计算机系统应用 ISSN 1003-3254, CODEN CSAOBN
Computer Systems & Applications,2022,31(12):51−58 [doi: 10.15888/j.cnki.csa.008861]

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

# 少样本条件下基于自监督改进 SimDet 模型的消毒场景目标检测①

蔡汝佳, 江文萱, 齐立哲, 孙云权

(复旦大学 工程与应用技术研究院, 上海 200082)
通信作者: 齐立哲, E-mail: qilizhe@fudan.edu.cn

**摘　要**: 日常消毒工作已经成了常态化的工作, 智能消毒机器人是非常有效的一种方式. 机器人通常通过视觉来感知周围环境, 但是基于监督学习的检测算法通常需要大量的标注数据进行训练, 当标注数据量多时, 标注成本非常高, 当标注数据量少时, 模型容易陷入过拟合, 因此少样本目标检测是一种有效的解决途径. 本文以 SimDet 模型为基础, 提出了 SimDet+模型. 第一, 针对消毒场景中的目标检测任务的特点, 增加了自监督预训练的过程, 第二, 因为存在查询图片可供参考, 对分类层进行了改进, 使用余弦相似度代替全连接层来计算置信度, 通过非参数化计算有效避免了过拟合现象. 针对消毒场景, 制作了一份 22 min 的视频数据集和包含 8 类物体的检测数据集, 分别用于两个阶段训练. 通过自监督预训练, 有效减少了数据标注成本, 同时下游任务的 mAP 从 0.216 2 提升到了 0.530 2.

**关键词**: 自监督学习; 少样本学习; 目标检测; 迁移学习; 机器人

引用格式: 蔡汝佳,江文萱,齐立哲,孙云权.少样本条件下基于自监督改进 SimDet 模型的消毒场景目标检测.计算机系统应用,2022,31(12):51–58. http://www.c-s-a.org.cn/1003-3254/8861.html

## Object Detection in Disinfection Scenes Based on Self-supervised Learning and SimDet Model under Condition of Few Samples

CAI Ru-Jia, JIANG Wen-Xuan, QI Li-Zhe, SUN Yun-Quan

(Academy for Engineering & Technology, Fudan University, Shanghai 200082, China)

**Abstract**: Intelligent disinfection robots are a highly effective way of daily disinfection as it becomes regular. Robots usually perceive the surrounding environment through vision, but object detection based on supervised learning usually requires a large amount of labeled data for training. When the amount of labeled data is large, the cost of labeling is very high, and when the amount of labeled data is small, the model is prone to overfitting. Therefore, few-shot object detection is an effective solution. On the basis of the SimDet Model, this study proposes the SimDet+ model. First, according to the characteristics of the object detection task in a disinfection scene, the process of self-supervised pre-training is added. Second, as there are query images for reference, the classification layer is improved, where the cosine similarity instead of the fully connected layer is employed for confidence level calculation, and thus the overfitting phenomenon is effectively avoided through non-parametric calculation. For the disinfection scene, a 22-minute video dataset and a detection dataset containing eight categories of objects are produced and used in two stages separately for training. Through self-supervised pre-training, the cost of data labeling is effectively reduced, and the mAP of downstream tasks is increased from 0.216 2 to 0.530 2.

**Key words**: self-supervised learning; few-shot learning; object detection; transfer learning; robot

① 基金项目: 复旦大学新型冠状病毒肺炎防治研究专项
收稿时间: 2022-04-17; 修改时间: 2022-05-22; 采用时间: 2022-05-28; csa 在线出版时间: 2022-08-12

机器人可以通过视觉、激光雷达、超声波等多种传感器感知周围的环境, 目前视觉是机器人感知最重要的一种方式, 目标检测是计算机视觉任务中极其重要和基础的一项任务, 目标检测就是检测出输入图像中是否存在特定类别的物体, 如果存在的话通常用一个矩形框标注出来并输出物体的位置信息. 目前主流的目标检测算法都是基于监督学习的方法, 虽然检测效果非常不错, 但是这些方法的训练都需要大量的标注数据, 当数据过少时非常容易陷入过拟合, 没有足够好的泛化性. 因此在数据标注较少的情况下, 少样本目标检测算法是一种非常有效的解决方案.

