计算机系统应用 ISSN 1003-3254, CODEN CSAOBN
Computer Systems & Applications,2022,31(5):137−146 [doi: 10.15888/j.cnki.csa.008462]

E-mail: csa@iscas.ac.cn
http://www.c-s-a.org.cn
Tel: +86-10-62661041

# 结合坐标注意力与自适应残差连接的 logo 检测①

王　林, 范亚臣

(西安理工大学 自动化与信息工程学院, 西安 710048)
通信作者: 范亚臣, E-mail: 1254954400@qq.com

**摘　要**: Logo 检测在品牌识别和知识产权保护等领域有着广泛的应用. 针对 logo 检测中存在小尺度 logo 检测性能差和 logo 定位不准的问题, 本文提出一种基于 YOLOv4 网络的 logo 检测方法, 将 YOLOv4 网络 PANet 模块中的 5 个连续卷积层用设计的自适应残差块替换, 增强浅层和深层的特征利用, 有侧重地进行特征融合, 同时优化网络训练; 并在自适应残差块之后使用坐标注意力机制, 通过精确的位置信息对通道关系和长期依赖性进行编码, 从融合的特征中过滤和增强对于检测更有用的特征; 最后采用 K-means++聚类算法得到更适合 logo 数据集的先验框, 并分配给不同的特征尺度. 实验结果表明, 本文提出的方法在 FlickrLogos-32 和 FlickrSportLogos-10 数据集上的平均精度达到了 88.09% 和 84.72%, 较原算法分别提高了 0.91% 和 1.40%, 在定位精度和小尺度 logo 检测上的性能都显著提升.

**关键词**: logo 检测; YOLOv4; 坐标注意力; 自适应残差连接

引用格式: 王林,范亚臣.结合坐标注意力与自适应残差连接的 logo 检测.计算机系统应用,2022,31(5):137–146. http://www.c-s-a.org.cn/1003-3254/8462.html

## Logo Detection Combining Coordinate Attention and Adaptive Residual Connection

WANG Lin, FAN Ya-Chen

(School of Automation and Information Engineering, Xi'an University of Technology, Xi'an 710048, China)

**Abstract**: Logo detection has a wide range of applications in areas such as brand recognition and intellectual property protection. In order to solve problems of poor detection performance on small-scale logo and inaccurate logo positioning, a logo detection method is proposed based on the YOLOv4 network. Five continuous convolutional layers in the PANet module of YOLOv4 network are replaced by the designed adaptive residual blocks to enhance the utilization of shallow and deep features and fuse features with emphasis and optimize the model training. And the coordinate attention mechanism is used after the adaptive residual blocks to encode channel relationship and long-term dependencies through precise location information, filter and enhance the more useful features from the fused features. The K-means++ clustering algorithm is used to obtain anchor boxes which are more suitable for the logo datasets and assign those to different feature scales. The experimental results show that the mean average precision of the proposed method on FlickrLogos-32 and FlickrSportLogos-10 datasets reaches 88.09% and 84.72%, which is 0.91% and 1.40% higher than the original algorithm, respectively. The performance of the proposed method in positioning accuracy and small-scale logo detection is significantly improved.

**Key words**: logo detection; YOLOv4; coordinate attention; adaptive residual connection

① 基金项目: 陕西省科技计划重点项目 (2017ZDCXL-GY-05-03)
收稿时间: 2021-07-13; 修改时间: 2021-08-24; 采用时间: 2021-08-31; csa 在线出版时间: 2022-04-11

Logo 是指徽标或商标, 通常由图形、文字或者图形和文字的组合构成. Logo 被用于各种物体表面对物体进行标识. Logo 检测的任务是定位和识别图像中的 logo, 它在知识产权保护、产品品牌识别、智能交通车辆标识检测、社交媒体产品品牌管理等领域有很多应用. Logo 检测虽然可以被视作目标检测的一种特殊类型, 但是由于大小、旋转、光照、遮挡和形变等因素的影响, 检测自然场景图像中的 logo 是具有挑战性的.

自然图像中的 logo 检测方法大致分为传统方法和基于深度学习的方法. 传统方法依赖于手工设计的特征和轮廓, 通过特征和轮廓匹配来识别和分类, 常见特征有尺度不变特征变换 (scale invariant feature transform, SIFT)[1] 特征、加速稳健特征 (speeded up robust feature, SURF) 和方向梯度直方图 (histograms of oriented gradients, HoG)[2] 特征. 在过去的 2007–2014 年间, 手工设计的特征是大多数 logo 检测和识别方法的核心. Kleban 等人[3] 提出了一种基于数据挖掘的方案, 将每幅图像视为一个事务, 在多分辨率下寻找关联规则, 以找到与 logo 对应的局部 SIFT 特征的频繁空间配置. Gao 等人[4] 提出通过空间光谱显著性来发现 logo 区域, 对查询图像中使用的这些区域提取 SURF 特征, 然后根据提取的 SURF 特征发现数据集图像与查询图像之间的相似度. Zhang 等人[5] 提出将图像随机分割并从中提取混合特征, 包括纹理特征、形状特征、梯度方向直方图特征、HoG 特征、SIFT特征和 SURF 特征, 然后应用随机森林分类器进行 logo 检测.

