工程技术与管理·第 04 卷·第 03 期·2020 年 03 月
DOI: https://doi.org/10.26549/gcjsygl.v4i3.3662

Improved Pedestrian Detection Based on YOLOv3

Xiaolu Ma　Shilin Hong

Anhui University of Technology, Maanshan, Anhui, 243000, China

Abstract

Aiming at the current problems of pedestrian detection anchor size mismatch and feature scale discontinuity that would affect detection accuracy, an improved pedestrian detection algorithm based on YOLOv3 was proposed. This paper uses Guided anchoring to generate anchors, incorporates adaptive spatial feature fusion. The test results on the intersection pedestrian dataset show that the mAP of the improved YOLOv3 pedestrian detection algorithm reaches 95.56%, which improves the detection accuracy.

Keywords

YOLOv3; guidance anchor; adaptive spatial feature fusion; pedestrian detection

基于 YOLOv3 改进的行人检测

马小陆　洪石林

安徽工业大学，中国·安徽 马鞍山 243000

摘　要

针对目前行人检测 anchor 尺寸不匹配和特征尺度不连续的会影响检测精度的问题，提出基于 YOLOv3 改进的行人检测算法。论文使用 Guided anchoring 指导生成 anchors，融入自适应空间特征融合。在路口行人数据集上的测试结果表明，改进后的 YOLOv3 行人检测算法 mAP 达到了 95.56%，提高了检测精度。

关键词

YOLOv3；指导锚；自适应空间特征融合；行人检测

1 引言

近些年行人检测技术被广泛地应用在各个领域中，如智能交通中的路口行人检测、车辆辅助驾驶、危险区域行人识别等。常用的基于深度学习方法的目标检测方法有 R-CNN 系列 [1-2]、SSD 系列 [3] 和 YOLO 系列 [4-6]。虽然将目标分类与定位一步回归的 YOLOv3 已经取得了较好的检测效果，但是在行人检测方面精度仍不够理想。主要体现在下面几个方面：(1)YOLOv3 训练、测试使用的数据集为 COCO 或者 VOC，此类数据集类别过多，在此数据集聚类得到的 anchor 对行人检测的适应性不高，而且使用聚类生成的 anchors 尺寸如果不合适将会阻碍行人检测的精度及速度。(2) 基于特征金字塔的一阶段行人检测模型，不同特征尺度之间的不连续性是影响行人检测准确率的另一个主要因素。因此，论文采用指导锚 (Guided Anchoring-GA)[7] 生成 anchors，并在模型中加入一种自动训练不同尺度特征融合权重的策略 (Adaptively Spatial Feature Fusion-ASFF)[8]。使得改进后的行人检测模型在依旧满足实时性的条件下，大大提高了行人检测的精度。

2 YOLOv3 原理

从 2016 年到 2018 年 YOLO 已经经历了 YOLOv1、YOLOv2、YOLOv3 三个版本的改进。应用最广泛的是 YOLOv3，首先采用类似 ResNet[9] 跳层连接的快捷连接，将原始数据跳过某些层而直接传到之后的层，在很大程度上地解决了因网络加深导致的模型过拟合的问题；其次采用多尺度预测，在三个不同的尺度上进行预测，使用类金字塔网络 [10] 从这些尺度中提取特征，然后与上采样后的特征进行合并，可以得到更有效的细粒度特征和更有意义的语义信息；并将 softmax 替换为二进制交叉熵损失函数 , 实现多种类别的预测。

一般 anchor 的生成方式会考虑对齐和一致性两个因素，即 anchor 中心和特征图像素的中心要对齐，一个特征图中不

同区域的感受野和语义范围要一致。YOLOv3 针对不同数据集和检测对象要制定不同尺寸及比例的 anchors, 而错误的制定会影响检测的精度与速度；而且目标检测在确保召回率的前提需要生成大量的 anchors，但其中大部分的 anchor 为假样本，会增加训练和推理的计算量。

YOLOv3 模型中除了 anchor 生成方式会影响检测精度之外，不同尺度特征之间的不一致性同样会影响。YOLOv3 模型中特征金字塔采用启发式引导的方式选择特征，即大尺寸的目标通常与上层特征图相关联，小尺寸目标通常与下层特征图相关联。但如果在某个级别特征图上一个目标被视为正例时，其他级别特征图中对应区域则会被视为背景。因此，若图像同时包含“小”和“大”两个尺寸的目标，则不同尺寸特征之间的冲突矛盾信息会占据“特征”金字塔的主要部分。这种特征尺度不一致会影响训练期间梯度的计算，并且会降低特征金字塔提取特征的有效性。

