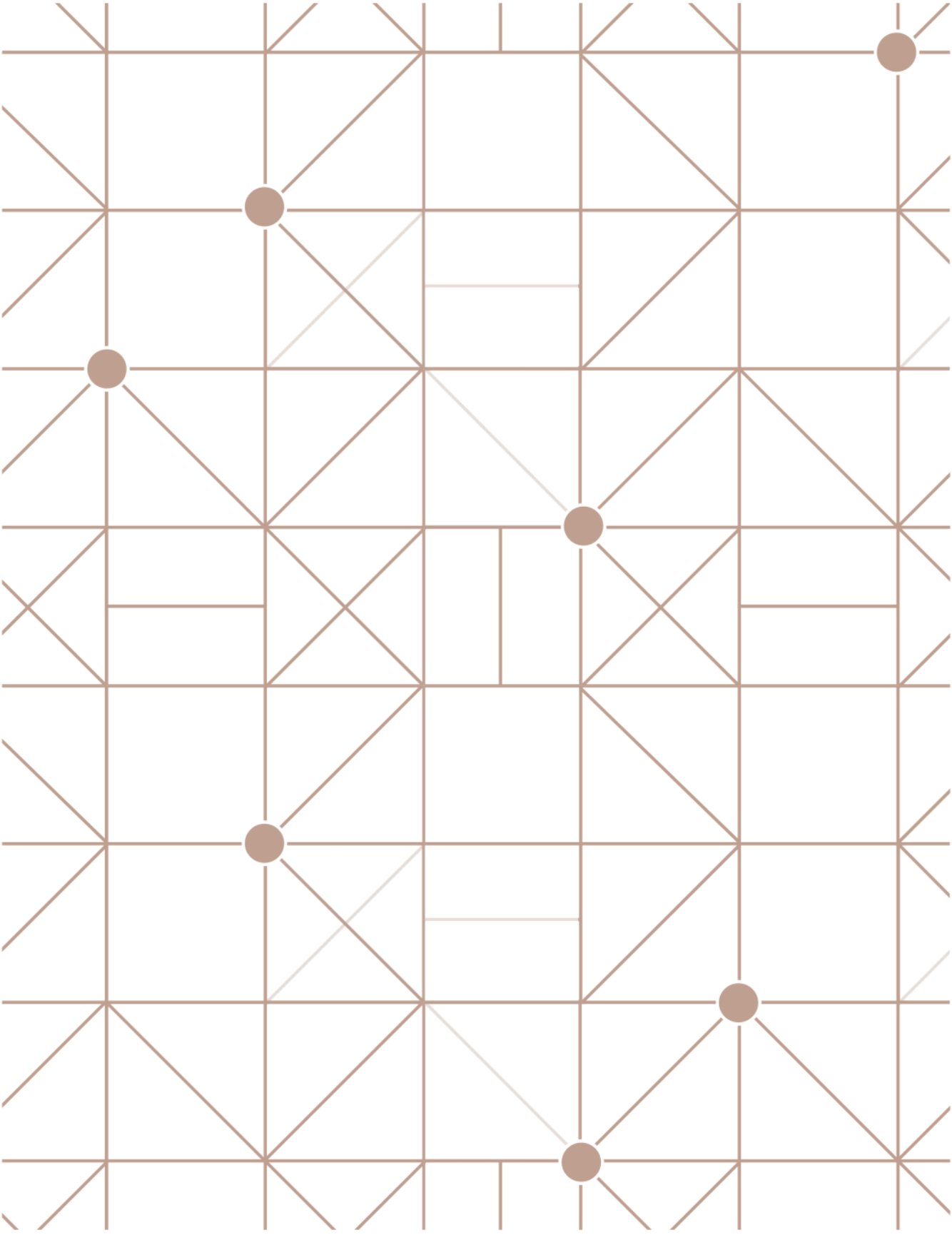
目标检测论文笔记

2022年5月17日 10:37

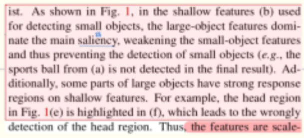


NETNet

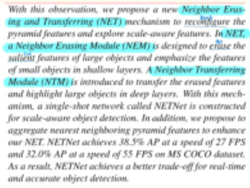
解决问题
- ①单级检测器中小目标容易丢失
- ②大目标局部的显著区域会被误检为一个目标.

