Object Detection

mAP

> AP is averaged over all categories. Traditionally, this is called “mean average precision” (mAP). We make no distinction between AP and mAP (and likewise AR and mAR) and assume the difference is clear from context.

mAP (mean Average Precision) for Object Detection - Jonathan Hui - Medium

CornerNet

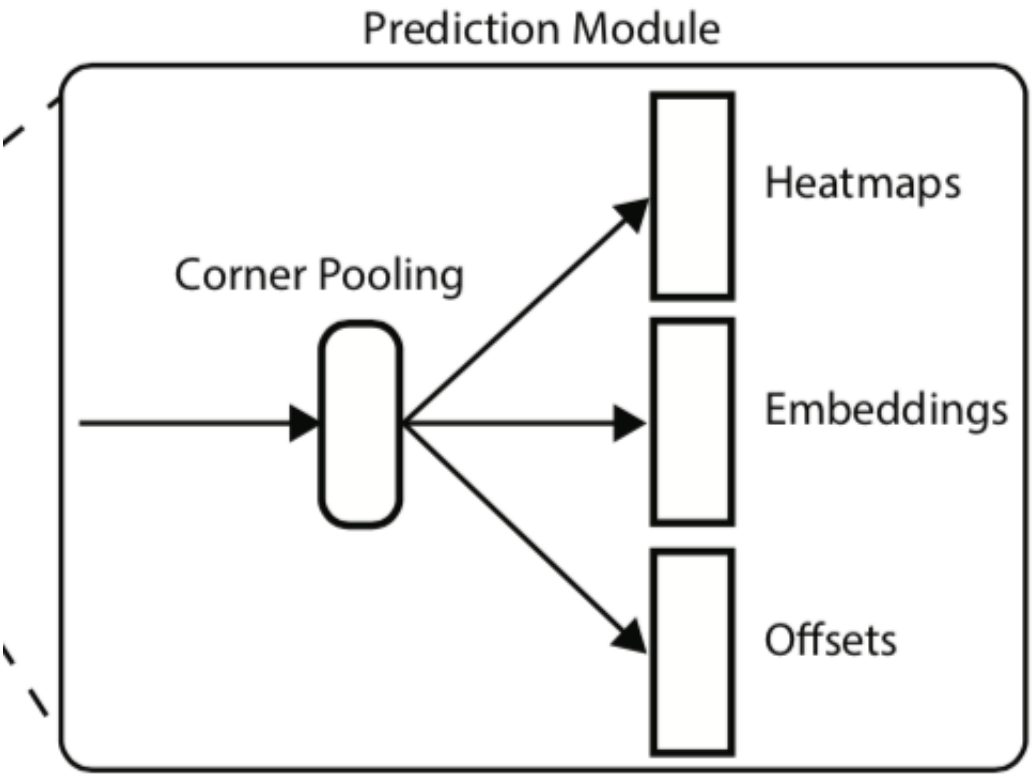
> anchor-free

Motivation

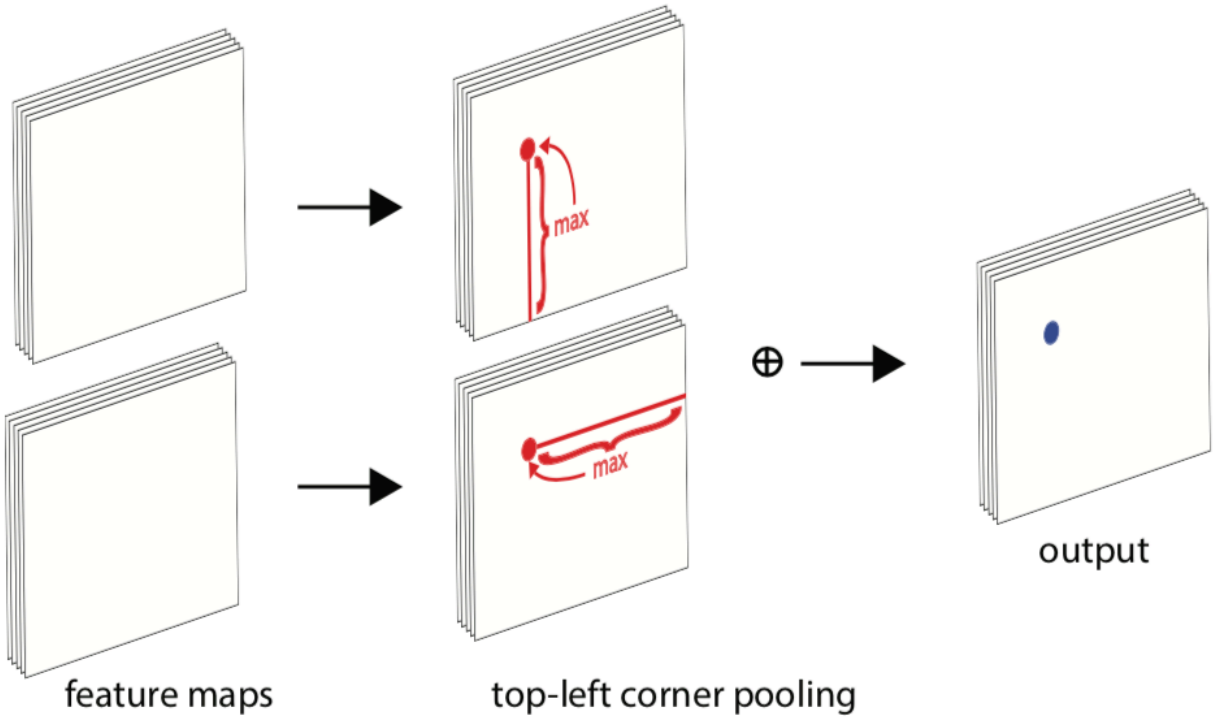
1. anchor box太多，只有少部分和GT重合
2. anchor 选择需要人为设计（数量，尺寸，比例），不同尺度anchor设置不同

两个部分： 1) 检测角网络，左上+右下 2) 嵌入网络，用于匹配角点

Method



检测corner网络：提特征之后，经过corner pooling产生每个类别的左上角和右下角的heat map。为正负样本匹配，只惩罚GT一定范围外的预测点（通过IoU threshold限制radius） **计算嵌入embed网络**：用于匹配同框的左上右下。损失函数：同框左上和右下接近（variance小），不同框的平均embed距离大。
Corner pooling：解决角点特征少，取一条线上的最大值pooling



Hourglass network： 提特征

two stage得到预测框之后采用RoI pooling/align提取检测框内部的信息，只有内部信息(approx.)的特征再进行一次框定和分类（refine步骤）；而一阶段的方法在提取到特征之后分成两支进行框定和分类，没有对近似的只包含框内物体的特征(局部特征)进行再一次提取，所以精度较差

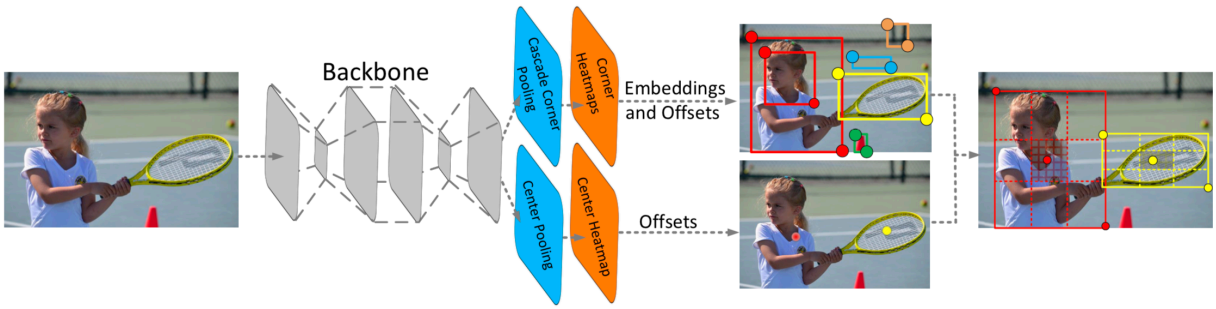
CenterNet

> anchor-free

Motivation

CornerNet只用到边缘的特征信息，没有用到内部的特征信息（造成不止对物体的边缘敏感，也对背景的边缘敏感）。内部信息对于决定两个keypoint是否是同一个框有帮助。 CenterNet预测三元组：左上，右下，中心

Method



二分支：产生corner点并match形成框；产生center点。如果center点在框的central region，计算框，否则删除 central region 确定判定的中心区域的大小：大框偏小，小框偏大，整体线性

$$
\begin{cases}
\mathrm{ctl_x} = \frac{(n+1)\mathrm{tl_x}+(n-1)\mathrm{br_x}}{2n} \\
\mathrm{ctl_y} = \frac{(n+1)\mathrm{tl_y}+(n-1)\mathrm{br_y}}{2n} \\
\mathrm{cbr_x} = \frac{(n-1)\mathrm{tl_x}+(n+1)\mathrm{br_x}}{2n} \\
\mathrm{cbr_y} = \frac{(n-1)\mathrm{tl_y}+(n+1)\mathrm{br_y}}{2n}
\end{cases}
$$

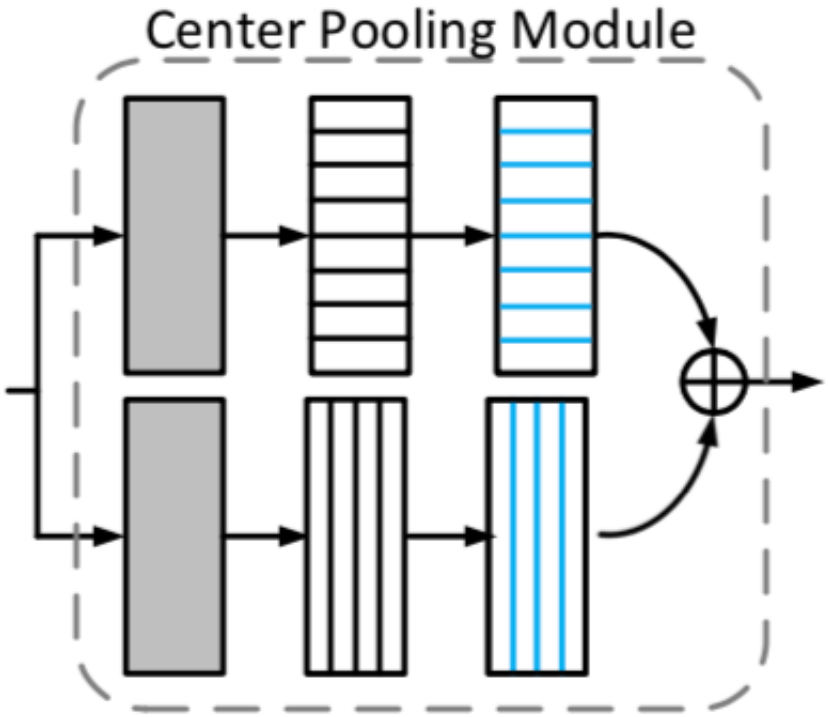
N离散变化，过threshold后变系数n。小→threshold→大，3→5

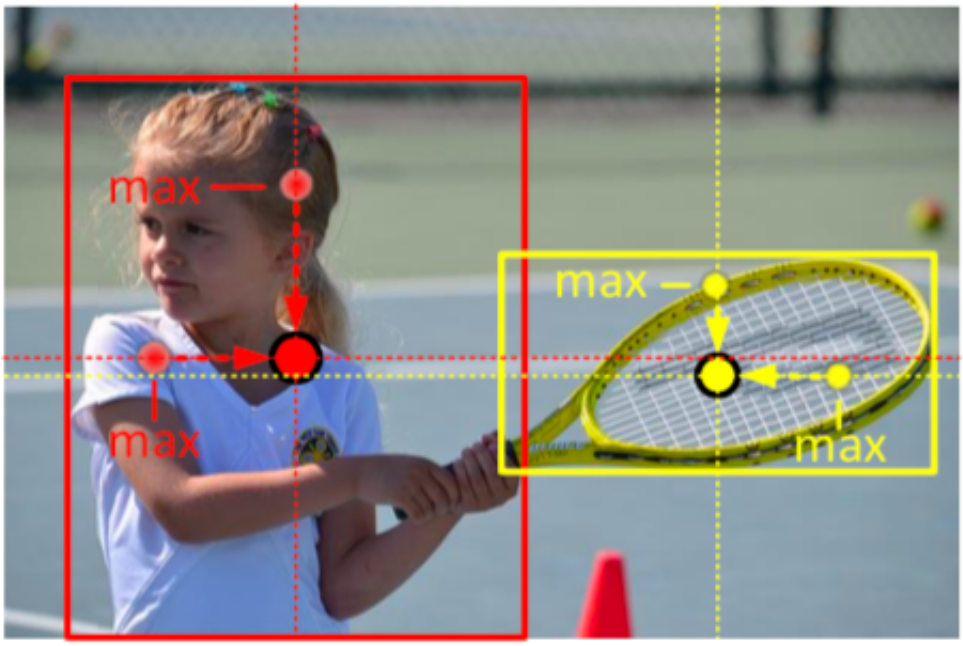
Enrich center&corner information

center pooling 用在预测中心点时，增加中心点的recognizable特征

```
1    例如找leftmost的（即把horizontal最大值传到最左边），每个点看自己到最右边的最大值，不断传，到最左可获得整条横线最大；找topmost（把vertical最大值传到最上面），每个点看自己到最下，不断传，到最上则可获得整条线最大
```

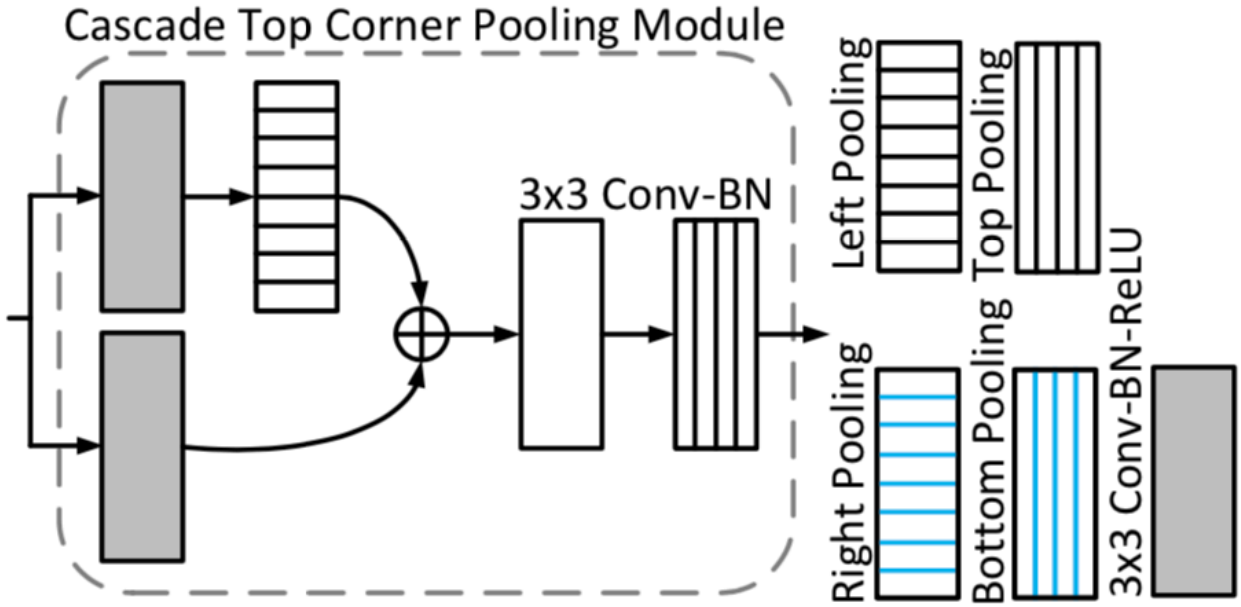
模块&示意图

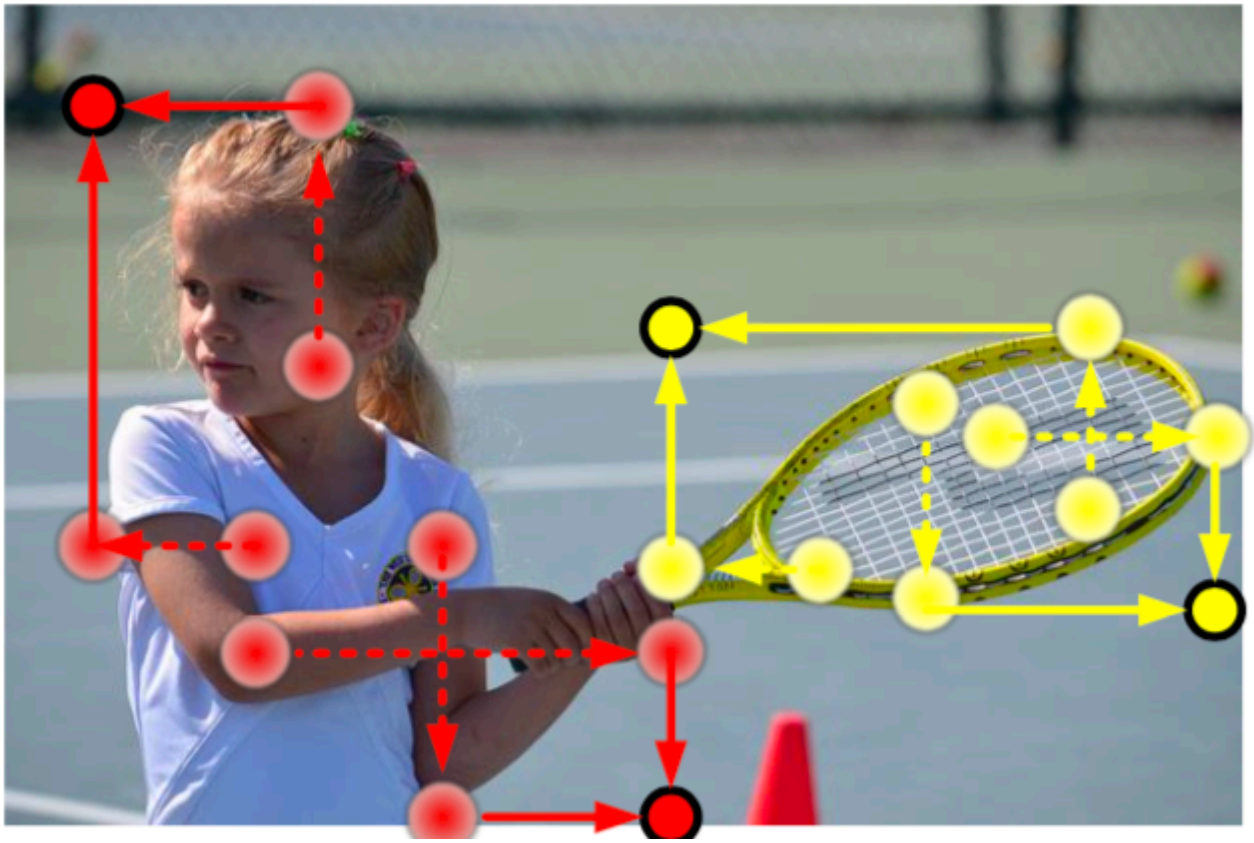




输出map表示是否为center点，然后找横向和纵向最大值

cascade corner pooling 增加角点的特征，相比corner pooling增加内部，使其不对边缘敏感 沿着边缘找边缘最大值，再从边缘最大值的位置 *向内找内部最大值* ，最后两个最大值相加 模块&示意图





> Q: how to classification?
> Need RoI align?

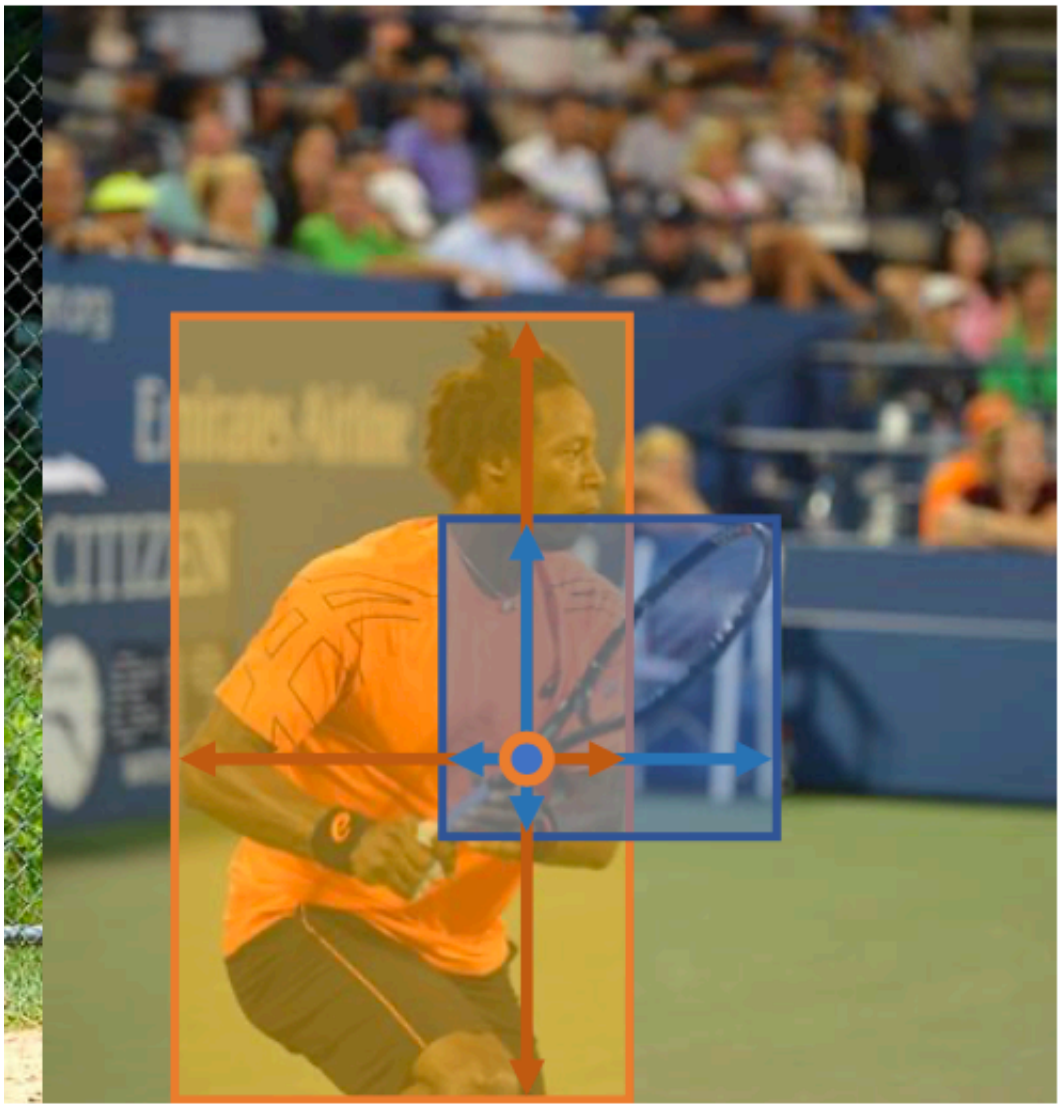
FCOS: Fully Convolutional One-Stage Object Detection

> anchor-free 消除anchor，减少IoU的计算和GT框的匹配。可以代替二阶段的RPN

按照像素进行预测per-pixel prediction，预测每一个像素点的四个维度的框 `(Left, Top, Right, Bottom)` 👇

- 对于特征图上每一个点，对应一个原图上的框。直接把特征图上的像素点看成训练样本而不是在点上铺不同长宽比和大小的anchor框

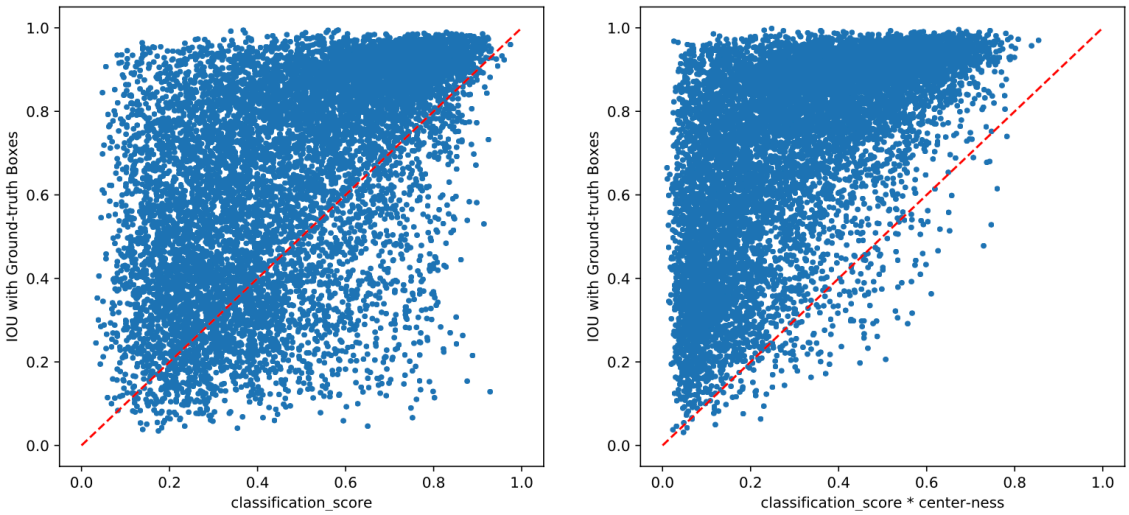
- 对于一个点落在多个GT框中（ambiguous samples）选择最小的bbox作为target。同时通过multi-level prediction来减少数量。👇



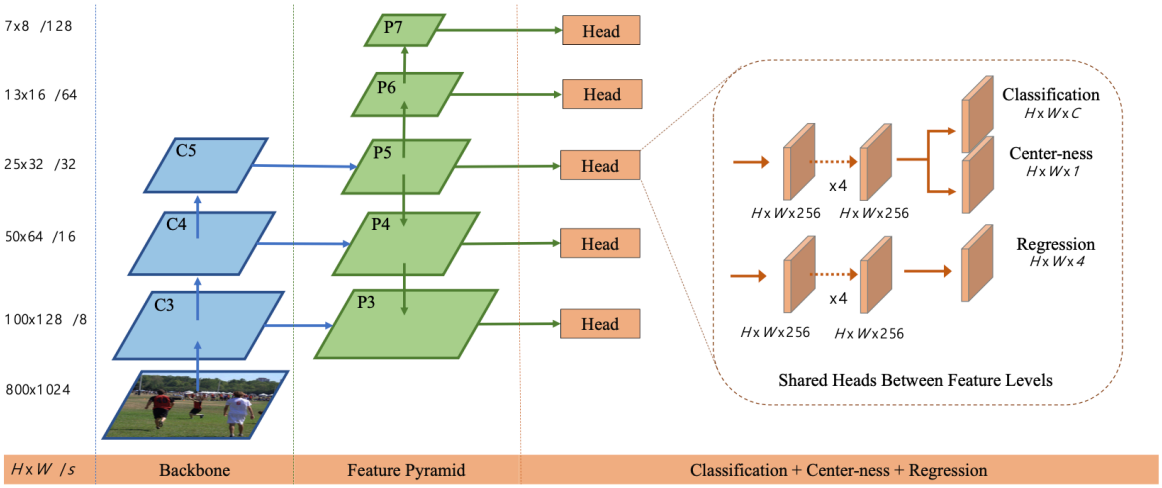
👆which bbox this location should regress?

- FCOS可以利用尽可能多的样本（特征图上的点形成的框）而不只是IoU足够大的anchor 来进行训练。每个点都去学习对框的预测，多个点共同产生多个类似的框，然后NMS选择最大的
- 对于ambiguous samples，采用多尺度特征图，每层限制4D vec中最大值的大小（**限制每层特征图产生的bbox的大小**），满足$m_i < \max(l,t,r,b) < m_{i+1}$。对于同一个点上多个框，因限制，所以在不同尺度特征图上构成的框进行regress，一个feature map上一个点只负责固定尺度的框回归。如果还出现重复，则选择尺寸最小
- 防止远离物体中心的点产生质量差的框，center-loss。$centerness^* = \sqrt{\frac{\min(l^*,r^*)}{\max(l^*,r^*)} \times \frac{\min(t^*,b^*)}{\max(t^*,b^*)}}$

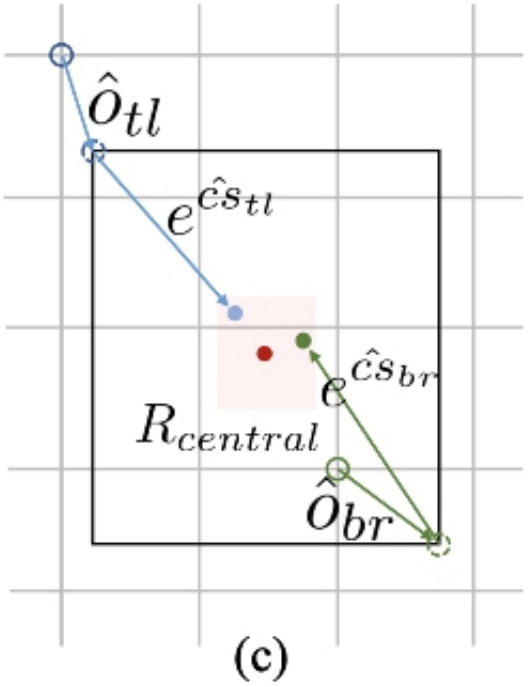
使**左和右，上和下的长度尽可能相等**。测试时centerness-weighted classification confidence，抑制偏远框 👇



网络结构 👇



CentripetalNet: Pursuing High-quality Keypoint Pairs for Object Detection



(c)

SaccadeNet: A Fast and Accurate Object Detector

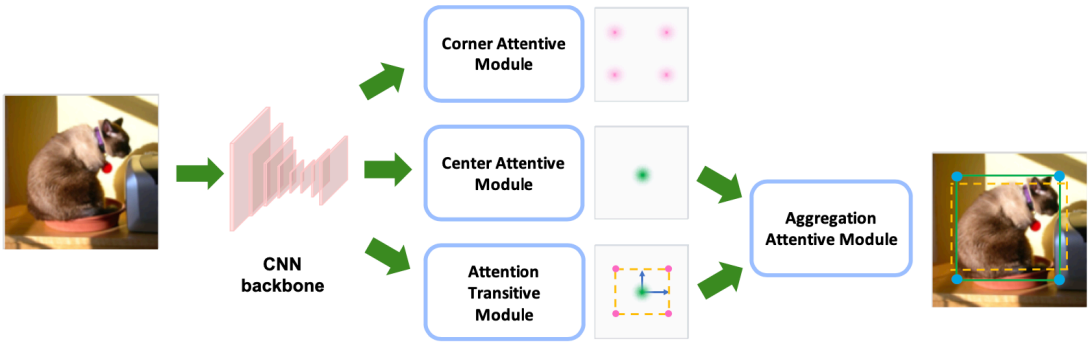


Figure 2. In SaccadeNet, we utilize 5 keypoints as informative parts for detection: the object center and 4 bounding box corners. After the CNN backbone, as in the middle branch, the Center Attentive Module focuses at predicting the object center keypoint; then the Attention Transitive Module in the bottom switches the attention from object center to estimate rough location of object corners. After that, the Aggregation Attentive Module uses information aggregated from both center and corner keypoints, and predicts a refined location of objects. Moreover, in order to obtain informative corner features, the Corner Attentive Module is used (in training only) to enforce the CNN backbone to pay more attention to object boundaries, as shown in the top branch.