3 基于 YOLOv3 训练优化

基于上述 YOLOv3 存在的问题，论文从 anchor 的生成方式和特征尺度不一致融合策略两个方面对模型进行优化。

3.1 anchor 生成方式优化

目标检测算法的好坏很大程度上依赖 anchors 的生成机制，大部分的算法都是通过事先定义好的尺寸和大小的 anchor 在空间上均匀移动采样得到。YOLOv3 通过聚类得到 anchors，聚类得到尺寸不合适将会阻碍检测的精度及速度。论文提出一种利用语义特征对 anchor 进行引导生成的方法 (GA)。由两个步骤生成 anchors：首先预测目标可能存在的位置，然后在可能存在的位置处预测该位置对应的形状。因此生成一系列稀疏的 anchors, 紧接着使用可变形卷积模块调整特征的连续性。

行人的位置及形状可以由（x,y,w,h）表示。(x,y) 代表行人位置的坐标。(w,h) 表示行人的宽高。在一帧图片上检测行人可表示为公式 (1)：

$$p(x,y,w,h \mid I) = p(x,y \mid I) * p(w,h \mid x,y,I) \quad (1)$$

上式 (2) 表示行人中心位置在图片中的 (x,y) 处，并且该位置处的 anchor 宽高与所在位置相关联。对于一帧图片 I，首先提取特征 map 称为 Fi，在 Fi 的头部设置 anchor 初始位置，即 YOLOv3 三个不同尺度通道对应的网格中心。接着预测针对每个位置的 anchor 形状 (w,h)。由于图像的宽、高的取值范围太大，直接进行预测较困难。论文做了如下映射。

$$w = \delta * s * e^{dw} \qquad h = \delta * s * e^{dh} \quad (2)$$

通过公式 (2)，将 w,h 的预测转换为对 dw,dh 的预测，为一个经验系数，论文取 8；s 表示特定尺度上的步长。该分支将附近的每一个 ground truth box 与 anchor 进行匹配，选择 IOU 得分最高的 anchor 宽度（$\hat{w}$）高度（$\hat{h}$）作为最优值。经过上述两个步骤生成一系列不同位置和形状稀疏 anchors。由于 anchor 形状不同，对应的特征编码区域也不同。此时生成的 Fi 已不再适合标准卷积分类 [11]。论文采用针对不同位置、anchor 尺寸来调整特征形状的模块，如下式 (3):

$$f' = N_T(f_i, w_i, h_i) \quad (3)$$

其式 (3) 中 ,f_i 、w_i 、h_i 分别图片中位置 i 处的特征、anchor 的宽、高。先通过可变形卷积 [12] 获取预测 anchor 形状的偏移量 (offest), 然后将最初的与 offset 结合获得新的特征。

3.2 特征尺度不一致优化

3.2.1 ASFF 介绍

基于特征金字塔的一阶段行人检测模型，不同特征尺度之间的不连续性对行人检测影响不容小觑。针对这一问题，一些模型采用了一些尝试性策略。比如将相邻级别上对应的特征图区域设置为忽略区域 [14]，但这种忽略可能会增加相邻级别特征的误检。

论文提出一种自动训练不同尺度特征融合权重的特征金字塔融合策略，自动学习融合权重并过滤空间上的冲突信息来改善特征尺度一致性。对于某个级别的特征，首先将其他级别的特征调整为与该级别相同的分辨率后再融合，然后训练以找到最佳融合权重。网络训练过程中使每个空间位置自动学习不同尺寸特征之间融合权重，即那些携带冲突矛盾信息位置的权重在学习过程中越来越小，并且这一操作几乎不引入额外的推理开销。

3.2.2 YOLOv3− ASFF 模型结构

下图 1 为 YOLOv3 模型加入 ASFF 模块的网络结构图，与使用逐元素求和级联集成多层特征的方法不同，论文自动地学习各个尺度特征图的空间融合权重。首先把不同尺寸的特征图缩放到相同的比例，接着将缩放后的各个尺寸的特征进行融合。

工程技术与管理 · 第 04 卷 · 第 03 期 · 2020 年 03 月
DOI: https://doi.org/10.26549/gcjsygl.v4i3.3662