原因
- 浅层特征图主要负责检测小目标，但浅层中仍旧存在大目标，对小目标造成干扰 (FN)
- 浅层特征图中，大目标的显著区域产生响应，导致大目标的重复检测 (FP)

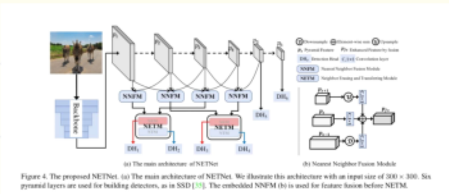
} Scale_confused


ist. As shown in Fig. 1, in the shallow features (b) used for detecting small objects, the large-object features dominate the main saliency, weakening the small-object features and thus preventing the detection of small objects (e.g., the sports ball from (a) is not detected in the final result). Additionally, some parts of large objects have strong response regions on shallow features. For example, the head region in Fig. 1(e) is highlighted in (f), which leads to the wrongly detection of the head region. Thus, the features are scale-

解决方法:

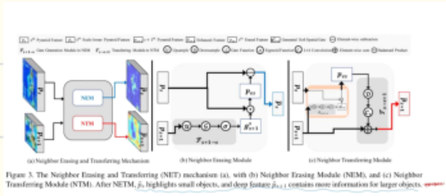

With this observation, we propose a new Neighbor Erasing and Transferring (NET) mechanism to reconfigure the pyramid features and explore scale-aware features. In NET, a Neighbor Erasing Module (NEM) is designed to erase the salient features of large objects and emphasize the features of small objects in shallow layers. A Neighbor Transferring Module (NTM) is introduced to transfer the erased features and highlight large objects in deep layers. With this mechanism, a single-shot network called NETNet is constructed for scale-aware object detection. In addition, we propose to aggregate nearest neighboring pyramid features to enhance our NET. NETNet achieves 38.5% AP at a speed of 27 FPS and 32.0% AP at a speed of 55 FPS on MS COCO dataset. As a result, NETNet achieves a better trade-off for real-time and accurate object detection.

- ①提出NET机制，构建 scale-aware features
- ②NEM: extract and erase 大目标 (从浅层fm上)
- ③NTM: 将浅层中 extract and erase 的大目标特征 transfer 到深层特征图中.
- ④NNFM: 实现 feature fusion.



(a) The main architecture of NETNet (b) Nearest Neighbor Fusion Module

Figure 4. The proposed NETNet. (a) The main architecture of NETNet. We illustrate this architecture with an input size of 300 × 300. Six pyramid layers are used for building detectors, as in SSD [35]. The embedded NNFM (b) is used for feature fusion before NETM.



(a) Neighbor Erasing and Transferring Mechanism (b) Neighbor Erasing Module (c) Neighbor Transferring Module

Figure 3. The Neighbor Erasing and Transferring (NET) mechanism (a), with (b) Neighbor Erasing Module (NEM), and (c) Neighbor Transferring Module (NTM). After NET, f_s highlights small objects, and deep feature $\hat{p}_{s+1}$ contains more information for larger objects.

存在问题
- NEM和NTM: 原理? 为什么简单的乘积操作就可以区分……?
- 与anchor-free detector 相比，精度相对较低
- NNFM的模块设计?

DSSD

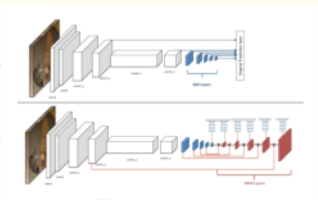
解决问题：
- 小目标 (SSD)
- 缺少上下文信息的融合。

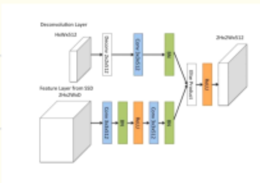
方法：
- 相对于SSD，使用ResNet101代替VGG16
- 使用反卷积 + skip connection 实现特征融合。
- 在预测框回归前，使用了Residual模块对特征图进行处理。

