人工智能之深度学习

目标检测(扩展)

主讲人：Vincent Ying

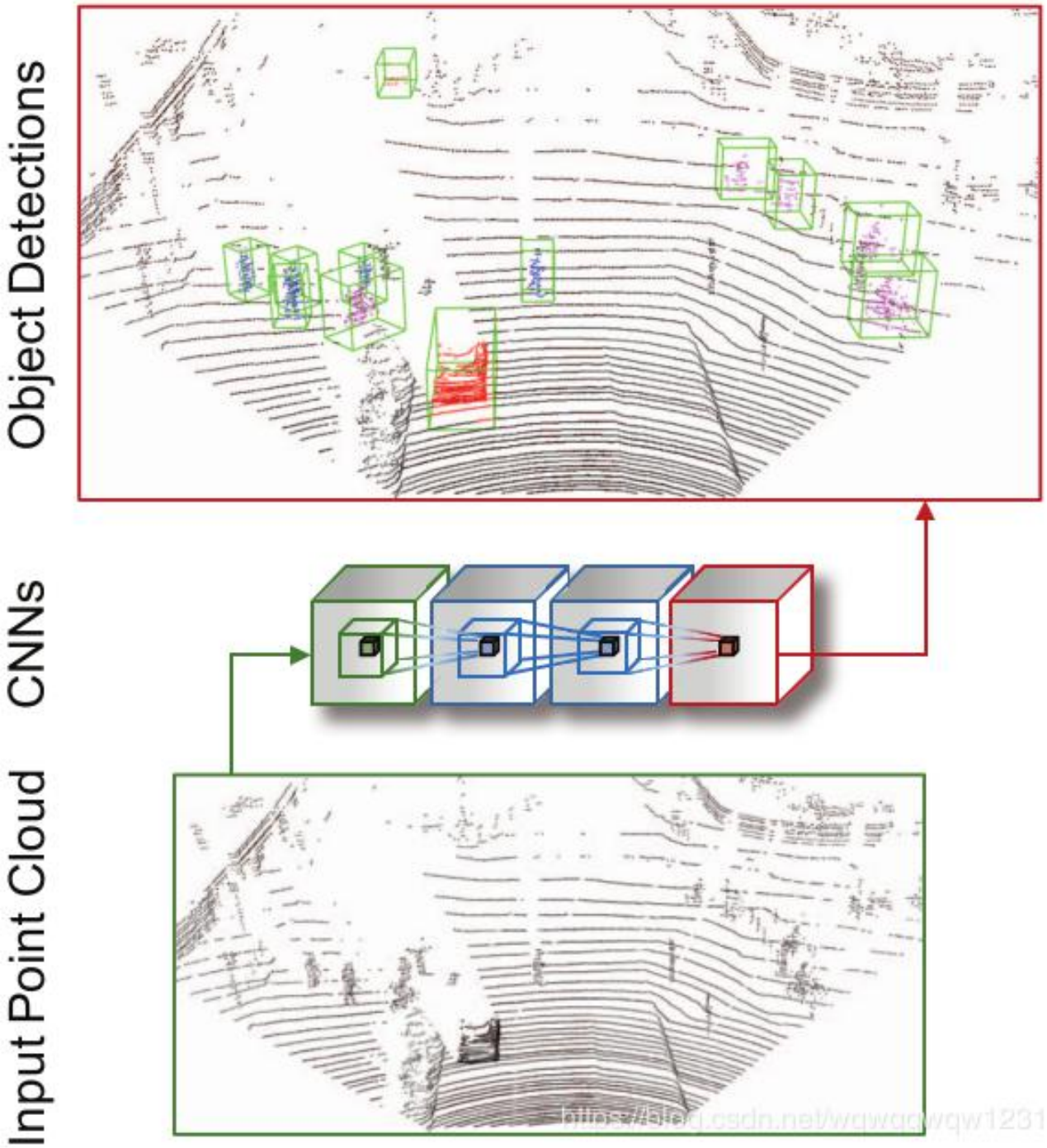
课程要求

- 课上课下“九字”真言
  - 认真听，**善摘录，勤思考**
  - **多温故，乐实践**，再发散
- 四不原则
  - 不懒散惰性，不迟到早退
  - 不请假旷课，不拖延作业
- 一点注意事项
  - 违反“四不原则”，不推荐就业

课程内容

- 3D目标检测
  - RGBD数据特征
  - 点云数据
- 基于Anchor Free的目标检测:
  - CornerNet、CenterNet、CornerNet-Lite

3D目标检测



3D目标检测

- 参考链接：
  - https://blog.csdn.net/wqwqqwqw1231/article/details/90693612
  - https://blog.csdn.net/Julialove102123/article/details/80196469
  - https://baijiahao.baidu.com/s?id=1629504685201571255&wfr=spider&for=pc
  - https://cloud.tencent.com/developer/article/1418687
  - https://cloud.tencent.com/developer/article/1478763

目标检测：Anchor Free_知识回顾

- 在SSD、YOLOv2、YOLOv3这些one-stage的算法以及Faster R-CNN算法中均采用了anchor box的思想。Anchor Box的出现，使得训练时可以预设一组不同尺度不同位置的锚框，覆盖几乎所有位置和尺度，每个锚框负责检测与其区域交叉比(IoU)大于阈值的目标边框，这样目标检测的问题就转换为”这个锚框中有没有目标，目标实际边框距离锚框有多远”这两个问题。
- NOTE: 锚框也就是固定的预选框。

目标检测：Anchor Free

- Anchor Free其实就是基于关键点的目标检测，在该方式中，不存在预选的Anchor Box，而是使用one-stage网络直接预测边框位置信息，通过关键点检测的方式可以消除现有的one-stage检测网络对于anchors的需求。
- 在Anchor Free类型的目标检测算法中，其核心思想为：**直接预测边框位置信息。**

目标检测：Anchor Free

- 属于Anchor Free类型的目标检测算法如下：
  - **YOLOv1**
  - UnitBox: An Advanced Object Detection Network
  - **CornerNet： Detecting Objects as Paired Keypoints**
  - Region Proposal by Guided Anchoring(GA-RPN)
  - FSAF(Feature Selective Anchor Free)
  - FCOS(Fully Convolutional One-Stage)
  - **CenterNet: Objects as Points**
  - **CenterNet: Keypoint Triplets for Object Detection**
  - **CornerNet-Lite: Efficient Keypoint Based Object Detection**

目标检测：Anchor Free_UnitBox

- 核心思想：**Intersection over Union (IoU) loss function for bounding box prediction(直接基于边框的IoU损失函数来构建模型)**
- 论文：https://arxiv.org/pdf/1608.01471

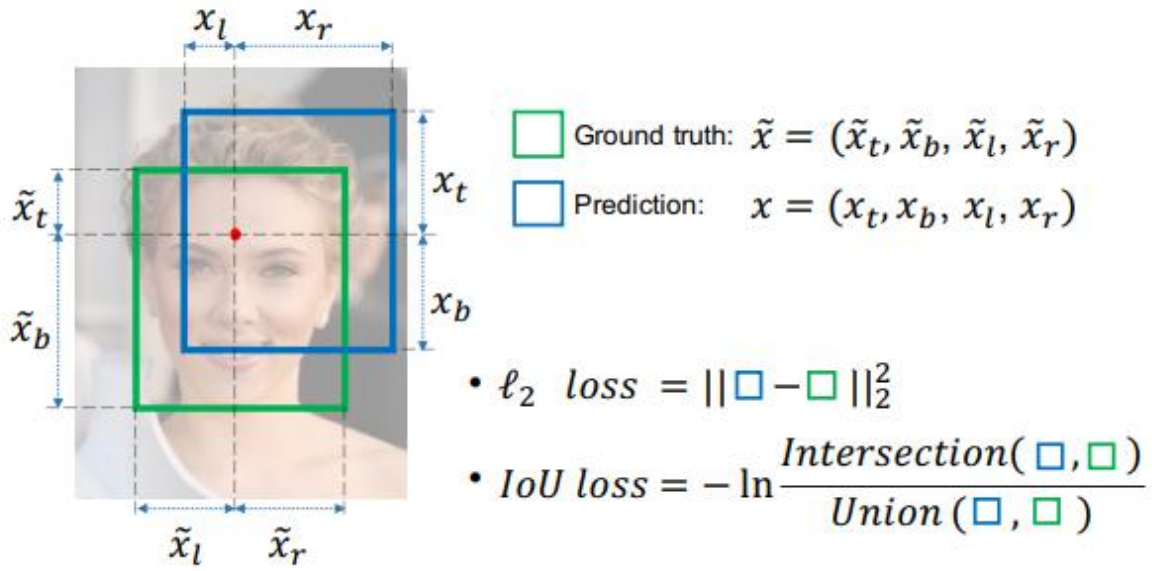


Figure 1: Illustration of IoU loss and ℓ_2 loss for pixel-wise bounding box prediction.

目标检测：Anchor Free_UnitBox

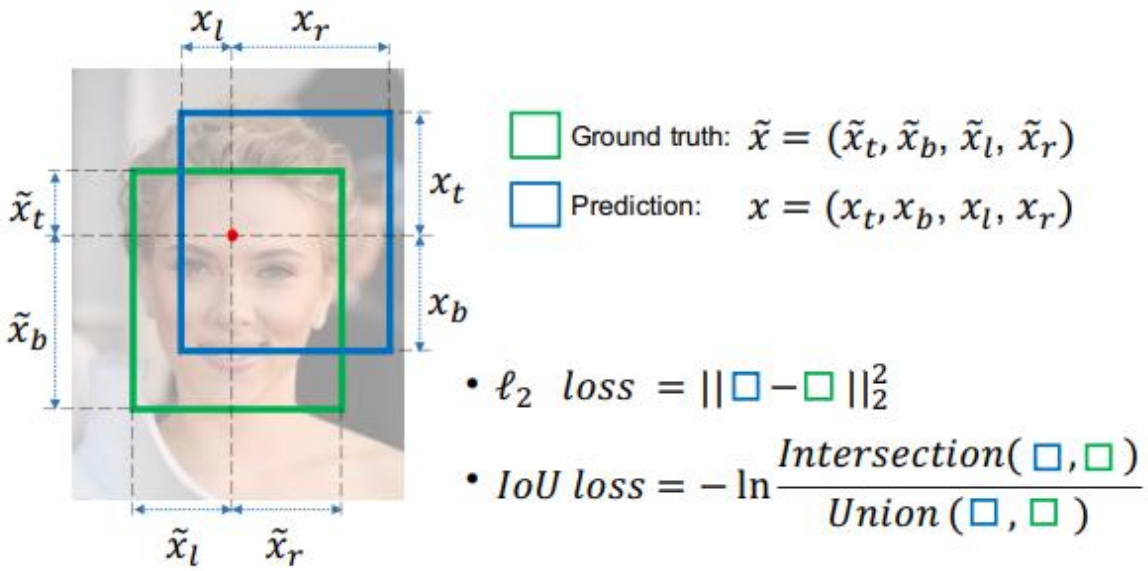


Figure 1: Illustration of IoU loss and ℓ_2 loss for pixel-wise bounding box prediction.