自监督学习因为不需要使用标签数据, 所以近几年广受欢迎. 自监督学习被 LeCun 等[1] 称为智能的暗物质, 它可以使人工智能系统从更大量级的数据中学习, 这对于理解和识别更为微妙、更不常见的世界表示模式很重要. 自监督学习通过构建一个代理任务 (pretext) 来学习特征表示, 进而迁移到下游任务, 比如说分类、目标检测、语义分割.

因为消毒场景多样化, 并且不同的场景可能需要对不同的物体进行消毒, 对于不同场景分别进行数据标注需要耗费大量人力成本, 但是机器人工作过程中采集的视频数据与检测物体数据集属于同源数据集, 因此增加了自监督预训练过程, 从预训练和微调两个阶段进行训练, 如图 1 所示. 消毒机器人在执行日常任务的过程中可以通过相机采集到很多真实消毒场景的视频数据, 因此, 第 1 阶段通过自监督学习利用视频数据来训练模型的骨干网络. 第 2 阶段将 MoCo 的骨干网络的 encoder 部分迁移到少样本目标检测算法 SimDet+ 中, 在此基础上继续训练, 保证了在降低人工标注成本的同时获得较好的检测效果.

## 1 相关研究

### 1.1 目标检测相关研究

目前的目标检测算法主要包含 anchor-based 和 anchor-free 算法. Anchor-based 算法是目前的主流目标检测算法, 需要通过预先设置 anchor 来得到最终的检测框, anchor-free 算法是近几年新提出的目标检测算法, 无需 anchor 的设置, 可以直接通过中心点或者角点回归得到检测框. Anchor-based 算法有单步 (one-stage) 和两步 (two-stage) 模型两类. 目标检测的单步模型是指没有独立地、显示地提取候选框 (region proposal), 直接由输入图像可以得到其中存在的特定物体的类别和位置信息的模型. 而两步模型通常第 1 步提取可能的候选框, 然后第 2 步对候选框进行分类和微调操作, 确定候选框内物体的类别和位置.

典型的单步模型有 OverFeat[2]、SSD[3]、YOLO 系列[4–7]、RetinaNet[8] 等, 两步模型有 R-CNN[9]、Fast R-CNN[10]、Faster R-CNN[11] 等.

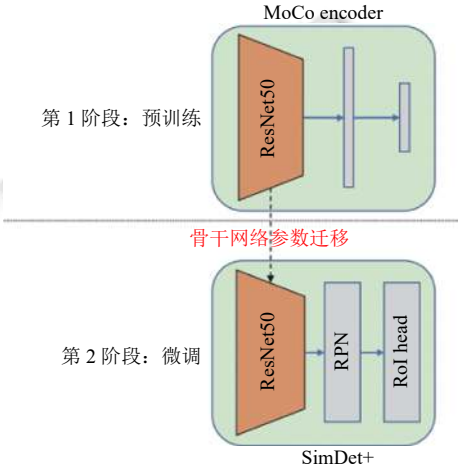


图 1 整体网络训练流程

Faster R-CNN 是在 Fast R-CNN 的基础上, 将其最耗时的候选框区域提取步骤 (即选择性搜索) 用一个区域候选框网络 (region proposal network, RPN) 代替, 并且这个 RPN 和用于检测的 Fast R-CNN 网络共享特征提取部分权值. 在 Faster R-CNN 中, 一幅图片先由 RPN 提取候选框, 再提取出各个候选框对应的特征图, 送入 Fast R-CNN (独立于 RPN 的后半部分) 进行物体分类和位置信息回归. Faster R-CNN 作为两步模型的代表之作, 第一次做到了实时的物体检测, 具有里程碑意义. 因为 Faster R-CNN 相比一阶段模型检测更加稳定和高效, 本文的少样本目标检测模型也是基于 Faster R-CNN 的.

对于目标检测任务而言, 检测类别是确定的, 比如说数据中包含 60 类数据, 那么模型预测也会是这 60 类中的任意一类. 但是对于少样本目标检测任务而言是不一定的, 在少样本目标检测中, 通常称待检测的图片为目标图片 (target), 称参考的图片称为查询图片 (query), 目标图片会根据查询图片的不同检测出不同的结果.