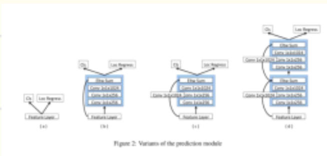
训练：
- 反卷积与主干融合时，使用的是 element_wise_product
- train_policy the same as SSD
- 通过使用k_means算法在pascal_voc 2007上实验，最终anchor比例设置为(1.6, 2.0, 3.0, 1)

结论：
- 精度相比于SSD更高
- 耗时较长：
  - ResNet101速度较慢，相对较大
  - 增加了额外的预测层和反卷积层
  - 使用了更多的先验框。

网络结构：







问题：
- 为啥不行？(只说是一种上下文融合方法，缺乏比较其他方法的说明)
- 当个模块都是通过对比实验得到的，缺乏可解释性。

DSOD

limited structure design space

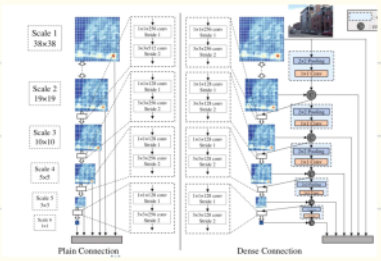
pre_trained model

Learning bias

Domain discrepancy

Layers		Output Size (Input 3 × 300 × 300)	DSOD
Stem	Convolution	64 × 150 × 150	3 × 3 conv, stride 2
	Convolution	64 × 150 × 150	3 × 3 conv, stride 1
	Convolution	128 × 150 × 150	3 × 3 conv, stride 1
	Pooling	128 × 75 × 75	2 × 2 max pool, stride 2
Dense Block (1)		416 × 75 × 75	[1 × 1 conv, 3 × 3 conv] × 6
Transition Layer (1)		416 × 75 × 75	1 × 1 conv
		416 × 38 × 38	2 × 2 max pool, stride 2
Dense Block (2)		800 × 38 × 38	[1 × 1 conv, 3 × 3 conv] × 8
Transition Layer (2)		800 × 38 × 38	1 × 1 conv
		800 × 19 × 19	2 × 2 max pool, stride 2
Dense Block (3)		1184 × 19 × 19	[1 × 1 conv, 3 × 3 conv] × 8
Transition w/o Pooling Layer (1)		1184 × 19 × 19	1 × 1 conv
Dense Block (4)		1568 × 19 × 19	[1 × 1 conv, 3 × 3 conv] × 8
Transition w/o Pooling Layer (2)		1568 × 19 × 19	1 × 1 conv
DSOD Prediction Layers		–	PlainDense

Table 1: DSOD architecture (growth rate $k = 48$ in each dense block).



FSSD

问题

- FPN的特征融合方式效率不高，需要多次进行fm融合操作，只融合相邻的层
- SSD在浅层检测小目标，缺少语义信息，效果差

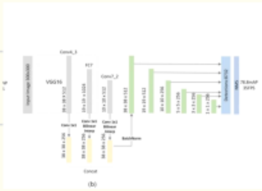
解决方法

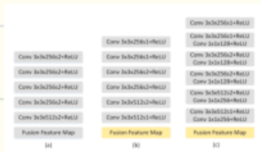
- 首先融合不同层特征（前三层特征融合）（conv7_3融合无效果，size<10的fm没有价值信息去融合）
- 再产生特征金字塔

结论

- FSSD 参数少，轻量级
- 小目标检测效果提升

网络结构：





RSSD

问题：SSD各层特征图独立检测，缺少不同特征图间关系的建立

小目标、重复检测

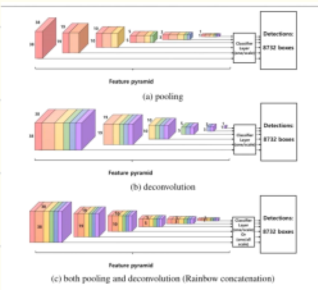
解决方法：通过 pooling 和 deconvolution 方法 增加每一层特征图的个数（channels）（增加各层关系建立）

由于用于检测的fm个数翻倍，因此可以共享检测conv（可以提高模型的泛化能力，避免因某尺寸样本过少，而使检测box训练不理想的情况发生）

结论：通过建立不同层不同尺度fm之间的关系，避免了同一物体重复检测的问题，小目标检测精度提高

使用单个分类检测器，精度稍有下降，但增加了模型的泛化能力

网络结构：



问题：channel↑，性能↑，why?