Algorithm 1: IoU loss Forward

Input: $\tilde{x}$ as bounding box ground truth
Input: x as bounding box prediction
Output: $\mathcal{L}$ as localization error
for *each pixel* (i, j) **do**
  if $\tilde{x} \neq 0$ **then**
    $X = (x_t + x_b) * (x_l + x_r)$
    $\tilde{X} = (\tilde{x}_t + \tilde{x}_b) * (\tilde{x}_l + \tilde{x}_r)$
    $I_h = min(x_t, \tilde{x}_t) + min(x_b, \tilde{x}_b)$
    $I_w = min(x_l, \tilde{x}_l) + min(x_r, \tilde{x}_r)$
    $I = I_h * I_w$
    $U = X + \tilde{X} - I$
    $IoU = \frac{I}{U}$
    $\mathcal{L} = -ln(IoU)$
  else
    $\mathcal{L} = 0$
  end
end

目标检测：Anchor Free_UnitBox

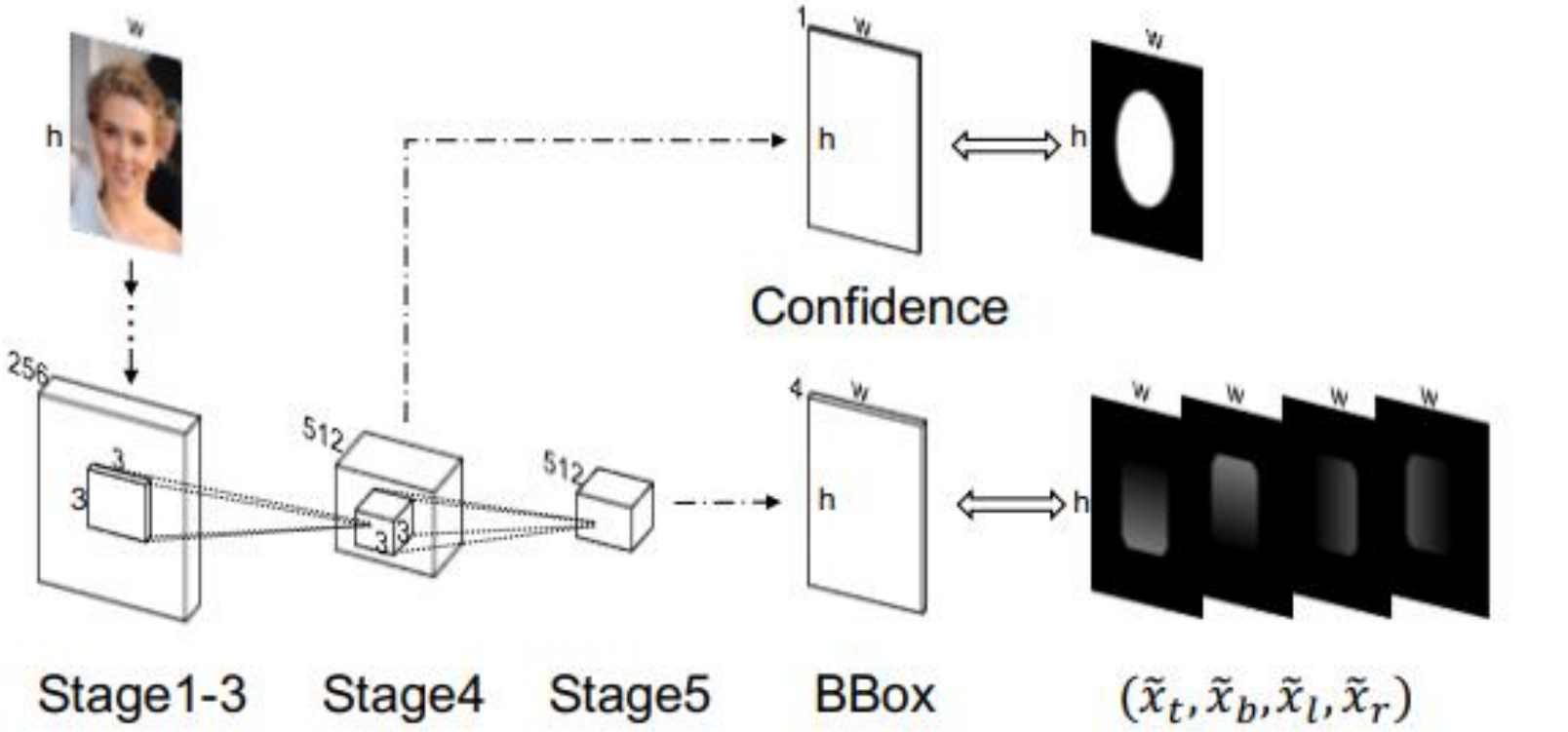


Figure 2: The Architecture of UnitBox Network.

目标检测：Anchor Free_UnitBox

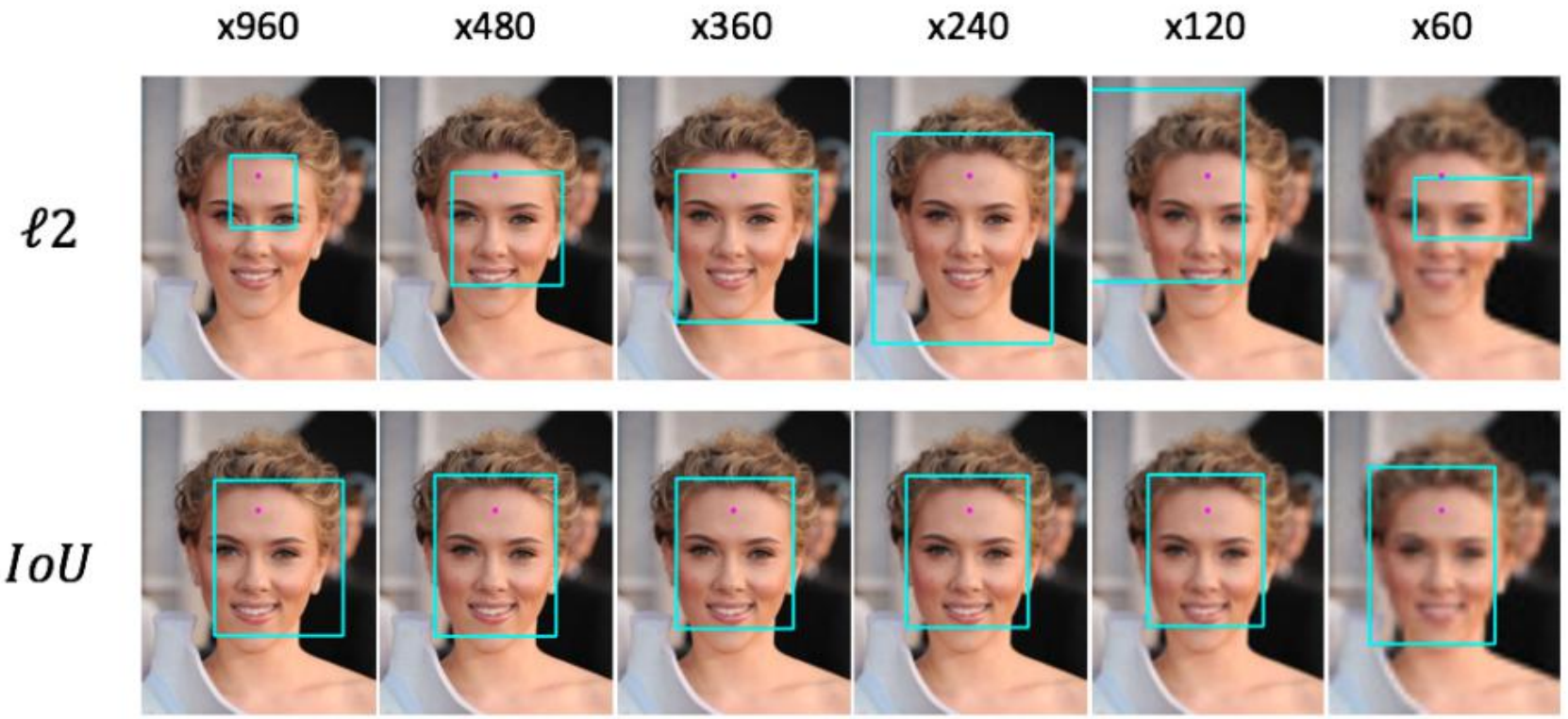


Figure 5: Compared to ℓ_2 loss, the IoU loss is much more robust to scale variations for bounding box prediction.

