该是它们内部的区域。过于关注边缘很容易引入大量的误检和错检。CenterNet增加了对中心点的检测，来帮助筛选候选框。

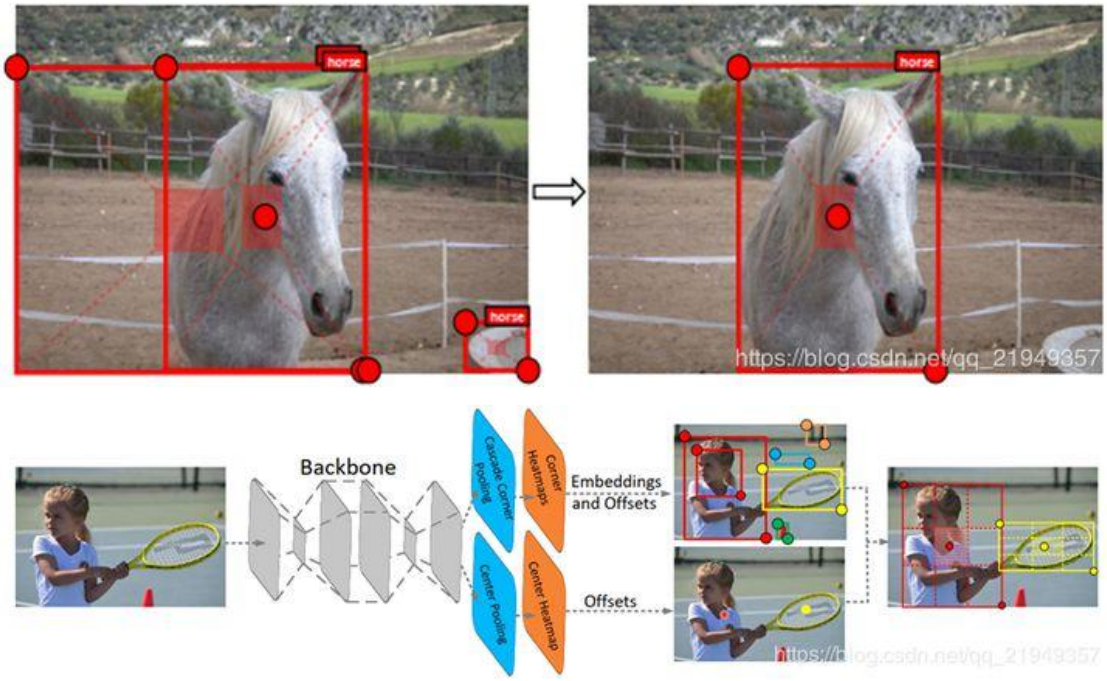



图 5 CenterNet 模型

CenterNet同样有top-left和bottom-right两条分支，而且和之前的CornerNet是相同的,它的网络的创新点在于,新增了一条没有embeddings的分支，

用于预测feature map上每个位置是中心点的概率。那么，在找出中心点以后要怎么用呢？作者的思路是找出匹配的top-left和bottom-right之后，按照图6所示，把box画成3x3个格子(如果box比较大可以划分成5x5的)，最中间的格子就被认为是中心区域。然后根据新增的预测center的分支找出来的中心点，看中心区域里面有没有中心点，如果有就检测成功，否则就失败；这一思路可以有力地筛去误检。

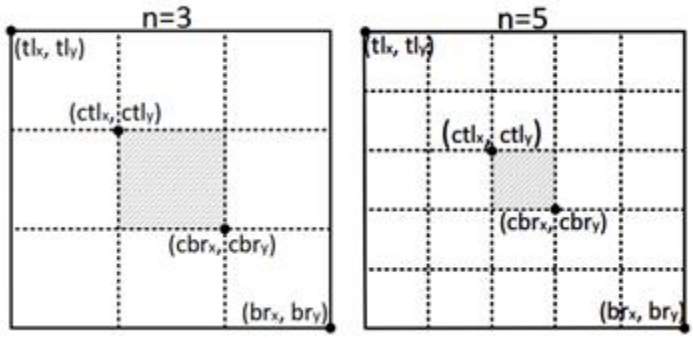



图 6 中心点匹配

## 6.ExtremeNet(2019)[5]

ExtremeNet 的 Motivation 似乎来源于一篇和标注相关的文章，它的观点是，CornerNet 的描述方式本质上还是画框，是 box 的另一种描述形式，没有改善 box 引入了大量干扰的问题；而且 Corner 点基本在 Obj 外，实际上这样检测效果是不好的，所以它采用了另一种点描述方式(top-most，bottom-most，left-most，right-most，center)，如图 7 所示，最上、最下、最左和最右的四个点提供了另一种描述 box 的方法，而补充的 center 点可以帮助判断 box 的真实性：