CenterNet (Object as point)

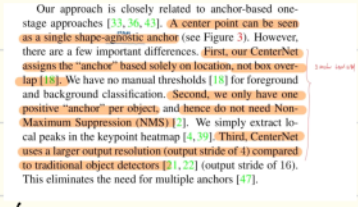
问题

- 往往的目标检测器大多都 枚举目标(锚框(先验框)), 费时, 效率不高, 还需要额外的后处理 (NMS)
- 往往检测器下采样因子比较大, 目标细节信息丢失多重

思想

- 将目标 => point (center point of bounding box)
- 利用 keypoint 预测 center point, 然后再回归目标其他属性 (size, orientation...)

CenterNet 与 anchor_based 区别:

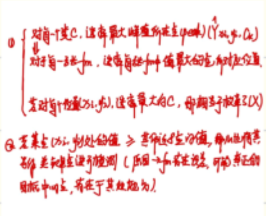


Our approach is closely related to anchor-based one-stage approaches [33, 36, 43]. A center point can be seen as a single shape-agnostic anchor (see Figure 3). However, there are a few important differences. First, our CenterNet assigns the "anchor" based solely on location, not box overlap [18]. We have no manual thresholds [18] for foreground and background classification. Second, we only have one positive "anchor" per object, and hence do not need Non-Maximum Suppression (NMS) [2]. We simply extract local peaks in the keypoint heatmap [4, 39]. Third, CenterNet uses a larger output resolution (output stride of 4) compared to traditional object detectors [21, 22] (output stride of 16). This eliminates the need for multiple anchors [47].

方法

- 单输入: $I \in R^{W \times H \times 3}$
- keypoint_heatmap
  - $\hat{Y} \in R^{\frac{W}{R} \times \frac{H}{R} \times C}$, $\hat{Y} \in [0, 1]$
  - $\hat{Y}_{x,y,c}=1$ (keypoint) $\hat{Y}_{x,y,c}=0$ (background)
- ground_truth
  - $p=(x,y)$, 类别为 C $\Rightarrow$ $\tilde{p}=\lfloor \frac{p}{R} \rfloor$
  - ground_truth heatmap: $Y^{\frac{H}{R} \times \frac{W}{R} \times C} \in [0,1]$ 通过高斯核 $Y_{xyc}=\exp\left(\frac{(x-\tilde{p}_x)^2+(y-\tilde{p}_y)^2}{2\sigma_p^2}\right)$
- 预测偏移: $\hat{O} \in R^{\frac{W}{R} \times \frac{H}{R} \times 2}$
- 预测 size: $\hat{S} \in R^{\frac{W}{R} \times \frac{H}{R} \times 2}$
- 损失函数: keypoint: Focal Loss offset: L_1 loss size: L_1 Loss
- 对于每一个 (x,y) 预测 $|C|+4$ 个值

检测：



问题：如果两个目标的关键点重合，那么只能检测其中一个。

基础网络：Hourglass，ResNet，DLA.

网络结构：

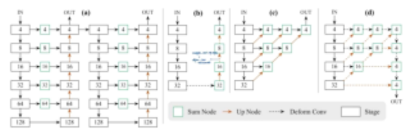


Figure 6: Model diagrams. The numbers in the boxes represent the stride to the image. (a): Hourglass Network [30]. We use it as is in CornerNet [30]. (b): ResNet with transpose convolutions [55]. We add one 3 × 3 deformable convolutional layer [63] before each up-sampling layer. Specifically, we first 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 → 16. We show these two steps together as a dashed arrow for 16 → 8 and 8 → 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.

SCRDet（一种适用于小、杂、旋转物体的多类别旋转检测器）

针对小目标：通过特征融合和锚点采样角度，设计了一种特征融合结构.

针对密集分布问题：设计了一种结合有监督像素注意力网络和通道注意力网络的结构，抑制噪声，突出目标特征.