近年来, 深度学习方法在计算机视觉的各个领域得到了广泛的应用, 随着深度学习的发展, 很多使用 CNNs 的 logo 检测模型被提出. Iandola 等人[6] 首次将 Faster R-CNN[7] 应用于 logo 检测, 在 GoogLeNet 结构中每个初始化层之后添加全局平均池化来辅助分类. Paleček 等人[8] 研究了优化算法、批量大小和学习率调度的具体设计对最终检测性能的影响. 同时, 对 4 种不同主干网的 3 种主要类型的探测器进行了实证评价. 通过实验观察到 Faster R-CNN 通常比 Mask R-CNN[9] 和以 RetinaNet[10] 为代表的 single shot 检测器表现得更好. 黄明珠等人[11] 考虑到 logo 的低分辨率导致的检测性能难以进一步提升, 在 Faster R-CNN 框架中结合了生成对抗模型, 利用网络先将分辨率较低的 logo 特征映射成高分辨率的表达能力更强的特征, 再送入完全连接层进行分类和回归, 从而提高检测的性能. 2020 年, Alsheikhy 等人[12] 将传递学习技术应用于深度卷积神经网络模型 DenseNet[13], 在较少的参数以及较小的计算开销下进行 logo 识别. Wang 等人[14] 引入最大的全标注 logo 检测数据集 LogoDet-3K, 并提出了一个强大的基线方法 Logo-Yolo, 它将 focal loss 和 CIoU 损失合并到最先进的 YOLOv3 (you only look once version 3) 框架[15] 中, 用于大规模的 logo 检测.

上述研究虽然在一定程度上提高了 logo 检测性能, 但仍存在一些不足. 目前存在的 logo 检测算法对小尺寸的 logo 检测不准, 并且对图像中的 logo 定位精度低, 无法在图像中准确地框出 logo 的位置, 因此本文基于 YOLOv4 算法[16] 提出了一种融合坐标注意力和自适应残差连接的 logo 检测方法, 可以提高 logo 定位精度和小尺寸 logo 的检测性能. 主要的改进包括两个方面: 一是使用设计的自适应残差块替换 5 个连续卷积层, 增强特征利用, 同时优化网络训练. 二是引入坐标注意力机制[17], 使用通道重要性和空间位置重要性来增强对于检测更有用的信息, 剔除冗余信息. 对 logo 检测数据集使用聚类算法获得最佳的先验框尺寸. 通过 FlickrLogos-32 数据集[18] 和 FlickrSportLogos-10 数据集进行训练和验证, 并在相同的环境下与 YOLOv3、YOLOv4 等检测算法进行比较, 验证改进算法的性能.

## 1 相关工作

### 1.1 YOLOv4 网络

YOLOv4 目标检测器的网络架构如图 1 所示, 主要包括输入 (input)、骨干网络 (backbone)、颈部 (neck) 和预测模块 (prediction) 4 个部分. 骨干网络是在 ImageNet 上进行预训练, 而颈部用于收集各个阶段的特征图, 预测模块用于预测物体的类别、置信度和边界框. 图 1 中, CBM 单元包含卷积层 (convolutional, CONV)、批归一化层 (BatchNormalization, BN) 和 Mish 激活函数; CBL 单元包含 CONV 层、BN 层和 LeakyReLU 激活函数; Res_unit 单元由两个 CBL 单元进行残差操作, 通过引入 BN 层和残差单元可以加快网络训练, 防止随着网络加深而出现的梯度消失以及网络退化问题; 3 个 CBL 单元和 $X$ 个 Res_unit 进行残差操作构成 CSPX 单元; SPP 单元由 4 个尺寸分别为 1×1、5×5、9×9和13×13的卷积核对输入进行最大池

化 (MaxPool) 操作, 然后拼接 (concat) 4 个分支的结果; 图中的*代表连续多个模块. YOLOv4 算法是一种端到端的目标检测算法, 相比于 YOLOv3 算法, 主要有以下 4 个方面的改进.