CoAE[12] 和 SimDet[13] 是两个非常有效的少样本

目标检测算法. CoAE[12] 是针对单样本目标检测 (one-shot object detection) 的方法, 通过非局部化 (non-local) 操作来建模查询和目标的关系来提升 RPN 的准确率, 此外通过 SCE 模块 (squeeze and co-excitation) 增强目标特征通道上与查询特征相关的部分. SimDet[13] 通过交叉相似度特征增强模块来提高相关位置的特征, 提高了检测效果. 但是上述算法都是针对 RPN 阶段进行改进的, 本文延续非参数化相似度的思想, 在最终的分类部分进行改进, 同时将自监督学习和少样本目标检测相结合, 提出了 SimDet+模型, 有效提高了在消毒场景的检测效果.

### 1.2 自监督学习相关研究

目前自监督学习的主流方法是基于生成式[14–16] 的方法和基于对比[17–20] 的方法. 基于生成式的方法主要关注重建误差, 比如针对一张图片, 随机对一部分盖住, 然后让模型去预测, 得到的结果与真实的图像之间的误差作为损失. 基于对比的方法一般是对同一张图片进行数据增强作为正样本, 其他的图片作为负样本, 希望模型在特征空间上可以分辨出样本是否为同一类别.

基于生成式的方法主要有 CPC[14]、MAE[15]、BEiT[16] 等. MAE 采用了一种非常简单的方法, 针对一张图片, 随机掩盖图片中的一部分块, 然后去预测重构这些被掩盖住的像素. 这些方法效果都非常显著, 但是因为使用了 Transformer, 参数量比较大.

基于对比学习的方法主要有 InstDisc[17]、InvaSpread[18]、SimCLR[19] 等. InstDisc 首次提出了实例判别 (instance discrimination) 代理任务, 为后续的相关工作提供了基础. InvaSpread 采用了端到端的学习, 只需要一个编码器即可, 在正负样本的选择上只采用了同一小批量数据中的样本, 将本身数据增强之后的样本作为正样本, 其他样本作为负样本. SimCLR 在 InvaSpread 的基础上, 做出了 3 点改进, 使得自监督学习的效果可以和监督学习相媲美, 分别是数据增强、增加模型的非线性表示和更长的训练时间. MoCo[20] 主要是针对负样本的问题做出了改进, 使用动量编码器 (momentum encoder) 和队列 (queue) 来解决这个问题. 本文采用了 MoCo 网络并根据下游任务和硬件设备的限制做了改动.

## 2 模型训练

本文的实验流程为 MoCo 自监督预训练和少样本目标检测两个阶段, 如图 1 所示, 首先通过自监督学习利用视频数据训练 MoCo 模型, 第 2 步将 MoCo 模型中编码器部分的参数迁移到下游检测任务中, 第 3 步在此基础上继续进行微调. 接下来将分别展开介绍.

### 2.1 MoCo 预训练

#### 2.1.1 MoCo 模型

MoCo 无需使用更大的批量大小, 对于硬件设备更加友好, 同时 MoCo 和下游任务模型有基本相似的骨干网络, 因此本文使用 MoCo 来预训练模型骨干网络, 便于参数迁移. MoCo 的网络结构如图 2 所示, 主要由编码器, 动量编码器和样本字典 (队列) 构成, 编码器和动量编码器的网络结构相同, 只是参数的更新方式不同, 都是负责对样本进行编码. 样本字典使用的是队列, 负责存储负样本, 队列中存储的是近期用于训练的批量的样本的特征向量. 在每次更新的时候, 旧的小批量的特征向量出队, 新的小批量特征向量入队列. 这个特征向量是图像经过编码器编码之后的图像特征. 编码器使用的是 ResNet50, 本文中设置的最后一层是 2 048 个节点的全连接层, 在此之后, 又接了一层 ReLU 激活函数层和一层 128 个节点的全连接层, 通过增加非线性层来增加了模型的表示效果. 在自监督的训练过程中使用的是编码器网络结构, 但是当迁移到下游的少样本目标检测任务的时候, 本文只使用了骨干网络中相关的模块, 没有使用最后的两层全连接层.