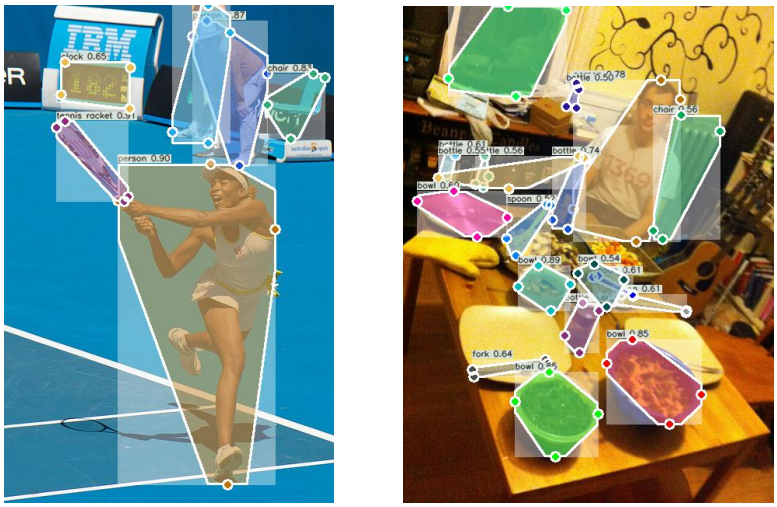



图 7 寻找关键点

剩下就没什么好说的了，检测点的思路基本和 Corner 相同，只不过 extreme net 检测的点不同，而匹配方法的话，extrmeNet 和 centernet 有点类似，但是它是没有 embedding 分支的，组合点的时候用的是穷举加 center 点辅助判断的方式，具体来说，从四张 heatmaps 选出所有候选点后，采用穷举而不是 embedding 的方法，对于任意一组点，也是计算它们的中心，然后到 center 的 heatmaps 上看中心位置的响应值是否大于阈值，具体步骤图 8 所示：