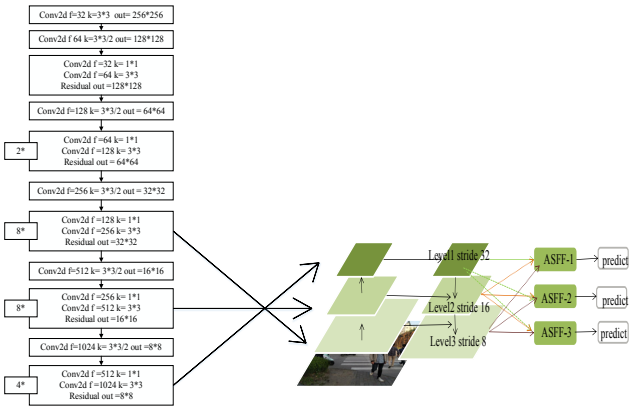


图 1 YOLOv3 加入 ASFF 模块网络结构图

图 1 中 conv2d 为标准卷积，f 为输入通道，k 为卷积核尺寸，out 为输出尺寸，s 为卷积步长，Residual 为残差结构，2* 为该模块重复次数。Level1 为第一个尺度特征，Level2、Level3 含义类同，stride 为特征卷积步长，ASFF1 为第一个尺度自动融合后的特征，ASFF2、ASFF3 含义类同，predict 为预测的目标框。将原 YOLOv3 的三尺度特征，经过相同的缩放和自适应融合后，网络自适应地学习各个比例下特征图的空间融合权重。其中 ASFF3 层特征融合的方式如下图 2。

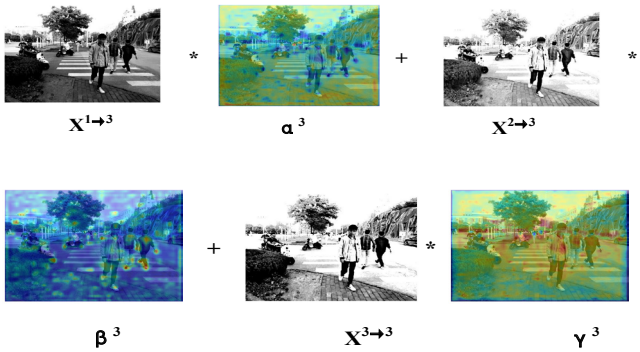


图 2 ASFF3 层特征融合

图 2 中 $x^{1\to3}$、$x^{2\to3}$、$x^{3\to3}$ 分别表示级别 1、2、3 调整到级别 3 尺寸后的特征，α、β、γ 分别为对应层的权重。首先调整特征尺寸，级别 l($l \in \{1,2,3\}$) 的特征表示为 x^l。对于级别 1，将非 l 级别 n（$n \neq l$）上的特征 x 调整为与 x^l 相同的尺寸。由于 YOLOv3 三个尺度特征图尺寸、通道数不同，特征尺寸调整的上下采样规则为：上采样时，使用 1*1 卷积层压缩通道，再通过插值来扩大分辨率。以 1/2 比率进行的下采样，使用步长为 2 的 3*3 卷积层同时修改通道数和分辨率。当下采样比率为 1/4 时，在步长为 2 的卷积之前再添加一个步长为 2 的最大池化层。

3.2.3 特征自适应融合

下面说明特征自适应融合过程。令 x_{ij}^l 表示从级别 n 到级别 l 调整尺寸后特征图中位于（i，j）处的特征向量。按公式 (6) 融合 l 层的特征：

$$y_{ij}^l = \alpha_{ij}^l * x_{ij}^{1\to l} + \beta_{ij}^l * x_{ij}^{2\to l} + \gamma_{ij}^l * x_{ij}^{3\to l} \qquad \alpha_{ij}^l + \beta_{ij}^l + \gamma_{ij}^l = 1, \quad \alpha_{ij}^l、\beta_{ij}^l、\gamma_{ij}^l \in [0,1] \tag{6}$$

公式 (6) 中 y_{ij}^l 表示输出特征 y^l 所有通道的第 (i,j) 个向量。α_{ij}^l、β_{ij}^l 和 γ_{ij}^l 为三个级别缩放到级别 l 上的空间重要性权重，由网络自适应学习得到并在所有通道之间共享。以 x^l 未缩放大小的特征图中（i，j）处的梯度来说明权重生成过程。根据链式求导规则，梯度的计算公式 (7)：

$$\frac{\partial L}{\partial x_{ij}^l} = \frac{\partial y_{ij}^1}{\partial x_{ij}^1} * \frac{\partial L}{\partial y_{ij}^1} + \frac{\partial x_{ij}^{1\to2}}{\partial x_{ij}^1} * \frac{\partial y_{ij}^2}{\partial x_{ij}^{1\to2}} * \frac{\partial L}{\partial y_{ij}^2} + \frac{\partial x_{ij}^{1\to3}}{\partial x_{ij}^1} * \frac{\partial y_{ij}^3}{\partial x_{ij}^{1\to3}} * \frac{\partial L}{\partial y_{ij}^3} \tag{7}$$