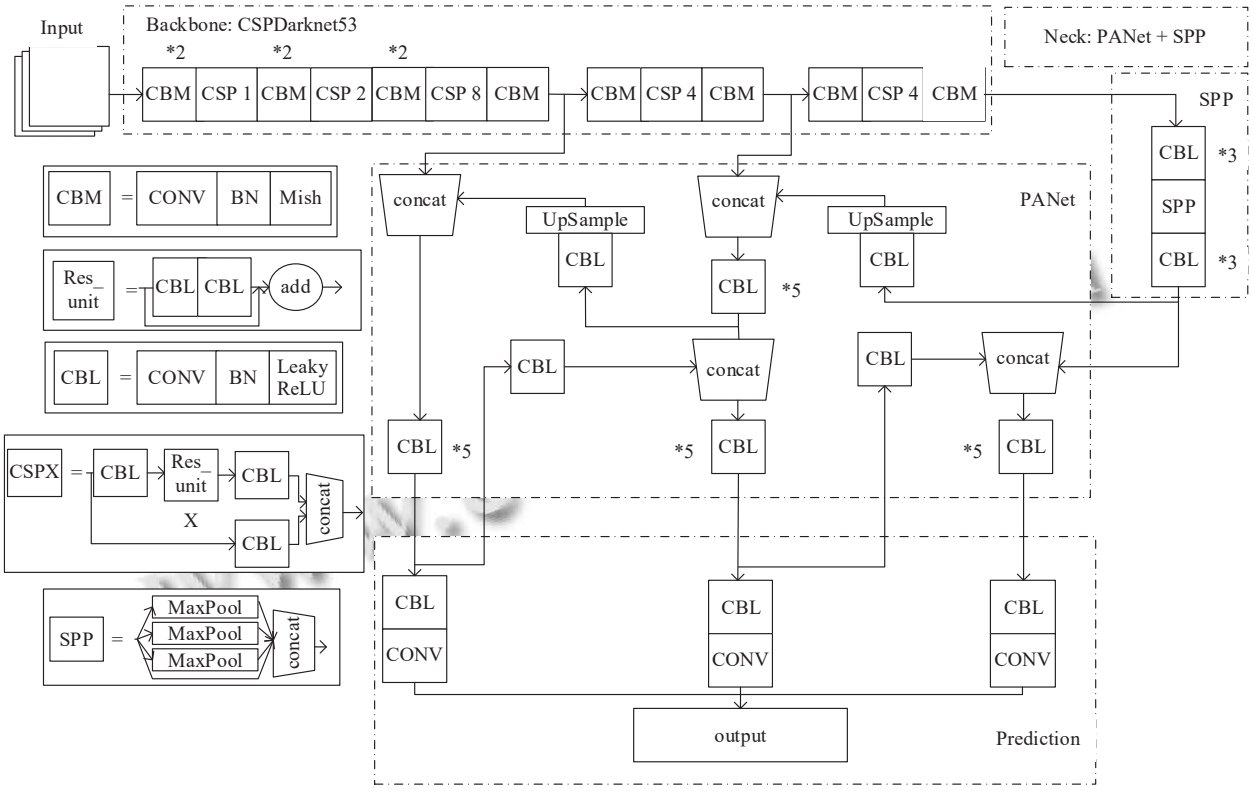


图 1 YOLOv4 网络架构图

(1) Input: 对输入数据进行 Mosaic 数据增强. 通过随机选取 4 张输入图像进行随机裁剪、拼接和排布从而丰富数据集, 一方面多样化目标可能出现的背景, 另一方面增加了小目标的数量. 这种数据增强方式扩充了数据集, 提高了模型的鲁棒性和对小目标的检测能力.

(2) Backbone: 使用更好的骨干网络 CSPDarknet53[19] 来提取输入的特征, 相比于 YOLOv3 的 Darknet53 多了 5 个 CSP 模块, 在骨干网络中使用 Mish 激活函数[20], 并使用 Dropblock 正则化方式来防止网络发生过拟合.

(3) Neck: 在 Backbone 和最后的预测模块之间添加了 SPP 模块和 PANet 模块[21], SPP 用于增大感受野, PANet 用于特征整合.

(4) Prediction: 训练时使用 CIOU 损失[22] 代替 MSE 损失, 在边界框回归问题上有更好的回归速度和准确率, 在测试阶段使用 DIOU 非极大值抑制策略.

### 1.2 注意力机制

人类的视觉系统会将有限的注意力放在重点信息上, 自动忽略不重要的信息, 注意力机制 (attention model, AM) 类似于人类的视觉系统, 它的核心思想是从关注全部到关注重点, 从而节约资源, 快速准确地获取最有效的信息. 注意力机制最初被应用于机器翻译任务中, 现在已被广泛应用在自然语言处理、统计学习、语音识别和计算机视觉任务中. 注意力模型能够显著提高神经网络性能和可解释性. 计算机视觉任务中注意力机制通常分为空间注意力、通道注意力和混合注意力, 如 SENet (squeeze-and-excitation network)[23] 和 CBAM (convolutional block attention module)[24] 等. 2021 年, Hou 等人[17] 考虑到 SENet 只考虑通道之间的信息而忽略了位置信息, 但是位置信息对于生成空间选择性注意力图非常重要, 因此作者引入了一种新的坐标注意块 (coordinate attention, CA), 它不仅仅考虑了通道间的关系还考虑了特征空间的位置信息.

## 2 YOLOv4-RCA 模型

为了获得更高的检测精度, 对 YOLOv4 网络的特征增强部分进行了改进, 提出了 YOLOv4-RCA 网络.

### 2.1 YOLOv4-RCA 网络架构

图 2 是 YOLOv4-RCA 的网络结构图. 与 YOLOv4 的结构相比, YOLOv4-RCA 在 PANet 部分有以下两个方面的改进.

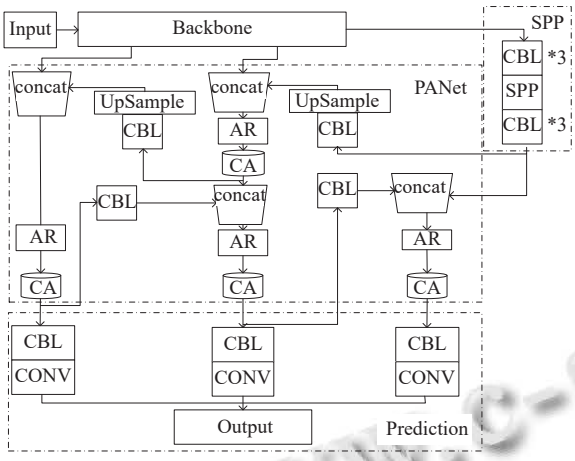


图 2 YOLOv4-RCA 架构

(1) 自适应残差块代替连续卷积

在网络中为了获得更大的感受野和更丰富的上下文信息, 通常会使用卷积来进行下采样操作, 但是下采样操作会导致原特征图中细节信息丢失. 残差单元[25]以跳层连接的形式实现, 将单元的输入直接与单元输出加在一起, 然后再激活. 残差连接可以将浅层特征送入深层网络, 在不增加过多成本的条件下融合了更多的特征信息, 增强了网络的特征表达能力, 同时很好地解决了深度神经网络的退化问题. 残差连接在一定程度上起到了细节补充的作用, 但是同时也带来了很多冗余信息, 因此本文设计了一种自适应残差连接的方式, 在融合浅层和深层特征来减少原特征图中细节信息丢失的同时能够减少冗余信息.

本文中的自适应残差 (adaptive residual) 连接方式首先对输入特征使用坐标注意力进行加权, 进一步提取出输入中的有用特征, 然后再与输出以通道相加的方式进行融合. 如图 3 所示, 对于 YOLOv4 网络 PANet 中 5 个连续卷积 CBL 块, 前两个 CBL 模块进行自适应残差连接, 第 3 和第 4 个 CBL 模块进行自适应残差连接, 这样构成设计的残差块 AR, 图 4 是 AR 的结构, 图中的 CA 代表对输入特征图通过坐标注意力进行加权, and 代表输入特征矩阵和输出特征矩阵通过逐元素相加来进行特征融合. 在图 2 中用 Res2C 模块替换了原网络架构中的 5 个连续卷积, 这样能有侧重地将输入的特征与输出特征进行融合, 增强浅层和深层的特征利用, 减少原特征图的细节信息丢失.