目标检测：Anchor Free_UnitBox

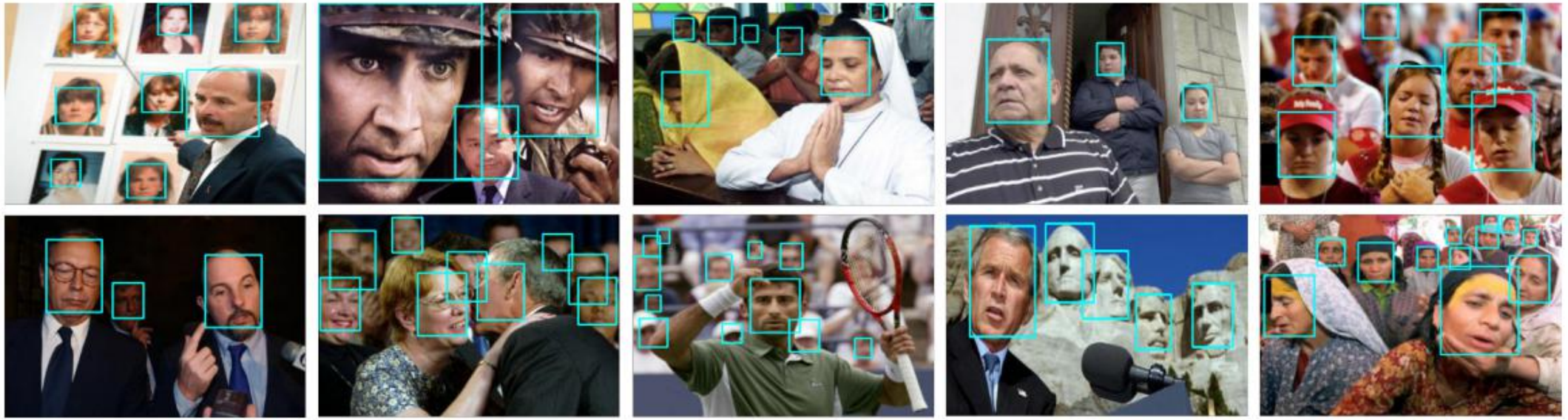
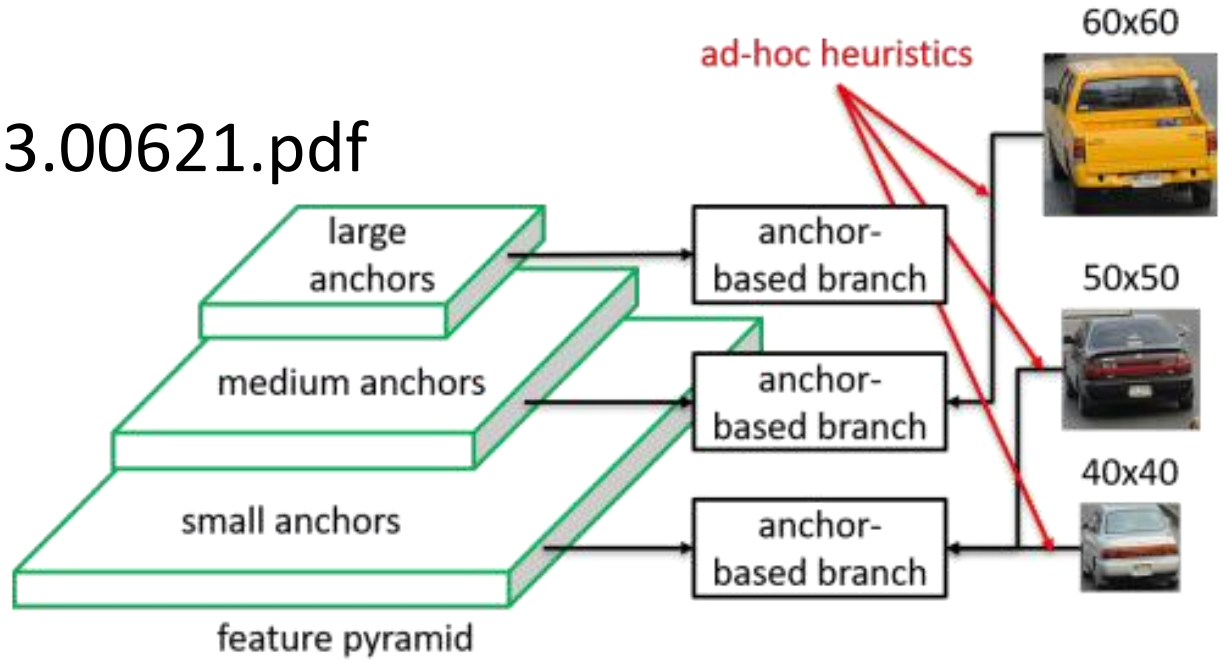


Figure 3: Examples of detection results of UnitBox on FDDB.

目标检测：Anchor Free_FSAF

- 核心思想：基于特征金字塔网络(Feature Pyramid Structure, FPN)的特征选择能力，在训练的时候可以动态分配每个实例的最合适的特征层，在预测的时候能够和带锚的模块分支一起工作，最后并行的进行预测输出。
- 论文：https://arxiv.org/pdf/1903.00621.pdf



目标检测：Anchor Free_FSAF

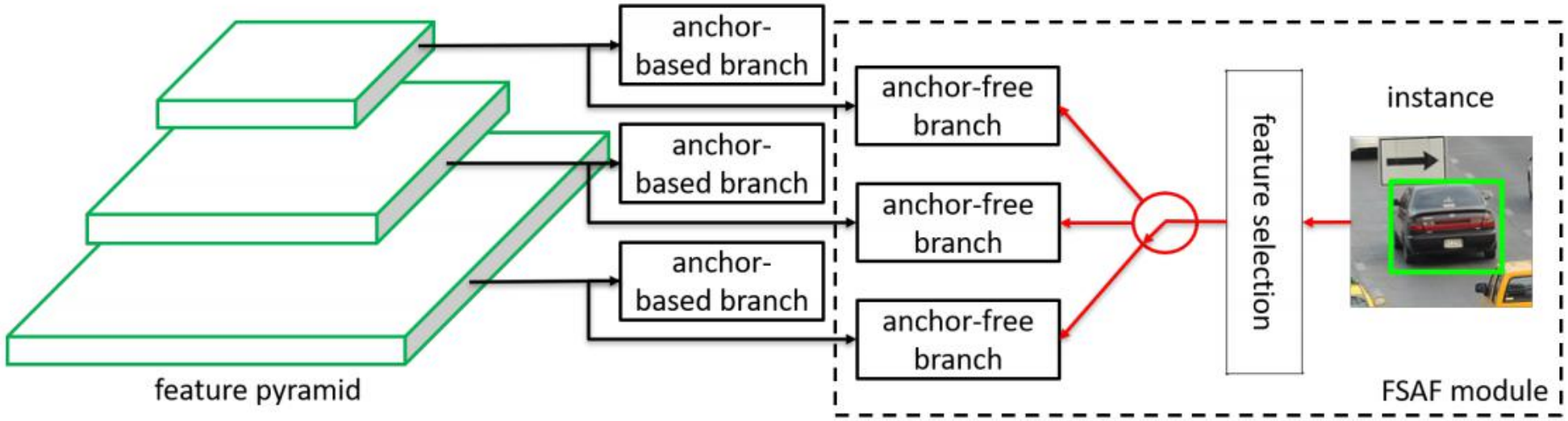


Figure 3: Overview of our FSAF module plugged into conventional anchor-based detection methods. During training, each instance is assigned to a pyramid level via feature selection for setting up supervision signals.

目标检测：Anchor Free_FSAF

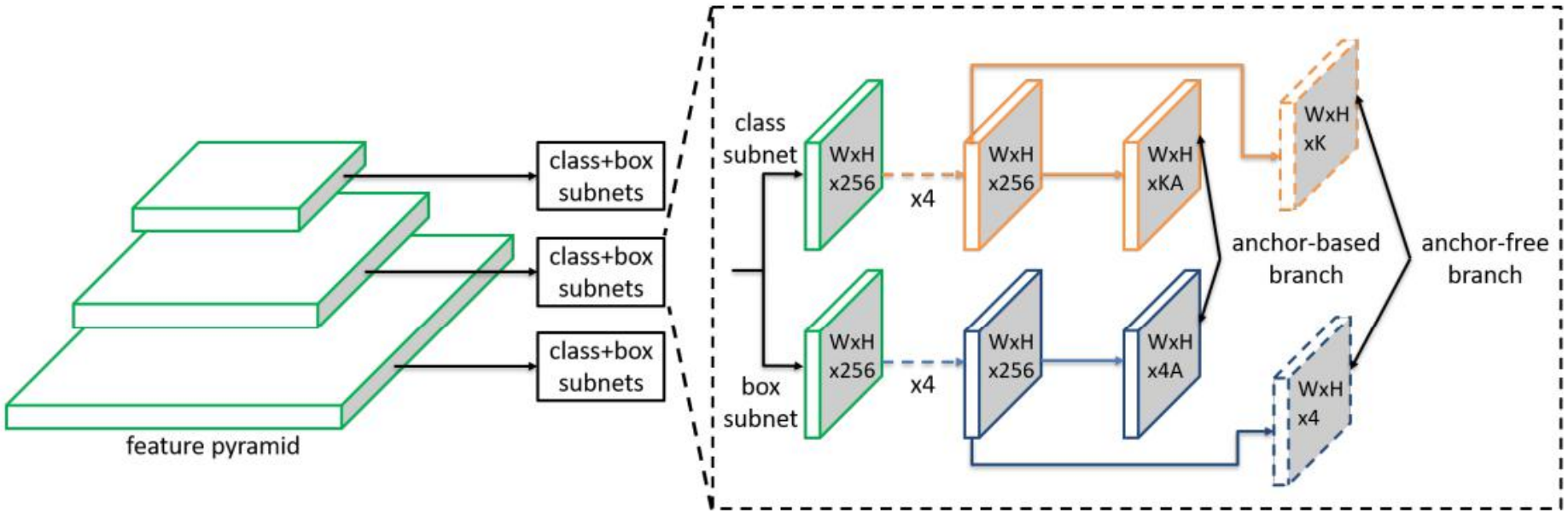


Figure 4: Network architecture of RetinaNet with our FSAF module. The FSAF module only introduces two additional conv layers (dashed feature maps) per pyramid level, keeping the architecture fully convolutional.