同时预测中心 `Center Attentive Module` 和角点 `Attention Transitive Module`，得到粗框，使用 `Aggregation Attentive Module` 双线性插值重新采样feature map，得到精细框，轻量级边框细化。 `Corner Attentive Module` 辅助训练。

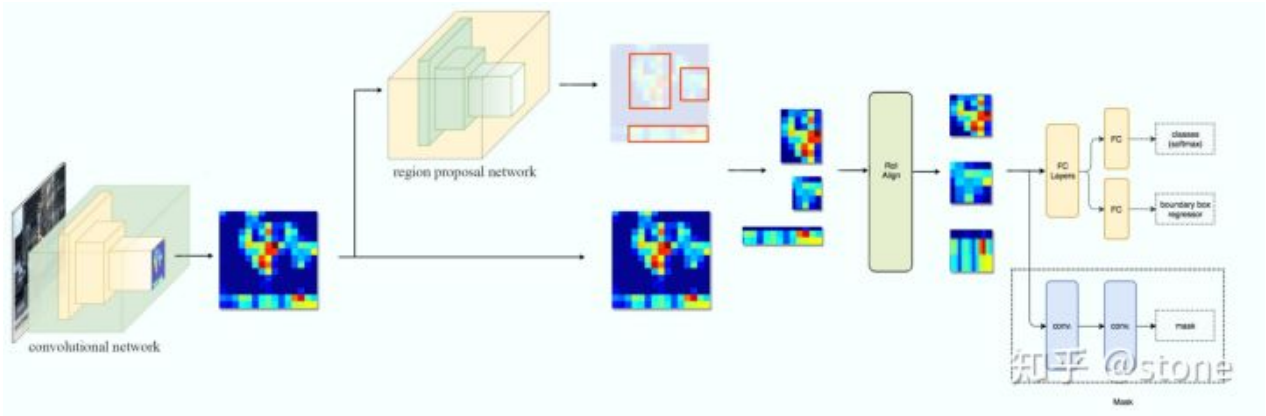
`Center-AM` 距离惩罚训练，采用Gaussian heat map作为GT $e^{\frac{\|X-X_k\|^2}{2\sigma^2}}$

相比CornerNet增加了中心特征，相比FCOS增加了边缘特征，相比CenterNet加速

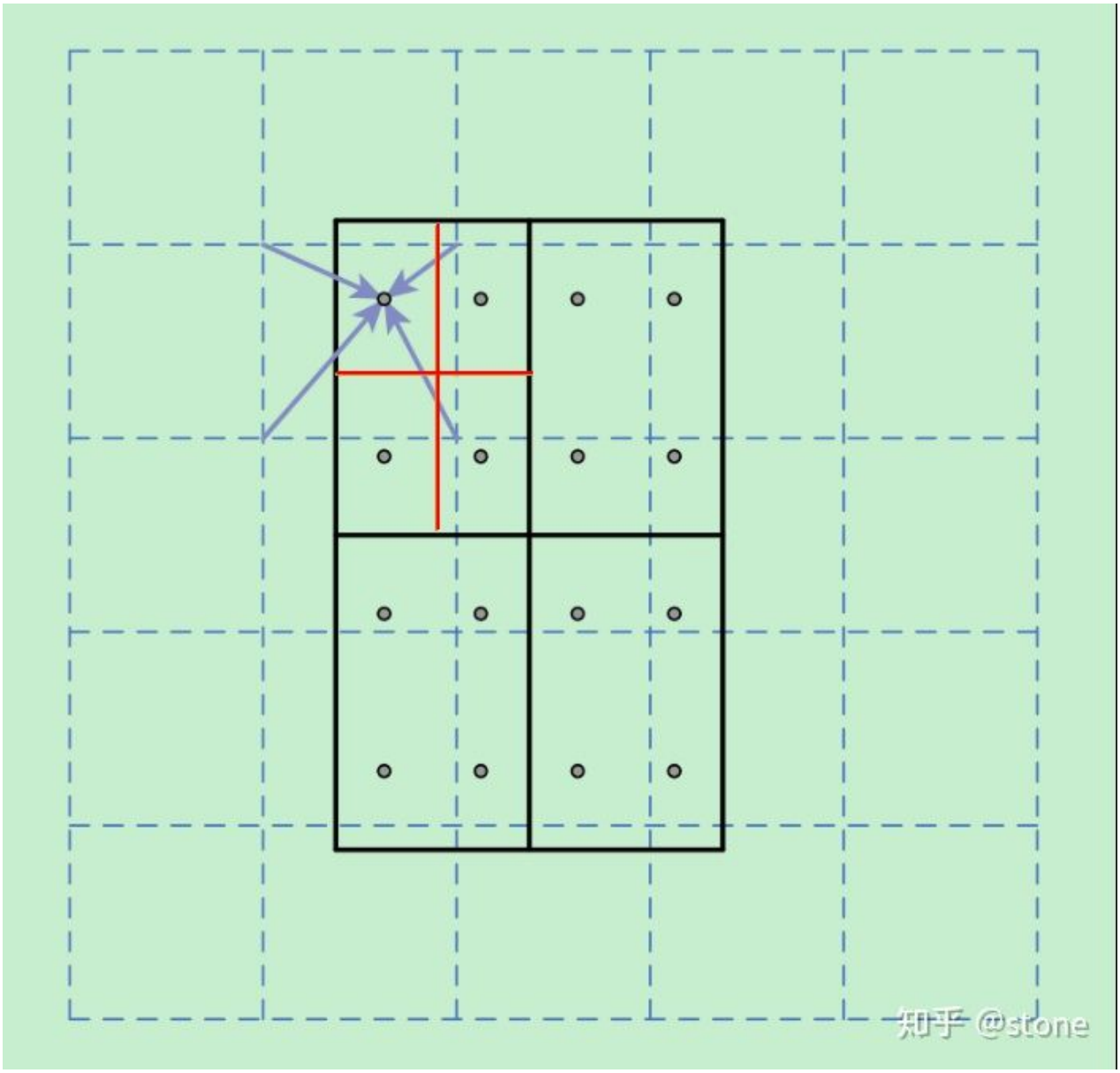
Mask RCNN

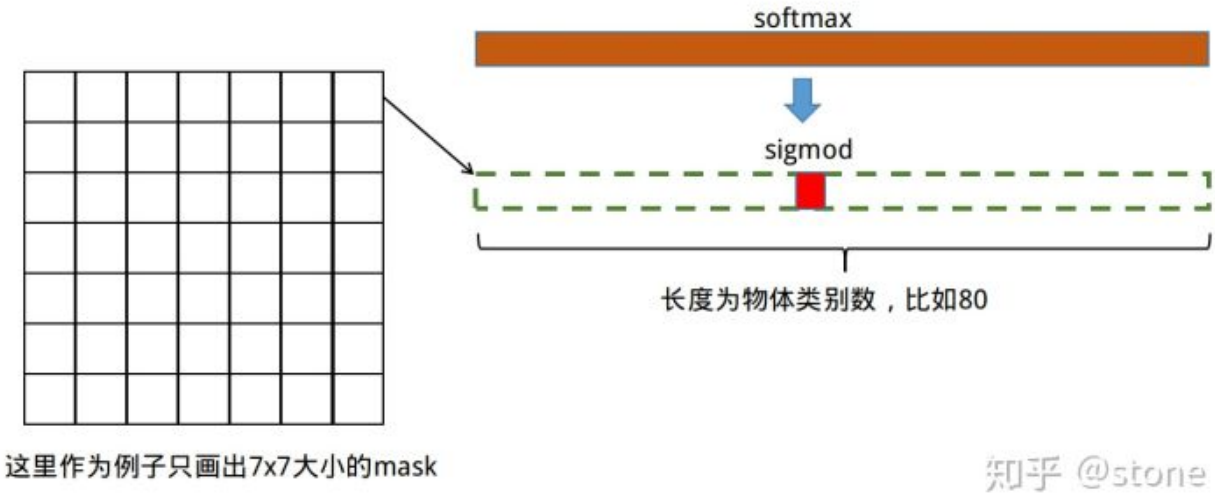
参考 https://zhuanlan.zhihu.com/p/37998710

网络结构



RoI Align





👆loss计算时 `w*h*c` 的mask输出，只计算分类分支预测的类别对应channel的sigmod输出作为损失「**语义mask预测与分类预测解耦**」

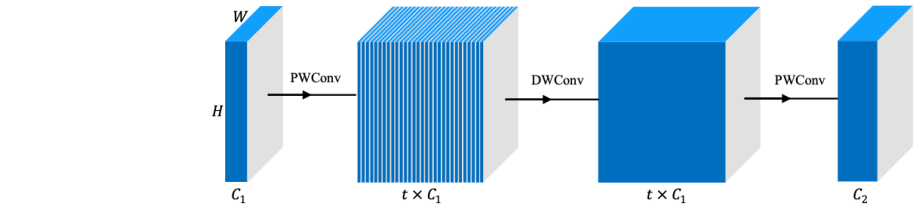
HBONet: Harmonious Bottleneck on Two Orthogonal Dimensions

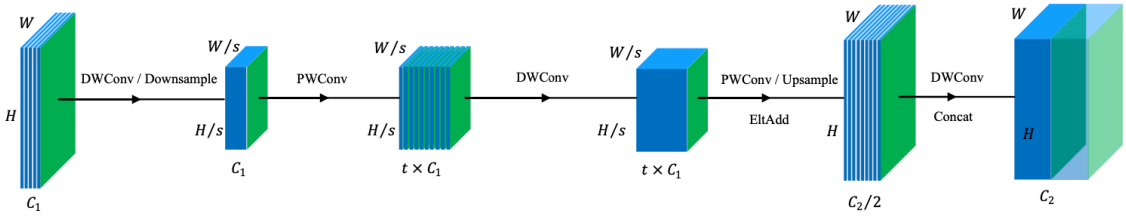
> light-weight

包含两个部分 *spatial contraction-expansion* 和 *channel expansion-contraction* ，独立作用在特征图的orthogonal dimension。前者通过减少特征图大小减少计算量，后者通过提升informative feature提升性能

> mobilenet通过分离成 *point-wise* 和 *depth-wise* 来分别不变尺寸变通道数和不变通道数变尺寸(up/down sampling) so called *depthwise separable conv*
> shufflenet通过group conv减少通道上的计算量，channel-shuffle来增加不同通道之间的连接
> 👆previous work focus on **channel transformation**, introduce **spatial feature dim(size)**

- 提升通道数可以提升信息，但是增加计算开销. 提出 *Two reciprocal components work on orthogonal dimension* .**通道数扩展时，尺寸减少**
- 相比mobilenet增加**spatial contraction缩小-expansion放大到输入的大小**，类似Squeeze-Excitation Network的先缩后放的思路 模块，对比mobilenet 👇



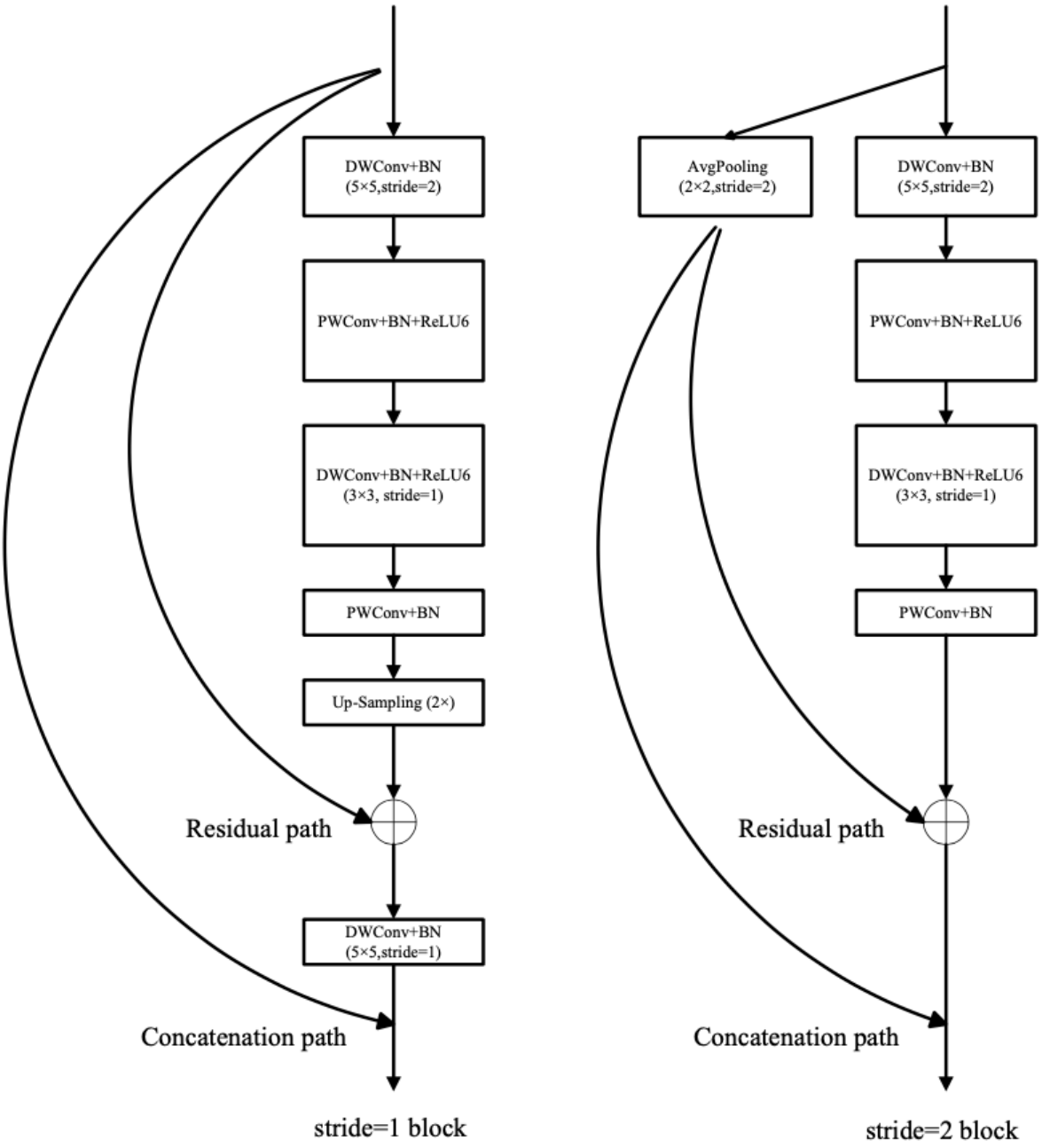


计算量 👇

$$B/s^2 + (h/s \times w/s \times c_1 + h \times w \times c_2) \times k^2, \qquad (2)$$

where B denotes the original computational cost of the blocks inserted between the spatial contraction and expansion operations. Spatial contraction-expansion component,

- 增加residual path，减少主干计算and **feature reuse**. Inverted residual with harmonious bottleneck 👇



- For Object detection: use **MobileNet V2 SSD** utilize the **warm-up** strategy which linearly ramps up the learning rate from a close-to-zero one 1e-6 to the normal initial learning rate of 1e-3 during the first 5 epochs.

SSD网络时间

1. 网络运行时间：0.002-0.003s 在GPU运行
2. detect（NMS为主）运行时间：0.013-0.016s 只在 *CPU* 上运行 **瓶颈**

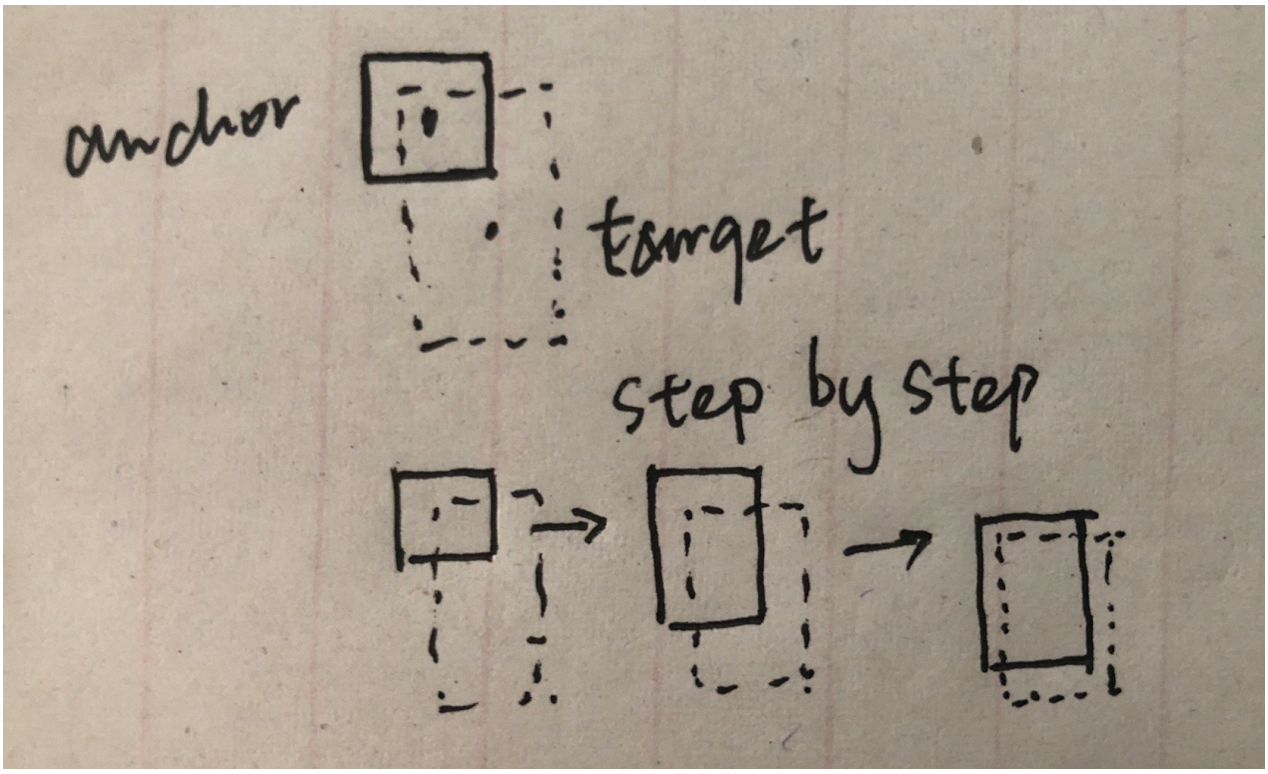
尺度减少，aspect ratio减少 Shallow feature map only for small Deep ONLY large 低秩简化 PointRend

特征融合 cascade

Cascade RPN

Single anchor per location + multi-stage refinement

> 一次回归不到（距离太远），多次回归
>
> 回归多次后anchor点处的特征和移动anchor所在位置特征不匹配→deformable conv
>
> 匹配的anchor位置不变(还是最初始点对应的anchor)，但是提取特征的位置改变👇👇



Motivation

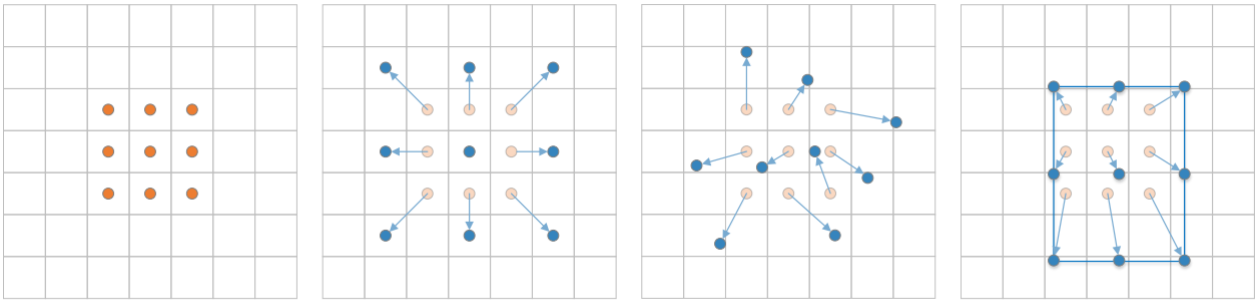
predefined anchor在GT和anchor对齐时限制性能/偏差，(#toread RoIPool RoIAlign)

1. **single anchor** + incorporates **criteria** of anchor/anchor-free in defining **positive** boxes
2. **adaptive convolution** to maintain the alignment between anchor boxes and features

Iterative RPN每次把anchor集合**看作新的anchor**进行refine，导致每次迭代后anchor位置和形状发生变化，anchor和表示anchor特征不匹配「**anchor中心点的特征(即表示anchor的特征)不发生变化，但是anchor的位置发生变化**，mismatch」👆使用deformable conv解决，but *no constraint to enforce* 🤷‍♂️

Adaptive Convolution

卷积采样的时候增加offset field `offset = center offset + shape offset` 中心的偏移和形状偏移(由anchor形状和kernel决定)



Convolution　Dilated Convolution　Deformable convolution　Adaptive convolution

👆对比deformable conv：偏移量由anchor和kernel决定，非网络学习➡️anchor和feature对齐

Sample Discrimination Metrics

每个位置只有一个anchor，然后迭代refine

> Determining whether a training sample is pos/neg as the use of anchor/anchor-free is adversarial 两种方法决定正负样本的方法不同

👆即anchor-free的决定方式宽松/数量多，anchor-based标准严格/数量少 Stage 1: anchor-free➡️更多正样本「解决正负样本不匹配」 Stage 2: amchor-based➡️严格，数量减少，IoU高 `anchor-free` 指FCOS，中心点在物体内为pos anchor `anchor-based` 指Faster RCNN，IoU threshold

Cascade RPN

前一个阶段的输出bridge到后一个阶段 由anchor计算出offset `o` ，再和feature `x` 输入regressor计算新的anchor (σ就是anchor回归的目标 eg. $(tx - ax)/aw$)

Algorithm 1. Cascade RPN

Input: sequence of regressors f^τ, classifier g, feature $\boldsymbol{x}$ of image I.
Output: proposal set $\mathbb{P}$.
Uniformly initialize anchor set $\mathbb{A}^1 = \{\boldsymbol{a}^1\}$ over image I.
for $\tau \leftarrow 1$ **to** T **do**
    Compute offset $\boldsymbol{o}^\tau$ of input anchor $\boldsymbol{a}^\tau$ on feature map using (7).
    Compute regression prediction $\hat{\boldsymbol{\delta}}^\tau = f^\tau(\boldsymbol{x}, \boldsymbol{o}^\tau)$.
    Compute regressed anchor $\boldsymbol{a}^{\tau+1}$ from $\hat{\boldsymbol{\delta}}^\tau$ using (3).
end
Compute objectness score $\boldsymbol{s} = g(\boldsymbol{x}, \boldsymbol{o}^T)$.
Derive proposals $\mathbb{P}$ from $\mathbb{A}^{\tau+1} = \{\boldsymbol{a}^{\tau+1}\}$ and $\mathbb{S} = \{\boldsymbol{s}\}$ using NMS (4).

DetNet

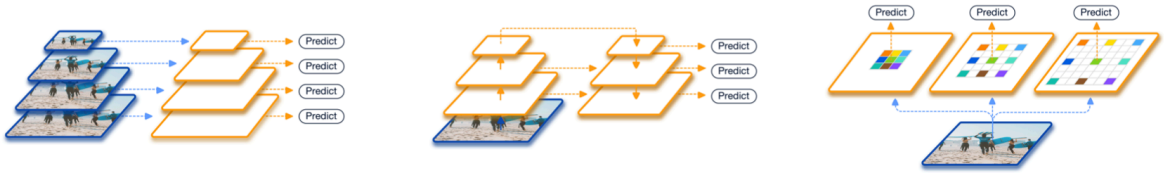
> 为检测任务设计backbone

现有的ImageNet backbone： 1. 网络stage需要增加，且未在imagenet训过 2. down-sample和stride损失空间信息，大目标边界模糊 3. 小目标「空间分辨率低」

TridentNet

> Scale variation → Different **receptive fields**

多分支网络，分支结构相同权重共享，每个分支不同的感受野对应检测不同尺度范围的物体 不同感受野使用 `dilated_conv` 实现👉参数相同 权重共享：减少参数量，inference时只选择一个主分支



(a) Image Pyramid (b) Feature Pyramid (c) Trident Network

Figure 1: (a) Using multiple images of several scales as input, the image pyramid methods perform feature extraction and object detection independently for each scale. (b) The feature pyramid methods utilize the features from different layers of CNNs for different scales, which is computational friendly. This figure takes FPN [25] as an example. (c) Our proposed Trident Network generates scale-aware feature maps efficiently by trident blocks with different receptive fields.

Image Pyramid (Multi-scale training&testing): time-consuming Feature Pyramid: use different params to predict different scale (not uniform)

trident block 👇

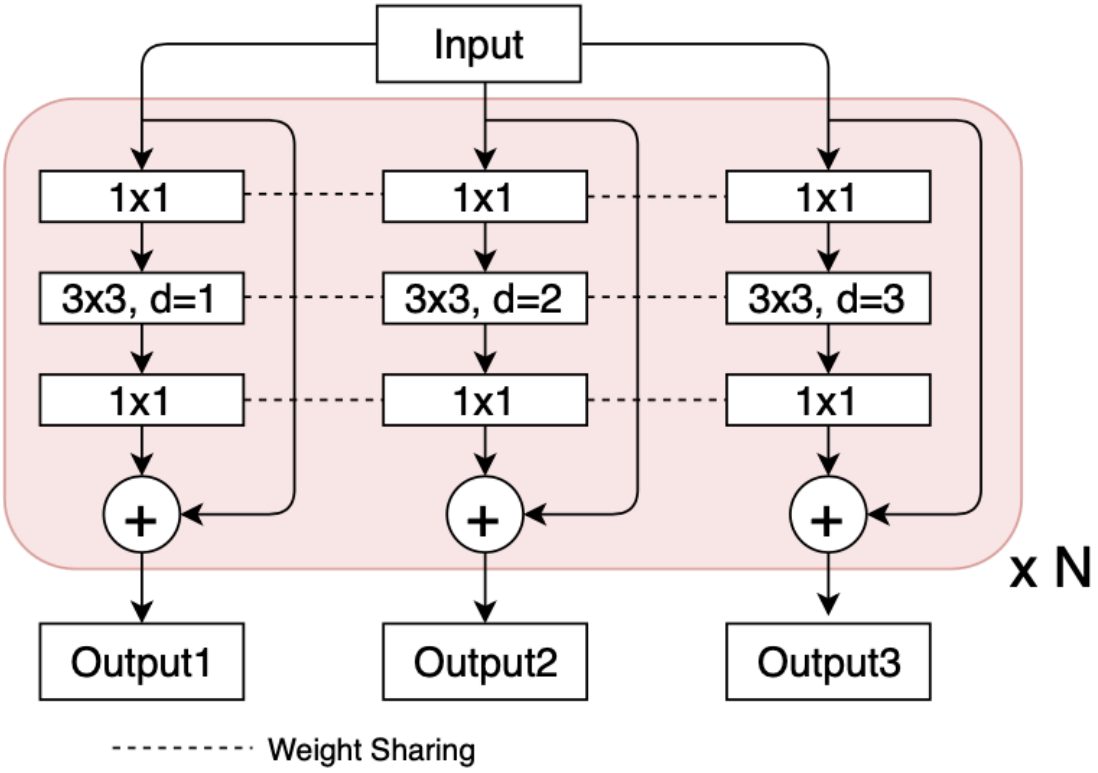


Figure 3: A trident block constructed from a bottleneck residual block.

Scale-aware Training Scheme：每个branch只对长宽在一定范围的proposal进行训练「一张图片使用不同branch(不同dilate rate)训练不同尺度的proposal」 其他参数相同(make sense?)

预测：计算每个分支的预测输出，filter out掉超过尺寸范围的box **TridentNet Fast**：预测只采用单分支→中间分支预测，得益于三分支权重共享，效果接近

SNIPER: Efficient Multi-Scale Training

> 解决多尺度问题, 不构建feature pyramid, **多尺度训练策略**, 尺寸适应网络

Scale Invariant: *RCNN* 将proposal缩放到同一个尺度，检测网络只需要学习一种尺度的检测。而为了适应不同尺度，多尺度训练的 *Faster RCNN* 对整个图片进行放缩，proposal也放大缩小，检测网络学习适应多种尺度。 通过网络capacity记忆不同scale的物体 ，浪费capacity

> *Process context regions around GT instances(chips) at appropriate scale*
> 截取固定尺寸的chip(eg `3x3, 5x5, 7x7`)对应不同尺度，然后resize到相同大小(low-res)去训练
> 小目标zoom-in，大目标zoom-out

Pos chips

Figure 1: SNIPER Positive chip selection . SNIPER adaptively samples context regions (aka chips) based on the presence of objects inside the image. Left side: The image, ground-truth boxes (represented by green lines), and the chips in the original image scale (represented by the blue, yellow, pink, and purple rectangles). Right side: Down/up-sampling is performed considering the size of the objects. Covered objects are shown in green and invalid objects in the corresponding scale are shown as red rectangles.

👆chip从最小的cover某个GT box开始，直到最多的box被这个chip cover到 「chip尺寸不变，围绕cover这个GTbox转，直到最大化cover的box数量」

1. 每个box至少被一个chip cover
2. 一个物体可能被多个chip cover
3. 一个物体在不同尺度chip中可能valid or not
4. 截断的物体保留

Neg chips

Figure 2: SNIPER negative chip selection. First row: the image and the ground-truth boxes. Bottom row: negative proposals not covered in positive chips (represented by red circles located at the center of each proposal for the clarity) and the generated negative chips based on the proposals (represented by orange rectangles).

👆只有pos chips会导致网络只对GT附近小范围的图片训练 *iconic* ，缺乏 背景 。增加**难样本**作为neg chips Metrics:

1. 如果区域没有proposal，认为是easy background，忽略
2. 去掉被pos chip cover的proposal「proposal和GT接近，易于区分」
3. 贪心选择至少cover M个剩余proposal的作为neg chips

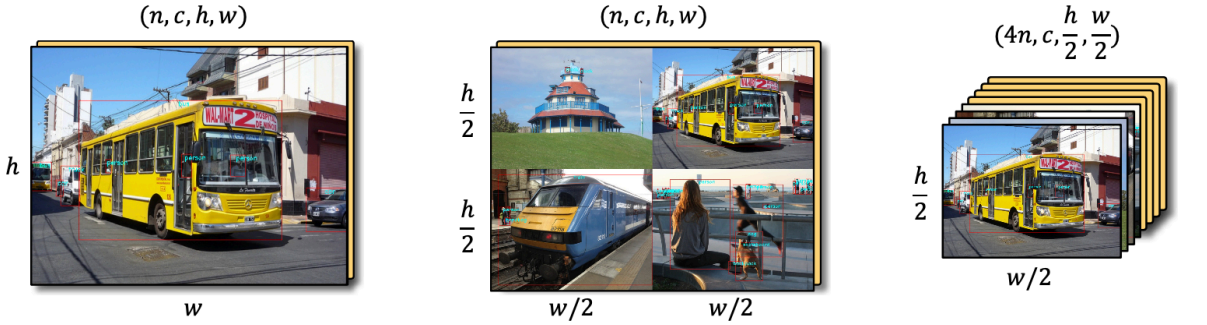
训练时可控制neg chip数量，类似OHEM 分辨率和准确率关系可能不大，过多context可能不必要 Ref: 目标检测-SNIPER-Efficient Multi-Scale Training-论文笔记 | arleyzhang

Stitcher: Feedback-driven Data Provider for Object Detection

> 小目标，粘贴构造训练样本

小目标数据集中分布不均匀(41.4%的小目标只出现在52.3%的图片中)，小目标在训练过程中贡献的loss低，学不好

把图片缩小，拼接在一起（和SNIPER切割相反）

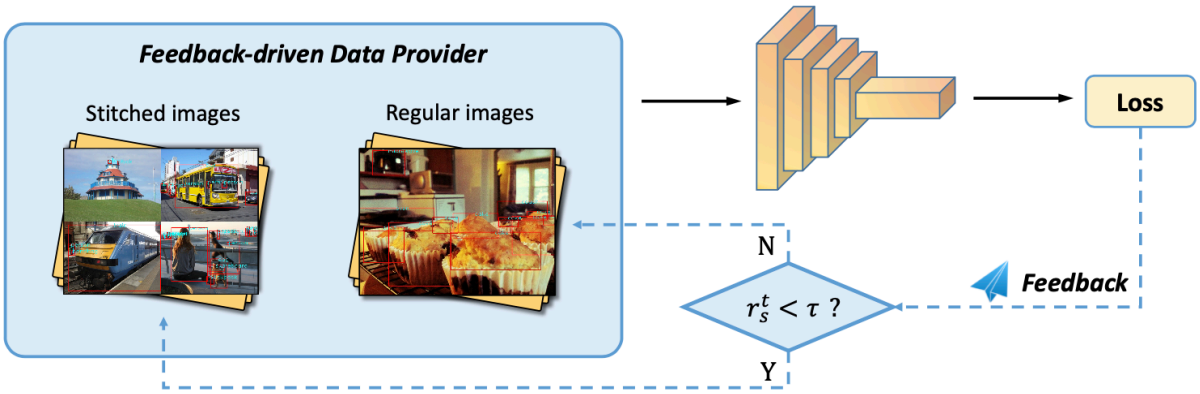


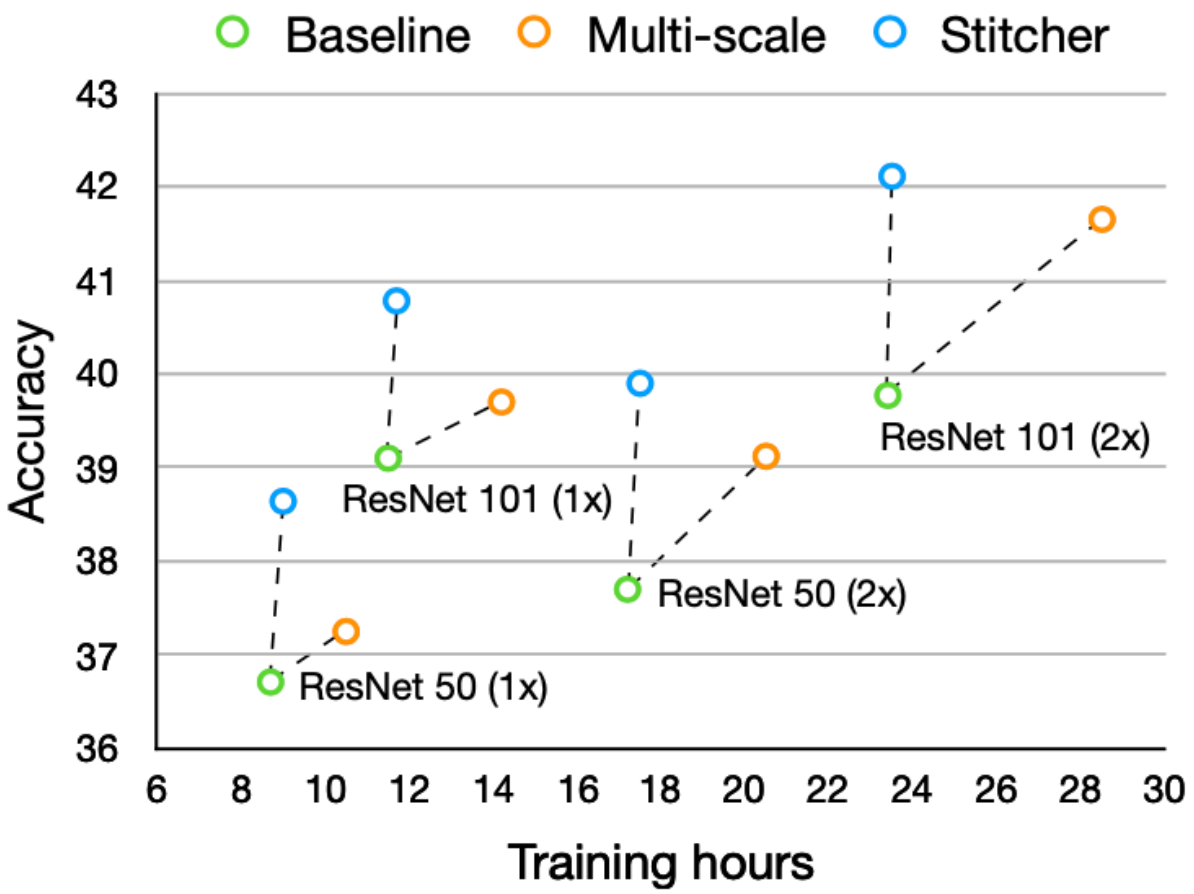
(a) Regular Images. (b) Stitch in spatial dimension. (c) Stitch in batch dimension.

把大物体和中物体都变成小物体，增加小尺度的分布

小目标：检测时放大❌，训练时缩小✅

loss作为反馈信号，**小目标产生loss不足($r_s^t < \tau$)则下个iter采用stitch，缺啥补啥**



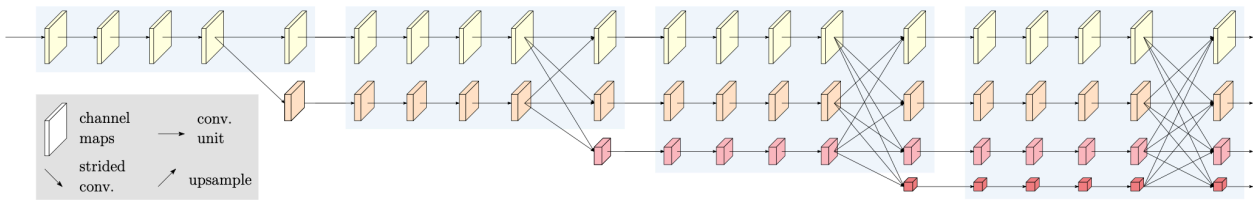


HRNet: Deep High-Resolution Representation Learning for Visual Recognition

> 处理过程中保持高分辨率「position-sensitive task」
> maintain high-res representation through the whole process
> 不同于skip connection：高分辨分支平行conv，通过fusion而不是add融合高低分支，多分辨率输出
> 不同于特征金字塔：高低分辨率平行计算（low-res增加分辨率下conv计算，不是通过high-res一次卷积downsample得到，**逐步平行**计算增加）

先前网络：encode high → low，recover low → high 提出网络：运算时**保持高分辨率**分支，**平行**的加入低分辨率分支；**multi-res fusion**

Parallel multi-res conv



👆每个stage 逐步加入一个低分辨率(eg 1/2) 分支，且保持原有分辨率分支 类似 group conv，通道分别 → 分辨率分别

Repeated multi-res fusion

每个stage(4个unit/block)交换不同分辨率的信息

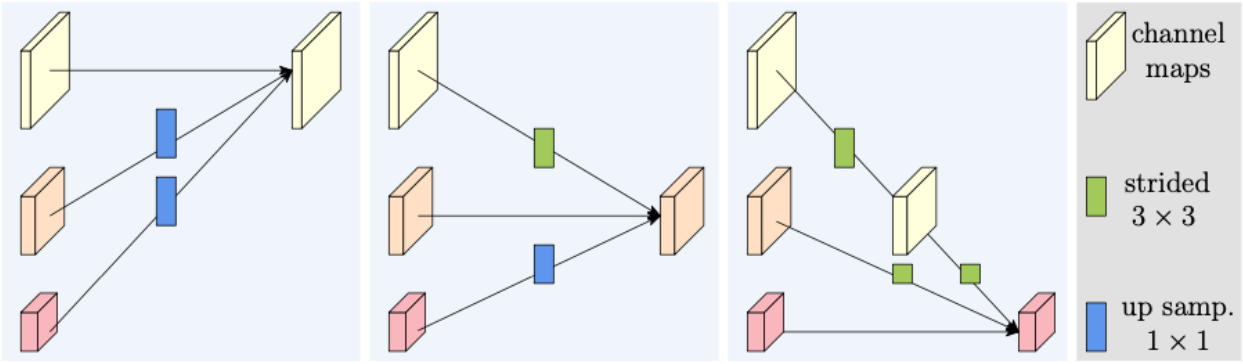
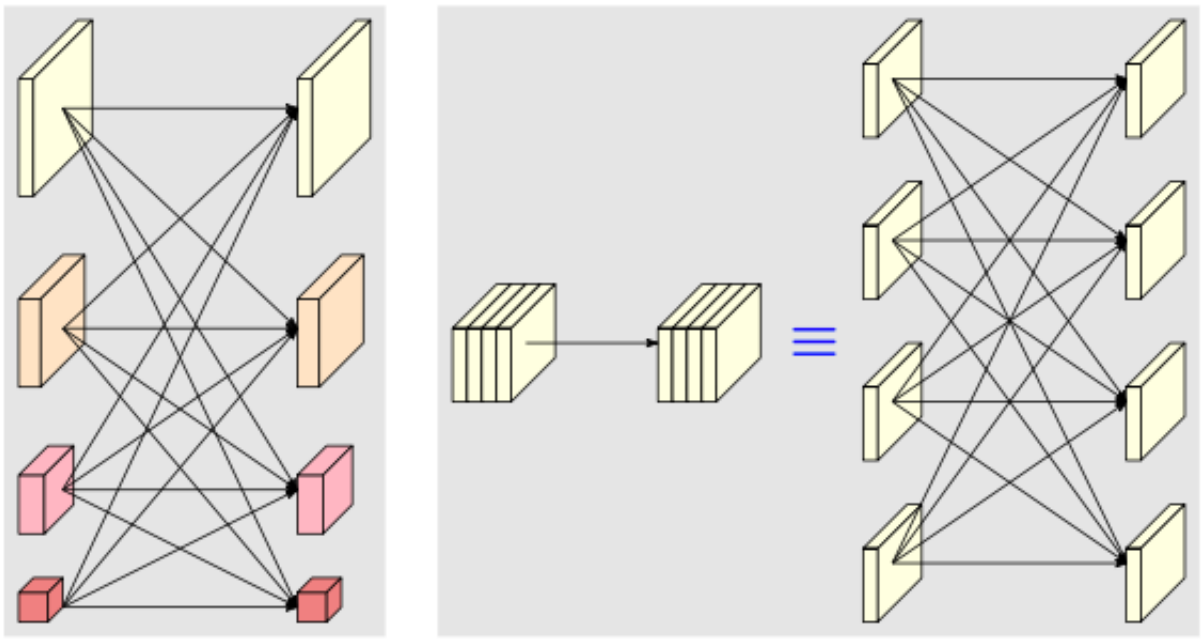


Fig. 3. Illustrating how the fusion module aggregates the information for high, medium and low resolutions from left to right, respectively. Right legend: strided 3×3 = stride-2 3×3 convolution, up samp. 1×1 = bilinear upsampling followed by a 1×1 convolution.