(2) 引入坐标注意力机制

坐标注意块给通道注意力中嵌入位置信息, 在重新权衡不同通道重要性的同时, 也考虑对空间信息进行编码. 这种编码方式可以使坐标注意力更准确地定位感兴趣对象的准确位置, 从而帮助整个模型更好地定位和识别. 如图 2 所示, 在 PANet 中 4 处 Res2C 模块之后添加坐标注意力 CA 模块. 这样可以从融合的特征中过滤和增强有用的特征, 同时抑制无用的特征, 将增强的特征送给预测部分来进行分类和定位. CA 模块不添加在主干网络中是为了不改变骨干网络 CSPDarknet53 的结构, 以使用在 ImageNet 上预训练的权重, 而无需从头开始训练网络. 添加 CA 模块可以在基本不增加计算量的同时, 提高模型区分背景和前景的能力.

### 2.2 YOLOv4-RCA 中的注意力模块

在通道注意力中通常使用全局池化来编码全局空间信息, 这种方式将全局空间信息压缩到单个通道描述符中, 因此很难保存通道中对象的空间位置信息. CA 注意力的核心思想是通过精确的位置信息对通道关系和长期依赖性进行编码, 如图 5 所示, 具体操作可分为坐标信息嵌入和坐标注意力生成两个步骤.

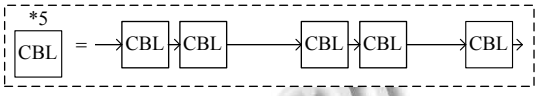


图 3 原 PANet 中的 5 个连续卷积

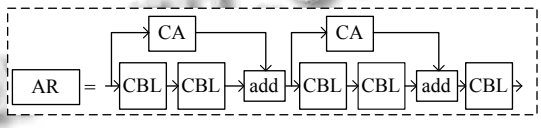


图 4 设计的自适应残差块 AR 的结构图

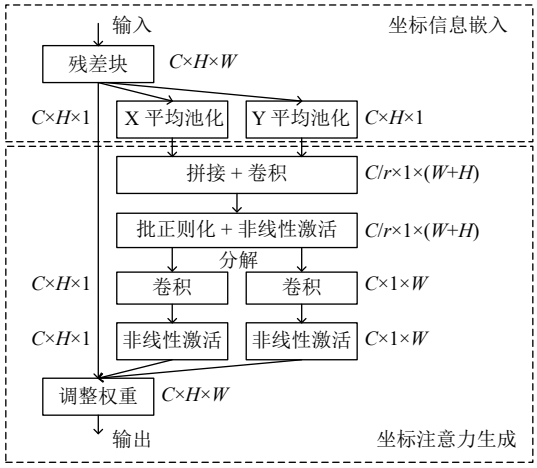


图 5 坐标注意力机制的操作过程

(1) 坐标信息嵌入

对于维度为$(C,H,W)$的输入$X$的每一个通道$x_c$, 按照式 (1) 将一个通道上的全局池化分解为沿X和Y方向两个一维特征编码操作.

$$z_c = \frac{1}{H\times W}\sum_{i=1}^{H}\sum_{j=1}^{W} x_c(i,j) \tag{1}$$

具体来说, 使用尺寸为$(H,1)$和$(1,W)$的池化核对每个通道分别沿着水平方向和垂直方向进行编码, 计算平均池化. 因此对于通道$x_c$, 高度为$h$第$c$个通道的编码输出为:

$$z_c^h(h) = \frac{1}{W}\sum_{0\leqslant i<W} x_c(h,i) \tag{2}$$

同样, 宽度为$w$的第$c$个通道的编码输出为:

$$z_c^w(w) = \frac{1}{H}\sum_{0\leqslant j<H} x_c(j,w) \tag{3}$$

这两种转换保证注意力模块捕捉到沿着一个空间方向特征的长期依赖关系, 并保存沿着另一个空间方向特征的精确位置信息, 这有助于网络更准确地定位感兴趣的信息.

(2) 坐标注意力生成

将两个方向的坐标信息嵌入进行拼接, 然后进行卷积、批正则化和非线性激活的操作, 如式 (4) 所示.

$$\boldsymbol{f} = \delta\left(F_1\left(\left[z^h,z^w\right]\right)\right) \tag{4}$$

其中, $[\cdot,\cdot]$为沿空间维度的拼接操作, $F_1$是卷积变换函数, $\delta$为非线性激活函数, 得到的$\boldsymbol{f}$为对空间信息在水平方向和垂直方向进行编码的中间特征映射. 然后将$\boldsymbol{f}$沿X和Y方向分解为张量$\boldsymbol{f}^h$和$\boldsymbol{f}^w$. 对两个张量分别进行卷积变换和非线性激活, 如式 (5) 和式 (6) 所示.

$$\boldsymbol{g}^h = \sigma\left(F_h\left(\boldsymbol{f}^h\right)\right) \tag{5}$$

$$\boldsymbol{g}^w = \sigma\left(F_w\left(\boldsymbol{f}^w\right)\right) \tag{6}$$

其中, $F_h$和$F_w$分别为对$\boldsymbol{f}^h$和$\boldsymbol{f}^w$的卷积变换函数, $\sigma$是Sigmoid 激活函数. 输出$\boldsymbol{g}^h$和$\boldsymbol{g}^w$作为注意力权重分别作用于输入$X$的水平方向和垂直方向. 最后, 输入特征$X$的一个通道$x_c$上高度为$i$宽度为$j$的特征$x_c(i,j)$, 经过坐标注意力模块的输出$y_c(i,j)$可以写成:

$$y_c(i,j) = x_c(i,j)\times g_c^h(i)\times g_c^w(j) \tag{7}$$

### 2.3 先验框尺寸

从 YOLOv2 开始, YOLO 系列算法引入了先验框 (anchor box) 的概念, 根据标注的真实框 (ground truth) 使用 K-means 聚类算法[26] 来获得 $K$ 个 anchor box, 用来提高检测的速度和准确率. YOLOv4 算法中先验框的尺寸是由 COCO (common objects in context) 数据集通过聚类算法得到的, 但是 COCO 数据集中包含了 80 个类别, 宽高比的差别较大, 检测对象的尺寸也较大, 而对于 logo 检测任务, 图像中 logo 的尺寸偏小, 同时 logo 的宽高比变化相对较少, 因此, 对于 logo 检测任务, 需要通过重新聚类来选取适合 logo 数据集的先验框尺寸.