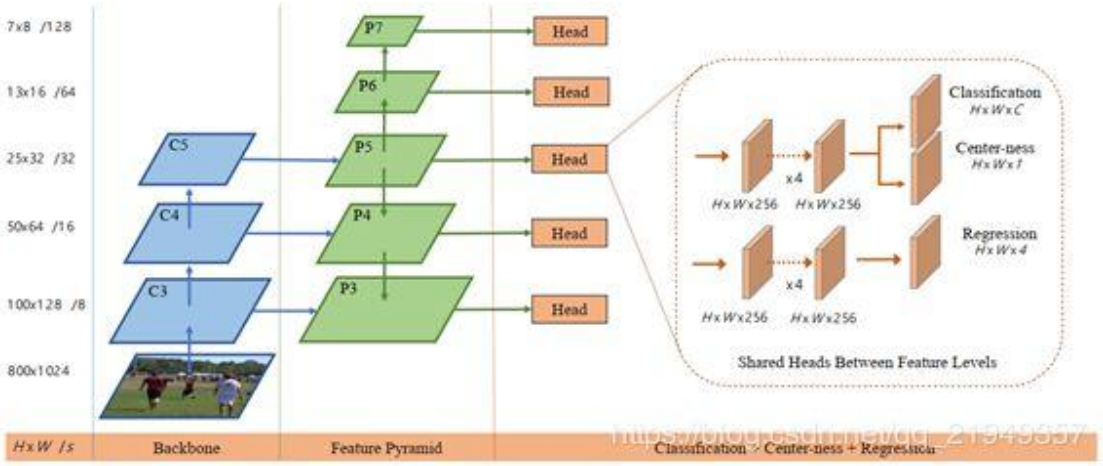
**Algorithm 1:** Center Grouping

**Input :** Center and Extremepoint 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: $\tau_c$ and $\tau_p$

**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** $\hat{Y}^{(c)}_{c_x,c_y} \geq \tau_c$ **then**
      Add Bounding box $(l_x, t_y, r_x, b_y)$ with score
      $(\hat{Y}^{(t)}_{t_x,t_y} + \hat{Y}^{(l)}_{l_x,l_y} + \hat{Y}^{(b)}_{b_x,b_y} + \hat{Y}^{(r)}_{r_x,r_y} + \hat{Y}^{(c)}_{c_x,c_y})/5$.
    **end**
  **end**
**end**

图 8 ExtremeNet 算法伪代码




## 7.FCOS（2019）[6]

图 9 FCOS 网络结构

按照 FCOS 的说法，它是把每个 location 都当做一个样本，如图 10 所示，可以看到，最左面的橙色点在棒球运动员的 box 内，这个点的 gt 实际上是该点到 box 的四个边缘的距离以及 box 的 obj 类别，所以最后预测出来的 output 是 HxWxC 以及 HxWx4，C 和 4 分别代表每个特征图的每个 location 要预测的该点所属于的类别和该点到 box 的边界距离。在通过这种方式得到 box 后，FCOS 回合 anchor based 的方法一样进行 NMS 等：