👆high $\rightarrow$ low: stride conv; low $\rightarrow$ high: bilinear upsampling + 1x1 conv an extra output for lower res output



👆融合类似FC

Multi-res representation head/不同任务不同输出模式

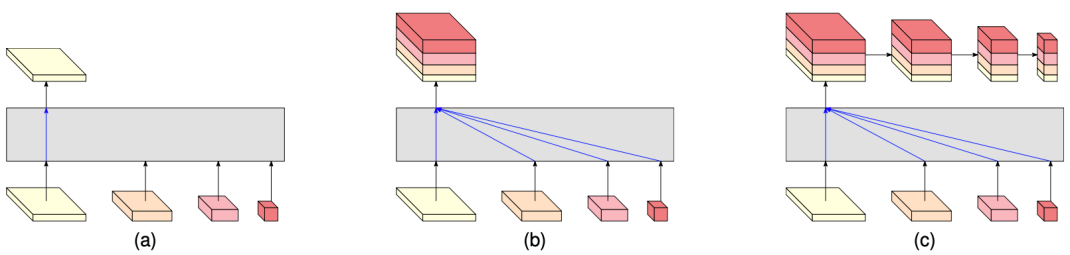


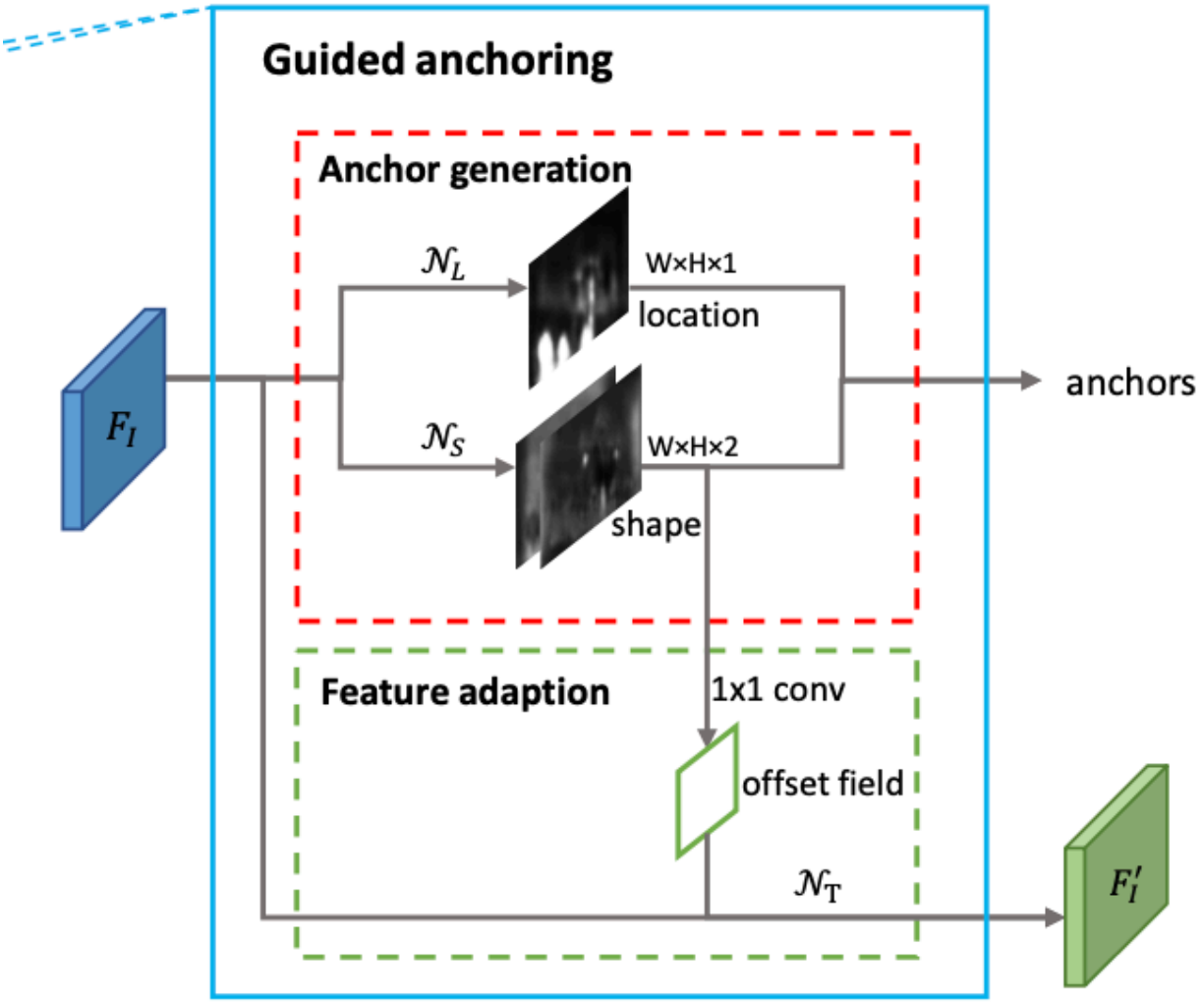
Fig. 4. (a) HRNetV1: only the representation output from the high-resolution convolution stream. (b) HRNetV2: Concatenate the (upsampled) representations that are from all the resolutions (the subsequent 1×1 convolution is not shown for clarity). (c) HRNetV2p: a feature pyramid formed from the representation by HRNetV2. The four-resolution representations at the bottom in each sub-figure are outputted from the network in Figure 2, and the gray box indicates how the output representation is obtained from the input four-resolution representations.

(a) 关键点检测 (b) segmentation (c) object detection

Region Proposal by Guided Anchoring

> 更好的anchor， **改进产生anchor的过程**「非密铺」
> anchor与feature: **alignment** + **consistency**

两个分支分别对anchor的中心点和长宽进行预测，防止offset偏移过大，anchor和点的feature不对应 采用**deformable conv**使feature的范围和anchor的形状对应，每个位置anchor形状不同而capture不同的特征「加offset以适应anchor形状」



Guided Anchoring

$p(x,y,w,h|I) = p(x,y|I)p(w,h|x,y,I)$ **分两步产生anchor**「减小同时预测xywh时出现的偏移不对应」

1. location：预测objectness，之后采用mask_conv **减少区域计算** 只对物体的中心(及附近)为pos训练，预测物体中心「边缘不容易回归框」
2. shape：预测每个位置上的best shape，位置不变只变长宽，不会misalign 预测$w = k \times e^{dw}$，预测dw，而不是w，范围更大 👇 $w = \sigma \cdot s \cdot e^{dw}$，$h = \sigma \cdot s \cdot e^{dh}$

选择高于thresh的location中，概率最高的shape，产生anchor

Feature adaptation

consistency: 每点对应的anchor长宽不同，所以学习到特征对应区域的长宽也应该不同
$\mathbf{f}_i' = \mathcal{N}_T(\mathbf{f}_i, w_i, h_i)$ 基于对应anchor的长宽，改变特征（xy不变，位置branch只预测objectness score）
👆使用deformable convolution实现

Anchor shape target

训练时anchor和gt box的匹配，训练目标。wh为变量，无法计算IoU

$$\mathbf{vIoU}(a_{\mathbf{wh}}, \mathbf{gt}) = \max_{w>0,h>0} \mathbf{IoU}_{normal}(a_{\mathbf{wh}}, \mathbf{gt})$$

方法：Sample常见的wh组合，计算和GT的IoU，得到vIoU👆，作为anchor和gt IoU的估计，之后采用常见anchor分配方法确定训练目标

High quality proposal

由于生成的anchor更好，pos样本数量更多。训练样本分布符合proposal分布 设置 更高正负样本比例，同时 更少样本 数量，即 更高IoU threshold

---

Soft NMS

> 解决密集 *相邻* 物体的检测框重叠IoU大，可能在NMS过程中 误删
> 密集物体检测有提升

NMS

按照置信度排序，选择最大的box i保留。其余box中，与I的IoU>threshold的删除(置信度置为0)。再从剩下box选择最大保留，重复 $s_i = \begin{cases} s_i, & \text{iou}(\mathcal{M}, b_i) < N_t \\ 0, & \text{iou}(\mathcal{M}, b_i) \geq N_t \end{cases}$,

Soft NMS

重叠IoU越大，置信度下降越多 置信度置为0变为更新IoU>threshold框的置信度

$$s_i = \begin{cases} s_i, & \text{iou}(\mathcal{M}, b_i) < N_t \\ s_i(1 - \text{iou}(\mathcal{M}, b_i)), & \text{iou}(\mathcal{M}, b_i) \geq N_t \end{cases},$$ 或

$$s_i = s_i e^{-\frac{\text{iou}(\mathcal{M}, b_i)^2}{\sigma}}, \forall b_i \notin \mathcal{D}$$

Ref: NMS与soft NMS - 知乎

Adaptive NMS Refining Pedestrian Detection in a Crowd

> 密集场景下NMS误删
>
> 通过预测crowd程度动态选择threshold

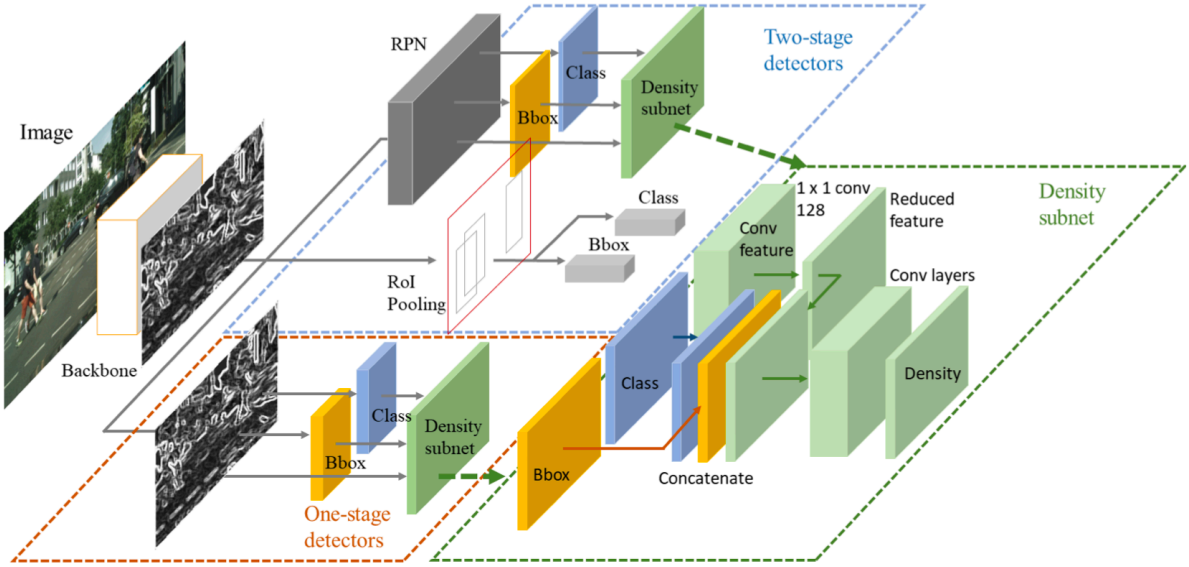
密集位置提高IoU阈值保留临近框，稀疏位置降低IoU阈值删除冗余框

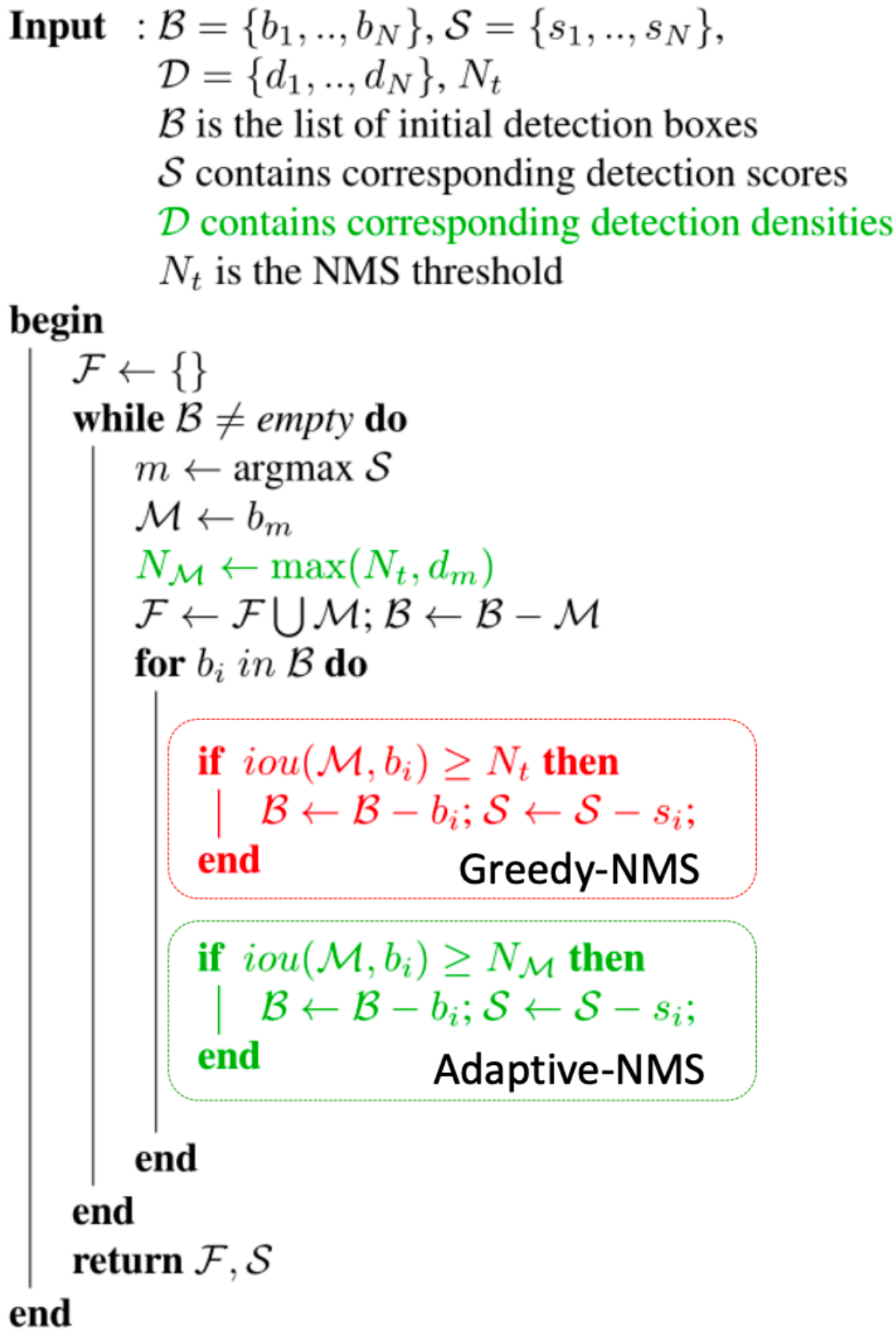
对于**每个物体**定义object density $d_i := \max_{b_j \in \mathcal{G}, i \neq j} \text{iou}(b_i, b_j)$

阈值计算过程 $N_{\mathcal{M}} := \max(N_t, d_{\mathcal{M}})$

NMS过程 $s_i = \begin{cases} s_i, & \text{iou}(\mathcal{M}, b_i) < N_{\mathcal{M}} \\ s_i f(\text{iou}(\mathcal{M}, b_i)), & \text{iou } (\mathcal{M}, b_i) \geq N_{\mathcal{M}} \end{cases}$

测试时density通过网络 `density subnet` 预测，objectness map + bbox预测 `concat` 作为输入， `5x5` 卷积(临近物体的信息)

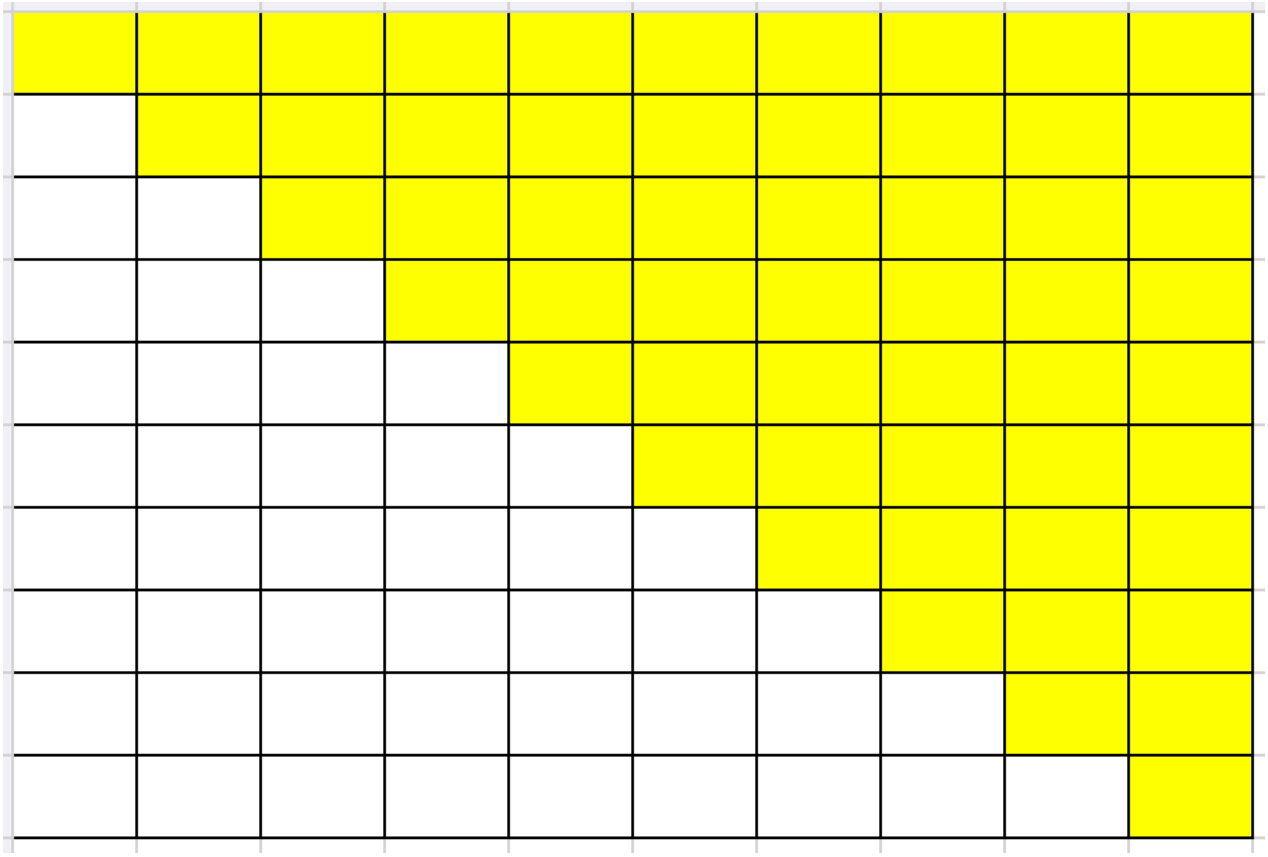




在cityperson和crowdhuman密集数据集效果好

Ref: https://www.starlg.cn/2019/05/20/Adaptive-NMS/

Fast NMS



按照conf顺序构建IoU矩阵，转为上三角。对B_i，如果$\exists x_{i,j} > \epsilon$，则去掉$B_i$，速度快

问题：没有去掉B_i时把之后的x_i.失效，「横向传播」，可能多删除框。B_i被删除后，之后的框和B_i的IoU仍被考虑计算

Cluster NMS

Enhancing Geometric Factors in Model Learning and Inference for Object Detection and Instance Segmentation

Cluster-NMS

改进 `Fast NMS`，增加remove row B_i的操作

Initialize $T = N$, $t = 1$ and $\boldsymbol{b}^0 = \mathbf{1}$
Compute IoU matrix $\boldsymbol{X} = \{x_{ij}\}_{N \times N}$ with $x_{ij} = IoU(\mathcal{B}_i, \mathcal{B}_j)$.
$\boldsymbol{X} = \mathtt{triu}(\boldsymbol{X})$ ▷ Upper triangular matrix with $x_{ii} = 0, \forall i$
while $t \leq T$ **do**
    $\boldsymbol{A}^t = \mathtt{diag}(\boldsymbol{b}^{t-1})$
    $\boldsymbol{C}^t = \boldsymbol{A}^t \times \boldsymbol{X}$
    $\boldsymbol{g} \leftarrow \max_j \boldsymbol{C}^t$ ▷ Find maximum for each column j
    $\boldsymbol{b}^t \leftarrow \mathtt{find}(\boldsymbol{g} < \varepsilon)$ ▷ $\begin{cases} b_j = 1, & \text{if } g_j < \varepsilon \\ b_j = 0, & \text{if } g_j \geq \varepsilon \end{cases}$
    if $\boldsymbol{b}^t == \boldsymbol{b}^{t-1}$ **then**
        $t^* = t$, break
    end if
    $t = t + 1$
end while
return $\boldsymbol{b}^{t^*}$

---

b^{t-1}表示$t-1$ iter的NMS indicator，t次iter时对C^t进行NMS

$A = diag(b)$表示根据上次NMS结果(indicator)，对已经被supressed的框去掉（i行置0，不考虑 $x_{i\,\cdot}$ ，和i框的IoU不计算）

$T = 1$同 `Fast NMS` ，$T = N$同 `NMS`

没有重合的box可以分成多个**cluster**并行处理

Score penalty reweigh + Cluster NMS

类似 `Soft NMS` 中不是直接去除box 「hard」，变成对score进行reweight「soft」，构成 `Cluster-NMS_S`

$$s_j = s_j \prod_i e^{-\frac{(\boldsymbol{A}\times\boldsymbol{X})_{ij}^2}{\sigma}}$$

j和其他box的IoU越大，score降低越多

不同于 `Soft NMS` ，只会被 和更高conf的box有大IoU 而受到惩罚，由于是上三角，只计算和更靠前box的IoU

Normalized central point distance + Cluster NMS

增加同 `DIoU` 类似的中心点距离D，构成 `Cluster-NMS_S+D`

$$s_j = s_j \prod_i \min\{e^{-\frac{(\boldsymbol{A}\times\boldsymbol{X})_{ij}^2}{\sigma}} + D^\beta, 1\}$$

Weighted NMS + Cluster NMS

`Weighted NMS` 根据IoU和conf加权 `merge` 重叠框，输出**全新的框**「速度慢」

$\mathcal{B} = \frac{1}{\sum_j w_j} \sum_{\mathcal{B}_j \in \Lambda} w_j \mathcal{B}_j$

$\mathcal{B}$是加权融合后的全新的框，$\Lambda = \{\mathcal{B}_j | x_{ij} \geq \varepsilon, \forall i\}$为重叠框，权重$w_j = s_j IoU\left(\mathcal{B}, \mathcal{B}_j\right)$， **weighted combination**

conf从高到低，找到IoU>threshold的框，根据IoU进行加权求和，得到融合框；再对其他IoU<threshold的框计算（https://github.com/sanch7/Weighted-NMS/blob/master/weighted_nms.py）

`Cluster-NMS_W`：

$\mathcal{C}' = s \otimes \mathcal{C}$

$\mathcal{B} = \mathcal{C}' \times \mathcal{B}$

TABLE VI
COMPARISON OF DIFFERENT NMS METHODS ON PRE-TRAINED YOLO V3 MODEL. THE RESULTS ARE REPORTED ON MS COCO 2014 VALIDATION SET.

Method	NMS Strategy	FPS	Time	AP	AP_{50}	AP_{75}	AP_S	AP_M	AP_L	AR_1	AR_{10}	AR_{100}	AR_S	AR_M	AR_L
YOLO v3 [8]	Fast NMS	85.5	11.7	42.2	62.2	45.1	24.4	47.2	53.4	34.8	56.0	60.1	43.0	65.1	73.4
	Original NMS	14.6	68.5	42.6	62.5	45.8	24.7	47.8	53.7	34.8	57.2	62.5	45.0	67.8	75.6
	Original NMS (TorchVision)	**95.2**	**10.5**	42.6	62.5	45.8	24.7	47.8	53.7	34.8	57.2	62.5	45.0	67.9	75.6
	Weighted-NMS	11.2	89.6	42.9	**62.7**	46.4	25.2	48.2	53.9	**35.0**	57.4	62.7	45.4	68.3	75.6
	Cluster-NMS	82.6	12.1	42.6	62.5	45.8	24.7	47.8	53.7	34.8	57.2	62.5	45.0	67.9	75.6
	Cluster-NMS_W	68.0	14.7	42.9	**62.7**	46.4	25.2	48.2	53.9	**35.0**	57.4	62.7	45.4	68.3	75.6
	Cluster-NMS_{W+D}	64.5	15.5	**43.1**	62.4	**46.8**	**25.3**	**48.3**	**54.1**	**35.0**	**58.0**	**63.7**	**46.4**	**69.3**	**76.7**

Feature NMS: NMS by Learning Feature Embeddings

密集重叠场景下，只通过IoU不能判断是否是对同一个物体的预测。**增加feature vec 距离判断是否是同一个物体的预测**，距离小删除

Algorithm 2 Proposed Non-Maximum Suppression. If the calculated value of the intersection over union is in a range that does not allow to make a definite decision we use a feature embedding similarity.

$\mathcal{P} \leftarrow$ GETPROPOSALS($image$)
$\mathcal{P} \leftarrow$ SORT($\mathcal{P}$)
$\mathcal{D} \leftarrow \emptyset$
while $\mathcal{P} \neq \emptyset$ **do**
  $p \leftarrow$ POP($\mathcal{P}$)
  for $d \in \mathcal{D}$ **do**
    $iou \leftarrow$ GETIOU(p, d)
    if $iou \leq N_1$ **then**
      PUSH($p, \mathcal{D}$)
    else if $iou < N_2$ **then**
      $embeddingDistance \leftarrow$ GETEMBEDDINGDISTANCE(p, d)
      if $embeddingDistance > T$ **then**
        PUSH($p, \mathcal{D}$)
      end if
    end if
  end for
end while

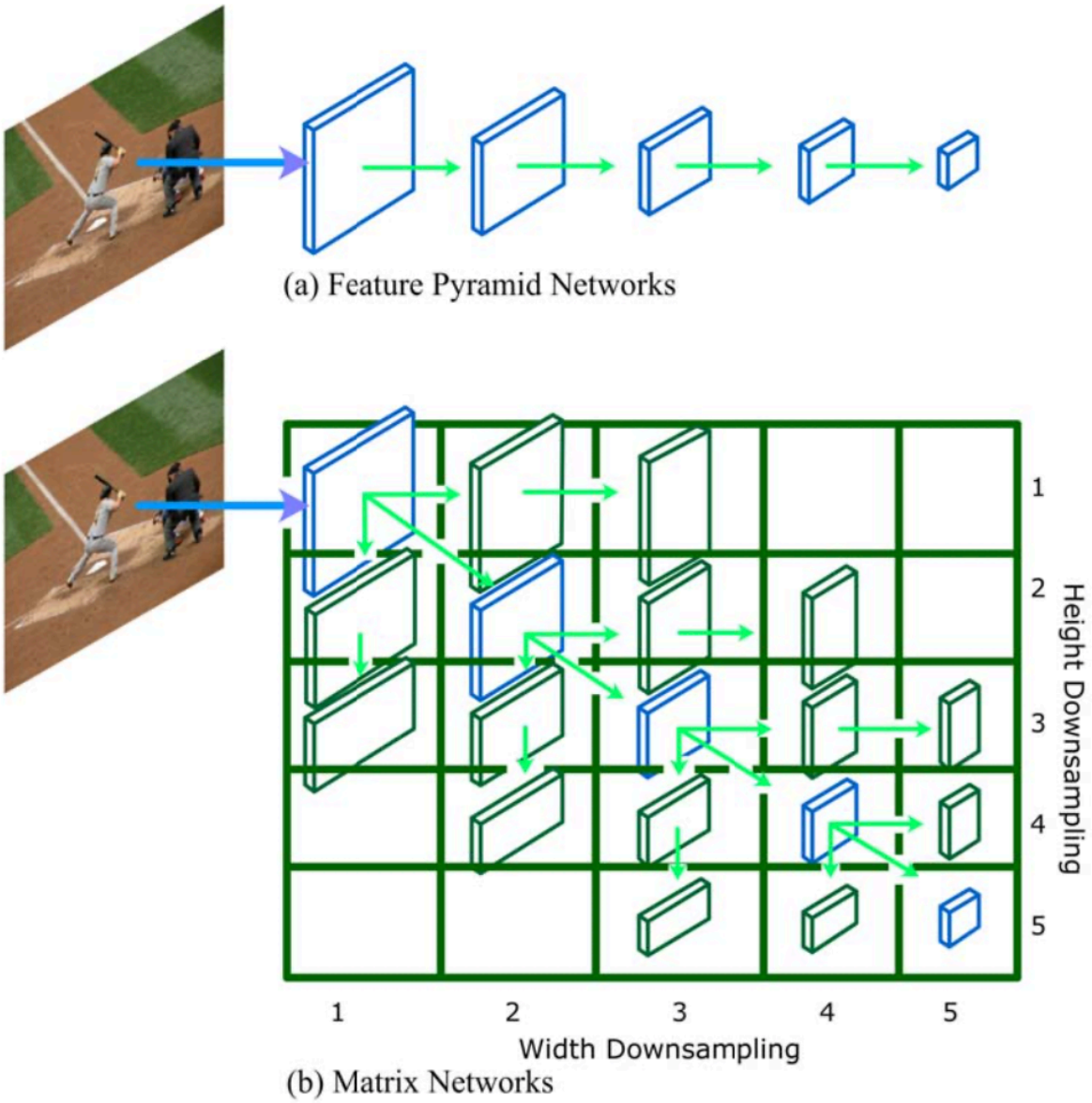
当IoU无法判断时使用embedding判断是否是同一个物体

训练使用Margin Loss: $L = \frac{\sum_{i \in \mathcal{A}} \sum_{j \in \mathcal{A} \setminus \{i\}} L'(i,j)}{|\mathcal{A}| \cdot (|\mathcal{A}|-1)}$

其中，pairwise loss: $L'(i,j) = \begin{cases} \max\left(0, \|\mathbf{f}_i, \mathbf{f}_j\|_2 - (\beta - \alpha)\right), & \text{if } obj(i) = obj(j) \\ \max\left(0, (\beta + \alpha) - \|\mathbf{f}_i, \mathbf{f}_j\|_2\right), & \text{otherwise} \end{cases}$

Matrix Nets: A New Deep Architecture for Object Detection (xNets)

> FPN处理不同大小的物体(特征金字塔)
> 👇本文增加不同长宽比物体的处理 (大小金字塔+aspect ratio金字塔)



高度，宽度减半。左下右上剪枝(物体不常见) 性能提升不明显，相比CenterNet参数量减少 Ref: **参数少一半、速度快3倍：最新目标检测核心架构来了**

IoU-Net

> Add localization confidence in NMS
>
> 可以看作一种精细化的前背景分类（soft）

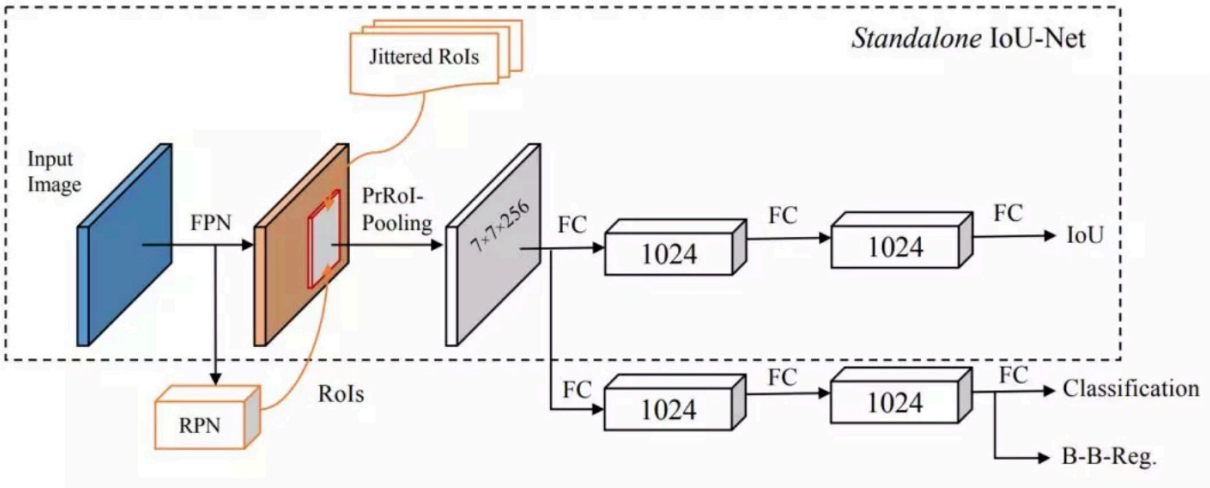
NMS

1. 选择最大classification confidence的框b_j，加入集合S。
2. 其他所有不再集合中的框，如果和b_j的IoU大于threshold，则删去，简化重复框。
3. 重复知道没有框，S为结果。

使用分类置信度作为最开始选择框的依据，IoU用于计算分类置信度最大的框和其他框之间的重合度，删去框。

而IoU-Net使用预测的框和GT的重合IoU，即定位置信度，选择最大作为依据。在inference阶段发挥作用。

预测IoU



通过网络预测IoU：使用FPN作为骨干网络，提特征。使用PrRoI pooling替代RoI pooling

IoU Guided NMS

Algorithm 1 IoU-guided NMS. Classification confidence and localization confidence are disentangled in the algorithm. We use the localization confidence (the predicted IoU) to rank all detected bounding boxes, and update the classification confidence based on a clustering-like rule.

Input: $\mathcal{B} = \{b_1, ..., b_n\}$, $\mathcal{S}$, $\mathcal{I}$, Ω_{nms}
$\mathcal{B}$ is a set of detected bounding boxes.
$\mathcal{S}$ and $\mathcal{I}$ are functions (neural networks) mapping bounding boxes to their classification confidence and IoU estimation (localization confidence) respectively.
Ω_{nms} is the NMS threshold.
Output: $\mathcal{D}$, the set of detected bounding boxes with classification scores.

$\mathcal{D} \leftarrow \varnothing$
while $\mathcal{B} \neq \varnothing$ **do**
  $b_m \leftarrow \arg\max \mathcal{I}(b_j)$
  $\mathcal{B} \leftarrow \mathcal{B} \setminus \{b_m\}$
  $s \leftarrow \mathcal{S}(b_m)$
  for $b_j \in \mathcal{B}$ **do**
    if $\text{IoU}(\text{b}_\text{m}, \text{b}_\text{j}) > \Omega_{\text{nms}}$ **then**
      $s \leftarrow max(s, \mathcal{S}(b_j))$
      $\mathcal{B} \leftarrow \mathcal{B} \setminus \{b_j\}$
    end if
  end for
  $\mathcal{D} \leftarrow \mathcal{D} \cup \{\langle b_m, s \rangle\}$
end while
return $\mathcal{D}$

Rank all detection bbox on localization confidence.