目标检测：Anchor Free_FSAF

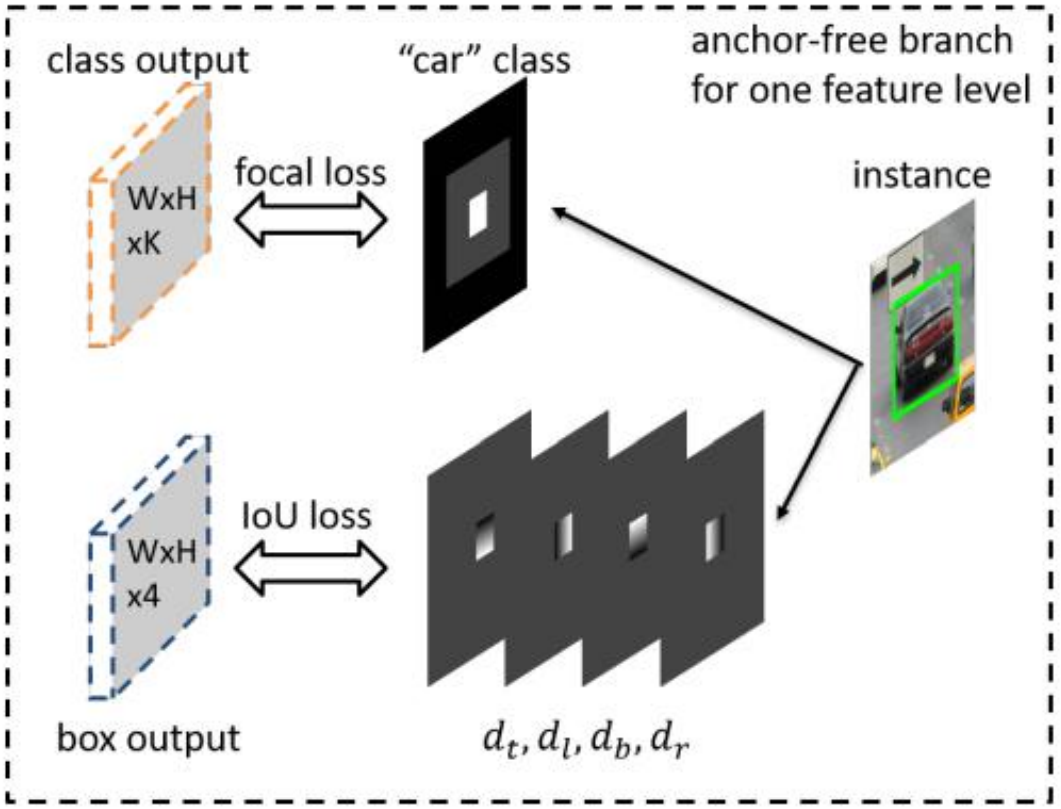


Figure 5: Supervision signals for an instance in one feature level of the anchor-free branches. We use focal loss for classification and IoU loss for box regression.

目标检测：Anchor Free_FSAF

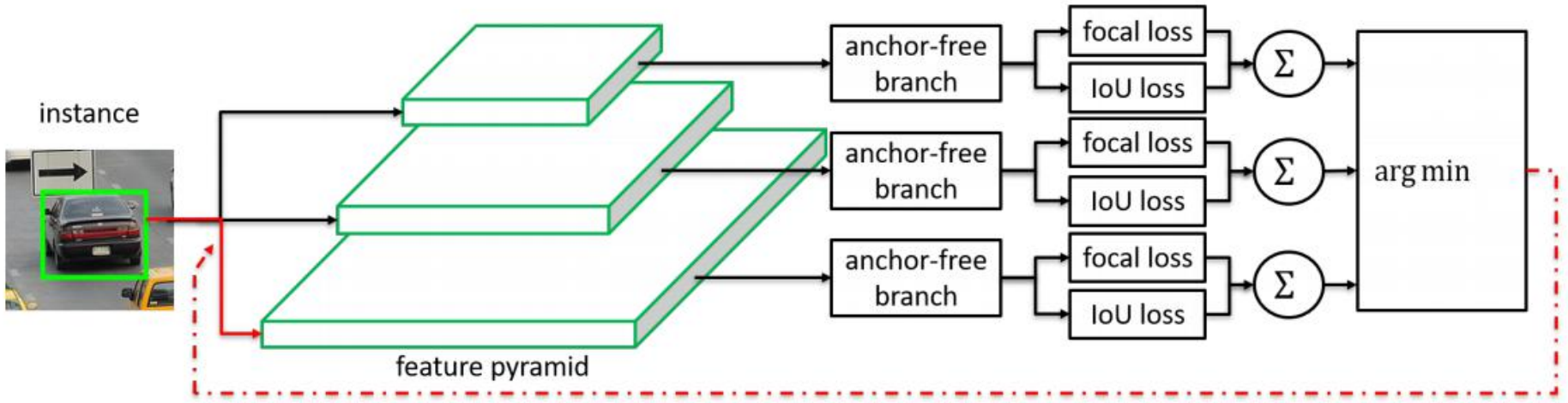


Figure 6: Online feature selection mechanism. Each instance is passing through all levels of anchor-free branches to compute the averaged classification (focal) loss and regression (IoU) loss over effective regions. Then the level with minimal summation of two losses is selected to set up the supervision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

目标检测：Anchor Free_FSAF

Backbone	Method	AP	AP_{50}	Runtime (ms/im)
R-50	RetinaNet	35.7	54.7	131
	Ours(FSAF)	35.9	55.0	107
	Ours(AB+FSAF)	37.2	57.2	138
R-101	RetinaNet	37.7	57.2	172
	Ours(FSAF)	37.9	58.0	148
	Ours(AB+FSAF)	39.3	59.2	180
X-101	RetinaNet	39.8	59.5	356
	Ours(FSAF)	41.0	61.5	288
	Ours(AB+FSAF)	41.6	62.4	362

Table 2: Detection accuracy and inference latency with different backbone networks on the COCO `minival`. **AB**: Anchor-based branches. **R**: ResNet. **X**: ResNeXt.

目标检测：Anchor Free_FCOS

- 核心思想：通过图像分割的思想来进行目标检测，通过”中心度(Center-ness)”来构建损失函数从而训练模型。
- 论文： https://arxiv.org/pdf/1904.01355.pdf

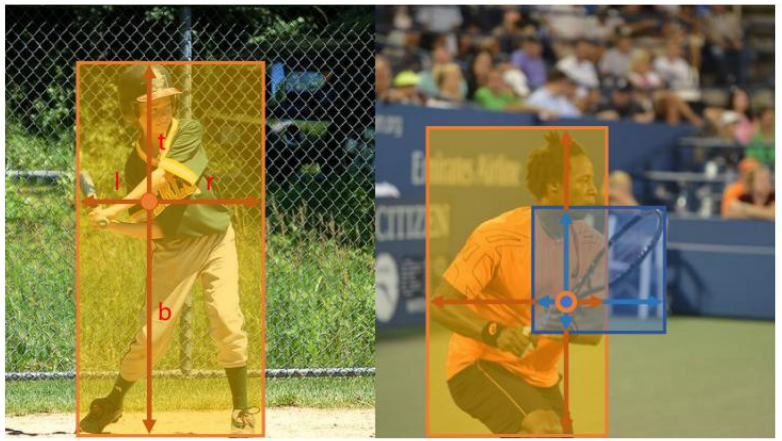
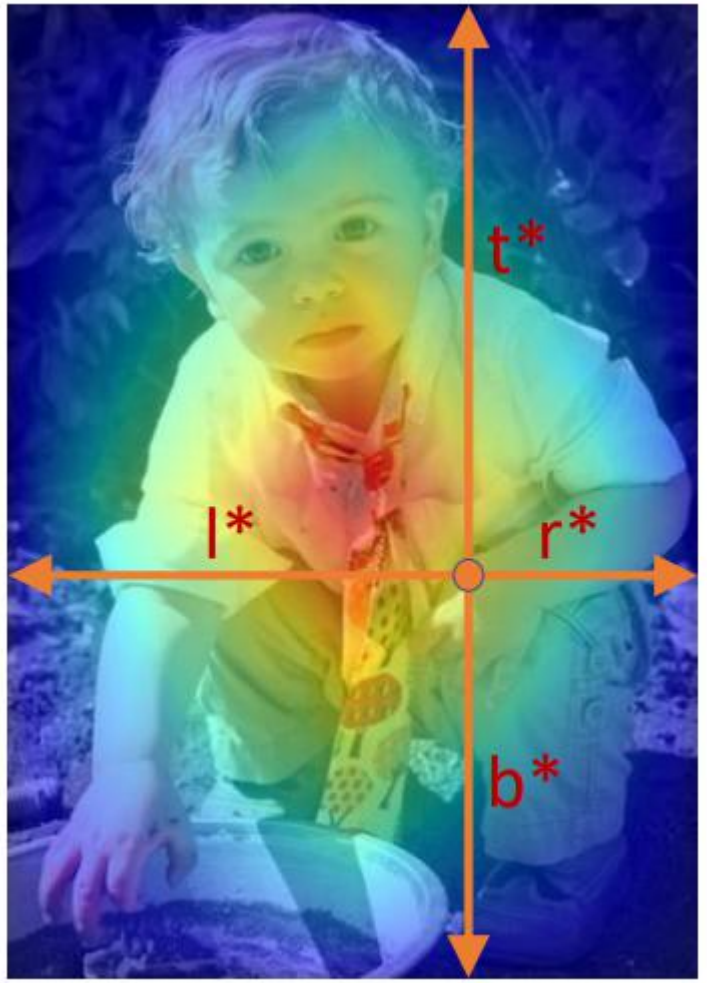


Figure 1 – As shown in the left image, FCOS works by predicting a 4D vector (l, t, r, b) encoding the location of a bounding box at each foreground pixel (supervised by ground-truth bounding box information during training). The right plot shows that when a location residing in multiple bounding boxes, it can be ambiguous in terms of which bounding box this location should regress.