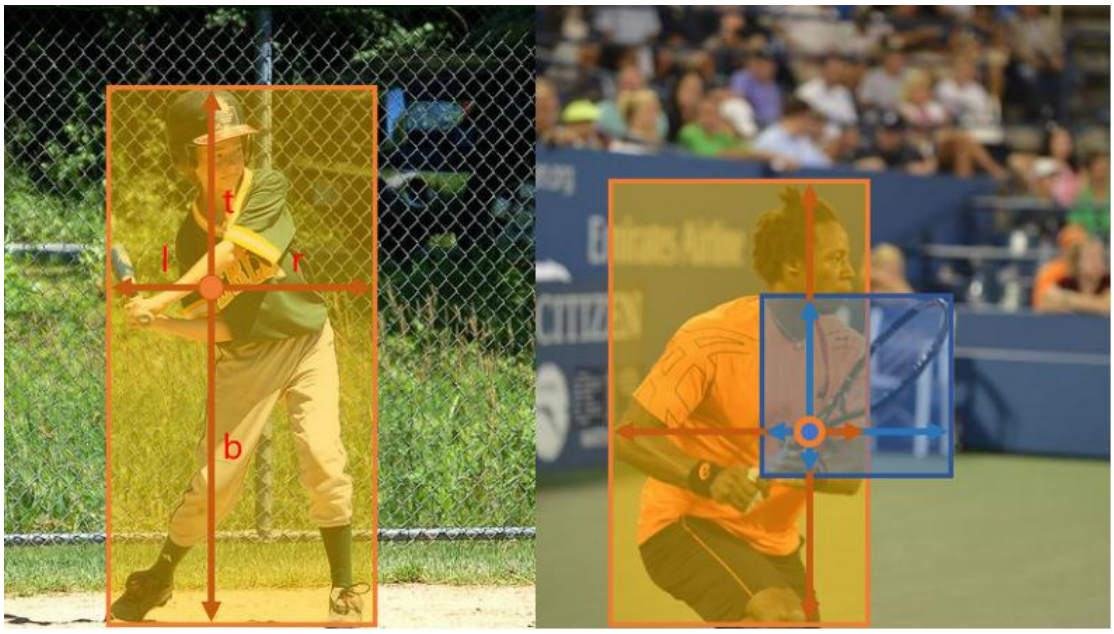


图 10 FCOS 模型示例

实际上如果不考虑 Classification 下面的 Center-ness 分支，就很像 retinanet 的网络图：

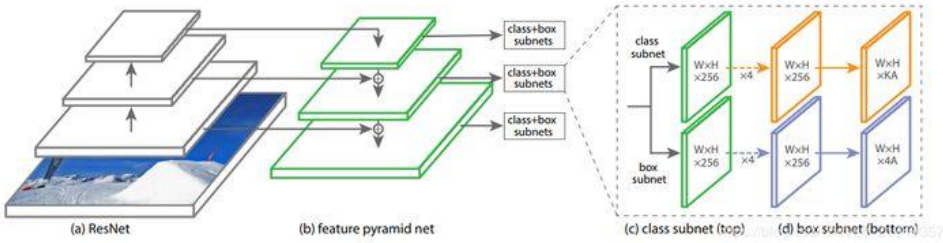



图 11 Retinanet 网络图

可以发现，两者最大的差别是最后输出的通道，Retinanet 输出的是 KA 和 4A(A 代表 anchors 数量，K 代表类别数量)，是对每个 location 位置的 A 个 anchors 预测它们的类别和相对偏移量，而 FCOS 则直接对格子所在的类别和产生 box 进行预测了，完全没有 box 的概念，整体上也非常接近语义分割的 segmentation 思想。 这样做的方法会有一个问题，就是 box 里面，越接近中心的位置往往效果越好，但是越靠近 Box 边缘，虽然理论上应该仍然是正类，但是因为往往落在 obj 外，预测效果不佳，对此，FCOS 的解决方法是引入一个新的分支 centerness，它的 gt 计算如下：

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

可以看到，如果 location 距离 box 的左边界距离和右边界距离相同，根号内第一项应该是 1，同理，当距离上下边界距离一样时候，根号内第二项是 1，此时，gt 值为 1，location 恰好处于中心位置。而如果 location 非常接近边缘，则 gt 会非常小。这个分支训练以后，在 inference 阶段将会和 cla

ssification 预测的值相乘作为最终 scores 得分，从而抑制接近中心点的位置。 此外，FCOS 还引入了多尺度的概念，如果在 FPN 的某个 level 上，t/b/l/r 中的最大值大于某个阈值，则认为这个 box 不适合当前 level 的 feature，从而进行排除。

## 8. Foveabox（2019）[7]

foveabox 的不同首先在于多尺度策略和 encoding 的方法。 foveabox 的多尺度策略是将不同大小的 box 根据面积分配到不同 level 的 feature map 上，且有重叠。FPN 的 P3~P7 的每个 leval 的 Pl 分别有一个基数 Sl，取 l=3 的时候，P3 对应的 S3 是 3232，取 l=4 的时候，P4 对应的 S4 是 6464，一直倍增。每个 level 负责的 box 的面积范围为，其中 n^2 是可变化的参数，如图 10，不同 leval 预测的范围会有重叠，这可以增加一定的鲁棒性：

ber of feature pyramidal levels. Each pyramid has a basic area ranging from $32^2$ to $512^2$ on pyramid levels $P_3$ to $P_7$, respectively. So for level $P_l$, the basic area $S_l$ is computed by

$$S_l = 4^l \cdot S_0. \quad (1)$$

Analogous to the ResNet-based Faster R-CNN system that uses C4 as the single-scale feature map, we set $S_0$ to 16 [40]. Within FoveaBox, each feature pyramid learns to be responsive to objects of particular scales. The valid scale range of the target boxes for pyramid level $l$ is computed as

$$[S_l/\eta^2, S_l \cdot \eta^2], \quad (2)$$

图 12 FoveaBox 计算公式

而考虑到不同 level 预测的 box 大小不同，预测的 box 位置坐标也是经过编码的，编码方式如下(z 代表系数，具体计算方式见论文公式)：

$$t_{x_1} = \log \frac{2^l(x+0.5) - x_1}{z},$$
$$t_{y_1} = \log \frac{2^l(y+0.5) - y_1}{z},$$
$$t_{x_2} = \log \frac{x_2 - 2^l(x+0.5)}{z},$$
$$t_{y_2} = \log \frac{y_2 - 2^l(y+0.5)}{z},$$

最后，也就是 foveabox 名称的由来，对于 box 内部离中心点比较远的抑制方法，foveabox 没有 centerness 那样的分支，而是用了另一个思路，那就是只有 box 内部比较靠近中心的点才被视作正样本(下图带黑色点的红色区域)，如果该点在 Box 内部但是离边缘比较近，则往往被视作灰色区域，即不算正样本，也不算负样本，梯度回传的时候不考虑(红色 box 内部白色区域)。正样本所在的矩形框和灰色区域的矩形框大小是由两个不同的伸缩系数控制的。