选择IoU最大的框，其他框重叠大于thres的框只使用他的最大conf score作为IoU最大框的conf「根据IoU选择，最大score修正conf」

Consider bounding box refinement as optimization

Algorithm 2 Optimization-based bounding box refinement

Input: $\mathcal{B} = \{b_1, ..., b_n\}$, $\mathcal{F}$, T, λ, Ω_1, Ω_2

$\mathcal{B}$ is a set of detected bounding boxes, in the form of (x_0, y_0, x_1, y_1).
$\mathcal{F}$ is the feature map of the input image.
T is number of steps. λ is the step size, and Ω_1 is an early-stop threshold and $\Omega_2 < 0$ is an localization degeneration tolerance.
Function PrPool extracts the feature representation for a given bounding box and function IoU denotes the estimation of IoU by the IoU-Net.

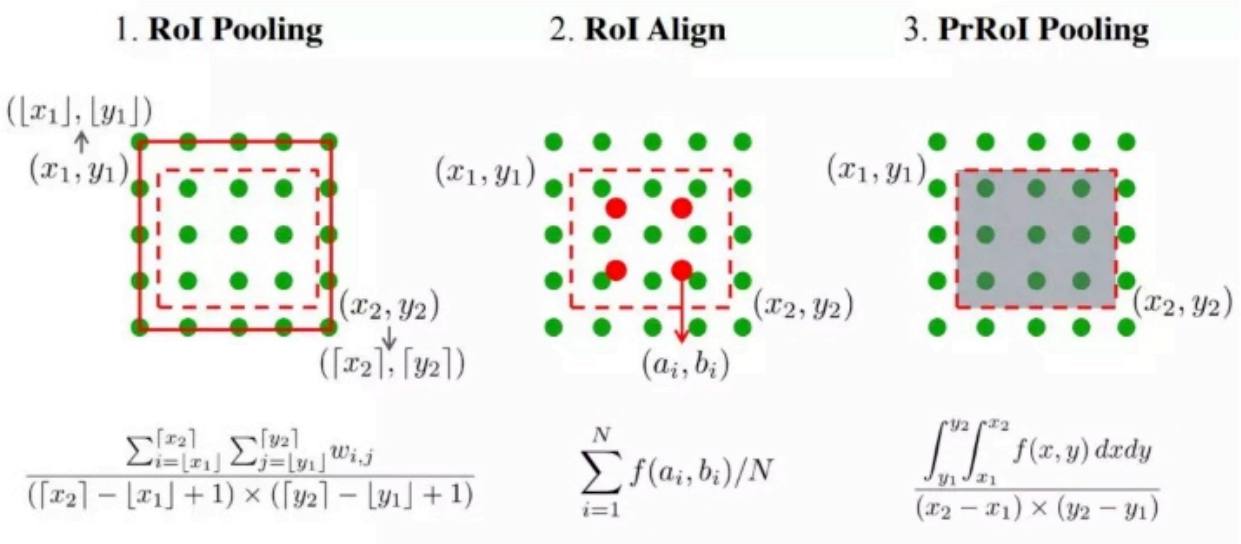
Output: The set of final detection bounding boxes.

$\mathcal{A} \leftarrow \varnothing$
for $i = 1$ *to* T **do**
  for $b_j \in \mathcal{B}$ and $b_j \notin \mathcal{A}$ **do**
    $\boldsymbol{grad} \leftarrow \nabla_{b_j}\text{IoU}(\text{PrPool}(\mathcal{F}, b_j))$
    $PrevScore \leftarrow \text{IoU}(\text{PrPool}(\mathcal{F}, b_j))$
    $b_j \leftarrow b_j + \lambda * scale(\boldsymbol{grad}, b_j)$
    $NewScore \leftarrow \text{IoU}(\text{PrPool}(\mathcal{F}, b_j))$
    if $|PrevScore - NewScore| < \Omega_1$ **or** $NewScore - PrevScore < \Omega_2$ **then**
      $\mathcal{A} \leftarrow \mathcal{A} \cup \{b_j\}$
    end if
  end for
end for
return $\mathcal{B}$

通过预测IoU并产生梯度，更新bounding box，并通过判断分数的提升和差值来更新边界框 // ToRead

Precise RoI pooling

使用双线性插值来连续化特征图，任意连续坐标(x,y)处都是连续的 $f(x,y) = \Sigma_{i,j} IC(x,y,i,j) \times w_{i,j}$
$IC(x,y,i,j) = max(0, 1-|x-i|) \times max(0, 1-|y-j|)$是插值系数，xy连续，ij为坐标像素点。RoI的一个bin表示为左上角和右下角的坐标对。通过二重积分进行池化（加权求和）
$PrPool(\{(x_1,y_1),\ (x_2,y_2)\},\ F) = \frac{\int_{y_1}^{y_2}\int_{x_1}^{x_2} f(x,y)dxdy}{(x_2-x_1)\times(y_2-y_1)}$



使用ResNet-FPN作为骨干网络，RoI pooling换成PrRoI pooling。同时IoU预测分支可以和R-CNN的分类和边界框回归分支并行工作.

FreeAnchor: Learning to Match Anchors for Visual Object Detection

Related 对于anchor生成/分配/选择的改进：Guided Anchoring, IoU-Net, MetaAnchor MetaAnchor - 简书

> 对于anchor和object的匹配方式的改进，Learn to match

之前采用IoU最大的anchor进行分配：细长物体，最representative的特征不在物体中心，IoU最大≠最representative

Assign策略需要满足：

1. **Recall**: 每个物体都能分配一个anchor
2. **Precision**: 区分background anchor
3. **Compatible NMS**: 高分类分数的anchor有好的localization

matching过程看作MLE过程，每个物体从bag of anchor中选likelihood probability最大的

Maximum Likelihood Estimation分析现有detector

训练损失函数，$C_{i,j}$表示j anchor和i 物体匹配「assign using IoU criterion」

$$\mathcal{L}(\theta) = \sum_{a_j \in A_+} \sum_{b_i \in B} C_{i,j} \mathcal{L}_{i,j}^{cls}(\theta) + \beta \sum_{a_j \in A_+} \sum_{b_i \in B} C_{i,j} \mathcal{L}_{i,j}^{loc}(\theta) + \sum_{a_j \in A_-} \mathcal{L}_j^{bg}(\theta)$$

把训练损失函数看作似然概率

$$\begin{aligned}
\mathcal{P}(\theta) &= \mathbf{e}^{-\mathcal{L}(\theta)} \\
&= \prod_{a_j \in A_+} \left(\sum_{b_i \in B} C_{ij} e^{-\mathcal{L}_{ij}^{cl_j}(\theta)} \right) \prod_{a_j \in B_+} \left(\sum_{b_i \in B} C_{ij} e^{-\beta \mathcal{L}_{j_j}^{loc}(\theta)} \right) \prod_{\substack{a_j \in A_- \\ a_j \in A_-}} \mathcal{P}_j^{bg}(\theta) \\
&= \prod_{a_j \in A_+} \left(\sum_{b_i \in B} C_{ij} \mathcal{P}_{ij}^{cls}(\theta) \right) \prod_{a_j \in A_+} \left(\sum_{b_i \in B} C_{ij} \mathcal{P}_{ij}^{loc}(\theta) \right) \prod_{a_j \in A_-} \mathcal{P}_j^{bg}(\theta)
\end{aligned}$$

映射非常巧妙，使$[0, +\infty)$的损失映射到$(0, 1]$，而且损失越小，$\mathcal{P}(\theta)$越大

因此，最小化损失的目标转换为最大化似然概率

改进detection似然函数

目标 recall，precision，compatible

Recall：每个obj构建bag of anchor，最大化其中anchor的cls和loc似然。每个obj一定存在一个anchor对应

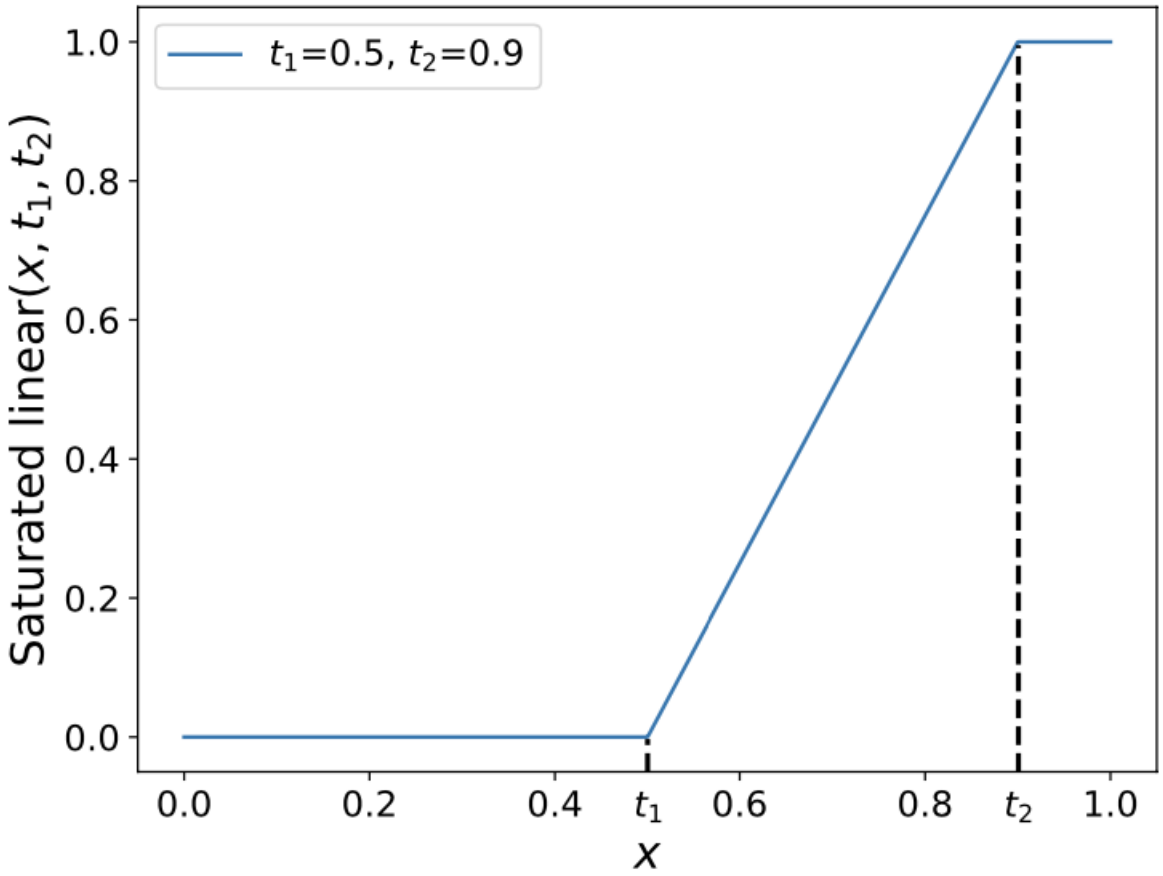
$\mathcal{P}_{\text{recall}}(\theta) = \prod_i \max_{a_j \in A_i} \left(\mathcal{P}_{ij}^{cls}(\theta) \mathcal{P}_{ij}^{loc}(\theta) \right)$

Precision：即对anchor区分前背景，把背景anchor分出

$\mathcal{P}_{\text{precision}}(\theta) = \prod_i \left(1 - P\{a_j \in A_-\} \left(1 - \mathcal{P}_j^{bg}(\theta) \right) \right)$

其中$P\{a_j \in A_-\} = 1 - \max_i P\{a_j \to b_i\}$ 表示anchor j不match任何物体。即anchor不match任何obj概率越高，anchor不属于背景的概率越低(1-)，才可以最大$\mathcal{P}_{\text{precision}}(\theta)$

Compatible: $P\{a_j \rightarrow b_i\}$表示j anchor匹配i obj概率，NMS按照cls分数选。所以改成loc分数「i j 的IoU」越大，匹配概率越高，P为关于IoU的saturated linear函数。步骤存在于$\mathcal{P}_{\text{precision}}(\theta)$中



横坐标为IoU

似然函数: Jointly maximize

$\mathcal{P}'(\theta) = \mathcal{P}_{recall}(\theta) \times \mathcal{P}_{precision}(\theta)$

改进似然函数推出Matching Mechanism

训练损失$\mathcal{L}'(\theta) = -\log \mathcal{P}'(\theta)$，使用FocalLoss

其中有max操作，但随机初始化的网络，所有anchor得分都低，max没有意义

改用**Mean-max**函数：$\text{Mean} - \max(X) = \frac{\sum_{x_j \in X} \frac{x_j}{1-x_j}}{\sum_{x_j \in X} \frac{1}{1-x_j}}$

训练不充分时接近mean，使用bag中所有anchor训练

训练充分时接近max，选择最好的anchor训练

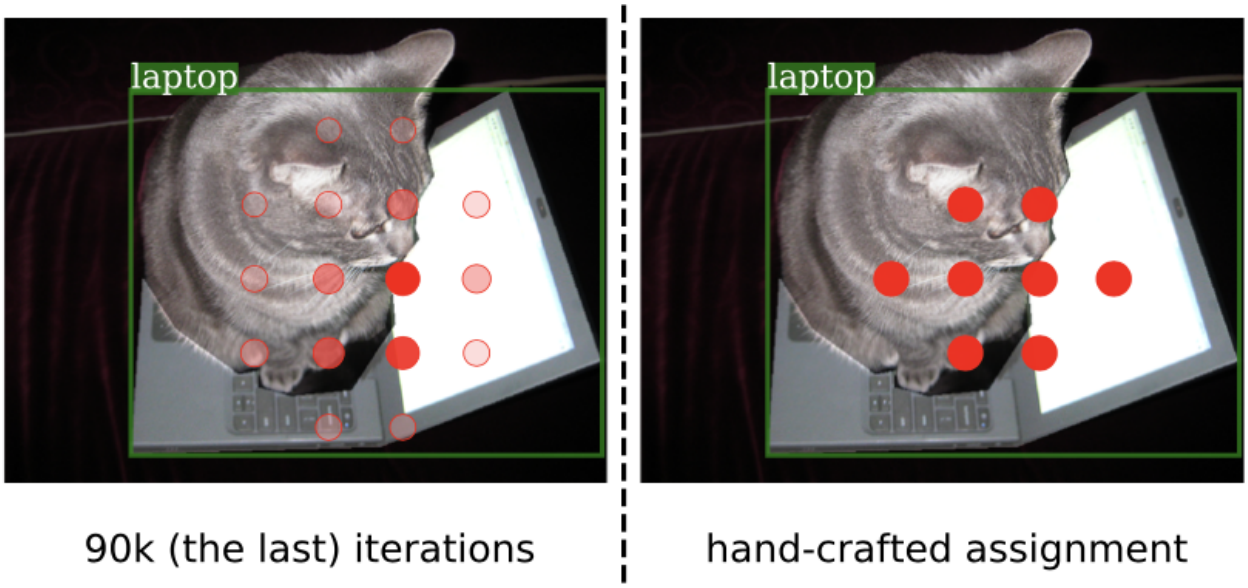
Algorithm 1 Detector training with FreeAnchor.

Input: I: Input image.
$\mathcal{B}$: A set of ground-truth bounding boxes b_i.
$\mathcal{A}$: A set of anchors a_j in image.
n: Hyper-parameter about anchor bag size .

Output: θ: Detection network parameters.

1: $\theta \leftarrow$ initialize network parameters.
2: **for** i=1:MaxIter **do**
3: **Forward propagation:**
Predict class a_j^{cls} and location a_j^{loc} for each anchor $a_j \in \mathcal{A}$.
4: **Anchor bag construction:**
$\mathcal{A}_i \leftarrow$ Select n top-ranked anchors a_j in terms of their IoU with b_i.
5: **Loss calculation:**
Calculate $L''(\theta)$ with Eq. 7.
6: **Backward propagation:**
$\theta^{t+1} = \theta^t - \lambda \nabla_{\theta^t} L''(\theta^t)$ using a stochastic gradient descent algorithm.
7: **end for**
8: **return** θ

可视化，anchor assign confident (laptop)



相比baseline有提升3%. 使用ResNeXt-64x4d-101，**为multi-scale

Detector	Backbone	Iter.	AP	AP_{50}	AP_{75}	AP_S	AP_M	AP_L
RetinaNet [7]	ResNet-101	135k	39.1	59.1	42.3	21.8	42.7	50.2
FoveaBox [22]	ResNet-101	135k	40.6	60.1	43.5	23.3	45.2	54.5
FSAF [23]	ResNet-101	135k	40.9	61.5	44.0	24.0	44.2	51.3
FCOS [13]	ResNet-101	180k	41.5	60.7	45.0	24.4	44.8	51.6
RetinaNet [7]	ResNeXt-101	135k	40.8	61.1	44.1	24.1	44.2	51.2
FoveaBox [22]	ResNeXt-101	135k	42.1	61.9	45.2	24.9	46.8	55.6
FSAF [23]	ResNeXt-101	135k	42.9	63.8	46.3	26.6	46.2	52.7
FCOS [13]	ResNeXt-101	180k	43.2	62.8	46.6	26.5	46.2	53.3
CornerNet [14]	Hourglass-104	500k	40.6	56.4	43.2	19.1	42.8	54.3
CenterNet [15]	Hourglass-104	480k	44.9	62.4	48.1	25.6	47.4	57.4
FreeAnchor	ResNet-101	180k	43.1	62.2	46.4	24.5	46.1	54.8
FreeAnchor	ResNeXt-101	180k	44.9	64.3	48.5	26.8	48.3	55.9
FreeAnchor*	ResNeXt-101	180k	46.0	65.6	49.8	27.8	49.5	57.7
FreeAnchor**	ResNeXt-101	180k	**47.3**	**66.3**	**51.5**	**30.6**	**50.4**	**59.0**

Ref: https://www.aminer.cn/research_report/5dedbde4af66005a4482453f?download=false

密集小目标

Paper: Benchmark for Generic Product Detection: A strong baseline for Dense Object Detection

Dataset	#Images	#Objects	#Obj/Img	Object Size (Mean)	Object Size (Std)	Avg Img Size
SKU110K-Test	2941	432,312	146	0.27%	0.21%	7.96
WebMarket	3153	118,388	37	1.20%	1.09%	4.40
TobaccoShelves	354	13,184	37	1.1%	0.65%	6.08
Holoselecta	295	10,036	34	0.99%	0.80%	15.62
GP	680	9184	13	3.66%	2.59%	7.99
CAPG-GP	234	4756	20	3.09%	3.04%	12.19

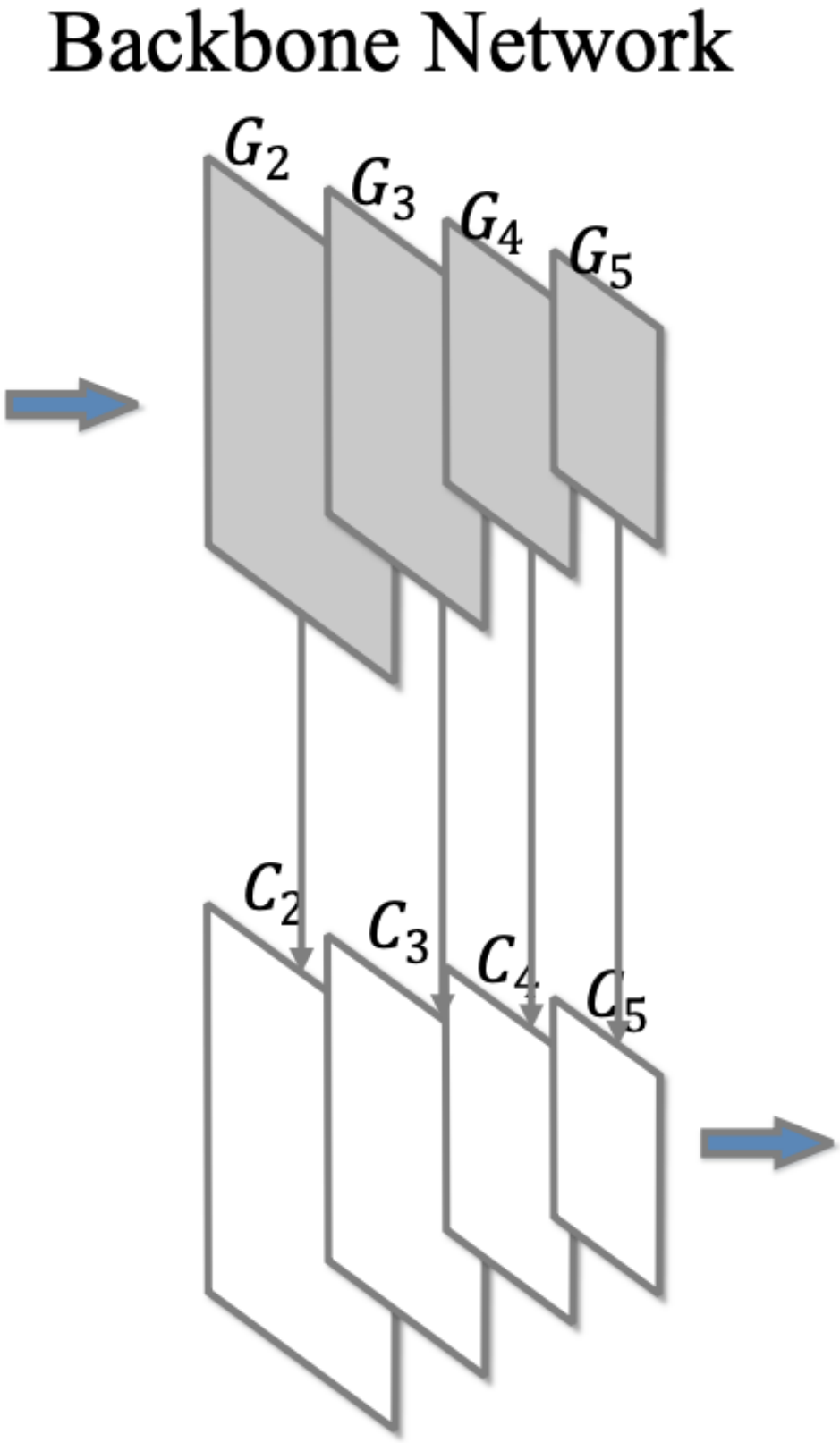
Table 1: Details of the datasets in the benchmark. # represents the count. Object sizes (Mean and Standard Deviation) are relative to the image size. Average Image size is shown in Megapixels

Scale Match for Tiny Person Detection [method+dataset]

Aligndet: Revisiting Feature Alignment for One-stage Object Detection

FPN & Variants

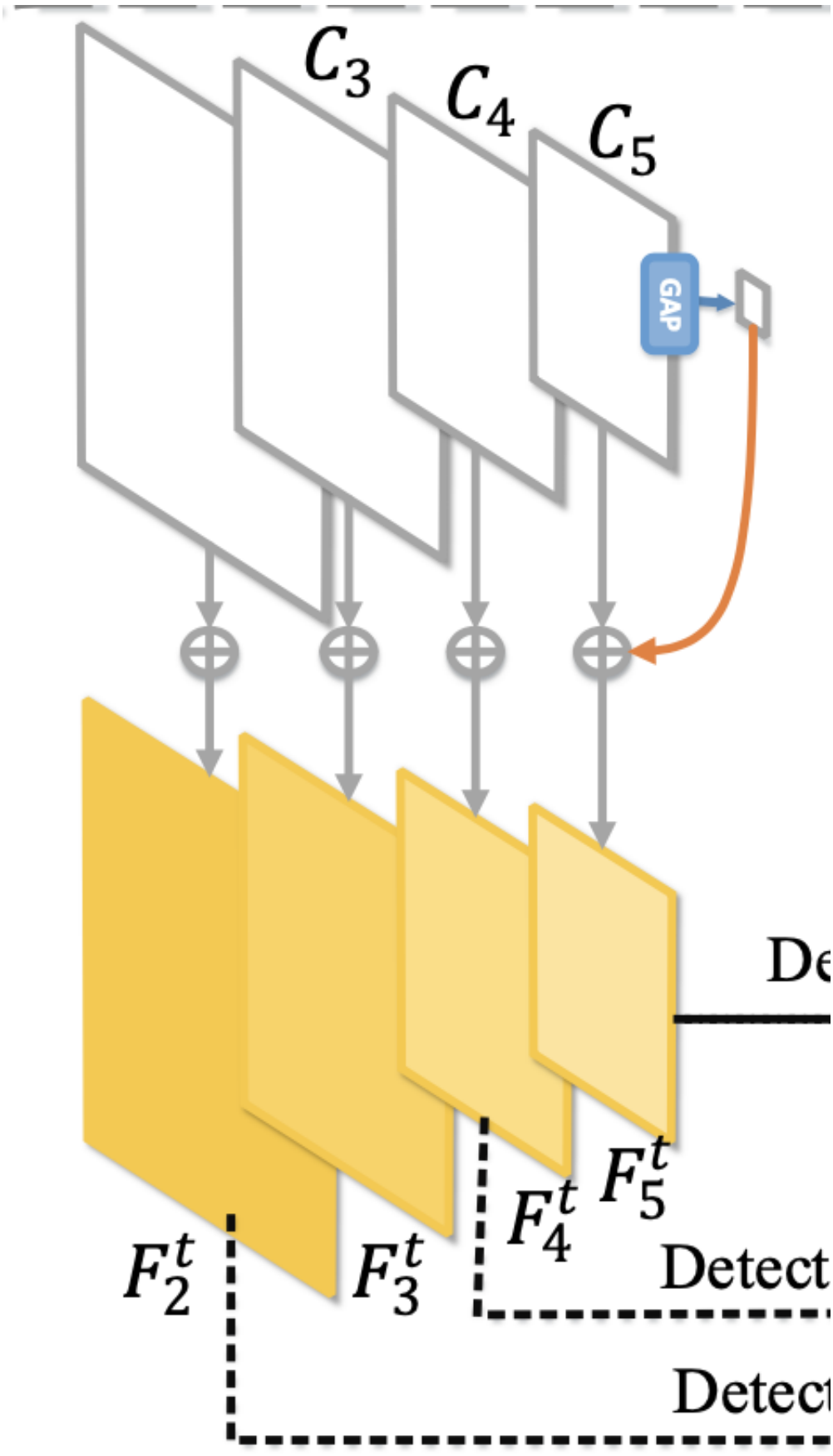
所有FPN都使用backbone的多层特征图（经过 1x1 卷积） 👇$C_{2..5}$



Top-down FPN

经典FPN，从最高层特征(semantic，low-res)经过upsample，和各同一级的特征图相加

给底层特征引入高层语义信息，益于小目标检测（低层特征图）

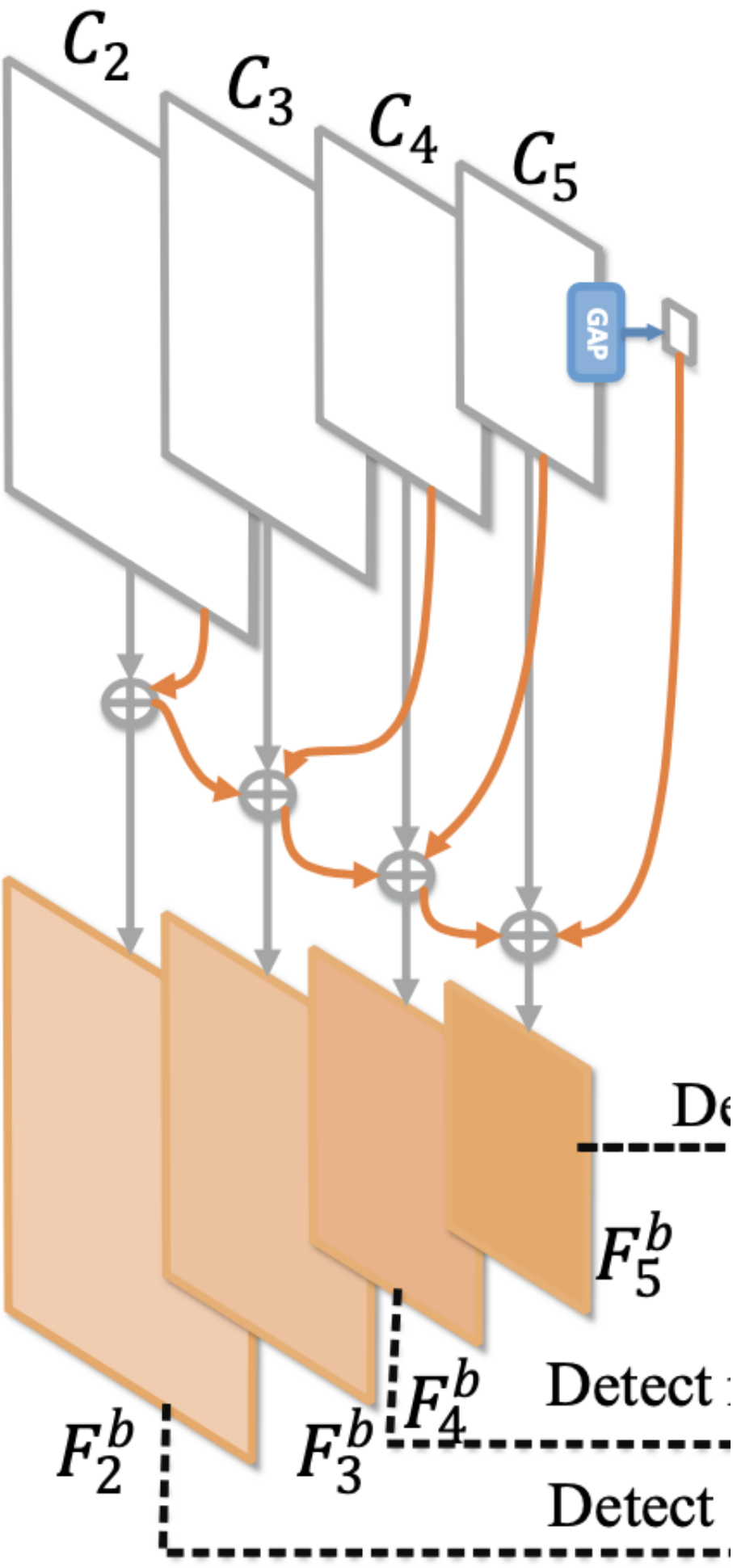


公式$F_i^t = \mathbf{W}_{i+1}^t \otimes \left(U\left(F_{i+1}^t\right) + C_i\right)$

Bottom-up FPN

从最底层(high-res)向上逐次产生FPN层，向高层特征图传播低层的空间细节信息(spatial)

从低到高，融合「本层特征，高一层特征，上一层FPN」


C_2
C_3
C_4
C_5
GAP
F_5^b
F_4^b
F_3^b
F_2^b
Detect
Detect

公式$F_i^b = \mathbf{W_i^b} \otimes \left(D\left(F_{i-1}^b\right) + C_i + U\left(C_{i+1}\right)\right)$

$D \rightarrow$ downsample, $U \rightarrow$ upsample

Fusing-splitting FPN

上述两个FPN顺序逐次产生，先产生的层会对之后层影响(unfair)

首先分组**fuse**高层和低层的临近两组特征

$\alpha_s = C_4 + U(C_5), \alpha_l = D(C_2) + C_3$

然后**merge**高层和低层的特征

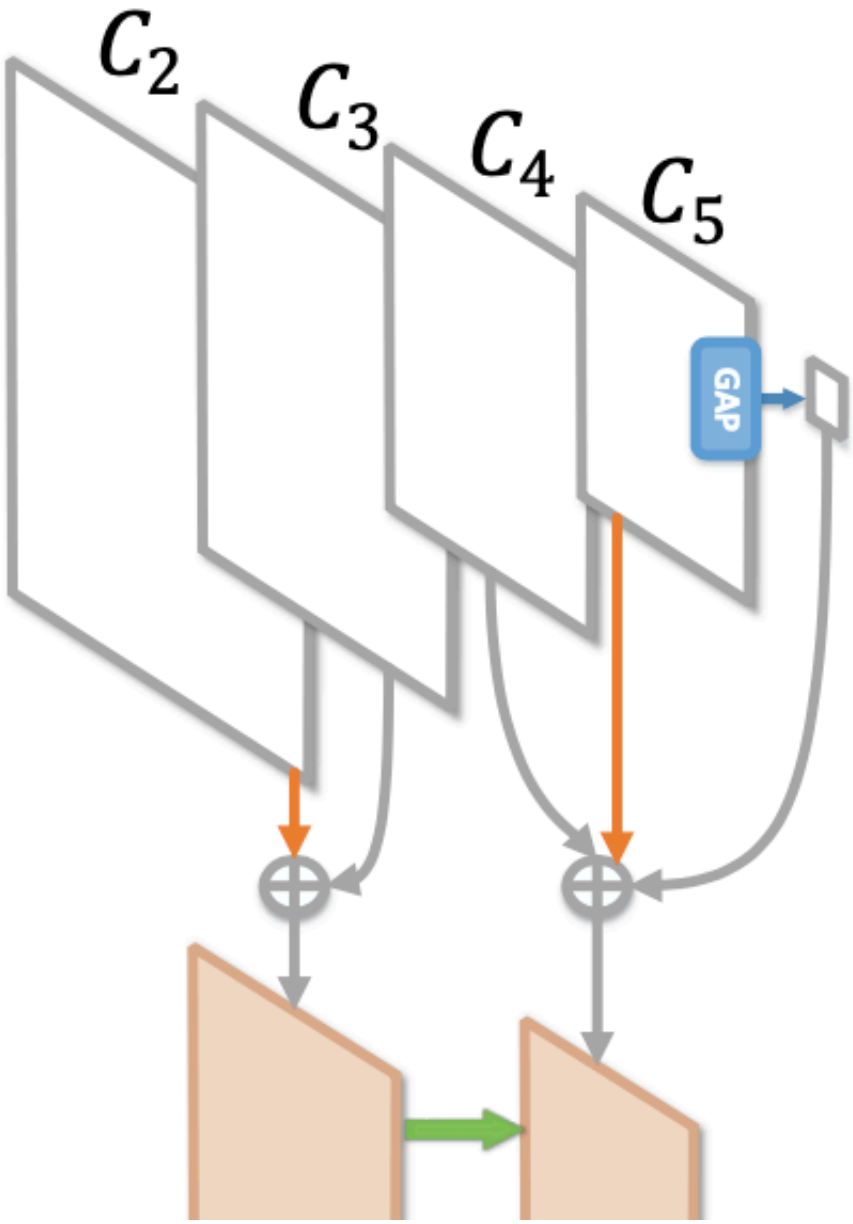
$\beta_s = \mathbf{W_s^f} \otimes \mathrm{cat}(\alpha_s, D(\alpha_l))$

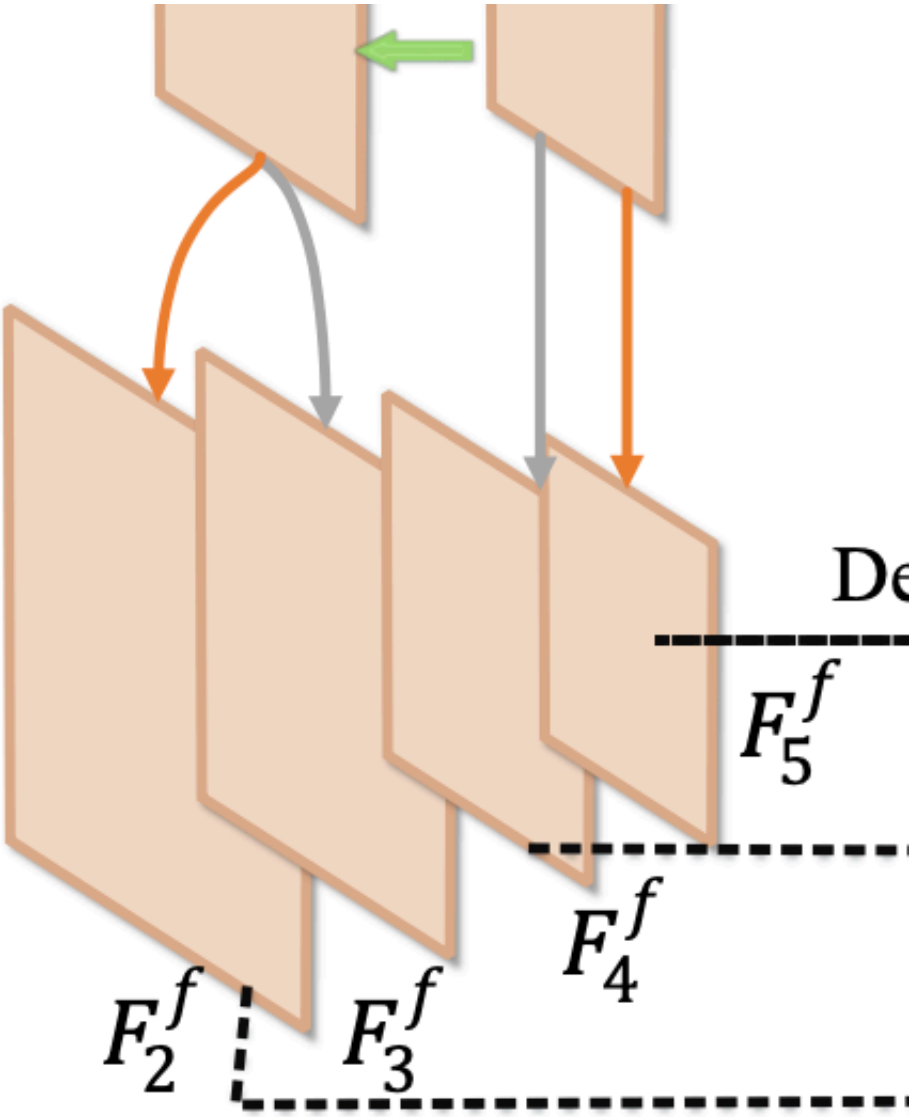
$\beta_l = \mathbf{W_l^f} \otimes \mathrm{cat}(U(\alpha_s), \alpha_l)$