本文选择 K-means++聚类算法对数据集中的标注框进行聚类获得 9 个先验框, 并为每个特征检测尺度分配 3 个检测框. 对 FlickrLogos-32 和 FLickrSportLogos-10 数据集先验框聚类和分配结果如表 1 所示. 相比于 YOLOv4 最初的先验框, 根据 logo 检测数据集通过聚类得到的先验框尺寸更符合训练集中 logo 的宽高比, 使用重新聚类获得的先验框对网络进行训练能够使得检测更加准确高效.

### 2.4 损失函数

总的损失包括类别损失、置信度损失和边界框回归损失 3 部分, 类别损失和置信度损失用二元交叉熵损失来计算, 对于边界框回归使用 CIOU 损失来代替 MSE 损失, 总损失的计算方法如式 (8) 所示.

$$\begin{aligned} Loss =& \lambda_{\text{coord}}\sum_{i=0}^{S\times S}\sum_{j=0}^{N} I_{ij}^{\text{obj}}\, l_{\text{CIOU}} \\ &-\sum_{i=0}^{S\times S} I_{ij}^{\text{obj}}\sum_{c\in cls}\left[p_i(c)\log\left(p_i(c)\right)+\left(1-\hat{p}_i(c)\right)\log\left(1-\hat{p}_i(c)\right)\right] \\ &-\sum_{i=0}^{S\times S}\sum_{j=0}^{N} I_{ij}^{\text{obj}}\left[C_i\log\left(C_i\right)+\left(1-\hat{C}_i\right)\log\left(1-\hat{C}_i\right)\right] \\ &-\lambda_{\text{noobj}}\sum_{i=0}^{S\times S}\sum_{j=0}^{N} I_{ij}^{\text{noobj}}\left[C_i\log\left(C_i\right)+\left(1-\hat{C}_i\right)\log\left(1-\hat{C}_i\right)\right] \end{aligned} \tag{8}$$

其中, 第 1 项是边界框回归损失, 第 2 项是类别损失, 第 3 项和第 4 项是置信度损失. $\lambda_{\text{coord}}$和$\lambda_{\text{noobj}}$是惩罚项, 分别设置 5 和 0.5, $S\times S$代表输入图像被划分成$S\times S$的网格, $N$代表每一个网格中锚框的数量. $p(c)$是一个对象属于类别$c$的概率, $C_i$是第 $i$ 个网格包含对象的置信度, 如果物体的中心点落入第 $i$ 个网格的第 $j$ 个框, 那么$I_{ij}^{\text{obj}}$为 1, $I_{ij}^{\text{noobj}}$为 0, 否则$I_{ij}^{\text{obj}}$为 0 , $I_{ij}^{\text{noobj}}$为 1. 当$I_{ij}^{\text{obj}}$为 0 时, 只计算置信度损失, 其他两项为 0. $l_{\text{CIOU}}$代表 CIOU 损失, 计算公式如式 (9) 所示, 它在 $IOU$ 损失的基础上添加了两项, 如式 (10), 式 (11) 所示.

表 1 先验框聚类和分配结果

| 特征图 | 76×76 | 38×38 | 19×19 |
|---|---|---|---|
| 感受野 | 小 | 中 | 大 |
| YOLOv4原本的先验框 | (12, 16), (19, 36), (40, 28) | (36, 75), (76, 55), (72, 146) | (142, 110), (192, 243), (459, 401) |
| FlickrSportLogos-10 | (15, 11), (24, 25), (40, 30) | (42, 58), (60, 35), (72, 49) | (94, 77), (118, 37), (200, 157) |
| FlickrLogos-32 | (14, 15), (23, 24), (48, 32) | (54, 62), (75, 113), (114, 73) | (132, 194), (172, 123), (278, 274) |

$$l_{CIOU} = 1 - IOU + \frac{d^2}{c^2} + \alpha v \tag{9}$$

$$v = \frac{4}{\pi^2}\left(\arctan\frac{w^{gt}}{h^{gt}} - \arctan\frac{w}{h}\right)^2 \tag{10}$$

$$\alpha = \frac{v}{(1-IOU)+v} \tag{11}$$

其中, $c$是真实框和预测框之间最小闭合度的对角线距离, $d^2$代表真实框和边界框中心点的欧式距离, $\alpha v$是影响因子, $\alpha$是一个用于权衡的参数, $v$用于衡量高宽比的一致性, $w$和$h$分别是预测框的宽和高, $w^{gt}$和$h^{gt}$分别是真实框的宽和高. 与 MSE 损失相比, CIOU 损失考虑了重叠面积、中心点距离和高宽比等尺度信息, 可以更好地表示真实框与预测框的拟合程度.

## 3 实验结果和分析

### 3.1 实验设置

实验在以下配置的计算机上进行: 处理器: Inter(R) Xeon(R) CPU E5-2640 v4 @2.4 GHz; 显卡: NVIDIA 1080Ti GPU, 显存 11 GB; 系统类型: 64 位 CentOS Linux 7 操作系统. 算法用深度学习框架 PyTorch1.4.0 实现, Anaconda 集成开发环境, Python 3.7 编程语言. 关于网络参数设置, 设置批训练量 batch_size 为 8, 图像在训练前尺寸调整为 608×608, 对骨干网络 CSPDarknet53 使用大型分类数据集 ImageNet 进行预训练, 获得参数初始化, 除此之外, 网络中其他结构均采用 normal 方法进行参数初始化; 优化算法使用 Adam 算法; 学习率使用余弦退火学习率衰减方法; 总共训练 100 个 epoch, 其中前 50 个 epoch 冻结骨干网络部分的权重; 使用 YOLOv4 系列算法时在训练阶段使用 Mosaic 数据增强.