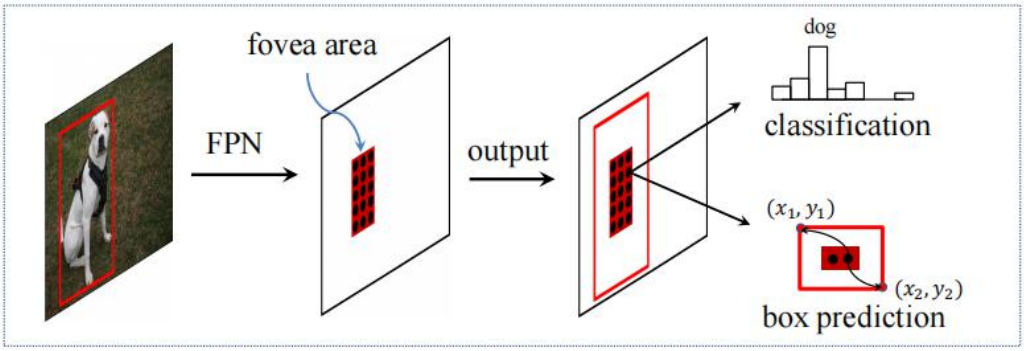



图 13 Faveabox 结构图

## 9. 总结

事实上，anchor free 的方法有很多，而且不像 anchor 方法那样整洁，本文也仅整理了一小部分内容。但无论是 anchor 还是 anchor free，检测任务基本就是这样的思路：

（1）表示：如何表示图像上的物体，如 bbox，conner，center，reppoints 等；

（2）分配：如何分配正负样本：IOU、高斯热图、centerness 等；

（3）分类：分类任务计算物体类别损失，解决样本不平衡的问题；

（4）回归：回归任务计算物体尺度、offset 等等，以进行一些修正。

Anchor free 可以看做是检测算法的进阶资料，因为 anchor 引入了先验框这种很强的假定，而 anchor free 则发散到了这套检测思路的本质，如何表示？如何分配？如何计算 loss？尽管 anchor free 的方法很杂，但都是在围绕这几个问题展开。

从 2018 年 CornerNet 问世，2019 年可以说是 anchor free 方法的井喷年，出现了非常多的 anchor free 方法，而且基本上达到了 SOTA 的水平(今年 CVPR 的 Libra R-CNN 给出的最好结果是 43%，而目前 anchor free 方法普遍在 42~43%，FSAF 联合了 anchor-free 和 anchor based 达到了 44%。 从设计来说，anchor free 方法确实更符合 Deeplearning 的思想与理念，也更方便和 keypoint 以及 segmentation 进行接轨，从而挖掘和借鉴其它的思路，应该会成为接下来的热潮。

## 10.参考文献

[1] Huang L, Yang Y, Deng Y, et al. Densebox: Unifying landmark localization with end to end object detection[J]. arXiv preprint arXiv:1509.04874, 2015.

[2] Redmon J, Divvala S, Girshick R, et al. You only look once: Unified, real-time object detection[C]//Proceedings of the IEEE conference on computer vision and pattern recognition. 2016: 779-788.

[3] Law H, Deng J. Cornernet: Detecting objects as paired keypoints[C]// Proceedings of the European conference on computer vision (ECCV). 20 18: 734-750.

[4] Zhou X, Wang D, Krähenbühl P. Objects as points[J]. arXiv preprint a rXiv:1904.07850, 2019.

[5] Zhou X, Zhuo J, Krahenbuhl P. Bottom-up object detection by groupin g extreme and center points[C]//Proceedings of the IEEE/CVF Conferenc e on Computer Vision and Pattern Recognition. 2019: 850-859.

[6] Tian Z, Shen C, Chen H, et al. Fcos: Fully convolutional one-stage o bject detection[C]//Proceedings of the IEEE/CVF International Conference on Computer Vision. 2019: 9627-9636.

[7] Kong T, Sun F, Liu H, et al. Foveabox: Beyound anchor-based object detection[J]. IEEE Transactions on Image Processing, 2020, 29: 7389-73 98.