目标检测：Anchor Free_FCOS

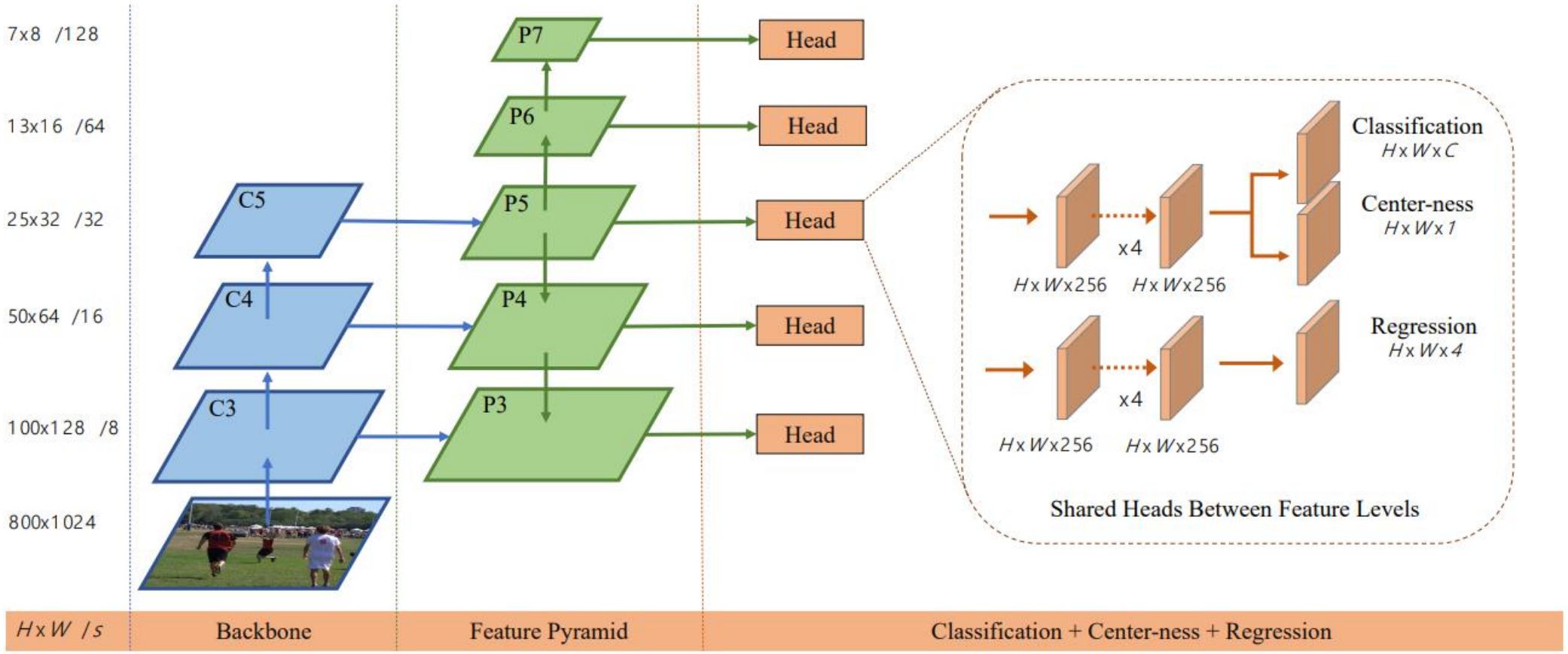


Figure 2 – The network architecture of FCOS, where C3, C4, and C5 denote the feature maps of the backbone network and P3 to P7 are the feature levels used for the final prediction. $H \times W$ is the height and width of feature maps. '/s' ($s = 8, 16, \ldots, 128$) is the down-sampling ratio of the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

目标检测：Anchor Free_FCOS

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

$$L(\{\boldsymbol{p}_{x,y}\}, \{\boldsymbol{t}_{x,y}\}) = \frac{1}{N_{\text{pos}}} \sum_{x,y} L_{\text{cls}}(\boldsymbol{p}_{x,y}, c^*_{x,y}) + \frac{\lambda}{N_{\text{pos}}} \sum_{x,y} \mathbb{1}_{\{c^*_{x,y}>0\}} L_{\text{reg}}(\boldsymbol{t}_{x,y}, \boldsymbol{t}^*_{x,y}),$$

- Centerness部分构建一个BCE(Binary Cross Entropy)损失，并添加到L中。最终让模型去学习那些位置是中心点位置。

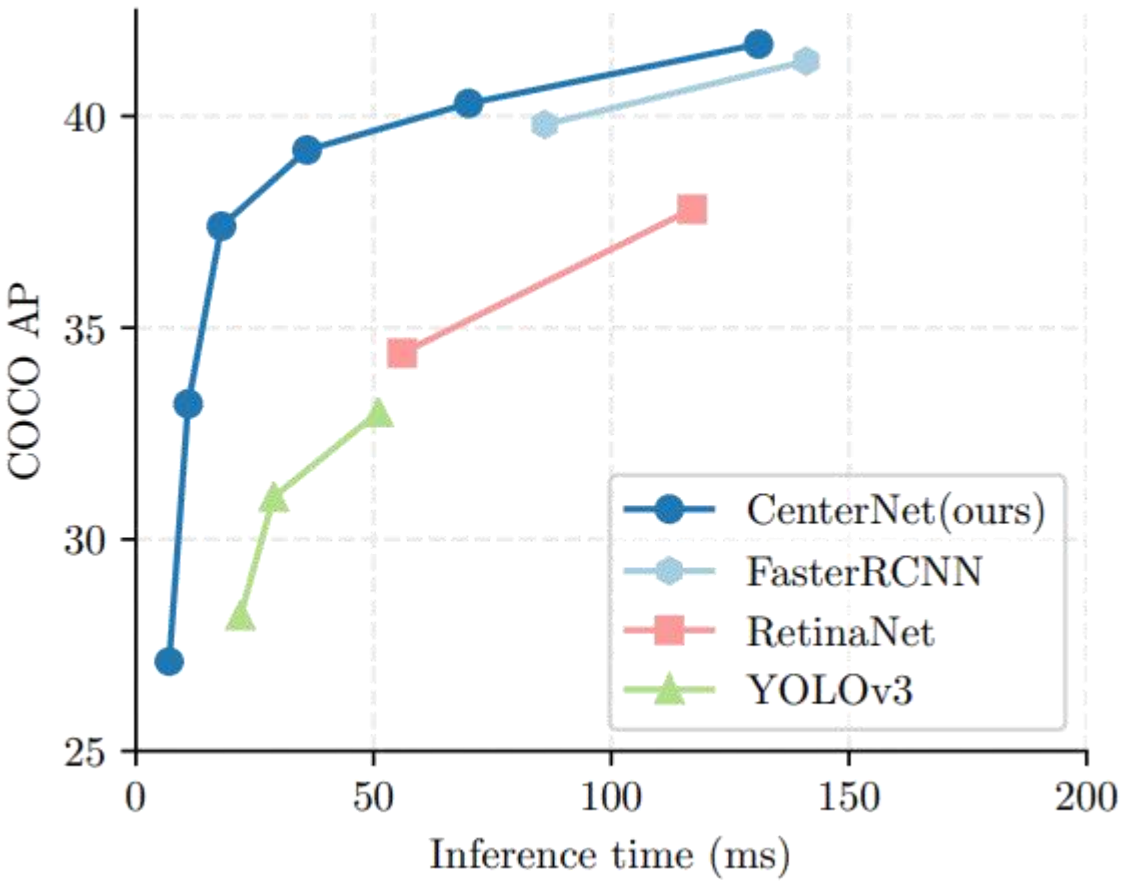
目标检测：Anchor Free_FCOS

Method	Backbone	AP	AP_{50}	AP_{75}	AP_S	AP_M	AP_L
Two-stage methods:							
Faster R-CNN w/ FPN [14]	ResNet-101-FPN	36.2	59.1	39.0	18.2	39.0	48.2
Faster R-CNN by G-RMI [11]	Inception-ResNet-v2 [27]	34.7	55.5	36.7	13.5	38.1	52.0
Faster R-CNN w/ TDM [25]	Inception-ResNet-v2-TDM	36.8	57.7	39.2	16.2	39.8	52.1
One-stage methods:							
YOLOv2 [22]	DarkNet-19 [22]	21.6	44.0	19.2	5.0	22.4	35.5
SSD513 [18]	ResNet-101-SSD	31.2	50.4	33.3	10.2	34.5	49.8
DSSD513 [5]	ResNet-101-DSSD	33.2	53.3	35.2	13.0	35.4	51.1
RetinaNet [15]	ResNet-101-FPN	39.1	59.1	42.3	21.8	42.7	50.2
CornerNet [13]	Hourglass-104	40.5	56.5	43.1	19.4	42.7	53.9
FSAF [34]	ResNeXt-64x4d-101-FPN	42.9	63.8	46.3	26.6	46.2	52.7
FCOS	ResNet-101-FPN	41.5	60.7	45.0	24.4	44.8	51.6
FCOS	HRNet-W32-5l [26]	42.0	60.4	45.3	25.4	45.0	51.0
FCOS	ResNeXt-32x8d-101-FPN	42.7	62.2	46.1	26.0	45.6	52.6
FCOS	ResNeXt-64x4d-101-FPN	43.2	62.8	46.6	26.5	46.2	53.3
FCOS w/ improvements	ResNeXt-64x4d-101-FPN	**44.7**	**64.1**	**48.4**	**27.6**	**47.5**	**55.6**

Table 5 – FCOS vs. other state-of-the-art two-stage or one-stage detectors (*single-model and single-scale results*). FCOS outperforms the anchor-based counterpart RetinaNet by 2.4% in AP with the same backbone. FCOS also outperforms the recent anchor-free one-stage detector CornerNet with much less design complexity. Refer to Table 3 for details of "improvements".

目标检测：Anchor Free_CenterNet—

- CenterNet: Object as Points
- 核心思想：
  - 将目标当作点来进行预测。
  - 模型直接预测HeatMaps(热力图，也就是认为属于中心点的概率值)
- 论文：
  - https://arxiv.org/pdf/1904.07850v1.pdf