由于上、下采样的实现方式分别为插值和池化，可假设 $\frac{\partial x_{ij}^{1\to l}}{\partial x_{ij}^l} \approx 1$ 并使 $\frac{\partial y_{ij}^l}{\partial x_{ij}^l} = 1$ 和 $\frac{\partial y_{ij}^l}{\partial x_{ij}^{1\to l}} = 1$ 将 (7) 简化为：

$$\frac{\partial L}{\partial x_{ij}^l} = \frac{\partial L}{\partial y_{ij}^1} + \frac{\partial L}{\partial y_{ij}^2} + \frac{\partial L}{\partial y_{ij}^3} \tag{8}$$

假设第 1 级特征中（i，j）处正好为目标的中心位置，则 $\frac{\partial L}{\partial y_{ij}^1}$ 为正样本梯度。之前的方法将 $\frac{\partial L}{\partial y_{ij}^2}$ 和 $\frac{\partial L}{\partial y_{ij}^3}$ 视为负样本梯度。这种不一致会扰乱 $\frac{\partial L}{\partial x_{ij}^l}$ 梯度计算并降低原始 1 层特征的训练效率。大部分网络将其他级别相应位置的梯度设置为忽略区域来解决该问题。比如令 $\frac{\partial L}{\partial y_{ij}^2}$ 和 $\frac{\partial L}{\partial y_{ij}^3}$ 为零。尽管消除了 x_{ij}^l 的冲突，但 y_{ij}^2、y_{ij}^3 中的松弛会引起更多次等水平的检测。对于 ASFF，梯度计算很简单，直接从式（6）和式（8）得到公式 (9)：

$$\frac{\partial L}{\partial x_{ij}^l} = \alpha_{ij}^1 * \frac{\partial L}{\partial y_{ij}^1} + \alpha_{ij}^2 * \frac{\partial L}{\partial y_{ij}^2} + \alpha_{ij}^3 * \frac{\partial L}{\partial y_{ij}^3} \qquad \alpha_{ij}^1、\alpha_{ij}^2、\alpha_{ij}^3 \in [0,1] \tag{9}$$

利用 $\alpha_{ij}^2 \to 0$、$\alpha_{ij}^3 \to 0$ 来解决上述问题，通过 α_{ij}^1、α_{ij}^2、α_{ij}^3 这几个参数来协调特征的不一致性。其次论文采用标准反向传播算法学习融合权重，能保留背景信息的监督信息。

为了进一步说明 ASFF 如何对特征进行自适应融合，论文在图 3 中对行人检测定性结果进行了可视化。

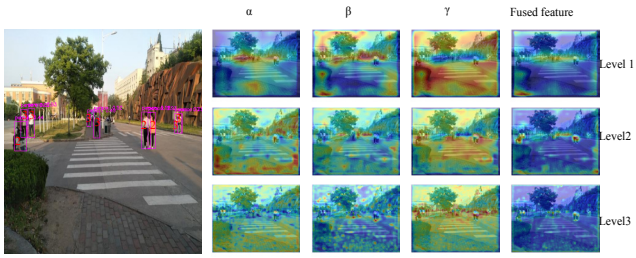


图 3 行人检测定性分析图

图 3 左侧为检测结果。右侧 α、β、γ 所对应的列为融合

权重标量的热图，Fused feature 对应的列为相应级别通道之间的值求和得到可视化激活图。检测结果中红色数字代表检行人融合特征所在的级别。为了说明问题，选取含有不同尺寸行人的图片作为自适应空间特征融合的对象。论文仅将目标在相应特征图中心处视为正例。由图 3 可知，部分行人分别由级别 2、3 特征图预测得到，还有一部分由级别 2 和级别 3 特征融合预测得到。由于待检测图中不包含大尺寸行人，该图的检测过程中级别 1 的特征被过滤掉，即将 1 级特征视为背景，并在训练过程中该级别的不会出现正例梯度。同时在级别 1 融合过程中，级别 2 和级别 3 缩放大小后的特征也被滤除，不会出现负例梯度。由图可知小尺寸行人更大程度由第 3 级别预测得到。因此借助 ASFF 模块，可以得到不同尺度的最佳融合权重。