### 3.2 数据集

为了验证我们提出的 YOLOv4-RCA 算法的性能, 我们在 FlickrLogos-32 数据集和 FlickrSportLogos-10 数据集上分别实验. 数据集包含的 logo 如图 6 所示. FlickrLogos-32 数据集包含了 Adidas、Aldi、Apple、Becks、Bmw 等 32 个 logo 类别, 每个类别有 70 张图像, 将官方提供的数据集标注的格式转换成 PASCAL VOC 数据集格式用来训练. FlickrSportLogos-10 数据集是一个包含 361、Adidas、Anta、Erke 和 Kappa 等 10 种体育运动品牌的数据集, 共有 2 038 张图片.

### 3.3 评估指标

为了评估所提出的算法对 logo 检测的有效性, 本文使用 COCO 评估指标, $AP$, $AP_{50}$, $AP_{75}$, $AP_S$, $AP_M$和$AP_L$, 其中, $AP$为 0.50–0.95 之间 10 个不同$IOU$设置下平均准确率的平均值, 该指标能描述模型对感兴趣对象的定位精度; $AP_{50}$是$IOU$为 0.5 时各个类别的平均准确度; $AP_{75}$更严格一些, 是$IOU$为 0.75 时各个类别的平均准确度; $AP_S$, $AP_M$和$AP_L$分别描述的是小、中和大目标的平均准确度. 同时, 使用 PASCAL 指标来评估算法在每个 logo 类别上的检测精度, 即每个类别上的$AP$和所有类别的平均检测精度$mAP$. 考虑在推理阶段的$FPS$来衡量模型的推理速度, 本实验中$FPS$是在单张 NVIDIA GTX1080ti GPU 上计算的.


图 6 FlickrLogos-32 和 FlickrSportLogos-10 数据集

### 3.4 实验过程

在 FlickrLogos-32 数据集和 FlickrSportLogos-10 数据集上分别实验, 使用 COCO 和 PASCAL 指标来评估引入自适应残差块 Res2C 和坐标注意力 CA 的 YOLOv4-RCA 算法的性能, 并将其与 YOLOv3 算法和 YOLOv4 算法进行比较.

在 FlickrLogos-32 数据集上的 COCO 评估结果如

表 2 所示. 根据表 2 的结果, 可以发现相比于 YOLOv3 算法, YOLOv4-RCA 算法在每一项指标上都有改进; 而相比于 YOLOv4 算法, YOLOv4-RCA 算法在牺牲 1.78%*FPS*的情况下, 除了指标$AP_{75}$, 在其余所有指标上都有不同程度的提高. 重点关注*AP*和$AP_S$这两个指标, 发现相比于 YOLOv4 算法, YOLOv4-RCA 算法的*AP*指标提高了 0.94, 说明 YOLOv4-RCA 算法提高了 logo 定位精度; $AP_S$指标提高了 7.76%, 表明 YOLOv4-RCA 算法在小尺寸 logo 的检测性能有了显著改善. 通过分析表 3, 可以在 FlickrSportLogos-10 数据集上得到类似的结论.

在 FlickrLogos-32 上的 PASCAL 评估结果如表 4 所示, 表 4 中展示了 3 种算法在每个类别上的准确度和在 32 个类别上的平均准确度. 从表 3 可以看出, 提出的 YOLOv4-RCA 算法在“Apple”“Dhl”和“Guiness”等 15 个 logo 类别上准确度达到最高; 32 个类别上的平均准确度*mAP*相比 YOLOv3 算法提高了 5.37%, 相比 YOLOv4 算法提高了 0.91. 通过分析表 5 也可以在 FlickrSportLogos-10 数据集上得到类似的结论.

表 2 FlickrLogos-32 数据集上的比较

| 算法 | *AP* (%) | $AP_{50}$ (%) | $AP_{75}$ (%) | $AP_S$ (%) | $AP_M$ (%) | $AP_L$ (%) | *FPS* (frame/s) |
|---|---|---|---|---|---|---|---|
| YOLOv3 | 50.57 | 81.64 | 56.24 | 20.58 | 37.20 | 57.15 | 18 |
| YOLOv4 | 59.29 | 86.97 | 72.91 | 24.94 | 54.75 | 63.30 | 19.71 |
| YOLOv4-RCA | **60.19** | **87.91** | 71.02 | **32.70** | **55.53** | **63.91** | 19.36 |

表 3 FlickrSportLogos-10 数据集上的比较

| 算法 | *AP* (%) | $AP_{50}$ (%) | $AP_{75}$ (%) | $AP_S$ (%) | $AP_M$ (%) | $AP_L$ (%) | *FPS* (frame/s) |
|---|---|---|---|---|---|---|---|
| YOLOv3 | 38.34 | 78.74 | 29.65 | 16.52 | 39.72 | 47.54 | 19.00 |
| YOLOv4 | 43.89 | 82.46 | 40.19 | 25.33 | 47.96 | 47.83 | 19.71 |
| YOLOv4-RCA | **44.60** | **84.23** | 39.72 | **25.47** | **48.72** | **49.55** | 19.36 |