针对任意方向问题：通过添加IoU常数因子设计了一种改进的Smooth L1损失，专门用于解决旋转边界框回归中的边界问题.

网络框架：

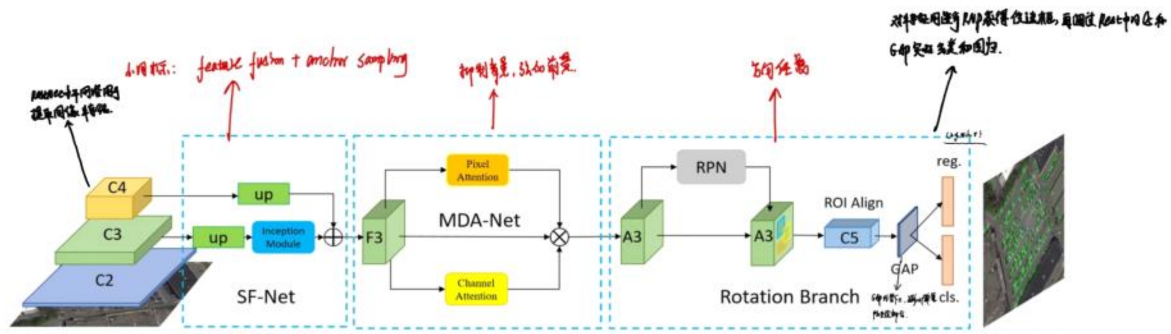


Figure 1: SCRDet includes SF-Net, MDA-Net against small and cluttered objects and rotation branch for rotated objects.

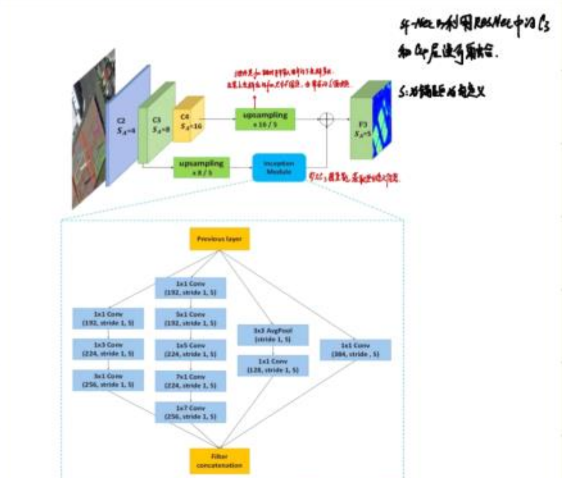


Figure 3: SF-Net. F3 has a small S_A, while fully considering the feature fusion and adaptability to different scales.

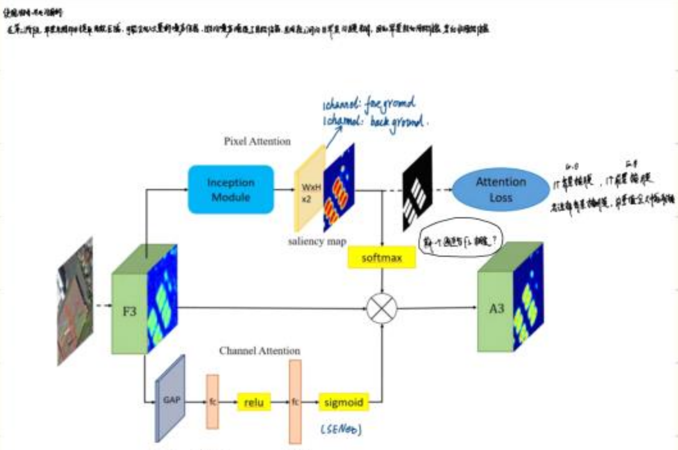
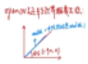


Figure 5: The devised MDA-Net consisting of channel attention network and pixel attention network.

SCRDet是一个two-stage目标检测器.

SCRDet由SF-Net, MDA-Net以及Rotation Branch三部分组成，SF-Net以及MDA-Net通过不断的强化目标提取网络特征，最后利用Rotation Branch进行角度回归.