再**split**产生不同层的特征

$F_2^f = U(\beta_l), F_3^f = \beta_l$

$F_4^f = \beta_s, F_5^f = D(\beta_s)$





Ref: *MFPN: A NOVEL MIXTURE FEATURE PYRAMID NETWORK OF MULTIPLE ARCHITECTURES FOR OBJECT DETECTION*

Learning Data Augmentation Strategies for Object Detection

> **数据增强**，通过搜索来**combine** transformations

数据增强角度：1. Learn a **generator** to create data 2. Learn a set of **transformations** applied to existing data(本文)

常用transformer：image mirror，multi-scale training，crop-and-erase (occlude)，cut-and-paste

自动学习数据集对应的数据增强方式：AutoAugment

Policy search问题：K=5个sub-policies，每个包含N=2个操作。训练时随机选择sub-policy，顺序执行N。

操作两个参数「执行操作的概率，操作大小程度」👇

Sub-policy 1. (Color, 0.2, 8), (Rotate, 0.8, 10)
Sub-policy 2. (BBox_Only_ShearY, 0.8, 5)
Sub-policy 3. (SolarizeAdd, 0.6, 8), (Brightness, 0.8, 10)
Sub-policy 4. (ShearY, 0.6, 10), (BBox_Only_Equalize,0.6, 8)
Sub-policy 5. (Equalize, 0.6, 10), (TranslateX, 0.2, 2)

Transform

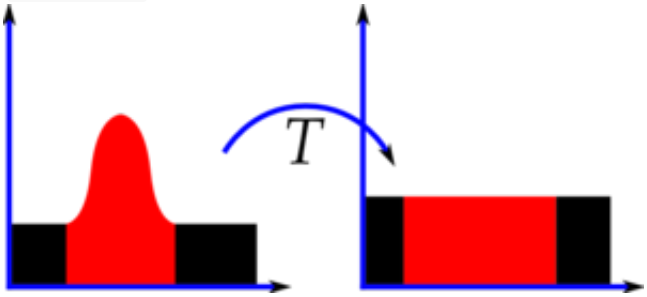
1. **Color operations**: 改变颜色通道，obj位置不变 `equalize, contrast brightness`
2. **Geometric ops**: 改变obj位置和大小 `rotate, ShearX, TranslationY`
3. **Bounding box ops.**: 只改变bbox内的图像 `BBox_Only_Equalize, BBox_Only_Rotate,`

`BBox_Only_FlipLR`

Results

`Rotate` 旋转图片和bbox（best）

`Equalize` 直方图均衡化(Histogram equalization)，平衡不同灰度像素出现概率，增大对比度👇

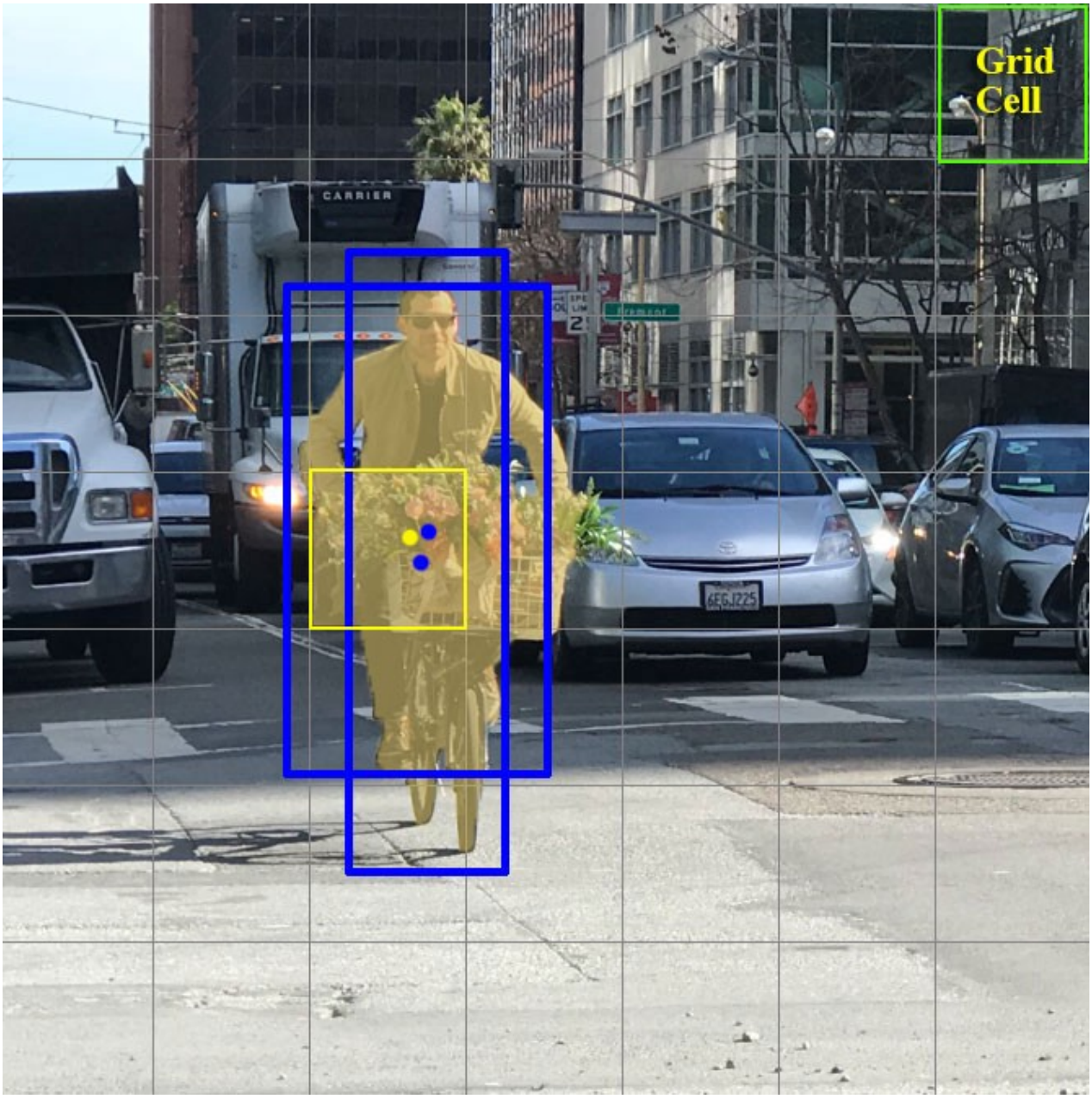


`BBox_Only_TranslateY` bbox内垂直变换，上下翻转

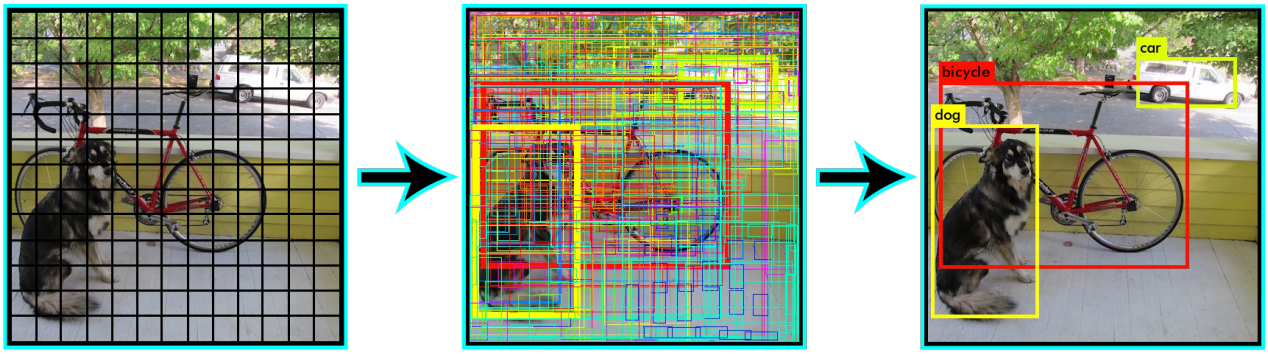
Operation Name	Description	Range of magnitudes
ShearX(Y)	Shear the image and the corners of the bounding boxes along the horizontal (vertical) axis with rate *magnitude*.	[-0.3,0.3]
TranslateX(Y)	Translate the image and the bounding boxes in the horizontal (vertical) direction by *magnitude* number of pixels.	[-150,150]
Rotate	Rotate the image and the bounding boxes *magnitude* degrees.	[-30,30]
Equalize	Equalize the image histogram.	
Solarize	Invert all pixels above a threshold value of *magnitude*.	[0,256]
SolarizeAdd	For each pixel in the image that is less than 128, add an additional amount to it decided by the magnitude.	[0,110]
Contrast	Control the contrast of the image. A *magnitude*=0 gives a gray image, whereas *magnitude*=1 gives the original image.	[0.1,1.9]
Color	Adjust the color balance of the image, in a manner similar to the controls on a colour TV set. A *magnitude*=0 gives a black & white image, whereas *magnitude*=1 gives the original image.	[0.1,1.9]
Brightness	Adjust the brightness of the image. A *magnitude*=0 gives a black image, whereas *magnitude*=1 gives the original image.	[0.1,1.9]
Sharpness	Adjust the sharpness of the image. A *magnitude*=0 gives a blurred image, whereas *magnitude*=1 gives the original image.	[0.1,1.9]
Cutout [9, 53]	Set a random square patch of side-length *magnitude* pixels to gray.	[0,60]
BBox_Only_X	Apply X to each bounding box content with independent probability, and magnitude that was chosen for X above. Location and the size of the bounding box are not changed.	

YOLO

分格子(grids)，每个格子只预测规定数量bbox，只有当gt box的中心点落入grid内时，此grid负责预测这个gt。（*潜在问题：密集物体，多个中心点落入同一个grid，漏检*）



网络输出 `(x,y,w,h), box_confidence_score` ，表示normalized长宽和中心点offset，以及置信度「表示objectness和位置准确性」

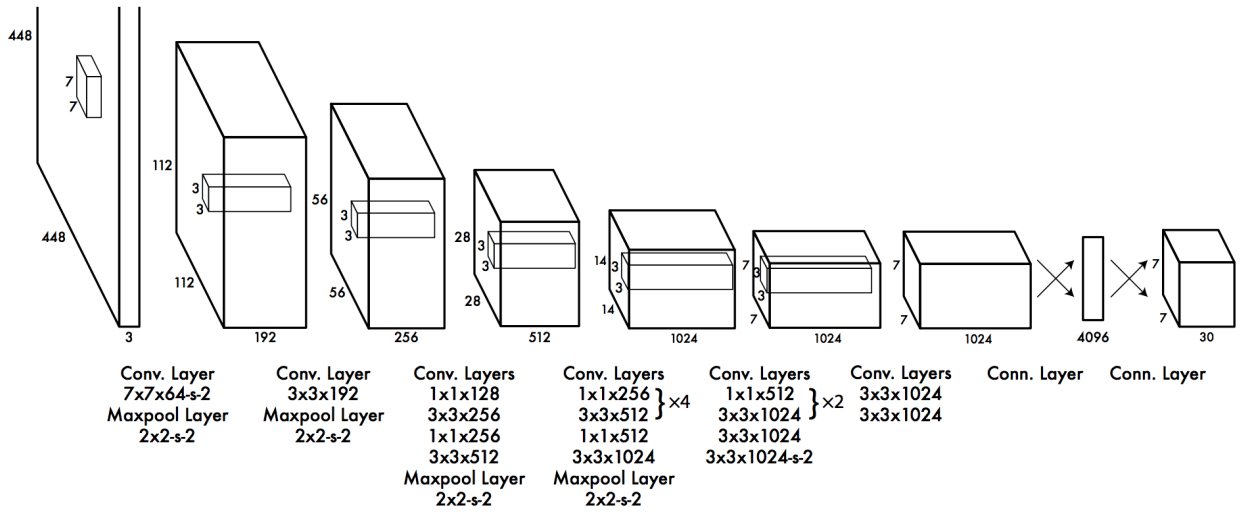


分grid→每个grid产生k个预测→保留高box_conf_score预测

其中$box_conf_score = P(object) \cdot IoU$，$conditional\_class\_prob = P(class_i|object)$

class confidence score: $box\_conf\_score \times cond\_cls\_prob = P(class_i) \cdot IoU$

表示分类和回归的准确率



网络结构👆：下采样+全连接回归预测

Loss function：包括分类损失，位置损失，objectness （$S \times S$grids, B bbox each grids ）

1. **Classification loss**：类别，cond_cls_prob

   $\sum_{i=0}^{S^2} 1_i^{\text{obj}} \sum_{c \in \text{ classes}} \left(p_i(c) - \hat{p}_i(c)\right)^2$

2. **Localization loss**：只计算匹配了gt的grid

   $$\lambda_{\text{coord}} \sum_{i=0}^{S^2} \sum_{j=0}^{B} 1_{ij}^{\text{obj}} \left[(x_i - \hat{x}_i)^2 + (y_i - \hat{y}_i)^2\right]$$
   $$+\lambda_{\text{coord}} \sum_{i=0}^{S^2} \sum_{j=0}^{B} 1_{ij}^{\text{obj}} \left[(\sqrt{w_i} - \sqrt{\hat{w}_i})^2 + (\sqrt{h_i} - \sqrt{\hat{h}_i})^2\right]$$

   where $1_{ij}^{\text{obj}} = 1$表示grid i的第j个box负责预测物体，预测根号来使大小物体误差值对loss函数贡献不同「见平方根函数，x小增长快，x大增长慢。小值误差增长快，大值误差增长慢」

3. **Confidence loss**：objectness，区分前背景，使用box_conf_score即C_i计算

   $\sum_{i=0}^{S^2} \sum_{j=0}^{B} 1_{ij}^{\text{obj}} \left(C_i - \hat{C}_i\right)^2$和$\lambda_{\text{noobj}} \sum_{i=0}^{S^2} \sum_{j=0}^{B} 1_{ij}^{\text{noobj}} \left(C_i - \hat{C}_i\right)^2$

采用**NMS**去掉重复框

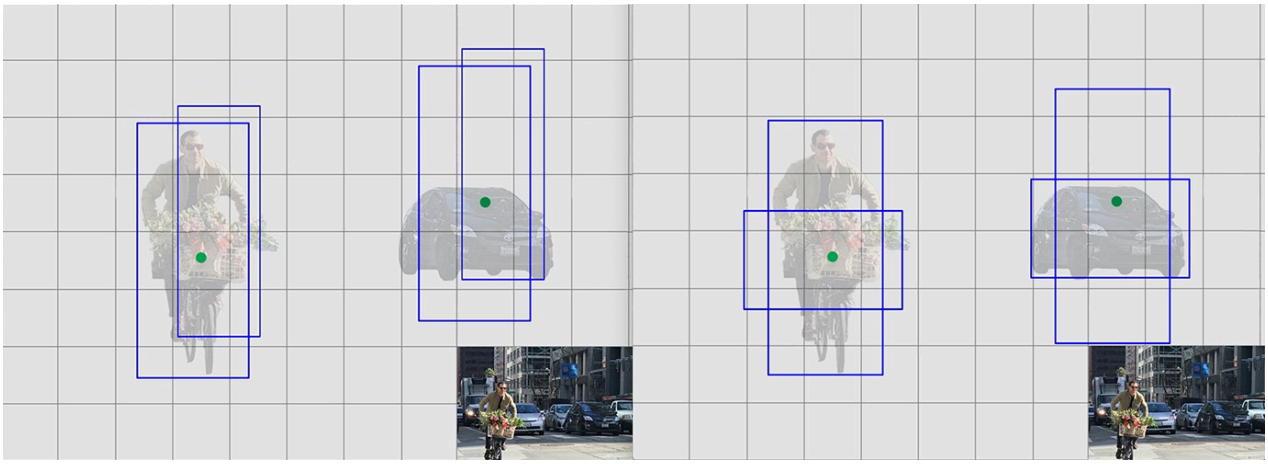
没有RPN可以让网络获得更多context，利于分类(fewer false pos.)

YOLOv2

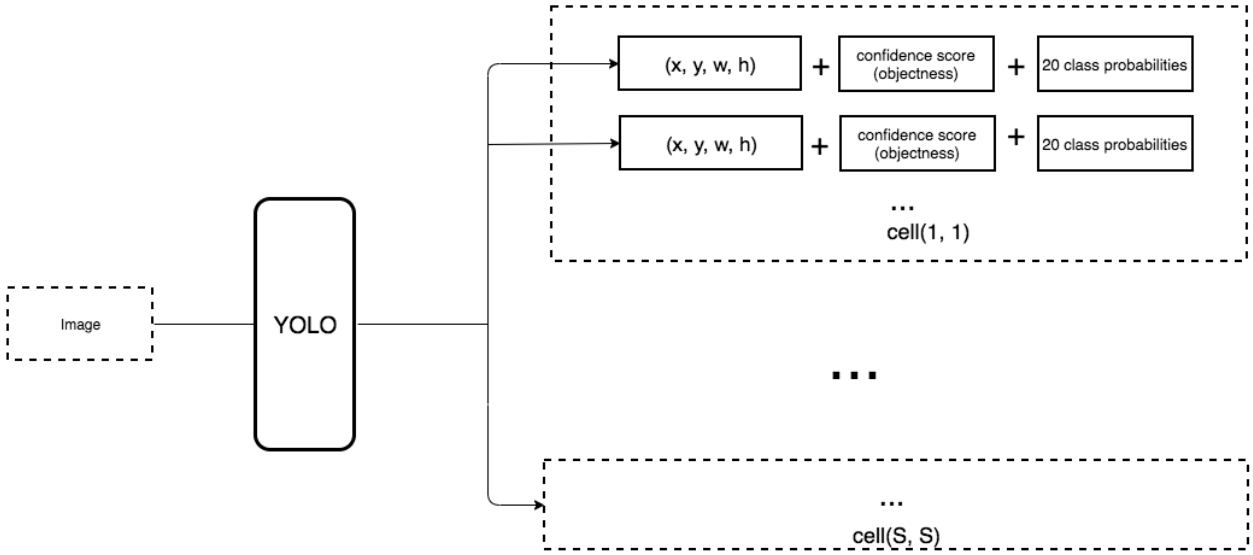
BN，**高分辨率**（`224x224` 上pretrain backbone，`448x448` 上finetune)

anchor box，对grid内B个box增加先验知识，规定初始scale和shape，focus on a specific shape，训练更稳定 👇从左到右

Anchor机制通过预定义scale和shape来引入先验知识，bbox has strong patterns



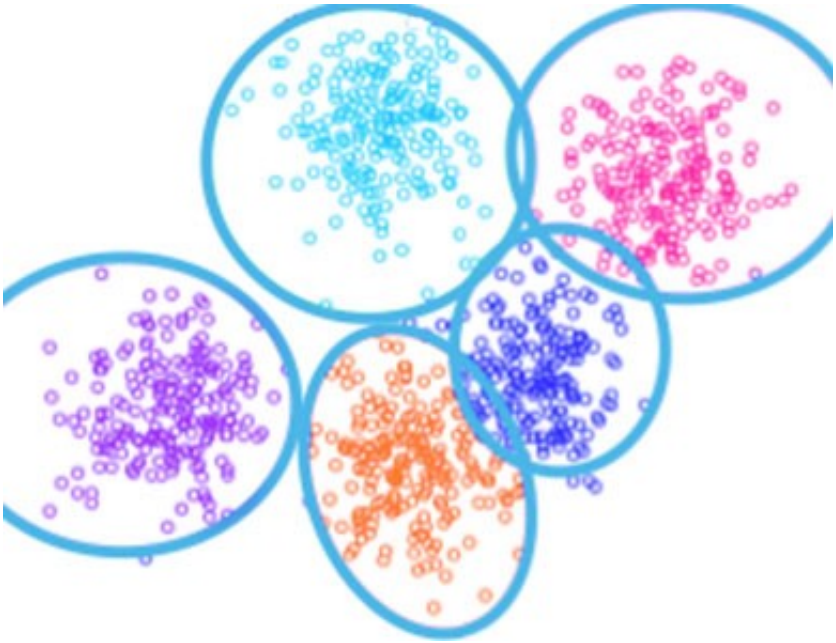
去掉FC层，使用 `1x1 conv` 改变通道为 `7x7x((4+1+20)x5)` ，grid内5个anchor



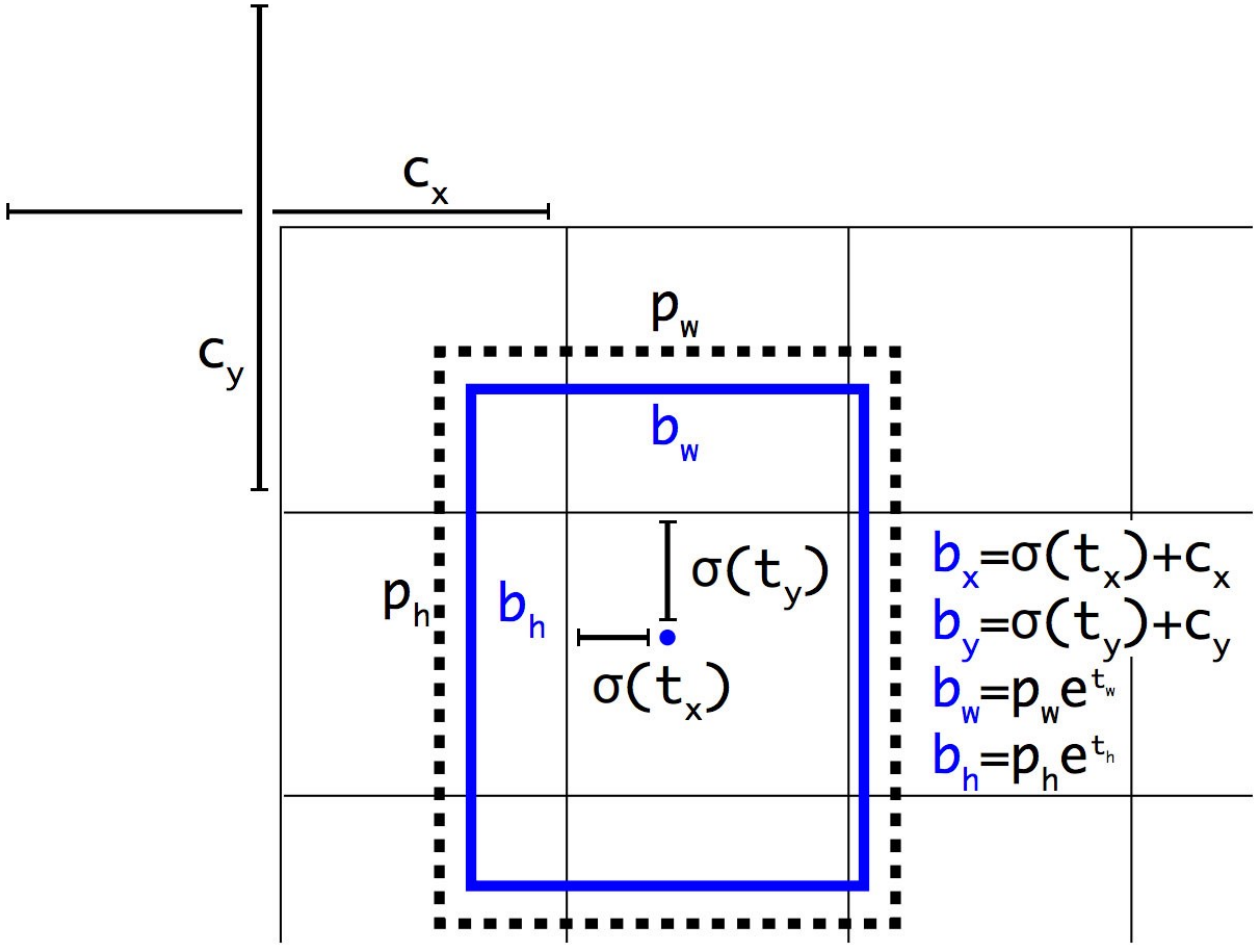
特征图奇数分辨率：大物体处于图片中心，奇数更好分which grid

去掉pooling

anchor**聚类**确定predefined scale&shape（👇数据点之间距离表示IoU大小，位置无意义）每类anchor配置看作一个cluster



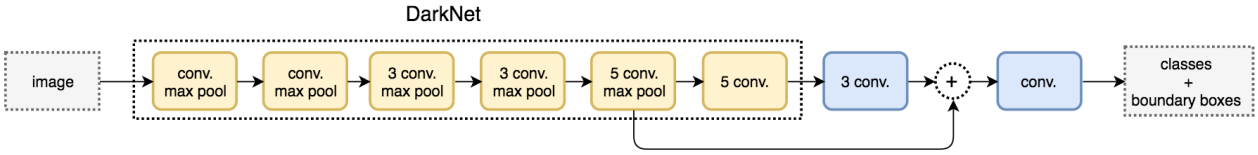
预测offset `[tx, ty, tw, th]` 减少网络预测取值范围，增大可表示的数值范围 👇



增加**passthrough**，类似skip-connection。和浅层特征图 `concat` 预测小目标

Multi-Scale Training采用多个尺度训练适应尺度变化 `320x320, 352x352,..., 608x608` 10个batch的不同尺度图片训练

使用**DarkNet**作为backbone 👇



YOLO9000

使用hierarichical classification训练yolo，使用WordTree将分类和检测数据集混合训练，分9418类

YOLOv3

Multi-label classification：输出一个label id而不是cat维向量，直接输出exclusive output，使用binary cross-entropy loss训练

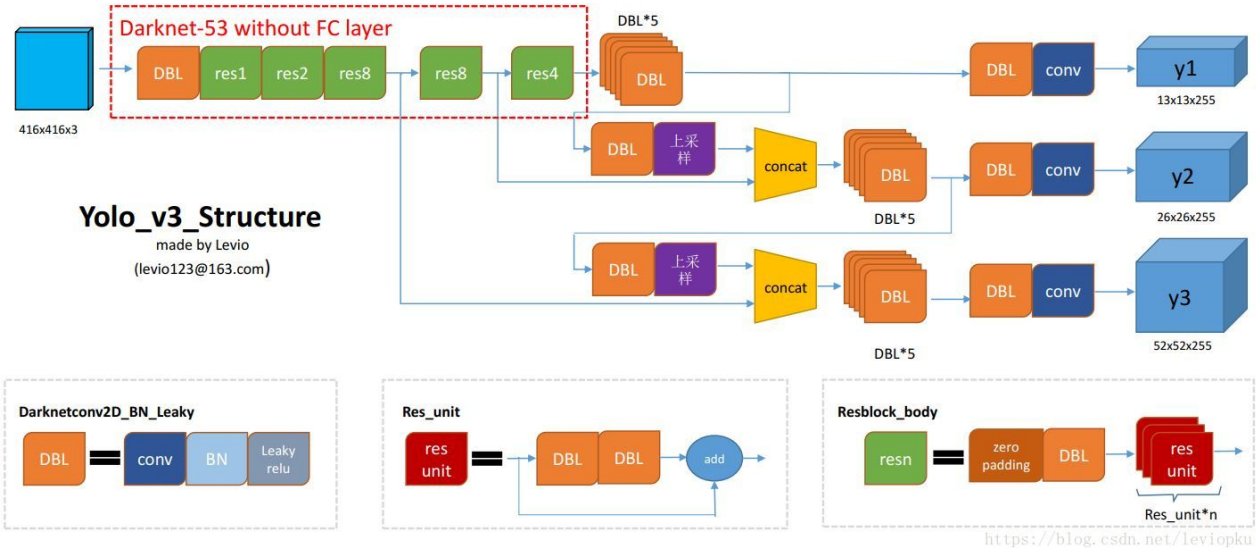
每个目标只匹配一个anchor，没有匹配的anchor不计算cls和loc损失，只计算objectness

FPN：在3个尺度的特征图上预测，每个grid预测3个anchor，一共9种anchor配置

Residual+DarkNet53：卷积增加skip-connection

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

FPN



Ref: https://medium.com/@jonathan_hui/real-time-object-detection-with-yolo-yolov2-28b1b93e2088

IoU

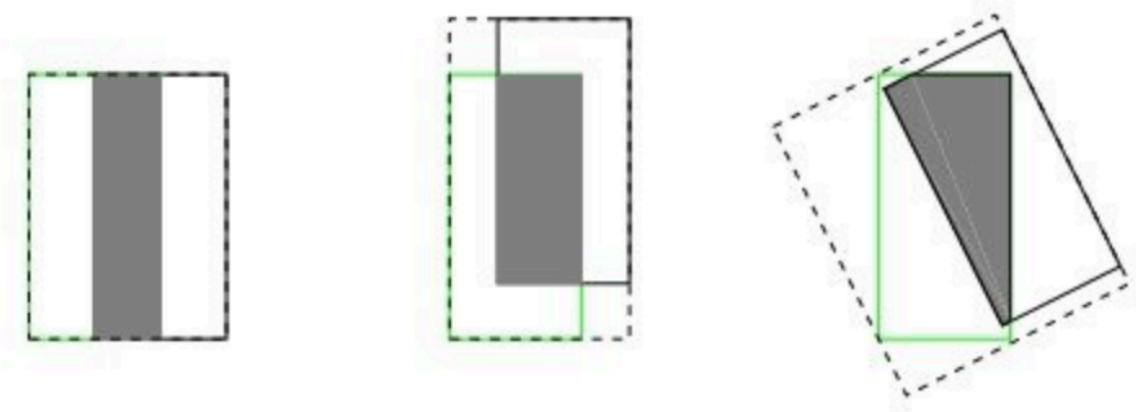
评价预测和gt的距离，回归目标。具有**尺度不变形**

$\text{IoU} = \frac{|A \cap B|}{|A \cup B|}$

训练$L_{IoU} = 1 - \text{IoU}$

问题：

1. 没有重叠时 IoU=0，没有梯度无法用IoU loss训练
2. 无法很好反映方向不一致时重叠



GIoU (Generalized)

$\text{GIoU} = \text{IoU} - \frac{|C \backslash (A \cap B)|}{|C|}$

其中C为包含A和B的最小凸多边形(enclosing convex)，多为矩形

A和B不重合时也可以优化，范围$[-1, 1]$，不重合时为负数(provide moving direction)

关注**形状之间缝隙减小**，如👆2,3中缝隙导致GIoU更小

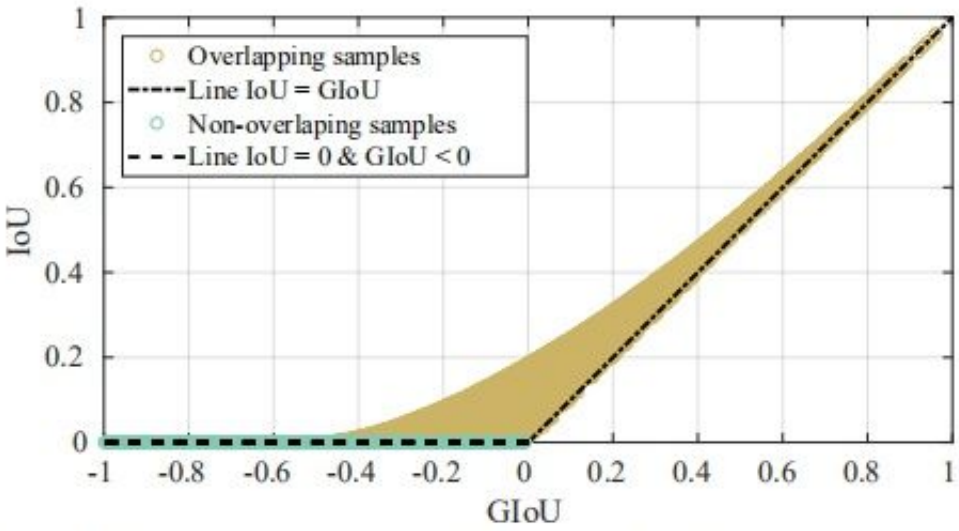


Figure 3. Correlation between GIoU and IOU for overlapping and non-overlapping samples.

👆不重叠样本，IoU=0，而GIoU为负值，有梯度

👆重叠样本，不断优化过程中GIoU$\rightarrow$IoU

使用IoU Loss L_{IoU}或L_{GIoU}训练，相比Sommoth-L1和MSE能带来性能提升

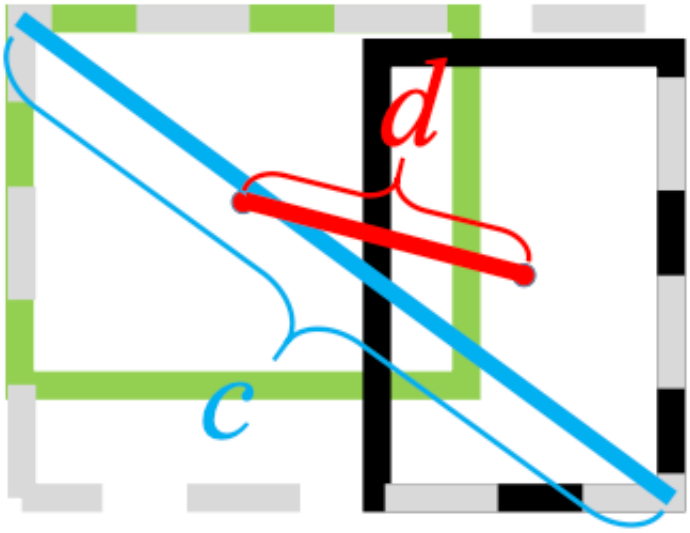
DIoU (Distance)

> 好的bounding box regres. 标准需要考虑三个因素： **Overlap, Center Distance, Aspect Ratio**

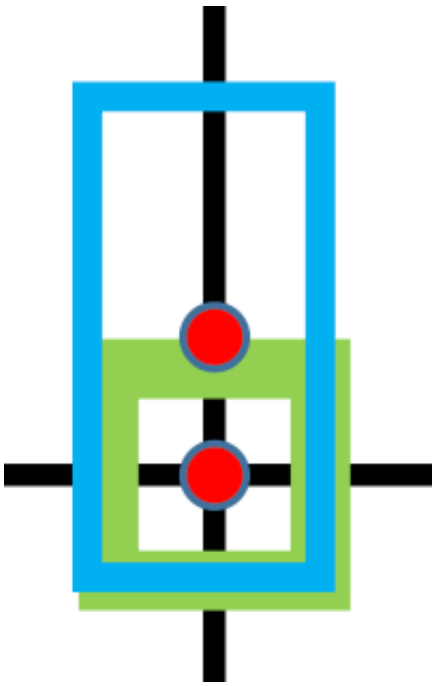
直接优化两个框中心点的距离

$$\mathrm{DIoU} = \mathrm{IoU} - \frac{\rho^2(b, b^{gt})}{c^2}$$

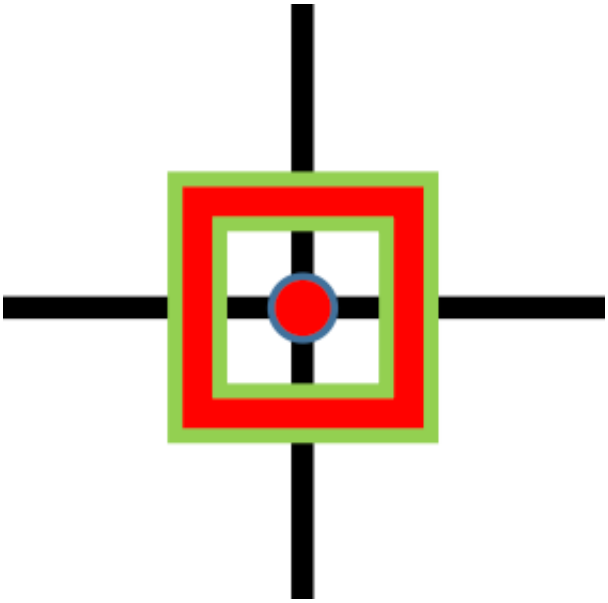
其中ρ (or 图中d)表示中心点的欧式距离（**L1可以吗？**），c表示包含两个框的最小闭包区域的对角线距离 👇

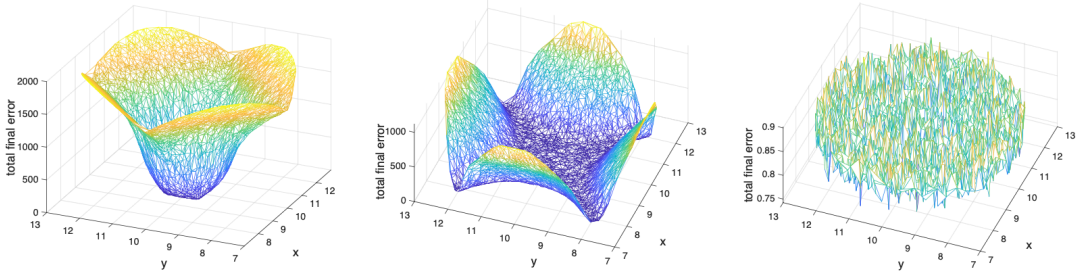


相比GIoU：GIoU更强调对齐，只要对齐之后没有梯度，如 👇，预测蓝色框和GT绿色对齐，没有缝隙，GIoU term=0「一框包括另一的情况」



而DIoU直接优化框中心的重合，距离近 👇





(a) $\mathcal{L}_{IoU}$ (b) $\mathcal{L}_{GIoU}$ (c) $\mathcal{L}_{DIoU}$

Figure 4: Visualization of regression errors of IoU, GIoU and DIoU losses at the final iteration T, i.e.. $\mathbf{E}(T, n)$ for every coordinate n. We note that the basins in (a) and (b) correspond to good regression cases. One can see that IoU loss has large errors for non-overlapping cases, GIoU loss has large errors for horizontal and vertical cases, and our DIoU loss leads to very small regression errors everywhere.