4 数据准备与实验验证

4.1 试验数据

论文使用的数据集为：采集的学校和街道路口行人，一共 5000 张，其中 3000 张作为训练集，1500 作为验证集，500 作为测试集。为了提高模型的检测精度，论文通过对原始图片和注释文件的相应矩形框进行缩放、水平翻转、随机裁剪和随机改变对比度等来增强图像数据；还采用一种视觉连贯的目标检测图像混合增强方法，称为 mixup，该方法本质是在高维空间对数据进行插值，从而增加更多的伪数据集，类似于正则化，一定程度上减少了模型的过拟合程度 [16]。

4.2 试验平台、训练方法、评估指标

本试验中使用的实验平台为：英特尔酷睿 i7−7700 2.80GHz 处理器、NVIDIA GTX 1060 显卡、Ubuntu16.04LTS 系统。使用 CUDA 9.0 和 CUDNN v7.0 的 PyTorch v1.0.1 框架来实现 YOLOv3 和加入 GA、ASFF 模块后的模型。由于模型使用 GA 指导 anchors 生成后，论文加入 anchor 损失 $Loss_{anchor}$。又由于预测边界框时，宽、高两个变量互相关联，加入 IOU 损失 $Loss_{anchor}$ [13]。梯度优化的方式为随机梯度下降 (SGD)，batchsize 为 6。学习率为 0.0001，$beta_1$=0.9，$beta_2$=0.999；模型训练了 300 个 epoch，并在最后 30 个 epoch 关闭 mixup 增强功能。

平均精度 (Average Precision,AP) 用来衡量模型对于每个种类分类的好坏。mAP (mean average precision) 则是针对所有类别而言。论文只有一类目标，两者意义相同，衡量行人检测精度。Precision−recall(PR) 曲线横坐标和纵坐标分别是召回率 (recall) 和精确度 (precision)[15]。通过取不同的阈值得到多组精度和召回率数值绘制得到。检测速度也是一个重要性能指标，每秒帧数 (Frame Percent Secind , FPS) 常用来衡量检测算法的速度。

4.3 试验结果

论文将添加 anchor 损失、IoU 损失、mixup 数据增强等模块统称为 trick。

表 1 YOLOv3 和改进后 YOLOv3 模型性能对比

模型	mAP（%）	FPS
YOLOv3	87.68	25
YOLOv3+GA+trick	92.59	25
YOLOv3+ASFF+GA+trick	95.56	25

由表 1 可知，YOLOv3 模型中加入 GA 和 trick 后 mAP 提高了 4.91%，且 FPS 保持不变；在上述模型基础上进一步加入 ASFF 模块，mAP 又提高了 2.97%，FPS 依旧不变。最终可知在实时性相同的条件下 YOLOv3+ASFF+trick 比原始 YOLOv3 的行人检测精度提高了 7.88%，说明 YOLOv3 在加入 mixup 增强方式、anchor 损失、IoU 损失，并使用 GA 指导 anchor 生成、和加入 ASFF 模块后可以在计算量增加不大的条件下有效的提高检测精度，解决检测过程中 anchor 不匹配和特征尺度不一致的问题。

为了评估论文算法的好坏，在两个不同的测试集上进行测试，测试结果如下表 2 所示。由表 2 可知，论文算法的泛化能力和鲁棒性较好。

表 2 YOLOv3+ASFF+trick 在不同数据集上的检测结果

模型	检测算法	mAP（%）	FPS
BDD	YOLOv3	86.75	14
	YOLOv3+ASFF+trick	94.36	14
Caltech	YOLOv3	85.62	14
	YOLOv3+ASFF+trick	92.96	14