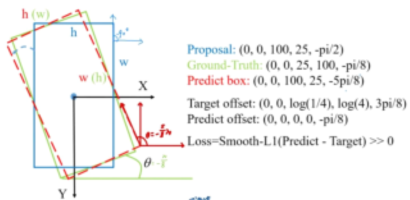


Figure 6: Boundary discontinuity of the rotation angle.

respectively (likewise for y, w, h, θ).

3.4. Loss Function

The multi-task loss is used which is defined as follows:

$$L = \frac{\lambda_1}{N}\sum_{n=1}^{N} t'_n \sum_{j\in\{x,y,w,h,\theta\}} \frac{L_{reg}(v'_{nj}, v_{nj})}{|L_{reg}(v'_{nj}, v_{nj})|}|-\log(IoU)| + \frac{\lambda_2}{h\times w}\sum_i^h\sum_j^w L_{att}(u'_{ij}, u_{ij}) + \frac{\lambda_3}{N}\sum_{n=1}^{N} L_{cls}(p_n, t_n) \quad (3)$$

where N indicates the number of proposals, t_n represents

AugFPN

问题：FPN存在的问题：

① before fusion：在fusion之前，需对用1x1 conv对图片进行降维，使channel相同时，才能相加。然而不同stage的感受野不同，包含的语义信息不同，把两个语义信息相差较大的fm直接相加，将削弱多尺度特征的表达能力。

② top-down feature fusion：高层的fm经过1x1 conv后才与底层fm相加，导致信息丢失

③ after feature fusion：特征融合后选择特定层进行检测，没有选择的层也包含丰富的信息，直接忽略会对检测结果有影响。

思想：Consistent supervision（一致性监督）：为缩小特征融合前，不同尺度特征之间的语义差距。

Residual Feature Augmentation（残差特征增强）：在特征融合中，通过残差特征增强提取比率不变的上下文信息，以减少特征映射在最高层特征金字塔中的信息丢失

Soft RoI selection：采用软RoI选择，在特征融合后自适应的选更好的RoI特征。

结构：

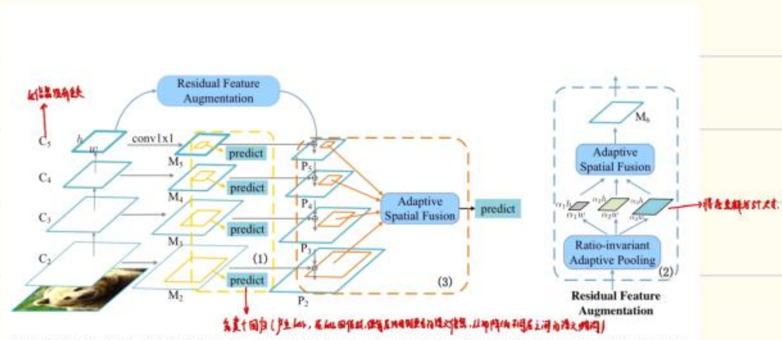


Figure 2. Overall pipeline of AugFPN based detector. (1)-(3) are three main components of AugFPN: Consistent Supervision, Residual Feature Augmentation, and Soft RoI Selection. For simplicity, the 3 × 3 convolution layers after feature summation are not shown.

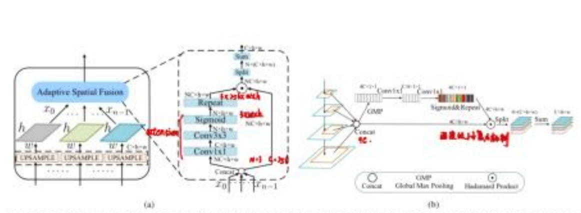


Figure 3. (a) is the process of fusing different context features and structure of Adaptive Spatial Fusion. (b) is the details of Adaptive Channel Fusion. In (a), it depends on the sizes of inputs to decide whether to use upsample blocks in the dashed box.

$$L_{rcnn} = \lambda \sum_{M=2}^{5} (L_{cls,M}(p_M, t^*) + \beta[t^* > 0] L_{loc,M}(d_M, b^*)) + \sum_{P=2}^{5} (L_{cls,P}(p, t^*) + \beta[t^* > 0] L_{loc,P}(d, b^*)). \quad (1)$$