👆IoU不重叠loss高，GIoU正垂直水平方向loss高（如👆👆👆图），DIoU较低

DIoU也可用于NMS中DIoU-NMS

CIoU (Complete)

$$\mathrm{CIoU} = \mathrm{IoU} - \frac{\rho^2(b,b^{gt})}{c^2} - \alpha v$$

$$v = \frac{4}{\pi^2}(\arctan\frac{w^{gt}}{h^{gt}} - \arctan\frac{w}{h})^2$$

$$\alpha = \frac{v}{(1-\mathrm{IoU})+v}$$

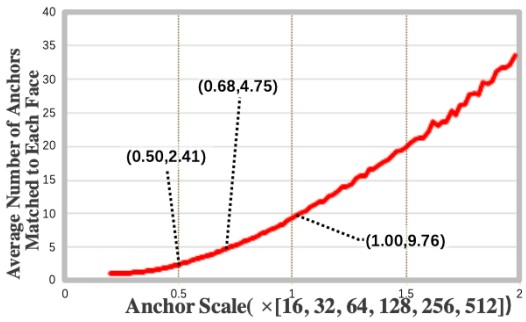
使用各种IoU loss训练👇

Loss / Evaluation	AP		AP75	
	IoU	GIoU	IoU	GIoU
$\mathcal{L}_{IoU}$	46.57	45.82	49.82	48.76
$\mathcal{L}_{GIoU}$	47.73	46.88	52.20	51.05
Relative improv. %	2.49%	2.31%	4.78%	4.70%
$\mathcal{L}_{DIoU}$	48.10	47.38	52.82	51.88
Relative improv. %	3.29%	3.40%	6.02%	6.40%
$\mathcal{L}_{CIoU}$	**49.21**	**48.42**	**54.28**	**52.87**
Relative improv. %	**5.67%**	**5.67%**	**8.95%**	**8.43%**
$\mathcal{L}_{CIoU}(D)$	**49.32**	**48.54**	**54.74**	**53.30**
Relative improv. %	**5.91%**	**5.94%**	**9.88%**	**9.31%**

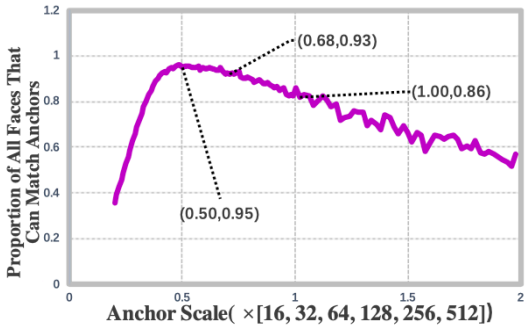
HAMBox: Delving into Online High-quality Anchors Mining for Detecting Outer Faces

> 在线匹配，先回归出框，再anchor-target匹配计算loss

实验/Motivation



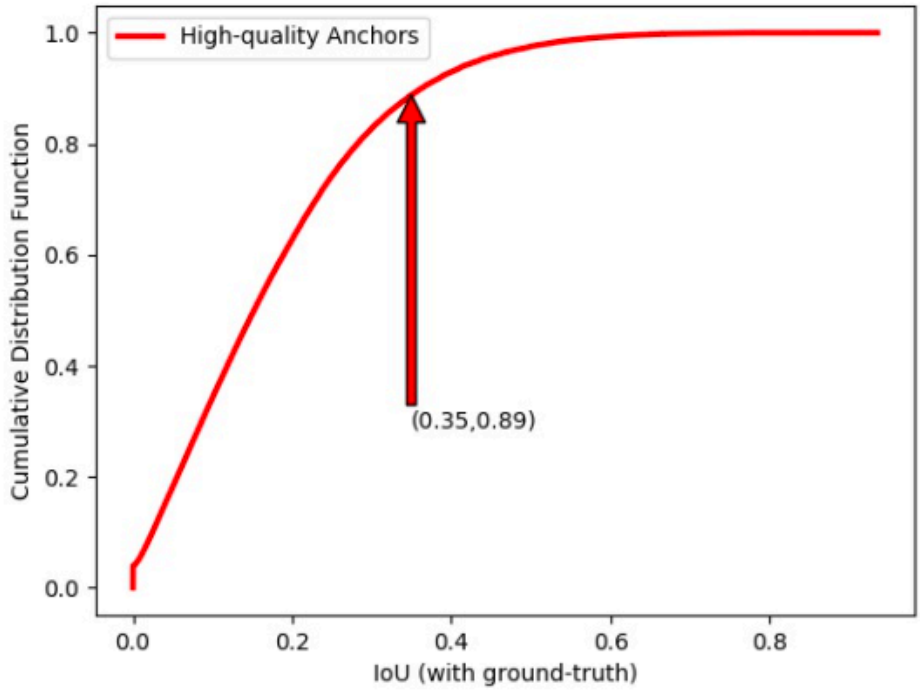
(a) Average Number of Anchors Matched to Each Face



(b) Proportion of Faces that can Match with Anchors

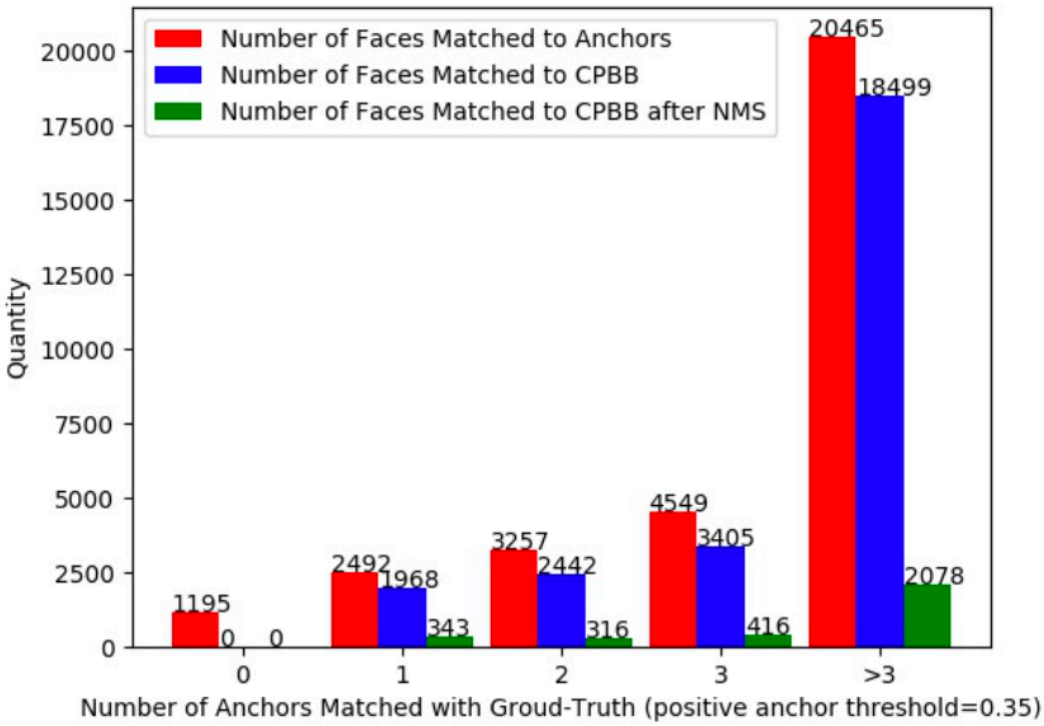
👆anchor大，每个人脸匹配到的数量变多，但是匹配到人脸占所有人脸中占比下降，人脸recall下降

👆anchor小，每个人脸匹配到的anchor数量下降，但是大多数人脸都有匹配anchor，人脸recall上升



(a) Cumulative Desity Curve of IoU

👆0.35为anchor和target match的threshold，所以89%的anchor都没有被match



(d) Performance of Matched High-quality Anchors

关键 纵坐标代表face数量

match 1个anchor的face有2492张，anchor能产生的正确预测框（`IoU>0.5`，Correctly Predicted Bounding Box）对应的人脸有1968张 → *大多数人脸都能通过anchor产生一个IoU高的预测框*

预测框经过NMS之后能保留下来的人脸只有343张 → *大多数人脸经过anchor产生的预测框都被NMS过滤掉，而导致这些face漏检*

但是NMS只删除一个位置重复的框(IoU过大)，对于漏检的人脸，只要有框cover，就一定会保留，NMS后同一个位置至少保留一个score最大的框 → NMS后导致漏检的1625张face，产生了CPBB(`IoU>0.5`)，但是NMS被删掉 → IoU足够大，但是得分太低（低于`cls_threshold`），NMS时过低score的忽略掉(不会考虑IoU) → *是由于训练的时候此anchor没有match，分类分支训练目标为BG，分类网络降低了score* 「本质为IoU和score的mismatch」

👆 **结论：低IoU而没有被match的anchor也能产生很好的预测框(CPBB)**，需要被match为物体，提高其分类score。这些anchor负责的face多为outer face(难样例)，上述也为outer face漏检的原因（低IoU框被unmatch，无法训练，分类网络不能分类为高分）

HAMBox (Online High-quality Anchor Mining)

选择大anchor，通过OHAM来进行弥补没有anchor match的face

传统match策略：一个face/target首先match所有和它IoU大于threshold的anchor。此后，对于没有anchor和它IoU高于阈值的face，选择和它IoU最大的anchor匹配进行补充（only one）

OHAM： 1) match所有anchor IoU大于threshold的face，对于没有anchor匹配的face，不进行compensate. 2) 对所有框回归计算bbox. 3) 对所有没有匹配anchor的face，计算预测框和face的IoU，对其进行弥补，$\mathrm{IoU} > \mathrm{threshold}'$ 「没有匹配或匹配数量不足 (K anchor bag)」

计算Loss学习时，使用回归后的bbox和target匹配，来弥补用原始anchor和target匹配的数量不足问题

Regression-aware Focal Loss

$$L_{cls}\left(p_i\right) = \frac{1}{N_{com}} \sum_{i \in \psi} F_i L_{fl}\left(p_i, g_i^*\right)$$
$$+ \frac{1}{N_{norm}} \sum_{i \in \Omega} \left(1_{(l_i^*=0)} 1_{(F_i<0.5)} + 1_{(l_i^*=1)}\right) L_{fl}\left(p_i, l_i^*\right)$$

其中ψ表示弥补的anchor，N_{com}为数量，p预测label，g=gt

Ω为matched anchor和unmatched low-quality (IoU<0.5) anchor「即unmatched hq anchor不进行训练，应该hq仍未被match表示简单样例上多余(>K)的anchor」，N_{norm}为数量，F_i表示IoU，l_i^*表示第一次match的label

$$L_{loc}\left(x_i\right) = \frac{1}{N_{com}} \sum_{i \in \psi} L_{SmoothL1}\left(x_i, x_i^*\right)$$
$$+ \frac{1}{N_{norm}} \sum_{i \in \Omega} L_{SmoothL1}\left(x_i, x_i^*\right)$$

Algorithm 1 Online high-quality anchor mining

Input: B, G, T, K, D, L, R, A

B is a set of regression bounding boxes, in the form of (x_0, y_0, x_1, y_1).

X is a set of ground-truth, in the form of (x_0, y_0, x_1, y_1)

T is an online anchor mining threshold (see details on Subsection 3.2)

K is a hyperparameter and represents the max number of anchors that F_{outer} can be matched with.

D is a Dict, key is ground-truth, item is the number of anchors that ground-truth can match in the first step of our HAMBox method.

L is a Dict, key is anchor index, item is a label that anchor index is assigned with in the final process of our HAMBox method.

R is a Dict, key is anchor index, item is encoded coordinates of the key during standard anchor matching strategy.

A is a Dict, key is anchor index, item is coordinates of

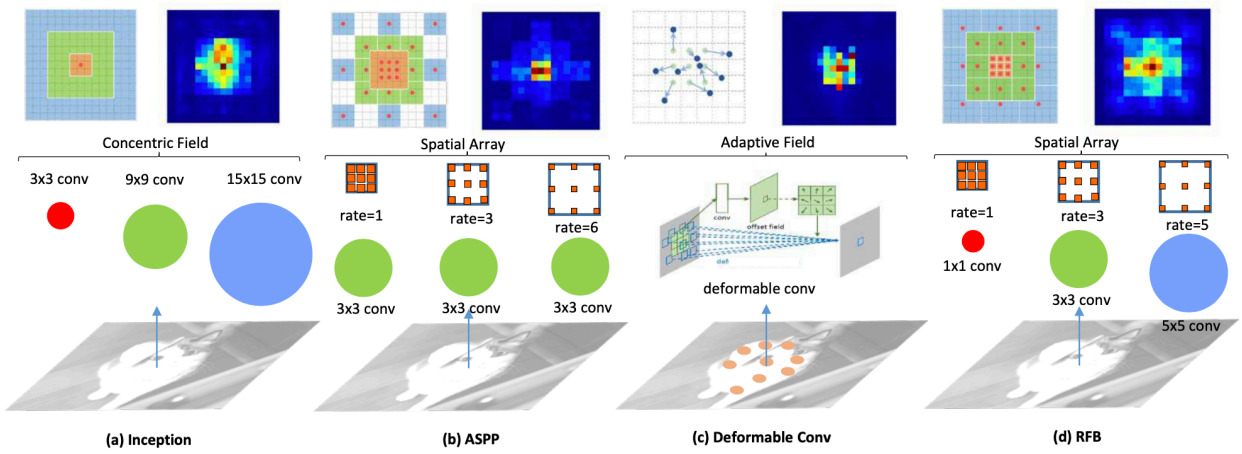
A is a Dict, key is anchor index, item is coordinates of the key, in the form of (x_0, y_0, x_1, y_1).

Output: R and L after using our HAMBox method.

for x_i in X **do**
  if $D(x_i) >= K$ **then**
    $continue$
  end if
  $CompensatedNumber = K - D(g_i)$
  $OnlineIoU \Leftarrow IoU(x_i, B), AnchorIdx$
  $SortedOnlineIoU = sorted(OnlineIoU, key = IoU, reverse = True)$
  for $IoU, AnchorIdx$ in $SortedOnlineIoU$ **do**
    if $L(AnchorIdx) = 1$ **then**
      $continue$
    end if
    $CompensatedNumber$ -= 1
    $L(AnchorIdx) = 1$
    $R(AnchorIdx) = encode(A(AnchorIdx), ground\text{-}truth)$
    if $CompensatedNumber = 0$ **then**
      $break$
    end if
  end for
end for
Return R, L

[RFB-Net] Receptive Field Block Net for Accurate and Fast Object Detection

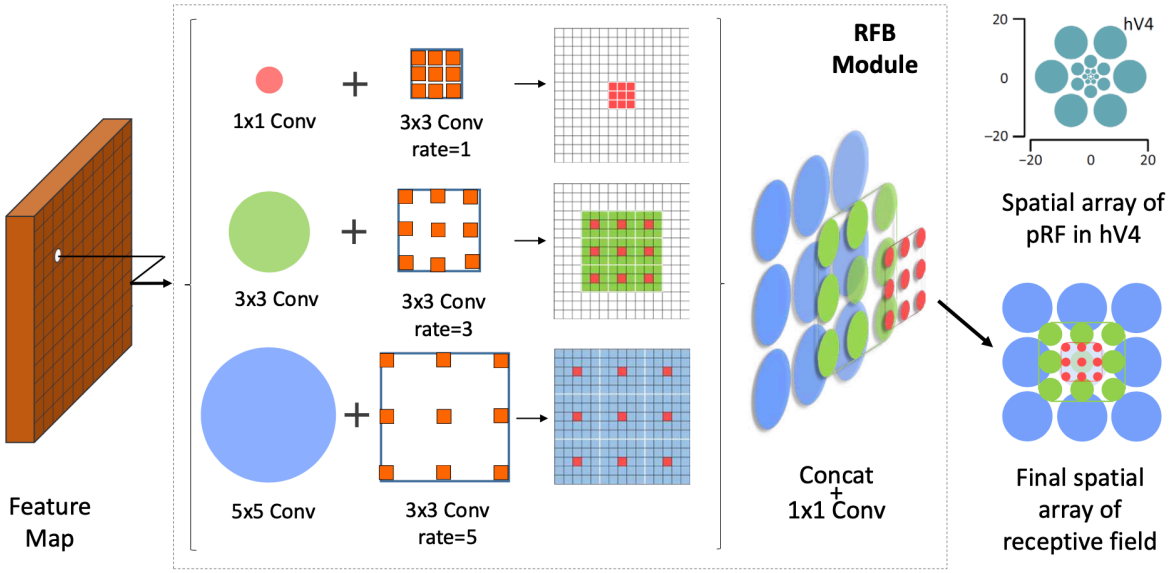
> 不同感受野，对应不同扩张(dilation)
>
> 不是多尺度的特征图，而是不同感受野大小的特征图



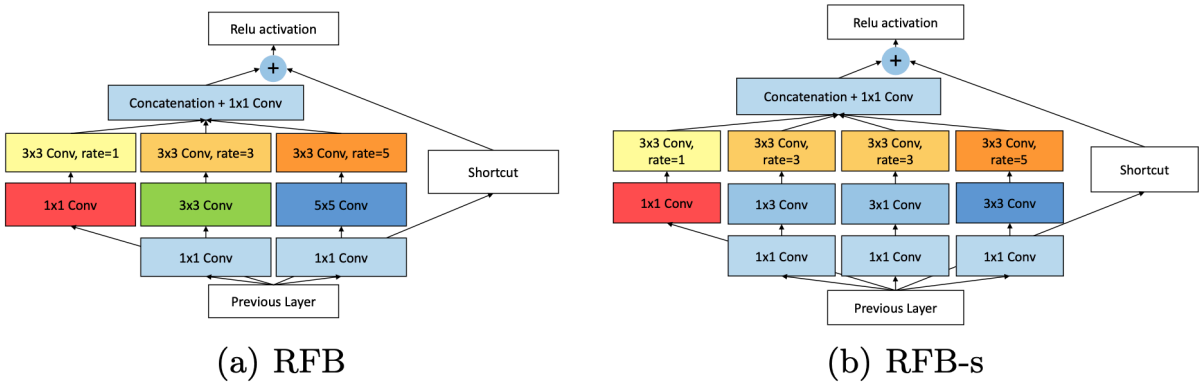
之前只变感受野(receptive field/kernel size)，或者只变扩张尺度(ASPP)

RFB提出感受野和扩张尺度应该同时变化「相互影响」

👇圈只表示感受野大小，大的kernel对应大的dilate，使感受野更大



实现上



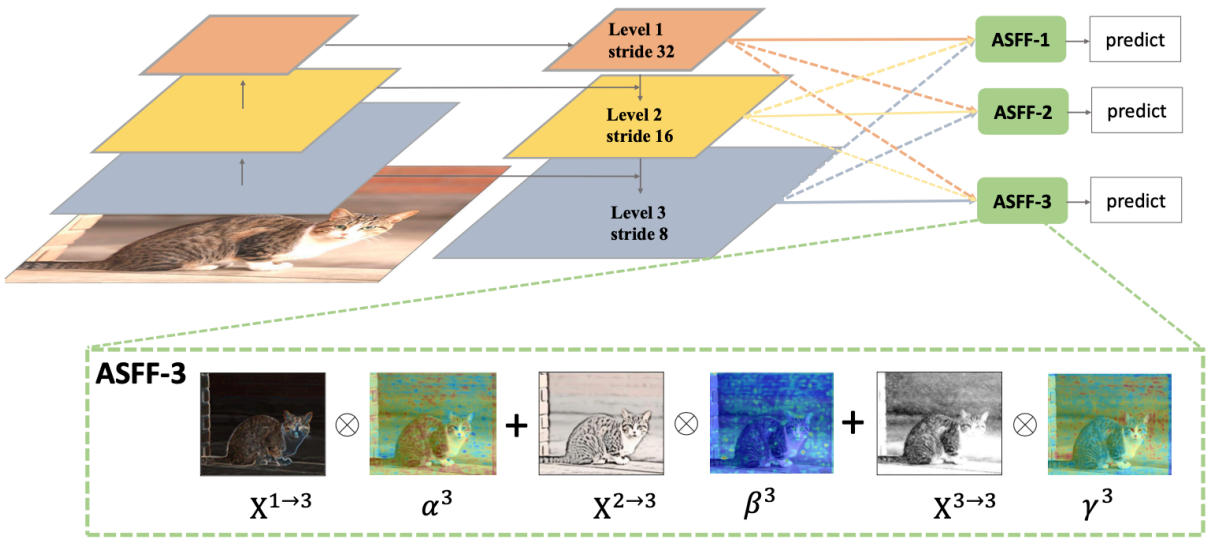
使用两个 `3x3` 代替 `5x5` 。**注意padding，所有都为same size(k=3, p=1)，每个分支产生的特征图大小相同**

```
1    self.branch0 = nn.Sequential(
```

```
                Conv(in_planes, 2*inter_planes, kernel_size=1, stride=stride),
                Conv(2*inter_planes, 2*inter_planes, kernel_size=3, stride=1,
padding=visual, dilation=visual, relu=False)
                )
        self.branch1 = nn.Sequential(
                Conv(in_planes, inter_planes, kernel_size=1, stride=1),
                Conv(inter_planes, 2*inter_planes, kernel_size=(3,3),
stride=stride, padding=(1,1)),
                Conv(2*inter_planes, 2*inter_planes, kernel_size=3, stride=1,
padding=visual+1, dilation=visual+1, relu=False)
                )
        self.branch2 = nn.Sequential(
                Conv(in_planes, inter_planes, kernel_size=1, stride=1),
                Conv(inter_planes, (inter_planes//2)*3, kernel_size=3, stride=1,
padding=1),
                Conv((inter_planes//2)*3, 2*inter_planes, kernel_size=3,
stride=stride, padding=1),
                Conv(2*inter_planes, 2*inter_planes, kernel_size=3, stride=1,
padding=2*visual+1, dilation=2*visual+1, relu=False)
                )
```

[ASFF] Learning Spatial Fusion for Single Shot Object Detection

> 多尺度特征图融合



特征首先经过resize，再融合。 *resize可使用deconv/conv, 插值/pooling*

$$\mathbf{y}_{ij}^{l} = \alpha_{ij}^{l} \cdot \mathbf{x}_{ij}^{1\to l} + \beta_{ij}^{l} \cdot \mathbf{x}_{ij}^{2\to l} + \gamma_{ij}^{l} \cdot \mathbf{x}_{ij}^{3\to l}$$

Where $\alpha_{ij}^{l} = \frac{e^{\lambda_{\alpha_{ij}}^{l}}}{e^{\lambda_{\alpha_{ij}}^{l}} + e^{\lambda_{\beta_{ij}}^{l}} + e^{\lambda_{\gamma_{ij}}^{l}}}$, etc

and $\lambda_\alpha^l, \lambda_\beta^l, \lambda_\gamma^l$ computed (`1x1` conv) from $\mathbf{x}^{1\to l}, \mathbf{x}^{2\to l}, \mathbf{x}^{3\to l}$ **respectively**

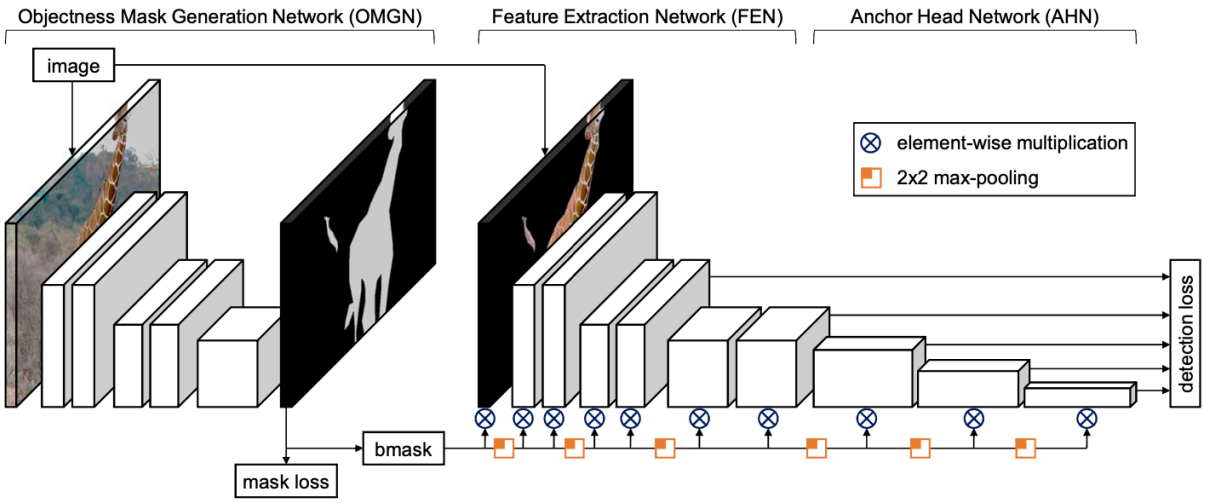
可以看作产生一个框feature pyramid多个特征图都用到，之前只用一个特征图产生一个框

训练tricks： mixup algorithm, cos lr, sync bn, bag of freebies

Accelerating Object Detection by Erasing Background Activation

> Objectness-aware object detection, 产生FG/BG的mask， 只对mask区域计算

图片只有小部分有物体，背景区域不需要特征提取计算， **只对前景mask区域计算特征**，分类&回归(次要)出bbox



OMG网络为Fast-SCNN 👇

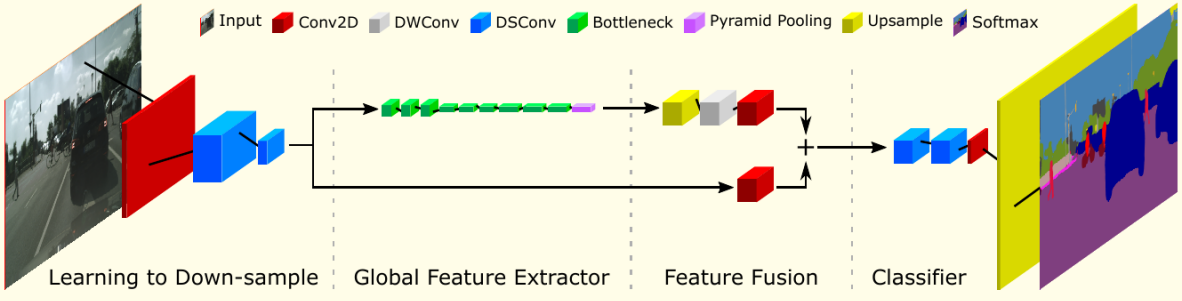


Figure 1. Fast-SCNN shares the computations between two branches (encoder) to build a above real-time semantic segmentation network.

使用element-wise mul来zero-out背景的feature map

OMG中有 `argmax` 操作，为了end2end training， 可以1) 使用 `soft-argmax` 代替 `argmax` 训练 2) 使用surrogate gradient，FP使用 `argmax` ，BP时使用 `soft-argmax` 代替获得近似梯度

$$\text{soft-argmax}(x) = \sum_i \frac{e^{\beta x_i}}{\sum_j e^{\beta x_j}} i$$

实验： 对比MAC，使用不同mask监督

Libra R-CNN: Towards Balanced Learning for Object Detection

> 训练方式，不平衡问题：hard example IoU分布不平衡，multi-level/res feature融合不平衡，不同loss样本产生的梯度不平衡
>
> 有梯度反推loss函数设计

目标检测器训练目标：

1. Selected region samples are representative
2. Extracted visual feature are fully utilized
3. Designed objective function is optimal

常见训练有三层次的imbalance

1. Sample-level: hard example需要多训练，但OHEM对噪声敏感
2. Feature-level: pyramid不同level/res的特征处理深度不同，高层处理多，浅/底层特征处理少
3. Objective-level: cls/reg两个任务损失函数协调

IoU-balanced Sampling

根据样本和GT的IoU，分成多个bin，每个bin均匀采样

第k bin中每个样本采样概率$p_k = \frac{N}{K} * \frac{1}{M_k}, \ k \in [0, K)$

样本各IoU均匀分布

Balanced Feature Pyramid

使用同样深的网络来处理不同层的特征

resize不同层feature，**取平均**；使用Gaussian **non-local attention**增强融合的特征。在resize会原先的尺度增强multi-scale特征👇

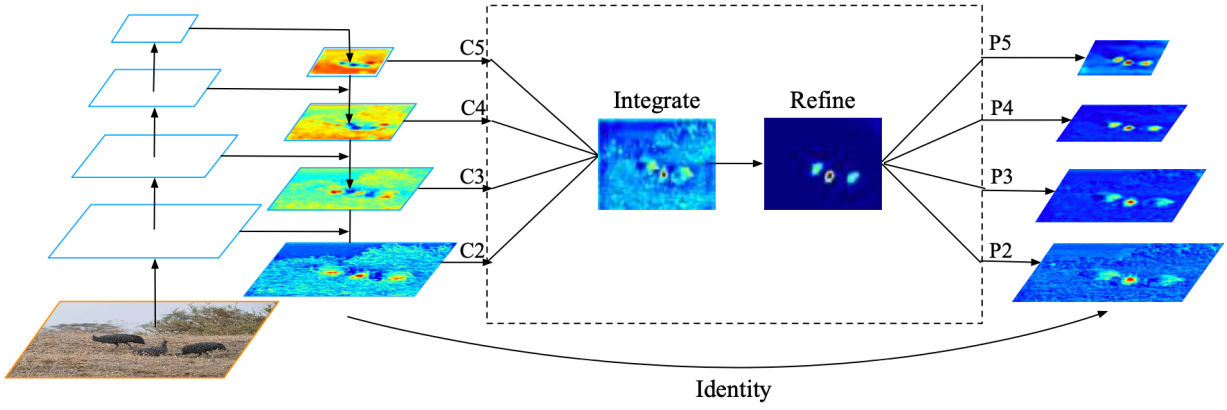
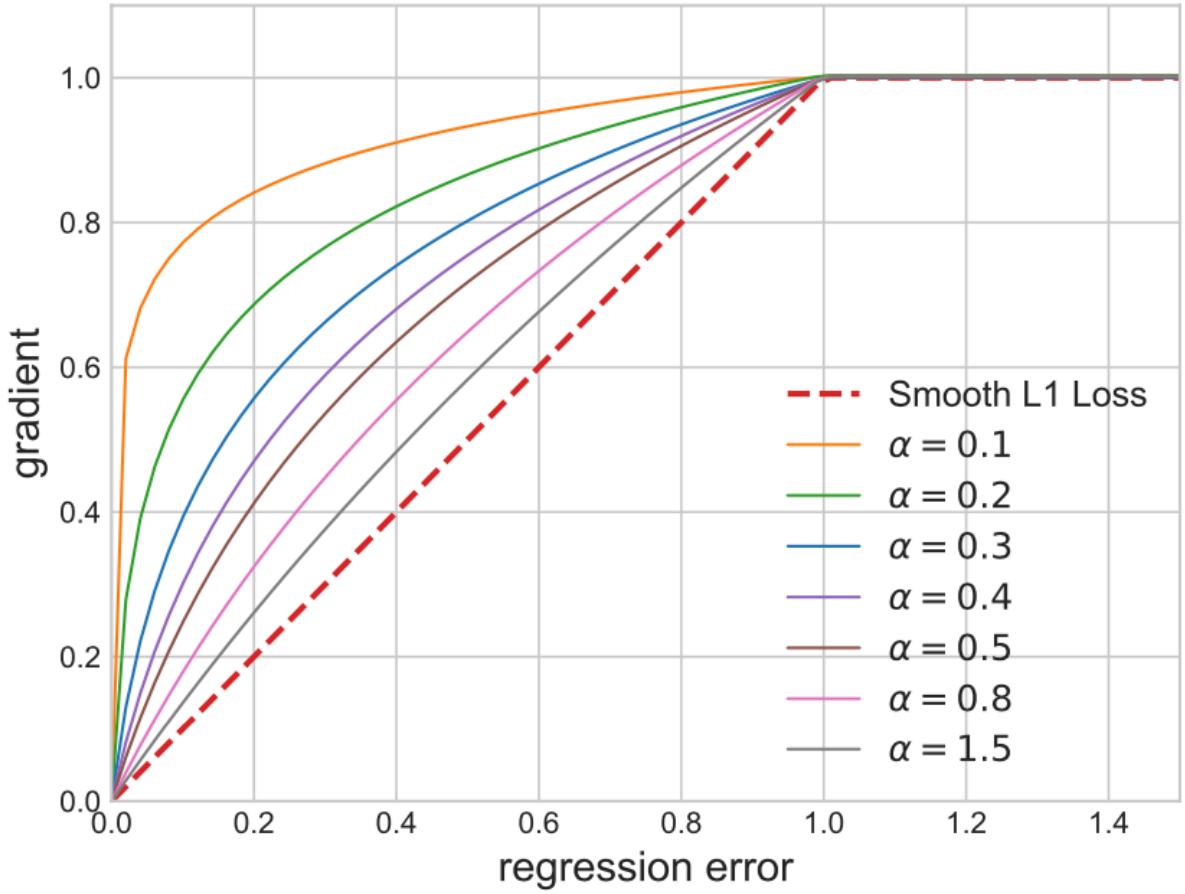


Figure 4: Pipeline and heatmap visualization of balanced feature pyramid.