目标检测：Anchor Free_CenterNet一

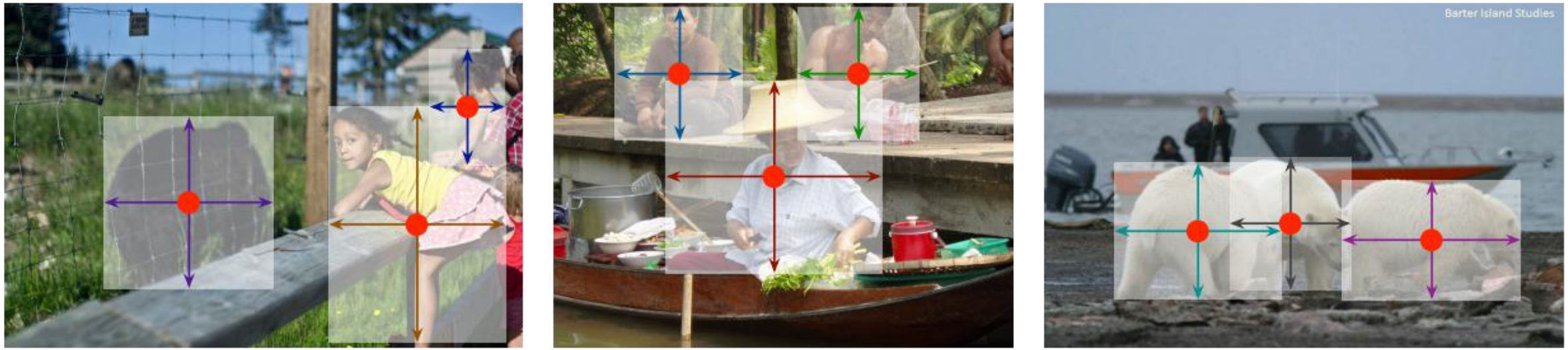
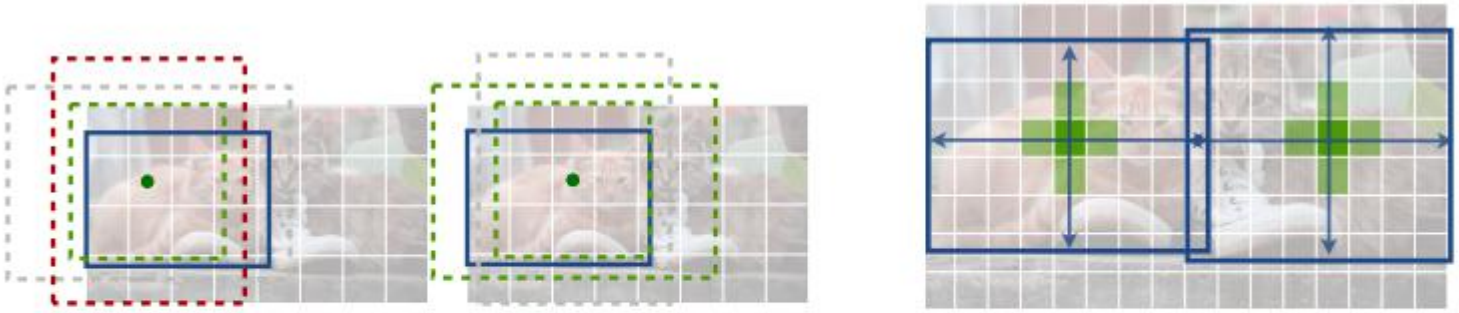


Figure 2: We model an object as the center point of its bounding box. The bounding box size and other object properties are inferred from the keypoint feature at the center. Best viewed in color.

目标检测：Anchor Free_CenterNet—



(a) Standard anchor based detection. Anchors count as positive with an overlap $IoU > 0.7$ to any object, negative with an overlap $IoU < 0.3$, or are ignored otherwise.

(b) Center point based detection. The center pixel is assigned to the object. Nearby points have a reduced negative loss. Object size is regressed.

Figure 3: Different between anchor-based detectors (a) and our center point detector (b). Best viewed on screen.

$$Y_{xyc} = \exp\left(-\frac{(x-\tilde{p}_x)^2+(y-\tilde{p}_y)^2}{2\sigma_p^2}\right) \quad \tilde{p} = \lfloor \frac{p}{R} \rfloor$$

目标检测：Anchor Free_CenterNet—

- Pixelwise Logistic Regression Loss:
  - N为Image中的keypoints的数目，α和β为超参数，论文中建议α=2，β=4.
  - 网络输出Feature形状为: [H/R, W/R, C]

$$L_k = -\frac{1}{N}\sum_{xyc}\begin{cases}\left(1-\hat{Y}_{xyc}\right)^{\alpha}\ln\left(\hat{Y}_{xyc}\right) & \text{if } Y_{xyc}=1\\ \left(1-Y_{xyc}\right)^{\beta}\left(\hat{Y}_{xyc}\right)^{\alpha}\ln\left(1-\hat{Y}_{xyc}\right) & \text{otherwise}\end{cases}$$

目标检测：Anchor Free_CenterNet—

- Object Local Offset Loss:
  - 由于图像下采样的过程中，GT的关键点会由于数据离散原因导致位置偏差，对每个中心点附加一个局部偏移，网络希望输出Feature形状为: [H/R, W/R, 2]，所有类别共享同一个偏移值。
  - NOTE: 只会在关键点(中心点)处进行损失的计算。

$$L_{off} = \frac{1}{N}\sum_{p}\left|\hat{O}_{\tilde{p}} - \left(\frac{p}{R} - \tilde{p}\right)\right|.$$

目标检测：Anchor Free_CenterNet—

- Object Bounding Box Size Loss:
  - 使用单一的尺寸预测，网络希望输出Feature形状为: [H/R, W/R, 2]
  - NOTE: 只会在关键点(中心点)处进行损失的计算。

Let $(x_1^{(k)}, y_1^{(k)}, x_2^{(k)}, y_2^{(k)})$ be the bounding box of object k with category c_k. Its center point is lies at $p_k = (\frac{x_1^{(k)}+x_2^{(k)}}{2}, \frac{y_1^{(k)}+y_2^{(k)}}{2})$. We use our keypoint estimator $\hat{Y}$ to predict all center points. In addition, we regress to the object size $s_k = (x_2^{(k)} - x_1^{(k)}, y_2^{(k)} - y_1^{(k)})$ for each object k. To limit the computational burden, we use a single size prediction $\hat{S} \in \mathcal{R}^{\frac{W}{R} \times \frac{H}{R} \times 2}$ for all object categories. We use an L1 loss at the center point similar to Objective 2:

$$L_{size} = \frac{1}{N} \sum_{k=1}^{N} \left| \hat{S}_{p_k} - s_k \right|. \tag{3}$$

目标检测：Anchor Free_CenterNet—

- 最终模型输出形状: [H/R, W/R, C+4]，也就是Keypoint regression、Local offset、Size Regression全部共享同一个backbone网络输出。
- 最终损失函数：
  - λ_size=0.1, λ_off=1.0

$$L_{det} = L_k + \lambda_{size} L_{size} + \lambda_{off} L_{off}.$$

目标检测：Anchor Free_CenterNet—

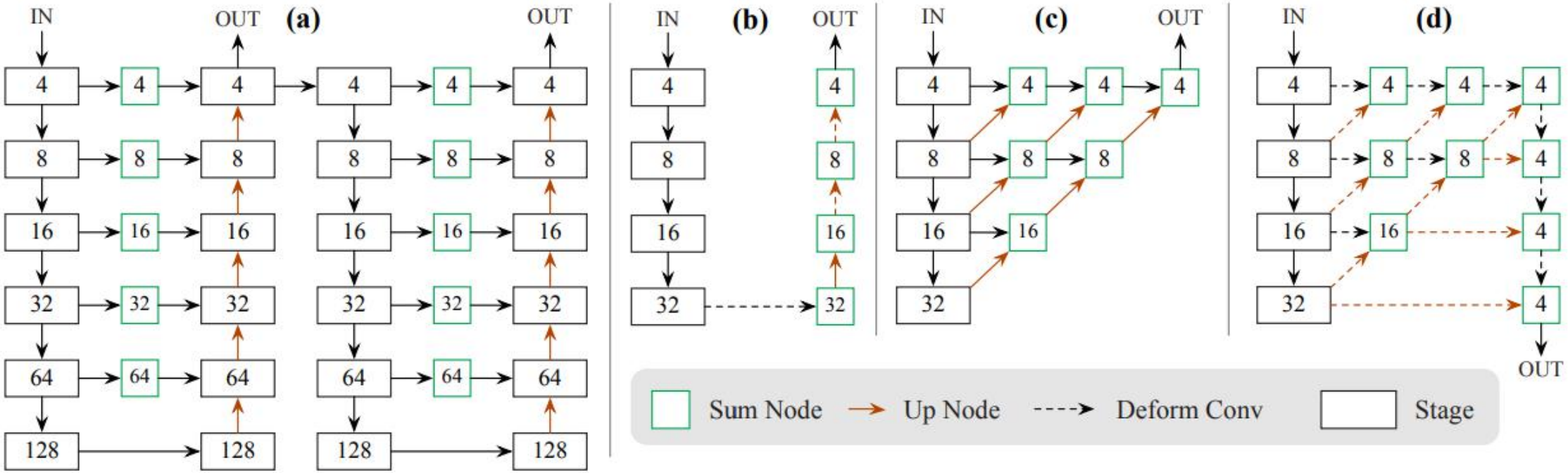


Figure 6: Model diagrams. The numbers in the boxes represent the stride to the image. (a): Hourglass Network [30]. We use it as is in CornerNet [30]. (b): ResNet with transpose convolutions [55]. We add one 3×3 deformable convolutional layer [63] before each up-sampling layer. Specifically, we first use deformable convolution to change the channels and then use transposed convolution to upsample the feature map (such two steps are shown separately in $32 \rightarrow 16$. We show these two steps together as a dashed arrow for $16 \rightarrow 8$ and $8 \rightarrow 4$). (c): The original DLA-34 [58] for semantic segmentation. (d): Our modified DLA-34. We add more skip connections from the bottom layers and upgrade every convolutional layer in upsampling stages to deformable convolutional layer.

目标检测：Anchor Free_CenterNet—

- CenterNet直接将三个分支的数据合并即可得到最终预测的边框位置，具体合并方式如下：
  - 从Heatmap中提取出每个类别的峰值点，也就是可能是物体中心点的区域，获取峰值点的方式类似NMS非极大值抑制：对于每个类别的Heatmap上的每个点，均考虑该点附近的八个点进行比较，如果该点的响应值大于或等于其相邻的八个点，那么当前点保留。最后获取所有保留下来的前N个响应值最大的点作为预测出来的中心点。N默认为100。
  - 结合Local Offset以及Bounding Box Size的回归预测值，从而产生最终边框。

dia

目标检测：Anchor Free_CenterNet—

模型预测的中心点坐标

模型预测的中心点偏移值

$$\left(\hat{x}_i + \delta\hat{x}_i - \hat{w}_i/2,\ \hat{y}_i + \delta\hat{y}_i - \hat{h}_i/2,\right.$$
$$\left.\hat{x}_i + \delta\hat{x}_i + \hat{w}_i/2,\ \hat{y}_i + \delta\hat{y}_i + \hat{h}_i/2\right),$$