YOLOv3、YOLOv3+GA+trick、YOLOv3+ASFF+GA+trick 模型的 PR 曲线如下图 5(a)、(b)、(c) 所示。

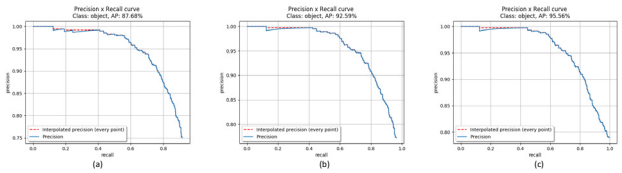
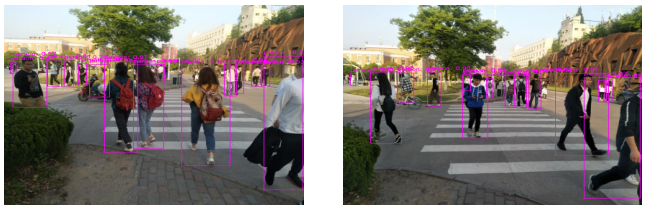


图 5 Precision−recall 曲线

为了测试模型的泛化能力，将算法用于实际场景中，检

工程技术与管理·第 04 卷·第 03 期·2020 年 03 月
DOI: https://doi.org/10.26549/gcjsygl.v4i3.3662

测结果如下图 6 显示。从图 6 可知对于近处和偏远处的行人检测效果好，且不同尺度的行人能被正确的检测到。



(a) (b)

图 6 实际场景检测效果图

5 结语

基于 YOLOv3 使用聚类生成的 anchors 尺寸不合适阻碍行人检测精度及速度这一问题，论文采用 GA 指导 anchors 生成；且基于特征金字塔的行人检测模型，不同特征尺度之间的不连续性影响行人检测准确率的问题，论文在模型中加入一种自动训练不同尺度特征融合权重的融合策略。实验验证表明在依旧满足实时性的条件下很大程度得提高了行人的检测精度。不过对于尺寸过小和遮挡的行人会出现漏检现象，后续会针对此做进一步研究。

参考文献

[1] Girshick R. Fast r−cnn[C]. Proceedings of the IEEE international conference on computer vision. 2015: 1440−1448.

[2] Girshick R, Donahue J, Darrell T, et al. Rich feature hierarchies for accurate object detection and semantic segmentation[C]. Proceedings of the IEEE conference on computer vision and pattern recognition. 2014: 580−587.

[3] Liu W, Anguelov D, Erhan D, et al. SSD: Single shot multibox detector [C]. European conference on computer vision. Springer, Cham, 2016: 21−37.

[4] Redmon J, Divvala S, Girshick R, et al. You only look once[C]. Proceedings of the IEEE conference on computer vision and pattern recognition. 2016: 779−788.

[5] Redmon J, Farhadi A. YOLO9000: better, faster, stronger [J]. arXiv preprint, 2017.

[6] Redmon J, Farhadi A. YOLOv3: An incremental improvement [J]. arXiv:1804.02767, 2018.

[7] Tao Kong, Fuchun Sun, Anbang Yao, Huaping Liu, Ming Lu,and Yurong Chen. Ron: Reverse connection with objectness prior networks for object detection. In CVPR, 2017.

[8] Songtao Liu Beihang University liusongtao Di Huang Beihang University Yunhong Wang Beihang University Learning Spatial Fusion for Single−Shot Object Detection [J]. arXiv:1911.09516v2.

[9] Research Kilian Q. Weinberger Cornell University CondenseNet: An Efficient DenseNet using Learned Group Convolutions[J]. arXiv:1711.09224v2.

[10] Lin Tsungyi, Dollár Piotr, Girshick Ross, et al. Feature pyramid networks for object detection[C]IEEE Conference on Computer Vision and Pattern Recognition. IEEE, 2017: 936 — 944.

[11] Shaoqing Ren, Kaiming He, Ross Girshick, and Jian Sun.Faster r−cnn: Towards real−time object detection with region proposal networks. In Advances in Neural Information Processing Systems, 2015.

[12] Jifeng Dai, Yi Li, Kaiming He, and Jian Sun. R−FCN: Object detection via region−based fully convolutional networks.[J] In Advances in Neural Information Processing Systems, 2016.

[13] Jiahui Yu, Yuning Jiang, Zhangyang Wang, Zhimin Cao, and Thomas Huang. Unitbox: An advanced object detection network. In ACMM, 2016.

[14] Chenchen Zhu, Yihui He, and Marios Savvides. Feature selective anchor−free module for single−shot object detection. In CVPR, 2019.

15] A 戴思达，‘准确率(Accuracy), 精确率(Precision), 召回率(Recall) 和 F1−Measure’2016,<https://www.cnblogs.com/sddai/p/5696870.html> (accessed 22 July 2016).

[16] L. Perez and J. Wang,The effectiveness of data augmentation in image classification using deep learning,2017, arXiv:1712.04621.