表 4 3 种算法在 FlickrSportLogos-10 数据集上的比较 (PASCAL 评估) (%)

| 算法 | Adidas Corona Google Ritt | Aldi Dhl Guin Shell | Apple Erdi Hein Sing | Becks Esso Hp Starb | Bmw Fedex Milka Stel | Carls Ferra Nvidia Texa | Chim Ford Paul Tsin | Coke Fost Pepsi ups | *mAP* |
|---|---|---|---|---|---|---|---|---|---|
| YOLOv3 | 83.12<br>97.02<br>100.00<br>93.36 | 91.03<br>84.33<br>59.23<br>96.21 | 100.00<br>94.22<br>83.36<br>92.32 | 100.00<br>90.03<br>85.23<br>62.20 | 100.00<br>92.32<br>47.22<br>92.30 | 82.23<br>90.20<br>75.33<br>82.00 | 97.21<br>90.03<br>77.10<br>48.30 | 81.32<br>64.36<br>48.33<br>72.30 | 82.72 |
| YOLOv4 | 99.01<br>99.02<br>100.00<br>96.22 | 100.0<br>76.36<br>60.46<br>100.0 | 100.00<br>100.00<br>80.14<br>92.36 | 100.00<br>99.02<br>98.26<br>59.33 | 100.00<br>85.03<br>61.36<br>92.15 | 83.12<br>100.00<br>84.12<br>82.14 | 97.01<br>97.12<br>100.0<br>54.36 | 97.20<br>69.36<br>53.36<br>83.36 | 87.18 |
| YOLOv4-RCA | 94.01<br>**100.00**<br>70.00<br>90.77 | 92.86<br>**97.33**<br>**76.47**<br>90.00 | **100.00**<br>**100.00**<br>66.67<br>91.67 | **100.00**<br>**100.00**<br>**99.36**<br>**100.00** | 80.00<br>82.43<br>**71.28**<br>91.67 | **90.63**<br>**100.00**<br>82.73<br>81.31 | 93.33<br>80.00<br>99.58<br>**58.07** | 85.71<br>**99.24**<br>**71.41**<br>82.45 | **88.09** |

表 5 3 种算法在 FlickrSportLogos-10 数据集上的比较 (PASCAL 评估) (%)

| 算法 | 361 Lining | Adidas Nb | Anta Nike | Erke Puma | Kappa xtep | *mAP* |
|---|---|---|---|---|---|---|
| YOLOv3 | 80.65<br>84.22 | 75.03<br>93.65 | 94.01<br>50.85 | 88.43<br>82.03 | 65.45<br>76.15 | 79.21 |
| YOLOv4 | 85.49<br>80.00 | 81.77<br>95.62 | 96.69<br>64.58 | 88.47<br>82.45 | 76.75<br>80.31 | 83.32 |
| YOLOv4-RCA | **86.25**<br>83.21 | **89.07**<br>**96.06** | 94.26<br>**67.29** | 83.63<br>**85.31** | **78.14**<br>**83.95** | **84.72** |

为了更直观地分析 3 种算法的检测性能, 从 FlickrLogos-32 数据集中选取 3 张具有代表性的图片来对算法进行测试, 检测对比结果如图 7 所示. 图 7(a) 中, 有很多小尺寸的 logo, 仅有 YOLOv4-RCA 算法可以检测到所有小尺寸 logo 目标. 图 7(b) 中, logo 尺寸中等, 但目标聚集, 多个 logo 之间距离接近, 容易产生混淆,

YOLOv4-RCA 算法可以避免混淆, 将每个 logo 单独检测出来, 并用非常准确的边界框框住. 图 7(c) 中, 大量 logo 被遮挡, YOLOv4-RCA 算法可以检测出图中所有被严重遮挡的 logo, 效果明显好于 YOLOv3 和 YOLOv4. 通过对比分析得出结论, 尽管仍然存在一些漏检的情况, YOLOv4-RCA 在处理小目标、目标密集和遮挡等复杂场景时检测性能更好.

### 3.5 消融实验

为了验证每个改进点对网络性能的优化作用, 本文进行了消融实验对比分析. 在 FlickrLogos-32 数据集上的实验统计结果如表 6 所示, 其中改进点 1 和改进点 2 分别对应用自适应残差块替代连续卷积和引入坐标注意力模块. 从表中可以看到, 使用残差块代替连续卷积, 仅牺牲了 4% 的 $FPS$, 将 $mAP$ 从 87.18% 提高到 87.96%. 坐标注意力的引入仅牺牲了 3% 的 $FPS$, 却将平均检测精度从 87.18% 提升到 87.71%. 同时增加这两个改进在仅牺牲 1.8% 速度的情况下, 将 $mAP$ 从 87.18% 提高到 88.09%, 取得了最好的效果, 这说明提出的方法在 logo 检测中相对可靠. 通过分析表 7 也可以在 FlickrSportLogos-10 数据集上得到类似的结论.

(1) YOLOv3 (2) YOLOv4 (3) YOLOv4-RCA
(a) 对照组 1

(1) YOLOv3 (2) YOLOv4 (3) YOLOv4-RCA
(b) 对照组 2

(1) YOLOv3 (2) YOLOv4 (3) YOLOv4-RCA
(c) 对照组 3

图 7 3 种算法在 FlickrLogos-32 数据集上的检测结果

## 4 结论与展望

针对 logo 检测对小尺寸 logo 检测效果差和对 logo 定位精度低的问题, 本文基于 YOLOv4 提出改进的 logo 检测算法 YOLOv4-RCA, 在特征融合阶段使用设计的

自适应残差块替换 5 个连续卷积层来有侧重地融合特征, 在增强浅层和深层特征利用的同时避免了特征的冗余, 增强了模型的特征融合和表达能力; 在自适应残差块之后引入坐标注意力机制通过精确的位置信息对通道关系和长期依赖性进行编码, 使用通道重要性和空间位置重要性来增强对于 logo 检测更有用的特征; 最后使用 K-means++聚类算法重新选取对于数据集效果最佳的先验框. 实验结果表明, 改进的 YOLOv4-RCA 算法满足实时 logo 检测的需求, 在 FlickrLogos-32 和 FlickrSportLogos-10 数据集上的平均精度分别提高了 0.91% 和 1.40%, 同时提高了模型整体的定位精度和小尺度 logo 的检测精度.