Balanced L1 Loss

从梯度的角度考虑 promoting the crucial gradient: 精确的样本(inlier)的梯度更重要，增强loss小的样本的梯度(正确的梯度，数量少要增强)，减弱loss大样本的梯度(难训练，大梯度导致训练不稳定) 👇



$$\frac{\partial L_b}{\partial x} = \begin{cases} \alpha \ln(b|x|+1) & \text{if } |x| < 1 \\ \gamma & \text{otherwise} \end{cases}$$

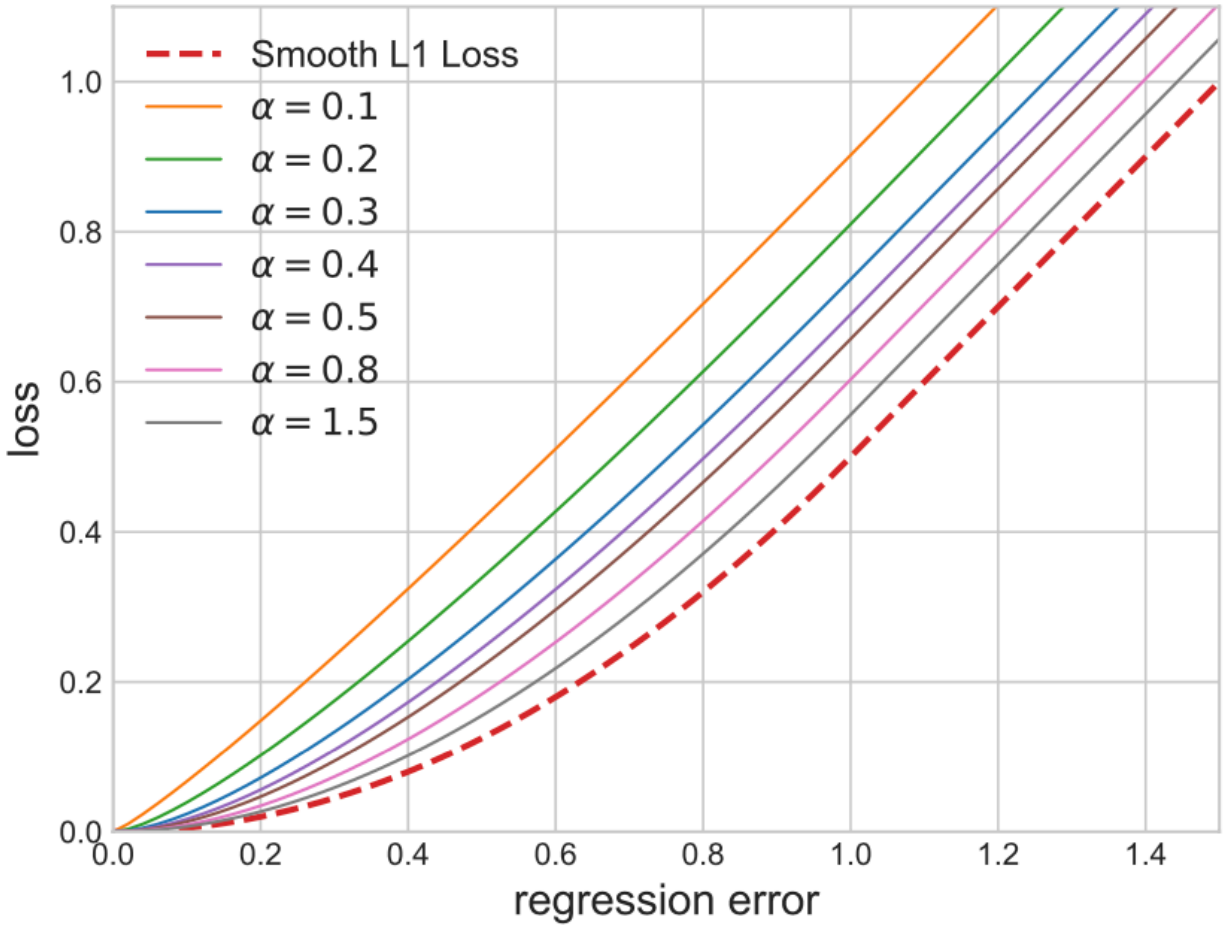
👆增强小loss的梯度，大loss的梯度clip。**大小loss样本产生的梯度平衡**

$(x, y) \rightarrow Loss \rightarrow gradient$，**通过分析梯度，反推回loss设计**

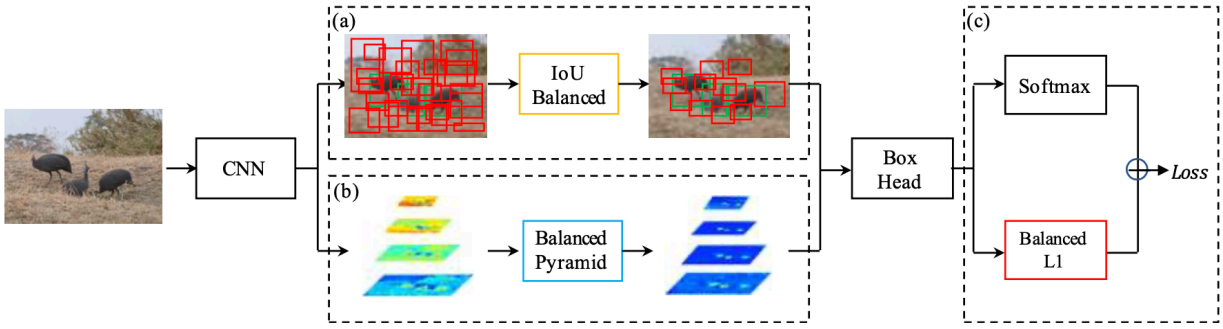
积分可得 👇

$$L_b(x) = \begin{cases} \frac{\alpha}{b}(b|x|+1)\ln(b|x|+1) - \alpha|x| & \text{if } |x| < 1 \\ \gamma|x| + C & \text{otherwise} \end{cases}$$

其中x为GT和pred的bbox坐标差距，γ为clip界 (大于γ，梯度恒定为1)，α控制对小loss的梯度增强，b为平衡项，求出每个位置loss后mean或sum，即$L_{loc} = \sum_{i \in \{x,y,w,h\}} L_b\left(t_i^u - v_i\right)$ 👇



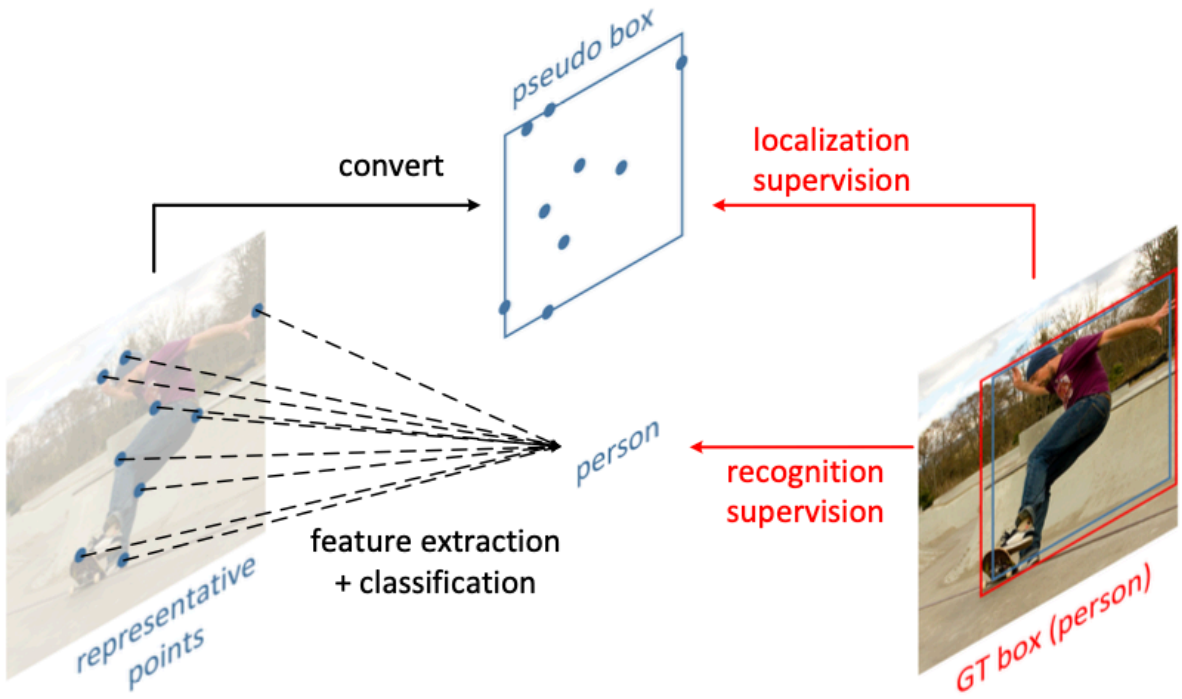
Pipeline



RepPoints: Point Set Representation for Object Detection

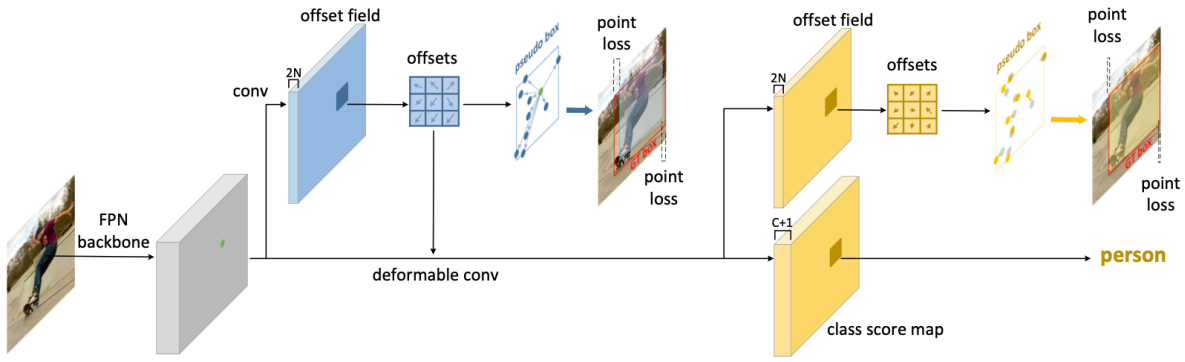
> Bbox特征不对齐问题，提供更好的object表示方法(不是中心点领域卷积)
>
> Deformable conv升级版，Representative Points
>
> 相比deformable，**直接使用变形后的结果作为预测的bbox位置（or anchor的偏移）**，而不是explicit回归xywh。中心点 + implicit-offsets + wh
>
> **采样点同时用来提取语义对齐的特征，又用来表示物体的几何形态**

大部分为背景，需要选择更representative的点求特征



👆**训练**：分类网络采用deformable conv即可，回归网络先把reppoints表示转为bbox表示(pseudo box)，然后计算和GT的offset（point loss）

转换方式：1) RepPoint set坐标最值 2) subset坐标最值 3) 使用mean和deviation估计，μ估计中心点，σ估计尺度(wh)



👆使用网络计算偏移量(offsets over the center points)，得到reppoint的点/物体特征表示 $\mathcal{R}=\{(x_k,y_k)\}_{k=1}^n$，n个点特征表示sample points/object的特征。学习$\mathcal{R}_r=\{(x_k+\Delta x_k,y_k+\Delta y_k)\}_{k=1}^n$

👆维护两个RepPoint set，二次refine

Learned via weak localization supervision from rectangular ground-truth boxes and implicit recognition feedback

使用基于中心点xy而不是xywh预测bbox，减少hypothesis space，一次只需要预测2D vec，更好训练

RPDet

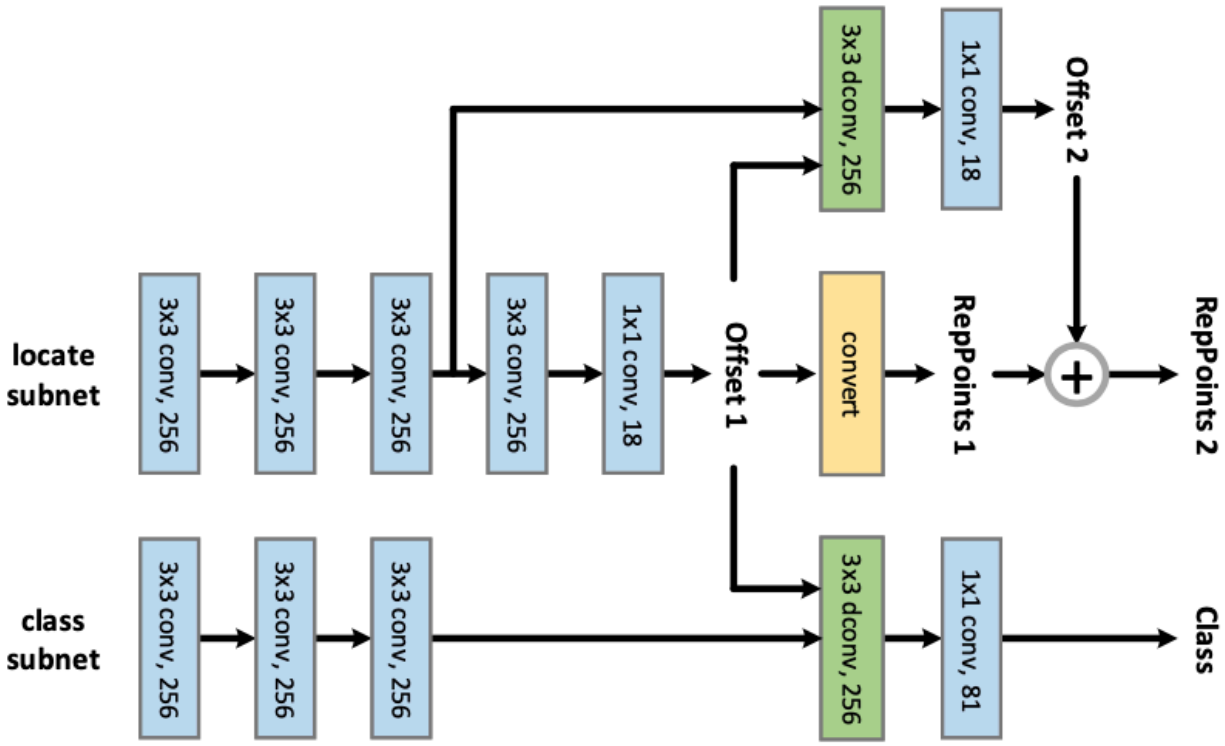


Figure 3. The head architecture of RPDet.

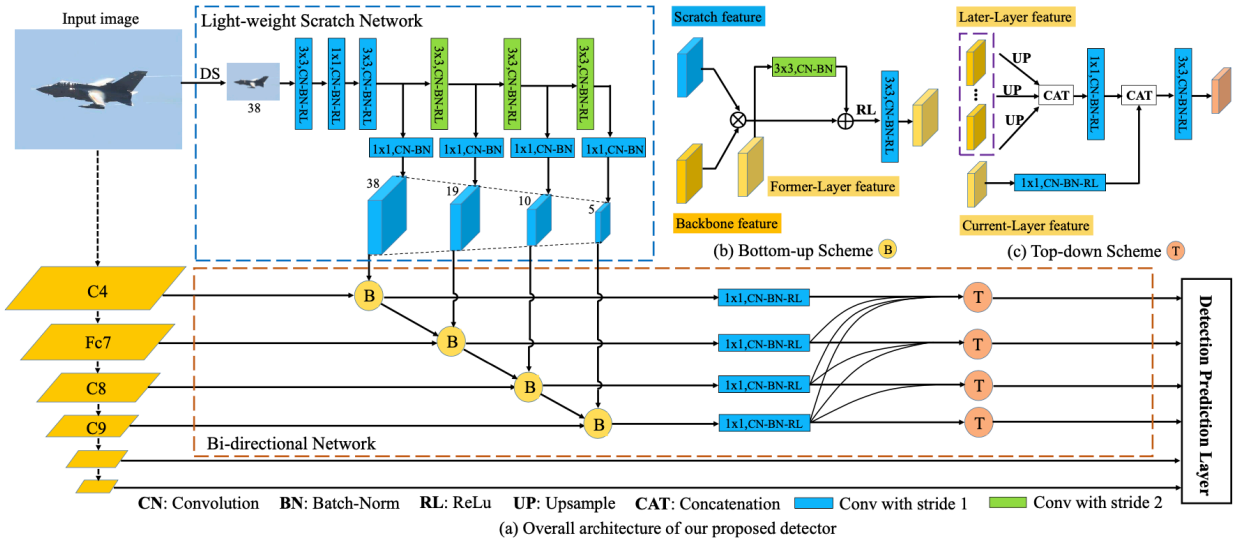
👆backbone FPN (多尺度即可解决基于点预测的重合物体问题，同FCOS)

👆两个分支：locate分支两次计算offset refine，class分支用offset变型卷积（dconv=deformable conv）

Ref: https://www.zhihu.com/question/322372759/answer/798327725

Learning Rich Features at High-Speed for Single-Shot Object Detection (LSN)

> 分类任务预训练和检测任务gap，feature pyramid融合
>
> Light-weight Scratch Network产生准确底层特征输入FPN
>
> 底层特征高层特征双向传播



(a) Overall architecture of our proposed detector

LSN

输入为downsample后的原始图片，低分辨率浅层网络 **train from scratch**

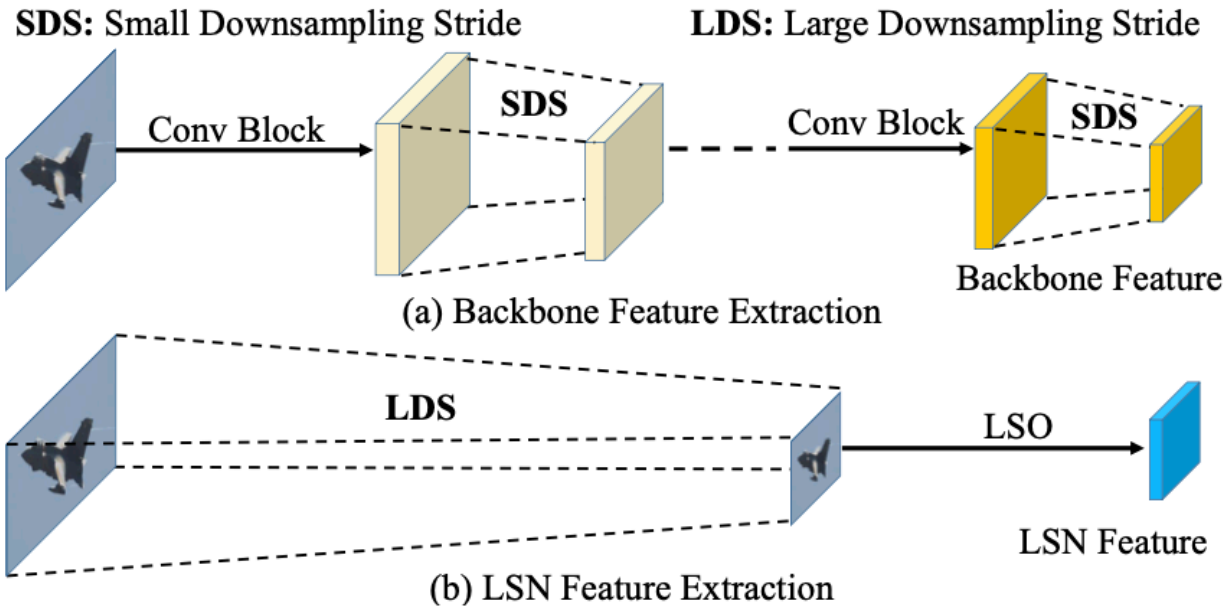


Figure 4: (a) Standard SSD feature extraction employs several convolution blocks together with small downsampling strides. (b) In our LSN, the input image is first downsampled to the target size followed by light-weight serial operations (LSO) to produce LSN features.

Bi-directional Network

Bottom-up Scheme 👆👆 (b)： $f_k = \phi_k\left((s_k \otimes o_k) \oplus (w_{k-1} f_{k-1})\right)$

s_k为LSN的特征输出，o_k为SSD (baseline)的输出，f_{k-1}为上一层特征，cascade依次计算

Top-down Scheme👆👆(c)：$b_k = \gamma_k \left(\sum \left(W_k f_k, W_{mk} \left(\sum_{k+1}^{n} \mu_k \left(W_i f_i \right) \right) \right) \right)$

W_k为 `1x1` conv降通道，W_{mk}为 `1x1` conv融合特征，dense融合所有上层特征(low-res)，μ_k为 `upsample`

先经过bottom-up再top-down，用top-down输出预测

Experiment

SSD	LSN	Bi-directional	AP	AP_{50}	AP_{75}	AP_s	AP_m	AP_l
✓			25.3	42.0	26.5	6.2	28.0	43.3
✓	✓		28.9	47.8	30.2	10.6	32.1	44.8
✓	✓	✓	**31.9**	**51.4**	**33.6**	**13.4**	**36.3**	**47.6**

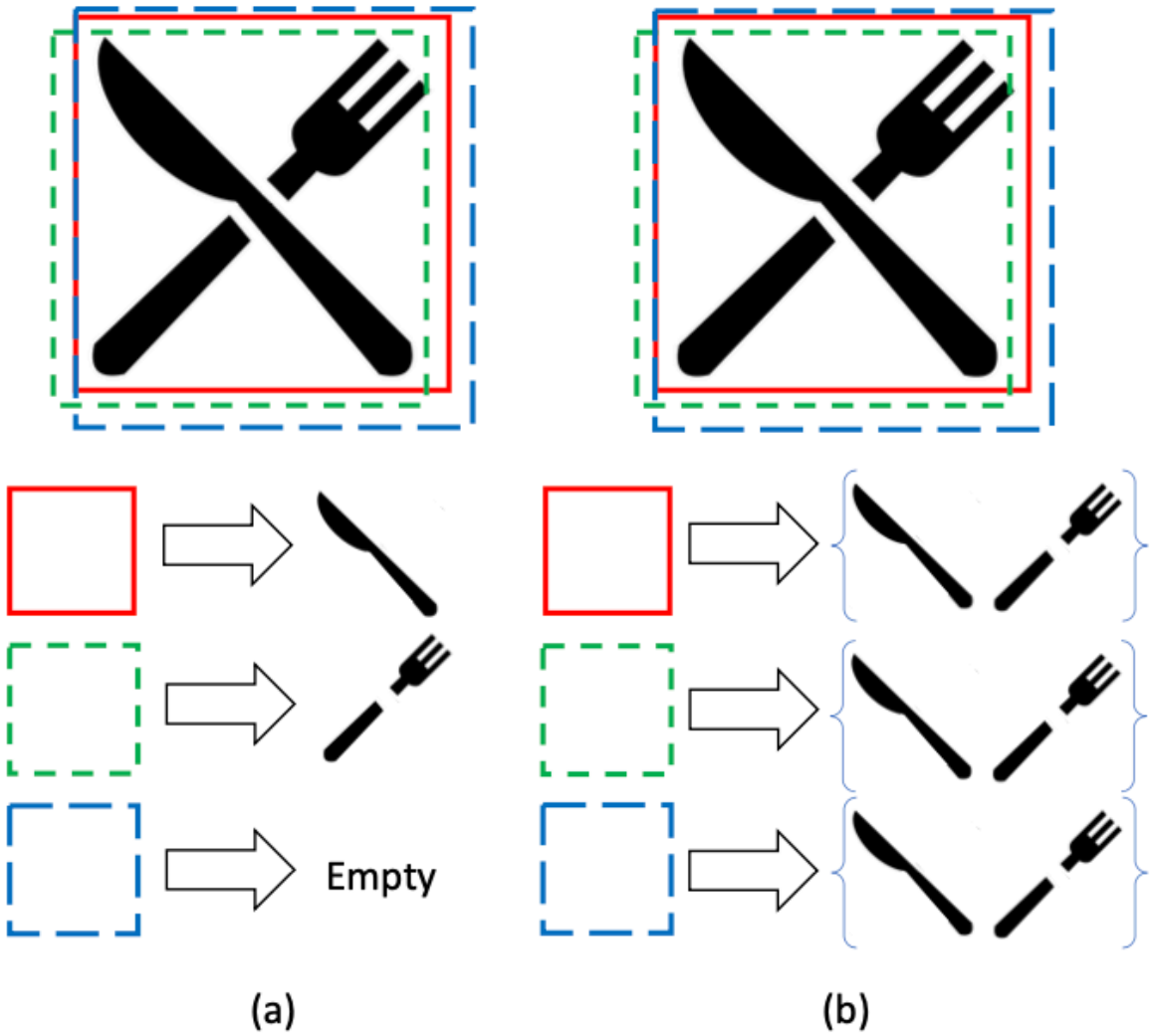
Table 2: Impact of integrating our different components (light-weight scratch network (LSN) and bi-directional network) in the standard SSD on MS COCO minival dataset. Our final detection framework improves the performance with an overall gain of 6.6% over the standard SSD.

Bottom-up Scheme	Top-down Scheme	AP	Time (ms)
Cascade	Cascade	31.0	**12**
Dense	Dense	31.3	16
Dense	Cascade	29.6	15
Cascade	Dense	**31.9**	13

Table 3: Impact of different design choices when constructing our bi-directional network. We obtain optimal performance in terms of speed and accuracy when using cascade connections for bottom-up scheme and dense connections for top-down scheme in our bi-directional network.

Detection in Crowded Scenes: One Proposal, Multiple Predictions

> 一个anchor/候选框负责预测多个物体。anchor-GT 一对多
>
> 密集行人检测。密集，重叠/遮挡



之前：一个anchor负责预测一个物体；提出：一个anchor预测一组

对于一个anchor/prior/proposal b_i，预测的GT：$\mathrm{G}\left(b_i\right)=\left\{g_j \in \mathcal{G} \mid \mathrm{IoU}\left(b_i, g_j\right) \geq \theta\right\}$

预测为set：$\mathrm{P}\left(b_i\right)=\left\{\left(\mathbf{c}_i^{(1)}, \mathbf{l}_i^{(1)}\right),\left(\mathbf{c}_i^{(2)}, \mathbf{l}_i^{(2)}\right), \ldots,\left(\mathbf{c}_i^{(K)}, \mathbf{l}_i^{(K)}\right)\right\}$，$c_i^(j)$表示第$i$个anchor预测的第$j$个框的类别和置信度，$l$为位置

匹配时K个物体，但预测时仍可能部分预测结果为背景「至多预测K个结果」 *是否可以扩展为预测更多?*

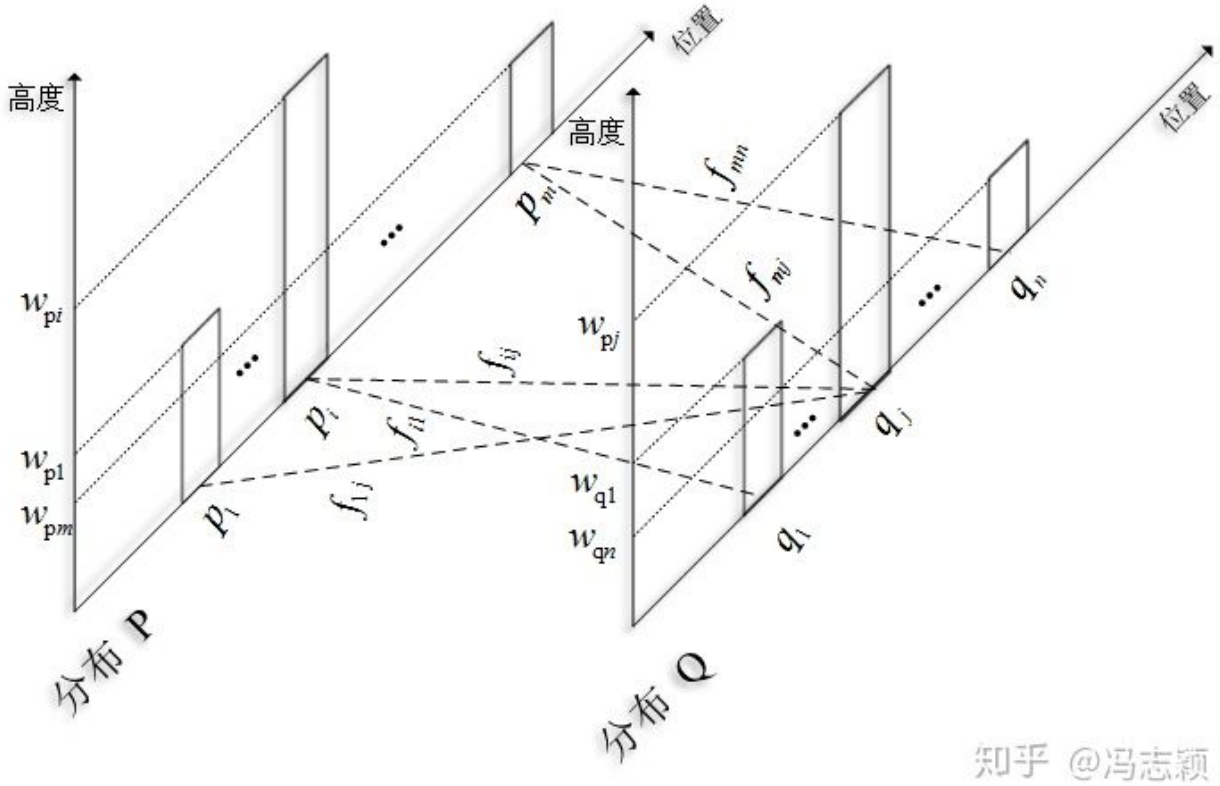
训练看作最小化预测集和GT集之间的推土机距离 $\mathrm{EMD}(P(b_i), G(b_i))$，和集合中位置无关，与分布有关

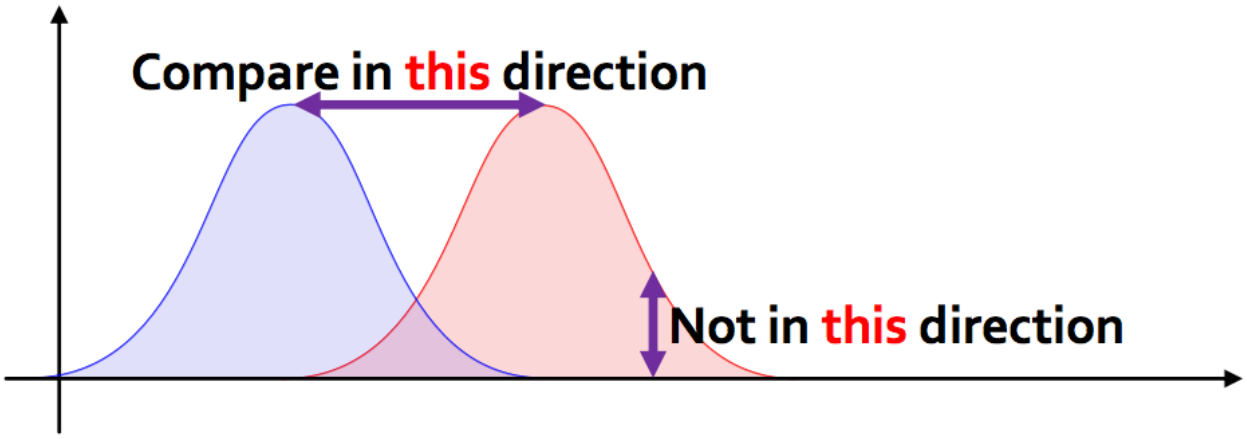
$\mathcal{L}\left(b_i\right)=\min _{\pi \in \Pi} \sum_{k=1}^K\left[\mathcal{L}_{cls}\left(\mathbf{c}_i^{(k)}, g_{\pi_k}\right)+\mathcal{L}_{reg}\left(\mathbf{l}_i^{(k)}, g_{\pi_k}\right)\right]$

预测的背景box计算$\mathcal{L}_{cls}$不计算$\mathcal{L}_{reg}$

推土机距离 (Earth Mover's Distance, Wasserstein)

两个分布间距离：从一个分布变化到另一个分布所需要的最小做功





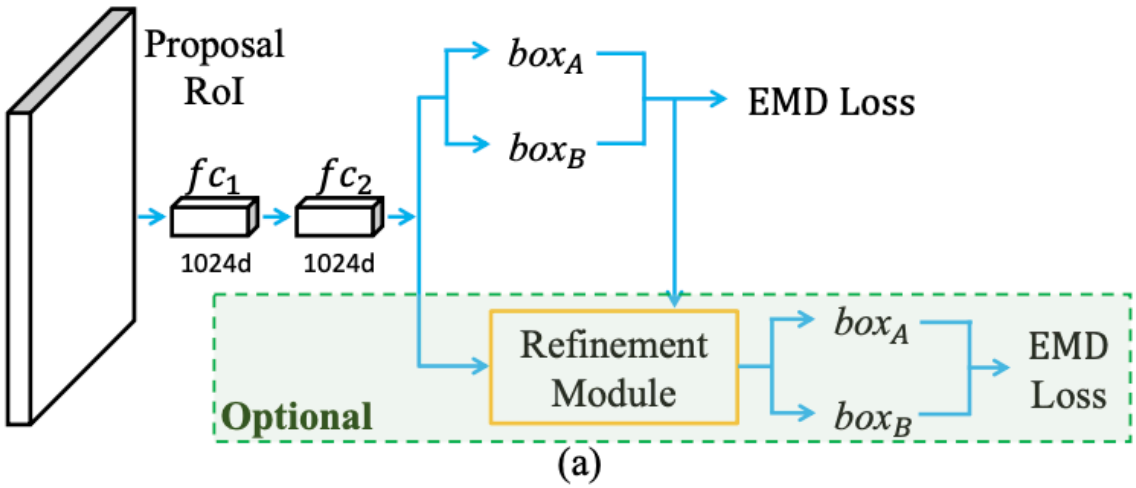
Ref: https://jeremykun.com/2018/03/05/earthmover-distance/ , https://zxth93.github.io/2017/09/27/KL散度JS散度Wasserstein距离 , https://zhuanlan.zhihu.com/p/74075915

Set NMS

一个anchor预测的多个物体是unique的，重复预测只可能出现在不同anchor预测集之间

NMS时增加：如果两个pred-box出自同一个anchor，则不进行抑制

网络架构

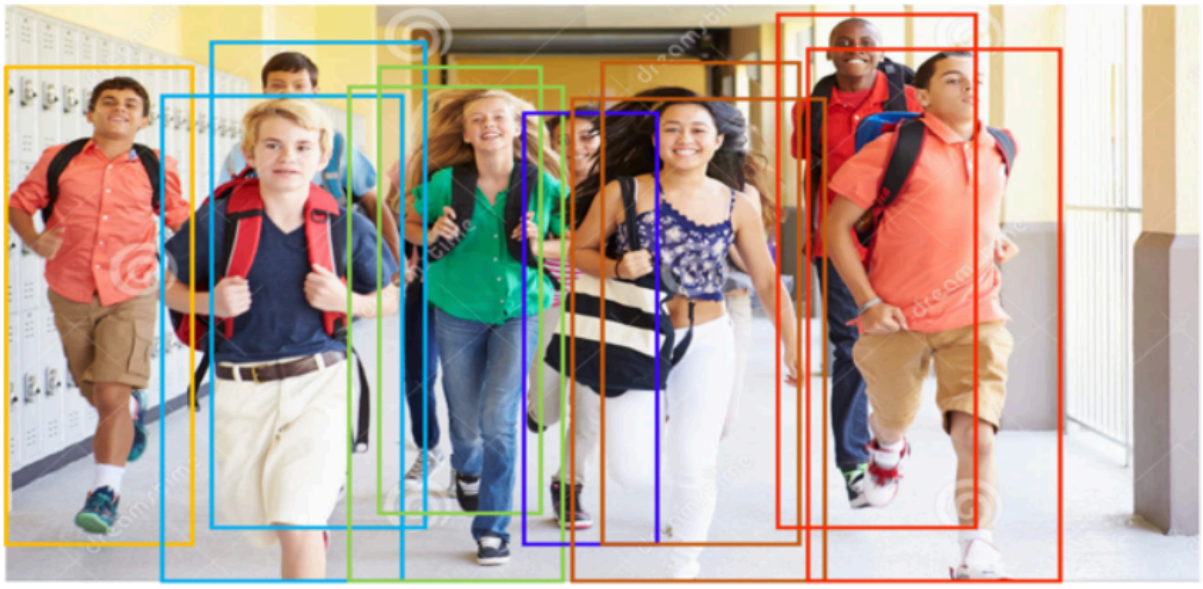


(a)