模型预测的边框大小

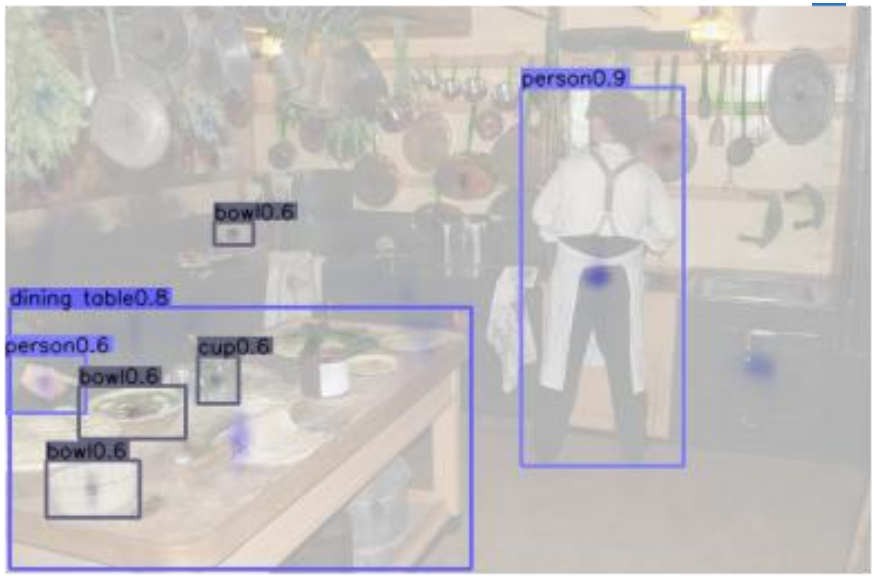
目标检测：Anchor Free_CenterNet一

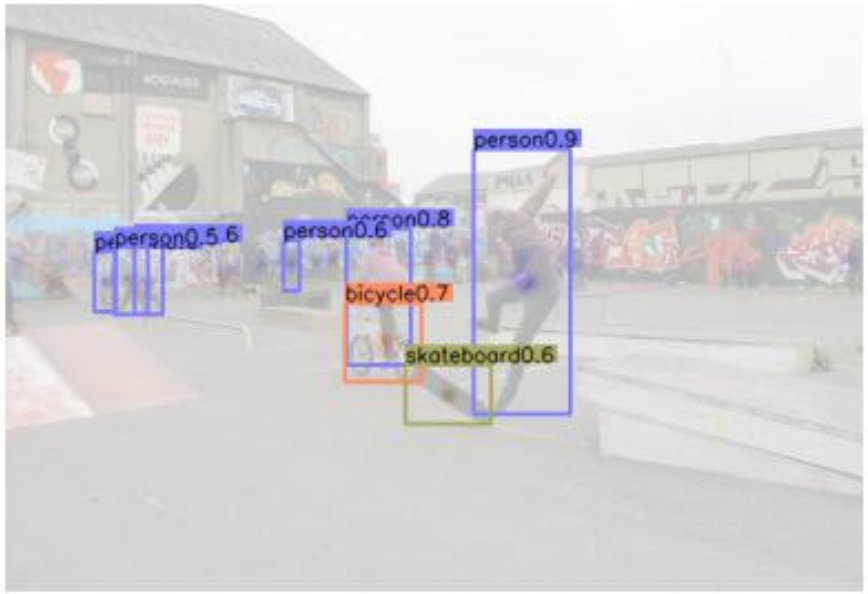
keypoint heatmap [C]

local offset [2]

object size [2]

目标检测：Anchor Free_CenterNet一





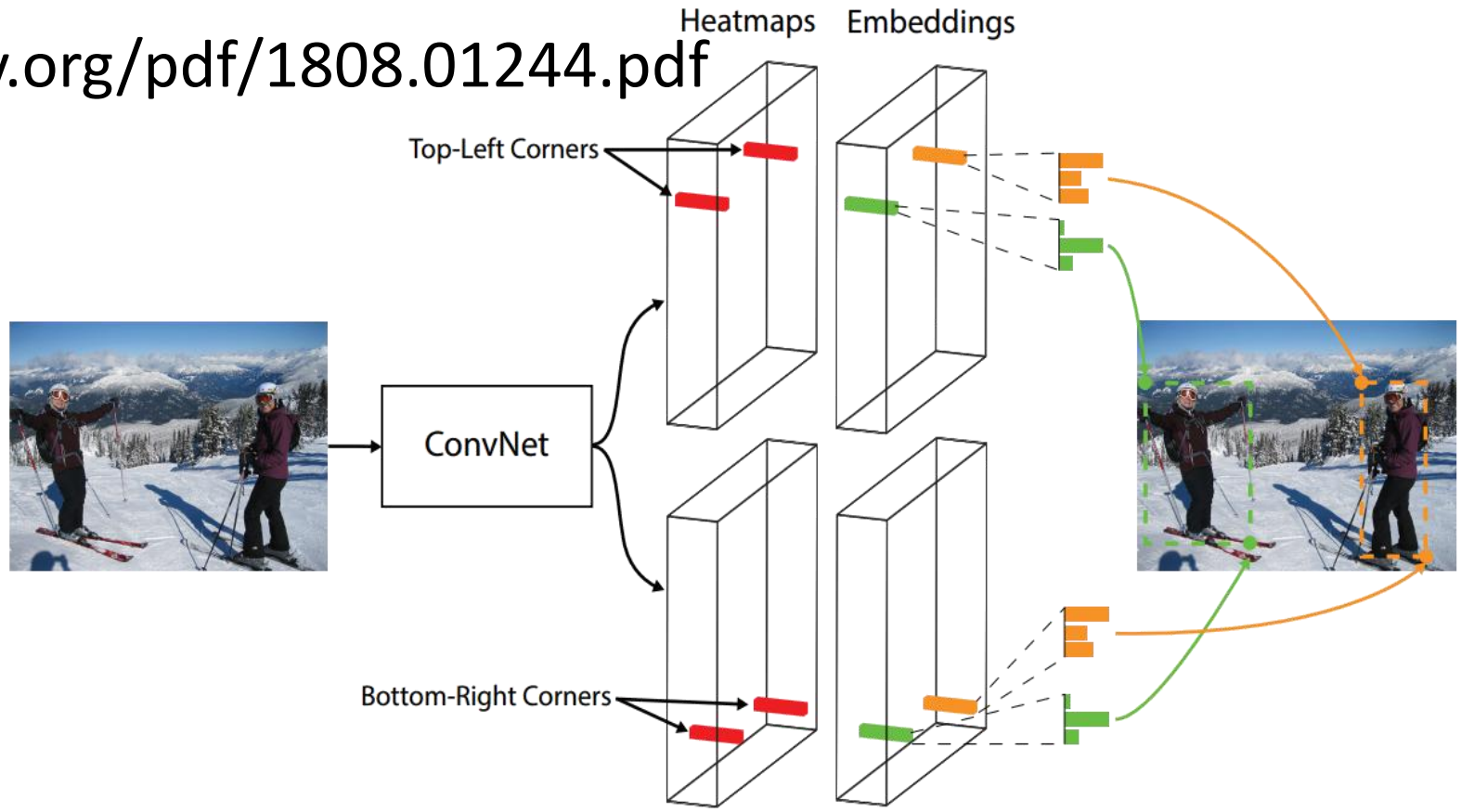
目标检测：Anchor Free_CenterNet—

	Backbone	FPS	AP	AP_{50}	AP_{75}	AP_S	AP_M	AP_L
MaskRCNN [21]	ResNeXt-101	**11**	39.8	62.3	43.4	22.1	43.2	51.2
Deform-v2 [63]	ResNet-101	-	46.0	67.9	50.8	27.8	49.1	59.5
SNIPER [48]	DPN-98	*2.5*	46.1	67.0	51.6	29.6	48.9	58.1
PANet [35]	ResNeXt-101	-	47.4	67.2	51.8	30.1	**51.7**	60.0
TridentNet [31]	ResNet-101-DCN	0.7	**48.4**	**69.7**	**53.5**	**31.8**	51.3	**60.3**
YOLOv3 [45]	DarkNet-53	20	33.0	57.9	34.4	18.3	25.4	41.9
RetinaNet [33]	ResNeXt-101-FPN	5.4	40.8	61.1	44.1	24.1	44.2	51.2
RefineDet [59]	ResNet-101	-	36.4 / 41.8	57.5 / 62.9	39.5 / 45.7	16.6 / 25.6	39.9 / 45.1	51.4 / 54.1
CornerNet [30]	Hourglass-104	4.1	40.5 / 42.1	56.5 / 57.8	43.1 / 45.3	19.4 / 20.8	42.7 / 44.8	**53.9** / 56.7
ExtremeNet [61]	Hourglass-104	3.1	40.2 / 43.7	55.5 / 60.5	43.2 / 47.0	20.4 / 24.1	43.2 / 46.9	53.1 / 57.6
FSAF [62]	ResNeXt-101	*2.7*	**42.9** / 44.6	**63.8** / **65.2**	**46.3** / 48.6	**26.6** / **29.7**	**46.2** / **47.1**	52.7 / 54.6
CenterNet-DLA	DLA-34	**28**	39.2 / 41.6	57.1 / 60.3	42.8 / 45.1	19.9 / 21.5	43.0 / 43.9	51.4 / 56.0
CenterNet-HG	Hourglass-104	7.8	42.1 / **45.1**	61.1 / 63.9	45.9 / **49.3**	24.1 / 26.6	45.5 / **47.1**	52.8 / **57.7**

Table 2: State-of-the-art comparison on COCO test-dev. Top: two-stage detectors; bottom: one-stage detectors. We show single-scale / multi-scale testing for most one-stage detectors. Frame-per-second (FPS) were measured on the same machine whenever possible. Italic FPS highlight the cases, where the performance measure was copied from the original publication. A dash indicates methods for which neither code and models, nor public timings were available.