表 6 FlickrLogos-32 数据集消融实验

| 模型 | 改进1 | 改进2 | *mAP* (%) | *FPS* (frame/s) |
|---|---|---|---|---|
| YOLOv4算法 | × | × | 87.18 | 19.71 |
| 优化模型1 | √ | × | 87.96 | 18.84 |
| 优化模型2 | × | √ | 87.71 | 19.07 |
| YOLOv4-RCA | √ | √ | **88.09** | 19.36 |

表 7 FlickrSportLogos-10 数据集消融实验

| 模型 | 改进1 | 改进2 | *mAP* (%) | *FPS* (frame/s) |
|---|---|---|---|---|
| YOLOv4算法 | × | × | 83.22 | 19.71 |
| 优化模型1 | √ | × | 83.34 | 18.84 |
| 优化模型2 | × | √ | 83.92 | 19.07 |
| YOLOv4-RCA | √ | √ | **84.72** | 19.36 |

## 参考文献


1 Lowe DG. Distinctive image features from scale-invariant keypoints. International Journal of Computer Vision, 2004, 60(2): 91–110. [doi: 10.1023/B:VISI.0000029664.99615.94]

2 Dalal N, Triggs B. Histograms of oriented gradients for human detection. 2005 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR '05). San Diego: IEEE, 2005. 886–893.

3 Kleban J, Xie X, Ma WY. Spatial pyramid mining for logo detection in natural scenes. 2008 IEEE International Conference on Multimedia and Expo. Hannover: IEEE, 2008. 1077–1080.

4 Gao K, Lin SX, Zhang YD, *et al*. Logo detection based on spatial-spectral saliency and partial spatial context. 2009 IEEE International Conference on Multimedia and Expo. New York: IEEE, 2009. 322–329.

5 Zhang YF, Zhu MM, Wang DL, *et al*. Logo detection and recognition based on classification. Proceedings of the 15th International Conference on Web-age Information Management. Macao: Springer, 2014. 805–816.

6 Iandola FN, Shen AT, Gao P, *et al*. DeepLogo: Hitting logo recognition with the deep neural network hammer. Computer Science. arXiv: 1510.02131, 2015.

7 Ren SQ, He KM, Girshick R, *et al*. Faster R-CNN: Towards real-time object detection with region proposal networks. IEEE Transactions on Pattern Analysis and Machine Intelligence, 2017, 39(6): 1137–1149. [doi: 10.1109/TPAMI.2016.2577031]

8 Paleček K. Deep learning for logo detection. 2019 42nd International Conference on Telecommunications and Signal Processing (TSP). Budapest: IEEE, 2019. 609–612. [doi: 10.1109/TSP.2019.8769038]

9 He KM, Gkioxari G, Dollár P, *et al*. Mask R-CNN. IEEE Transactions on Pattern Analysis and Machine Intelligence, 2020, 42(2): 386–397. [doi: 10.1109/TPAMI.2018.2844175]

10 Lin TY, Goyal P, Girshick R, *et al*. Focal loss for dense object detection. 2017 IEEE International Conference on Computer Vision (ICCV). Venice: IEEE, 2017. 2999–3007.

11 黄明珠, 黄文清. 基于改进 Faster R-CNN 的 Logo 目标检测方法. 计算机系统应用, 2019, 28(2): 41–48. [doi: 10.15888/j.cnki.csa.006766]

12 Alsheikhy A, Said Y, Barr M. Logo recognition with the use of deep convolutional neural networks. Engineering, Technology & Applied Science Research, 2020, 10(5): 6191–6194.

13 Huang G, Liu Z, van der Maaten L, *et al*. Densely connected convolutional networks. 2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR). Honolulu: IEEE, 2017. 2261–2269. [doi: 10.1109/CVPR.2017.243]

14 Wang J, Min WQ, Hou SJ, *et al*. LogoDet-3K: A large-scale image dataset for logo detection. arXiv: 2008.05359v1, 2020.

15 Redmon J, Farhadi A. YOLOv3: An incremental improvement. arXiv: 1804.02767v1, 2018.

16 Bochkovskiy A, Wang CY, Liao HYM. YOLOv4: Optimal speed and accuracy of object detection. arXiv: 2004.10934v1, 2020.

17 Hou QB, Zhou DQ, Feng JS. Coordinate attention for efficient mobile network design. arXiv: 2103.02907, 2021.

18 Romberg S, Pueyo LG, Lienhart R, *et al*. Scalable logo recognition in real-world images. Proceedings of the 1st ACM International Conference on Multimedia Retrieval. Lisboa: ACM, 2011. 25.

19 Wang CY, Liao HYM, Wu YH, *et al*. CSPNet: A new backbone that can enhance learning capability of CNN.

Proceedings of the 2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops (CVPRW). Seattle: IEEE, 2020. 1571–1580.

20 Misra D. Mish: A self regularized non-monotonic neural activation function. Machine Learning, 2019, 18(5): 5–21.

21 Liu S, Qi L, Qin HF, *et al*. Path aggregation network for instance segmentation. 2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition. Salt Lake City: IEEE, 2018. 8759–8768. [doi: 10.1109/CVPR.2018.00913]

22 Zheng ZH, Wang P, Liu W, *et al*. Distance-IoU loss: Faster and better learning for bounding box regression. Proceedings of the 34th AAAI Conference on Artificial Intelligence. Palo Alto: AAAI, 2020. 12993–13000.

23 Hu J, Shen L, Albanie S, *et al*. Squeeze-and-excitation networks. IEEE Transactions on Pattern Analysis and Machine Intelligence, 2020, 42(8): 2011–2023. [doi: 10.1109/TPAMI.2019.2913372]

24 Woo S, Park J, Lee JY, *et al*. CBAM: Convolutional block attention module. Proceedings of the 15th European Conference on Computer Vision. Munich: Springer, 2018. 3–19.

25 He KM, Zhang XY, Ren SQ, *et al*. Deep residual learning for image recognition. Proceedings of 2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR). Las Vegas: IEEE, 2016. 770–778.

26 Arthur D, Vassilvitskii S. K-means++: The advantages of careful seeding. Proceedings of the 8th Annual ACM-SIAM Symposium on Discrete algorithms. New Orleans: Society for Industrial and Applied Mathematics, 2007. 1027–1035.