`FPN`，增加`RoIAlign`

更多预测，可能出现更多False positive，可增加`Refinement Module`进行二次预测refine

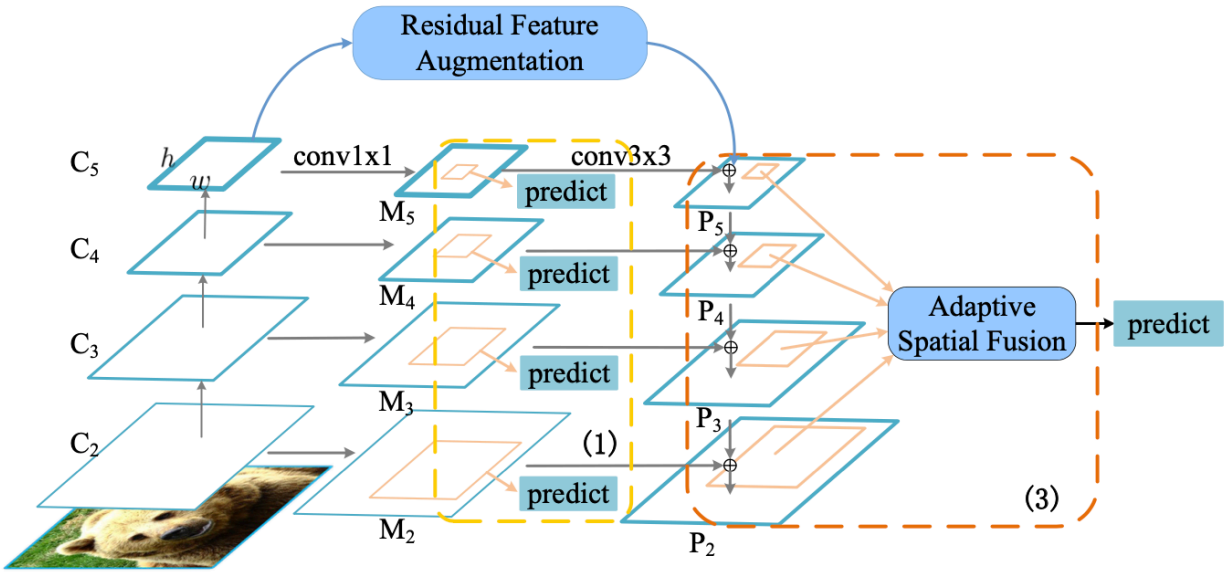

(a) Baseline


(b) Ours

Cityperson, CrowdHuman效果好

AugFPN: Improving Multi-scale Feature Learning for Object Detection

> FPN改进，特征融合



Consistent Supervision

不同尺度的特征图有semantic gap，增加一个监督信号来限制学习到的特征的差异

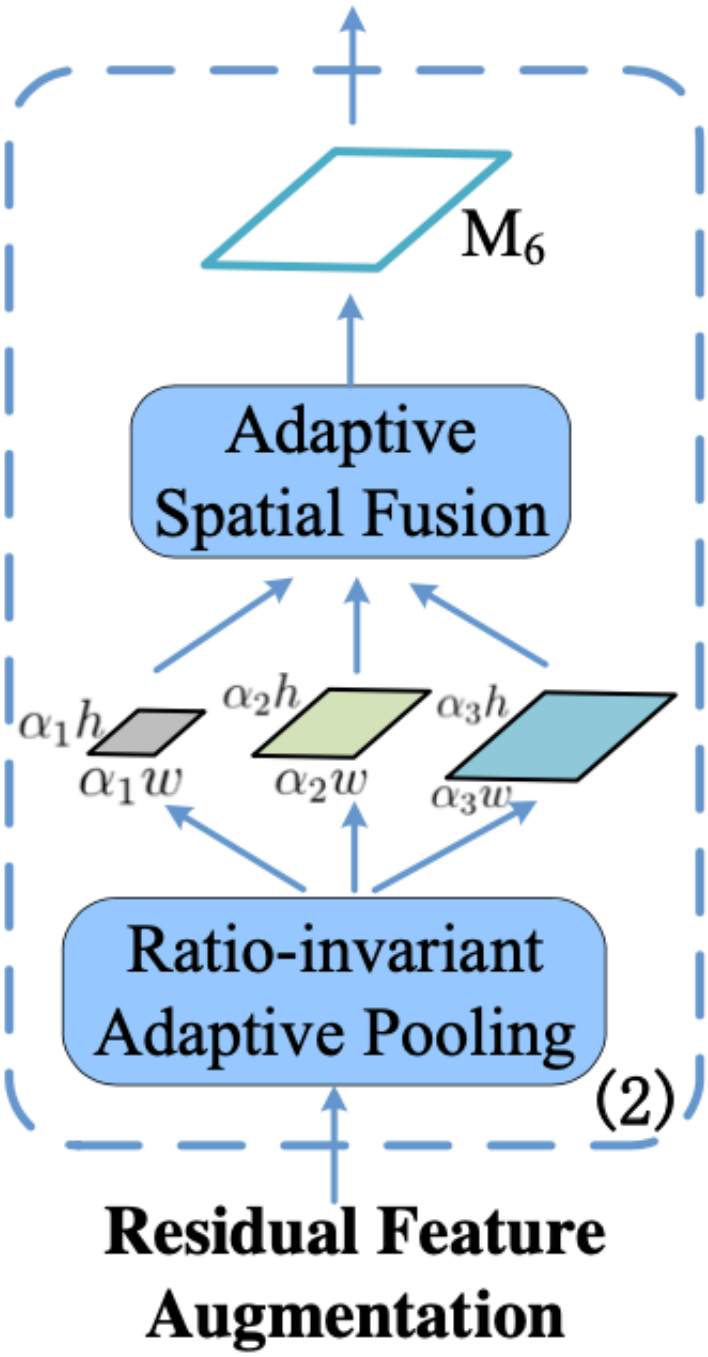
增加多个共享权重的预测头(detect head)对不同尺度特征图($M_{1..5}$)上的proposal进行预测，加监督信号「**multi-head prediction**」

$$L_{rcnn} = \lambda \left(L_{cls,M}\left(p_M, t^*\right) + \beta\left[t^* > 0\right] L_{loc,M}\left(d_M, b^*\right) \right) + L_{cls,P}\left(p, t^*\right) + \beta\left[t^* > 0\right] L_{loc,P}\left(d, b^*\right)$$

其中p_M，d_M表示中间层的预测，p，d表示最终层的预测，t^*，b^*表示GT的label和box

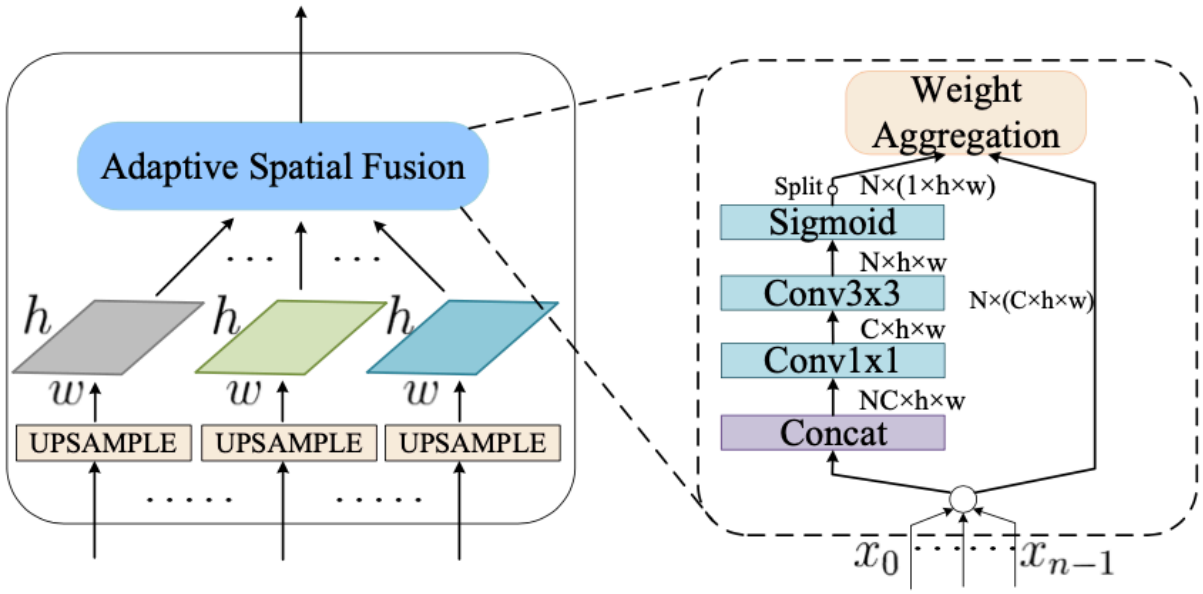
Residual Feature Augmentation

最高层特征没有上层特征与其融合。采用不同尺度的C_5特征进行组合得到M_6，并融合到M_5中，来**增强最高层特征**



`Ratio-invariant Adaptive Pooling` 为把C_5 @S pooling到不同尺度$(\alpha_1 \times S, \alpha_2 \times S, \ldots, \alpha_n \times S)$

其中 `Adaptive Spatial Fusion` 为



(a)

Soft RoI Selection

two RoIs with similar sizes may be assigned to different levels

使用 ASF 对多层特征进行加权融合，作为RoI的特征

融合为了使anchor-feature的匹配不只是一对一，临近尺度的特征图也**参与预测**，一个样本**学习信号**也传到多个尺度特征图

Learning to Separate: Detecting Heavily-Occluded Objects in Urban Scenes (SGE/Serial R-FCN)

> 密集检测，embeding+NMS (类似feature NMS)，cascade

密集检测中不同类物体的区分和同一类不同物体的区分

Semantics-Geometry Embedding & SG-NMS

增加将检测框映射到隐空间中 $e = \mathbf{s}^T \cdot \mathbf{g}$，其中g即为检测框$(x, y, w, h)$，s为语义嵌入向量。可以看作**将位置信息以语义信息作为权重进行线性变换得到embed**

box和GT匹配时，对每个proposal b_i，选择最大IoU的物体b_j^*，如果i和j的IoU大于阈值，则认为i proposal匹配到j物体。**Select max then thresholding**

损失函数增加 1. Group：proposal的embed和匹配的物体的embed距离尽可能小 2. Sep：proposal的embed和与其第二大IoU的物体的embed距离增大（**第一大的obj: 距离减小，第二大的obj: 距离增大**）

NMS时，IoU小于阈值N_T的保留(Greedy)，大于阈值的计算**embed的距离**(SG)，embed距离大于Φ的保留，$\Phi \propto \mathrm{IoU}$，**IoU大的两个物体需要有更大的embed距离**

for b_i IN B **do**

$\tau \leftarrow \text{IoU}(b_m, b_i)$

if $\tau \geq N_t$ **then**

$\text{B} \leftarrow \text{B} \setminus \{b_i\}; \text{S} \leftarrow \text{S} \setminus \{s_i\}$

end

Greedy-NMS

if $\tau \geq N_t$ AND $d(e_m, e_i) \leq \Phi(\tau)$ **then**

$\text{B} \leftarrow \text{B} \setminus \{b_i\}; \text{S} \leftarrow \text{S} \setminus \{s_i\}$

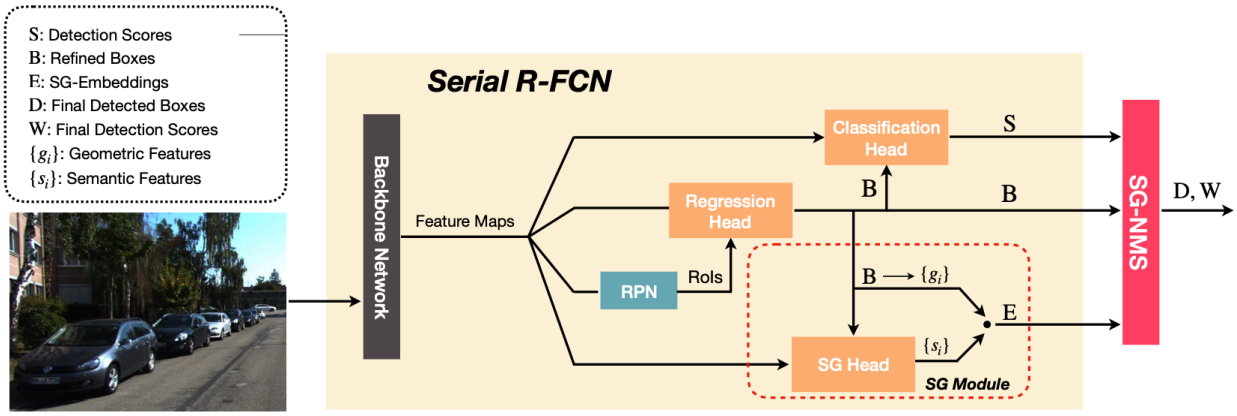
end

SG-NMS

end

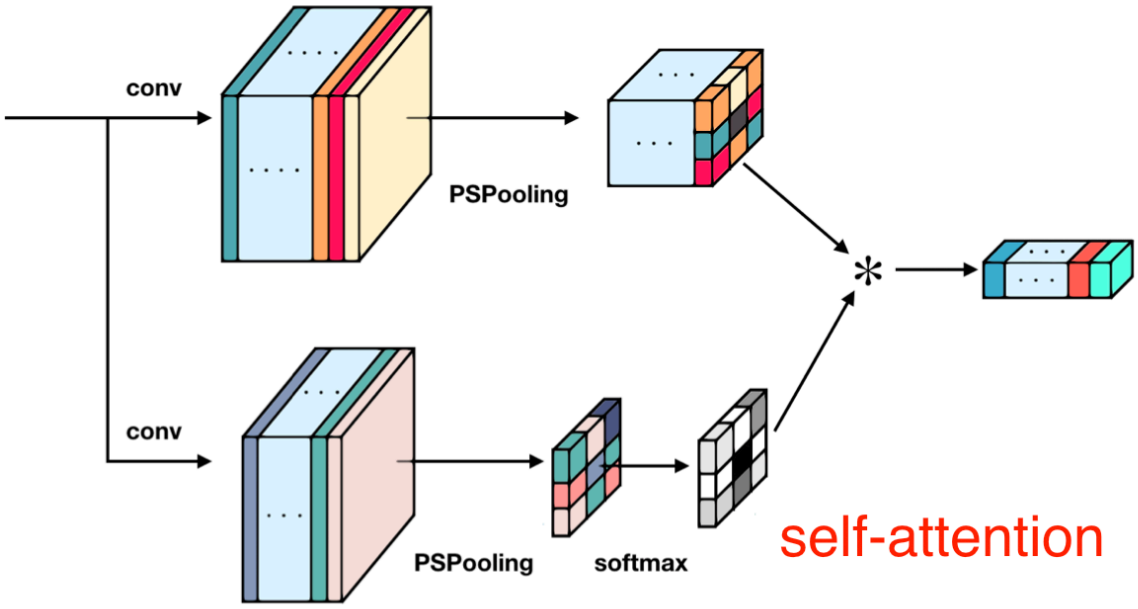
由于FPN作为backbone有多层，在FPN的每一层进行Greedy+SG NMS。在不同层之间只进行Greedy-NMS，且一个box只会被FPN其他层的box抑制

Serial R-FCN

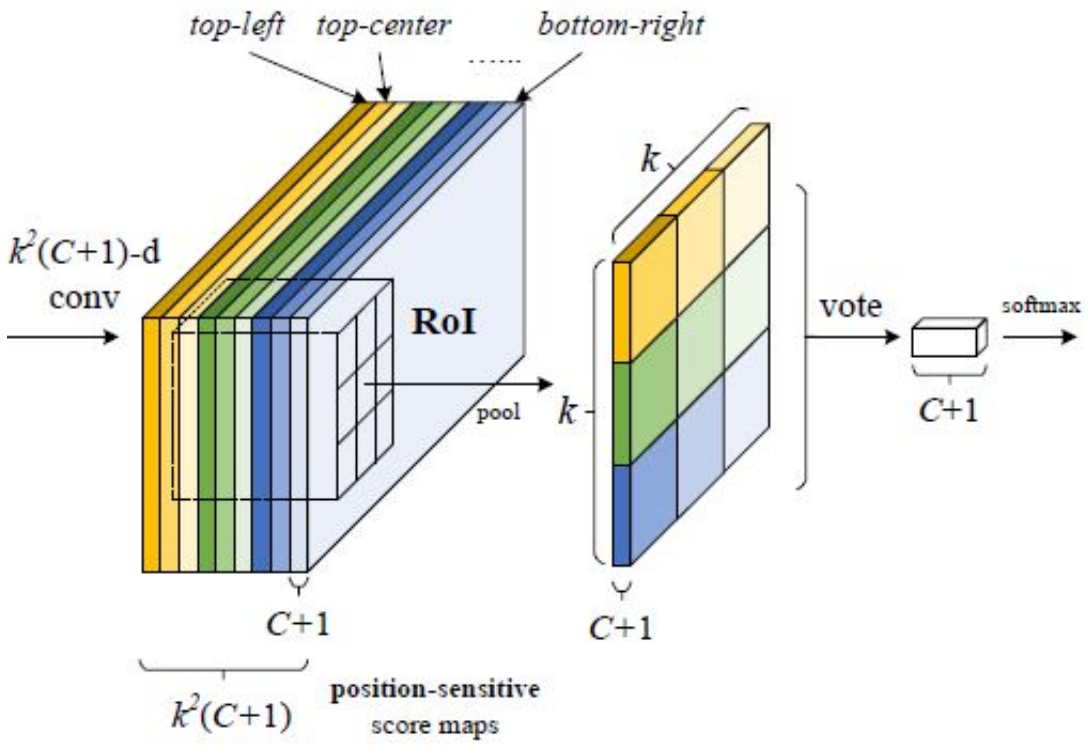


分类分支和SG计算分支在回归分支之后cascade进行。**直接使用refined-box而不是RoI/proposal进行特征提取**，可以使用更高的IoU阈值来训练分类分支

分类分支辨别回归分支回归的refined-box属于类别/BG。随着回归分支能力增强，BG类别样本数量减少，需要**hard negative mining**。在refined-box上增加随机噪声输入分类分支，作为接近且低于IoU阈值的难负样本



所有分支都采用Position Sensitive RoI-Pooling 👇，并增加self-attention👆

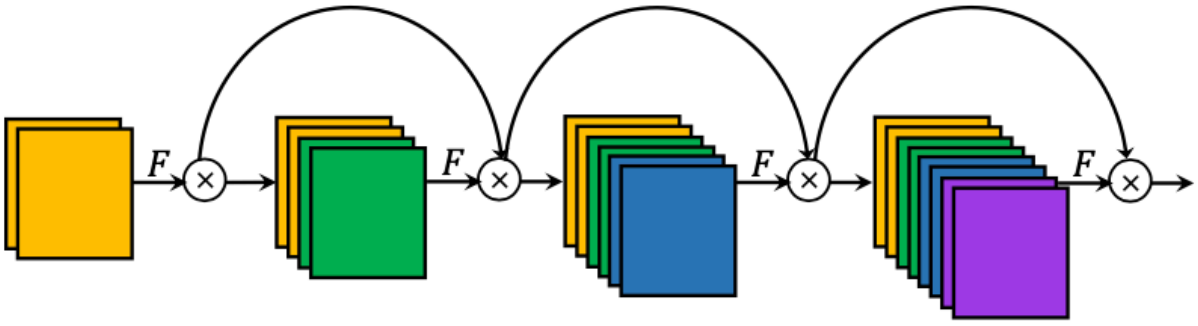


proposal/RoI区域内的每个位置有一个特征图，在对应特征图上RoI区域内pooling得到结果对应位置的响应值

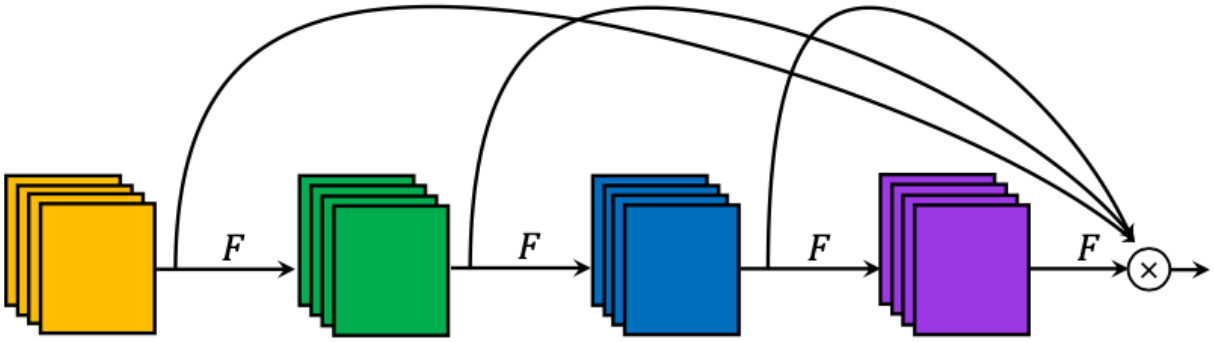
Ref: https://zhuanlan.zhihu.com/p/30867916

VoVNet

相比densenet，只进行一次特征融合操作



(a) Dense Aggregation (DenseNet)



(b) One-Shot Aggregation (VoVNet)

轻量级网络平DenseNet性能（不明显 $\approx 33@all$）

VoVNetV2 增加残差连接和SE-block

DRConv: Dynamic Region-Aware Convolution

> 动态选择卷积核，不是receptive field。类似空域和通道上的attention

空间的动态卷积核，卷积核区域间不同，区域内共享

传统卷积核：通道间不同，区域间完全相同（共享卷积核）W_c

局部卷积：不同位置**pixel**卷积核不同$W_{u,v,c}$

$Y_{u,v,o} = \sum_{c=1}^{C} X_{u,v,c} * W_{u,v,c}^{(o)} \quad (u,v) \in S$

DRConv：不同区域卷积核不同，同一个区域内共享$W_{t,c}$

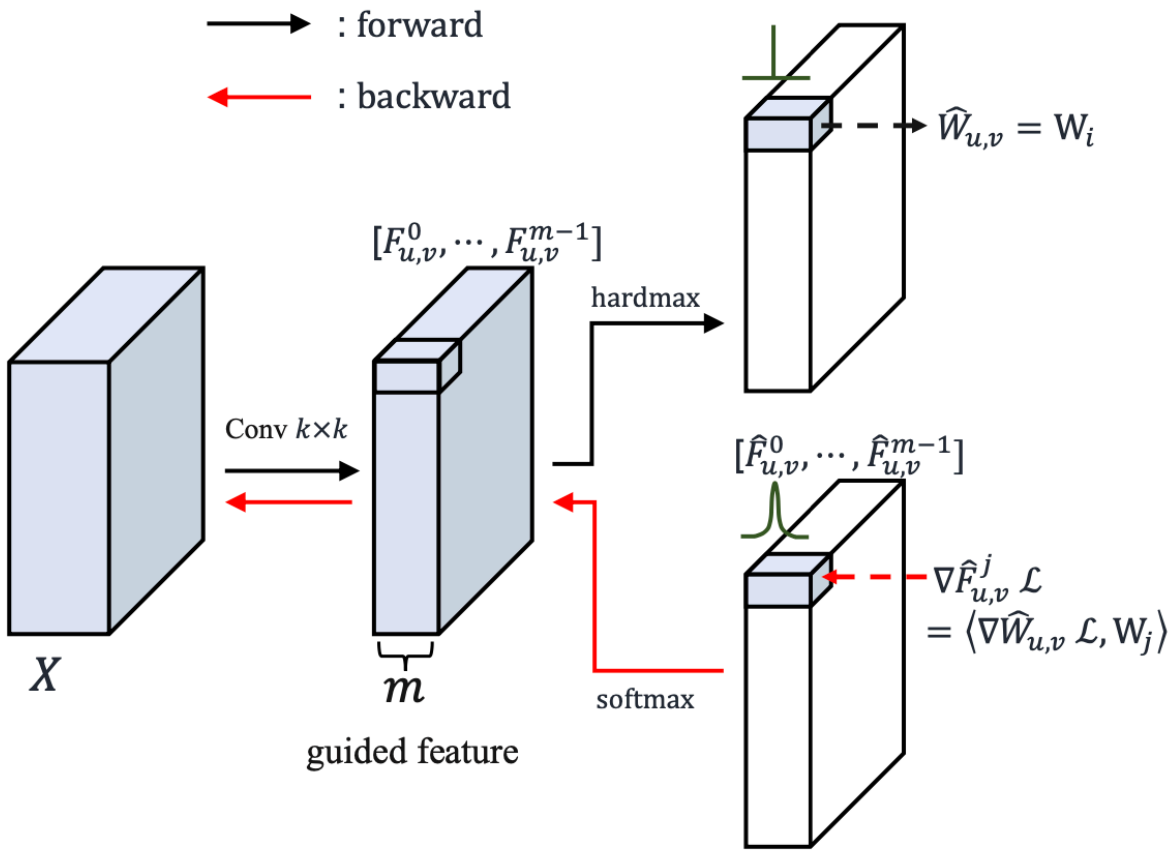
$Y_{u,v,g} = \sum_{c=1}^{C} X_{u,v,c} * W_{t,c}^{(o)} \quad (u,v) \in S_t$

首先学习划分区域mask，之后在每个区域内进行动态卷积

Learnable guided mask

学习分区，学习卷积核在特征图上的分布

使用普通卷积计算特征图（kernel空域上相同，通道维不同），**在特征图通道维选择最大的对应的卷积核**作为该位置上使用的卷积核



(a) Optimization of learnable guided mask

$M_{u,v} = \mathrm{argmax}\left(F^0_{u,v}, F^1_{u,v}, \cdots, F^{m-1}_{u,v}\right)$，m个channel

选择**通道维**最大的kernel作为区域的kernel (同样大小不同参数)

由于argmax没有梯度(mask $M_{u,v}$是one-hot向量)，所以反向传播时使用softmax取代$M_{u,v}$

$\hat{F}^j_{u,v} = \frac{e^{F^j_{u,v}}}{\sum_{n=0}^{m-1} e^{F^n_{u,v}}} \quad j \in [0, m-1]$

Dynamic Filter

根据输入特征动态产生每个区域的卷积核

类似通道+空域的attention机制（每个区域选择最大通道对应的卷积核）

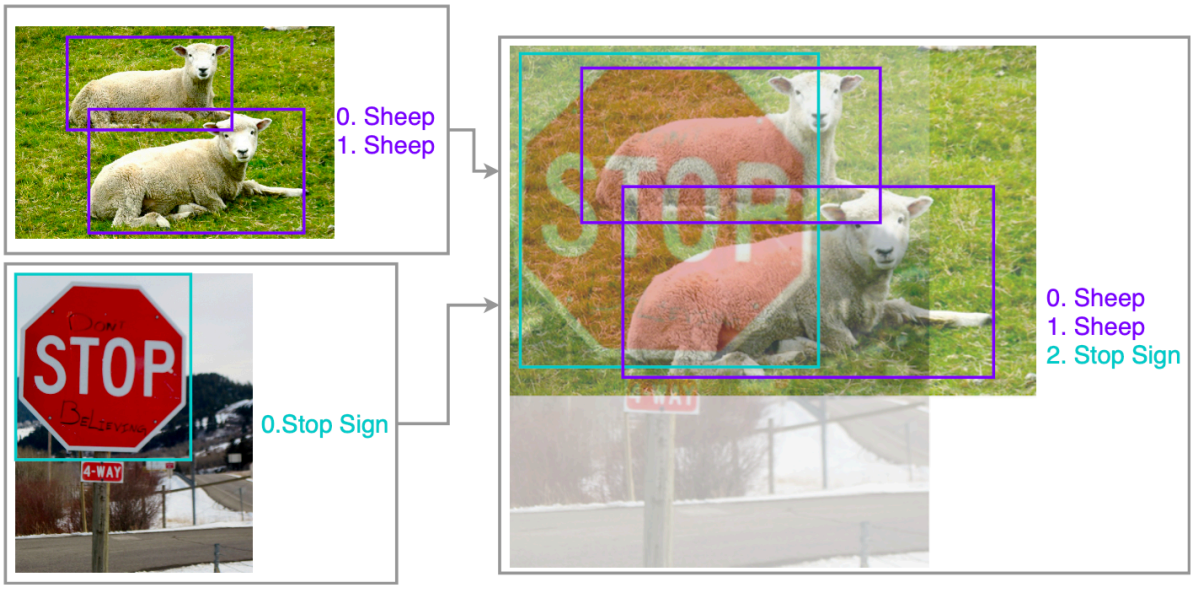
性能提升1-2点 Mask-RCNN

Object Detection Tricks

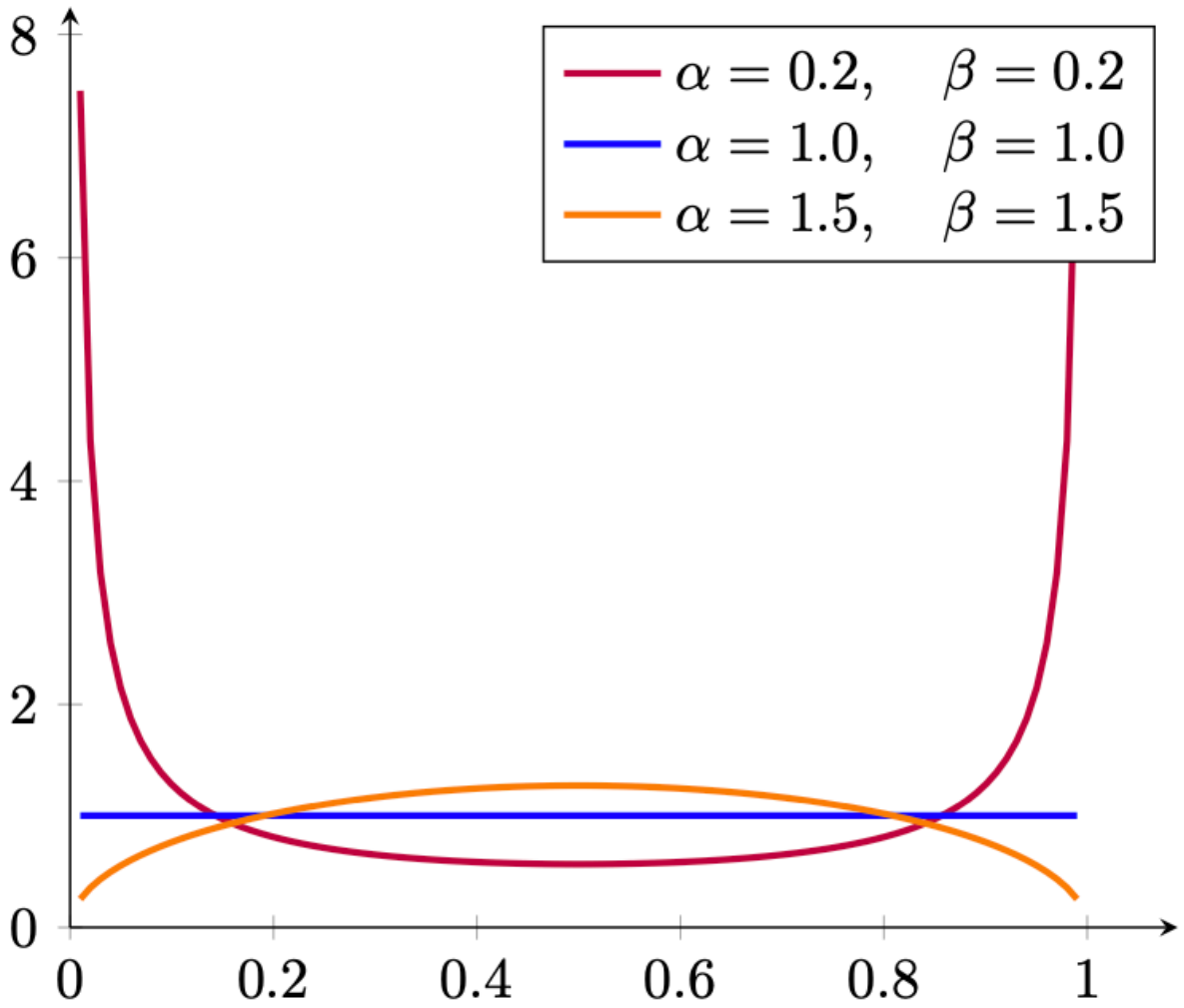
Bag of Freebies for Training Object Detection Neural Networks

Visually Coherent Image Mixup Training

按照beta分布融合两张图片训练，位置信息不变（geometry preserved）求loss时按照融合占比加权



Beta distribution



效果：解决 **unprecedented** scenes (如屋中大象) 和 very **crowded** object group，但可能会使置信度降低

Label Smoothing

分类头上使用，增加CE-Loss中错误label的梯度，防止模型too-confident & over-fitting

$$q_i = \begin{cases} 1-\varepsilon & \text{if } i = y \\ \varepsilon/(K-1) & \text{otherwise} \end{cases}$$

Data Augmentation

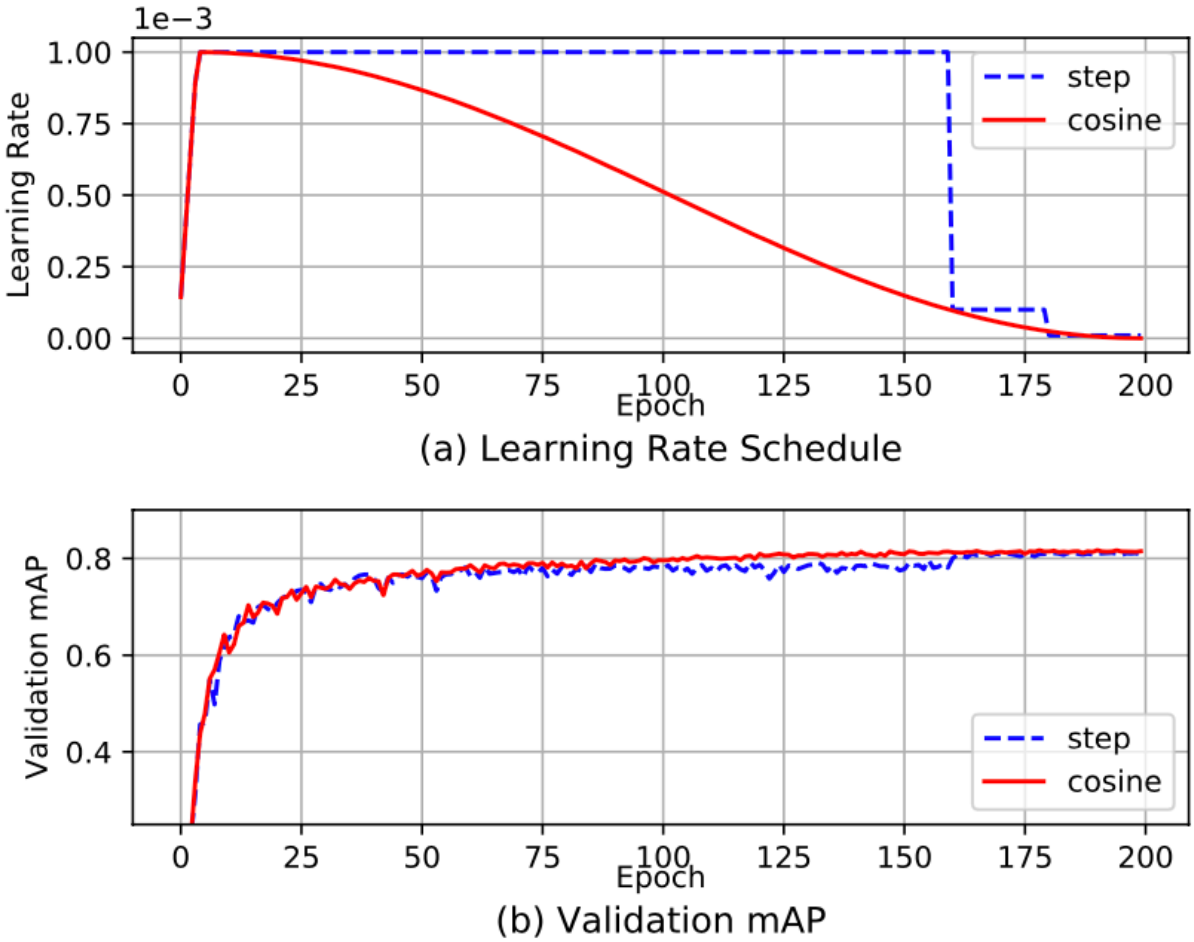
- Random geometry transformation: crop, expansion, flip, resize
- Random color jittering: brightness, hue, saturation, contrast

二阶段有proposal的剪裁不需要geometry transformation

Training Schedule

采用cosine学习率，防止step scheduler剧烈变化不稳定

warm-up防止训练初期梯度爆炸



Sync BN

Batch-size 对性能影响大

```
model = apex.parallel.convert_syncbn_model(model)
```

Multi-scale

Multi-scale training → Image level pyramid

Multi-level/stage feature → Feature pyramid

相比image pyramid，特征金字塔只提取一次图像特征，不同stage输出（多尺度），速度更快

Multi-scale training + Feature pyramid → all in