目标检测：Anchor Free_CornerNet

- 核心思想：直接检测目标框的左上角坐标点以及右下角坐标点，并使用角点池化(Corner pooling)技术更好的进行角点定位。
- 论文：https://arxiv.org/pdf/1808.01244.pdf



目标检测：Anchor Free_CornerNet

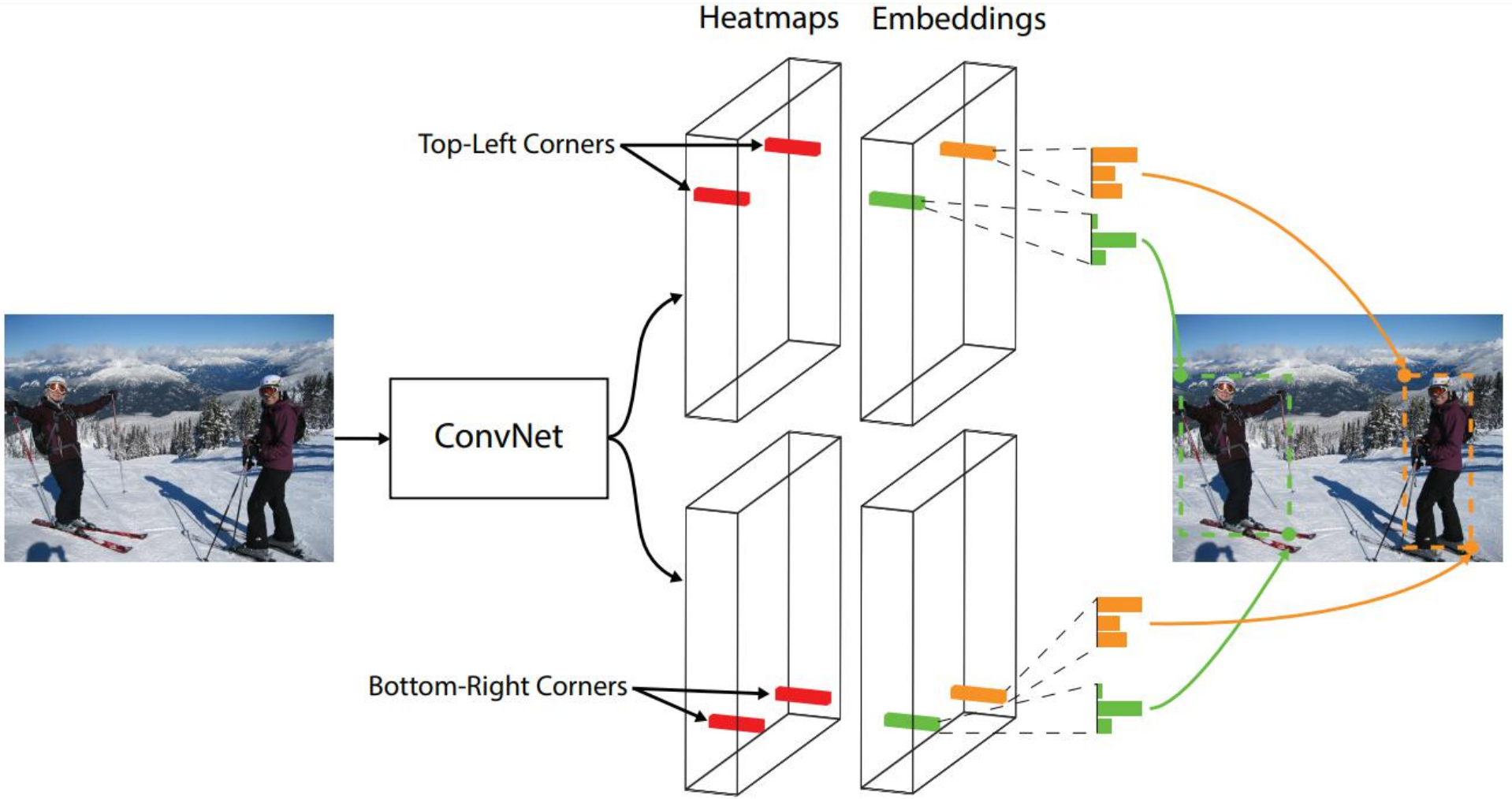


Fig. 1 We detect an object as a pair of bounding box corners grouped together. A convolutional network outputs a heatmap for all top-left corners, a heatmap for all bottom-right corners, and an embedding vector for each detected corner. The network is trained to predict similar embeddings for corners that belong to the same object.

目标检测：Anchor Free_CornerNet

- 参考链接：
  - https://blog.csdn.net/weixin_40414267/article/details/82379793
  - https://www.cnblogs.com/yumoye/p/10964916.html

目标检测：Anchor Free_CenterNet二

- 参考：
  - https://www.e-learn.cn/content/qita/2340220
  - https://blog.csdn.net/diligent_321/article/details/89736598

目标检测：Anchor Free_CornerNet-Lite

- 参考：
  - https://www.e-learn.cn/content/qita/2340220
  - https://www.cnblogs.com/gawain-ma/p/10868852.html

目标检测：Anchor Free_GA RPN

- 核心思想：利于语义特征来生成锚框，然后一起预测目标中心位置以及不同位置的尺度和长宽比。有点类似直接使用RPN网络产生预测框。
- 论文：https://arxiv.org/pdf/1901.03278.pdf

目标检测：Anchor Free_GA RPN

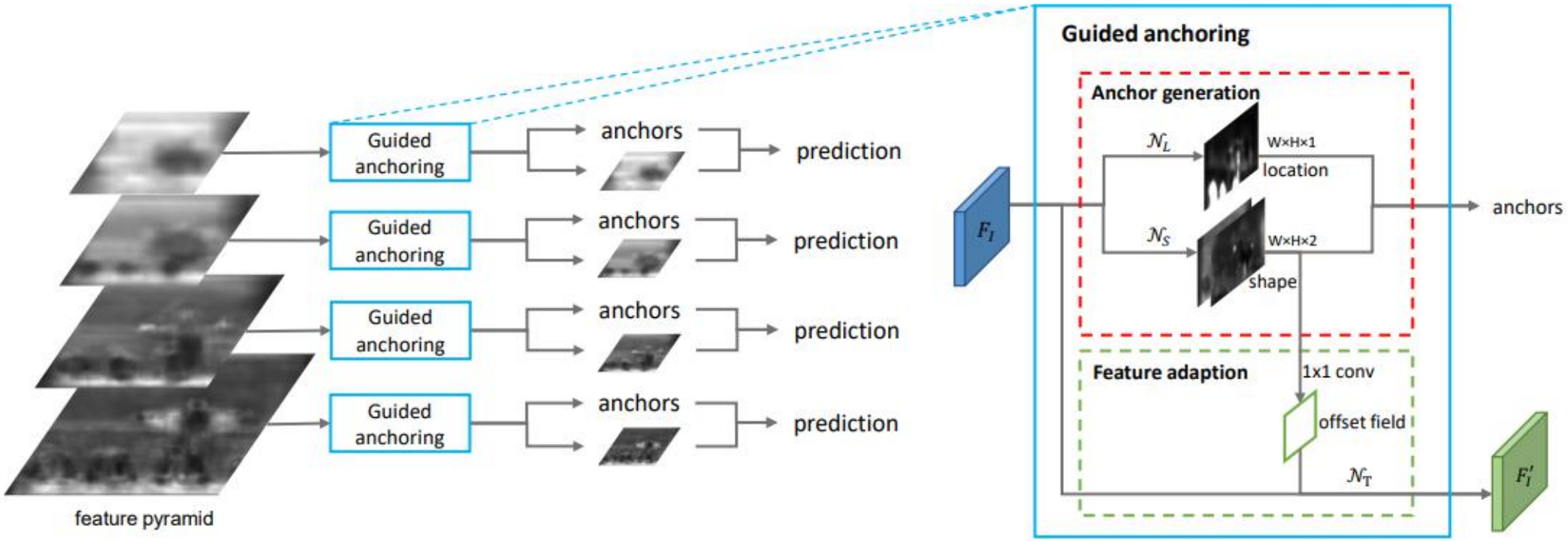


Figure 1: An illustration of our framework. For each output feature map in the feature pyramid, we use an anchor generation module with two branches to predict the anchor location and shape, respectively. Then a feature adaption module is applied to the original feature map to make the new feature map aware of anchor shapes.

目标检测：Anchor Free_GA RPN

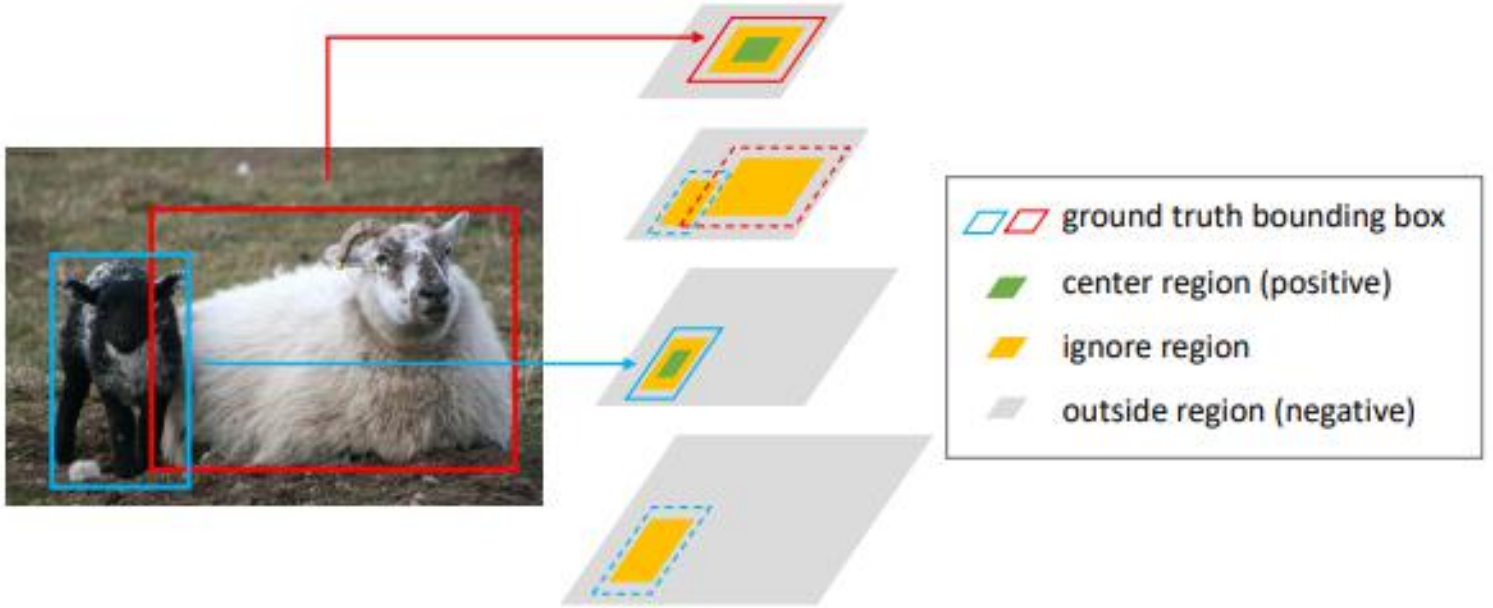


Figure 2: Anchor location target for multi-level features. We assign ground truth objects to different feature levels according to their scales, and define CR, IR and OR respectively. (Best viewed in color.)

目标检测：Anchor Free_GA RPN