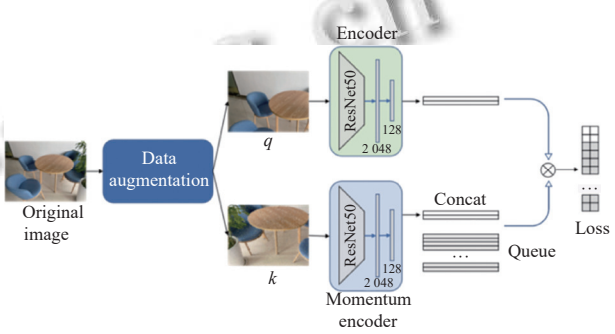


图 2 MoCo 网络结构图

MoCo 模型中使用了混合批量归一化 (shuffle batch normalization, shuffle BN), 只是在动量编码器提取特征的过程中使用的. 所谓混合批量归一化, 就是在分布式特征提取之前将样本随机打散, 提取完特征之后再恢复到原来的顺序. 保证了每个 batch 中的正样本是随机的, 防止在批量归一化的过程中发生信息泄漏. 但是在本实验过程中使用的是一张 GPU, 因此没有使用混合批量归一化, 使用的还是 ResNet 中的批量归一化.

2.1.2 动量参数更新

MoCo 使用队列来存储负样本, 在学习的过程中, 正样本是数据增强之后的样本, 负样本直接从队列中顺序出队, 然后通过计算损失值来更新编码器, 对于动量编码器则采用动量更新, 既可以拥有较大的负样本采样空间, 又可以基本上保持负样本更新的一致性.

$$\theta_k \leftarrow m\theta_k + (1-m)\theta_q \tag{1}$$

其中, $\theta_k$代表动量编码器的参数, $\theta_q$代表编码器的参数, $m \in [0,1)$表示控制动量更新的系数, 在实验中设置的是 0.999. 由于$\theta_q$的参数更新是直接使用反向传播来更新的, 因此$\theta_k$的参数更新要比$\theta_q$的更新更加平滑.

2.1.3 损失函数

MoCo 中使用的对比损失函数为 InfoNCE loss, 其公式为:

$$L_q = -\log \frac{\exp\left(q \cdot \frac{k_+}{\tau}\right)}{\sum_{i=0}^{K} \exp\left(q \cdot \frac{k_i}{\tau}\right)} \tag{2}$$

其中, $q$代表样本本身, $k_+$代表正样本, $k_i$代表负样本, 负样本就是队列中的全部数据, $q$和$k_+$的点积表示样本本身与正样本的相似性, $q$和$k_i$的点积表示样本与负样本的相似性, $\tau$为控制分布浓度水平的温度超参数. 对比学习即希望与正样本有更高的相似性, 与负样本有更低的相似性.

### 2.2 参数迁移

MoCo 的骨干网络采用的是 ResNet50, 下游检测任务 SimDet+的骨干网络采用的也是 ResNet50, 因此可以进行参数迁移, 下游任务无需从零开始训练, 有利于避免在少样本条件下的过拟合现象. MoCo 包含两个分支, 分别为编码器部分和动量编码器部分, 编码器负责对查询图片编码, 动量编码器负责对正样本编码, 参数迁移的过程中使用的是编码器的参数. ResNet50 包含 4 个 layer 层, 在迁移之后的微调阶段, 冻结了前两个 layer 层, 只针对后面两个 layer 层和后续检测头进行继续训练. 前两层的参数保持了和预训练阶段一致, 保证了预训练对于检测任务的作用性, 同时也不会导致模型针对检测任务过拟合.

### 2.3 少样本目标检测模型: SimDet+

本文的少样本目标检测算法 SimDet+模型是以 SimDet[14] 为基础进行改进的. 在少样本目标检测任务中, 采用了元学习的方法, 即构造一个个的任务来进行训练, 每个任务包含查询图片 query 和目标图片 target. 目前此类方法的主要问题就是目标特征和查询特征交互较少, 因此最终的检测效果较差. SimDet 使用查询特征和目标特征做相似度计算, 然后来增强目标特征中可能存在物体的区域特征, 但是 SimDet 是在 RPN 阶段之前做的. 本文延续了少样本 SimDet 中相似度计算的思想, 因为相比较目标检测算法, 本任务中包含查询图片可供参考, 因此在 RoI 检测头分类的过程中抛弃了全连接层, 利用查询特征和 RoI Align 之后的特征通过余弦相似度来计算物体的置信度, 同时将交叉熵损失函数换成均方差损失函数. 余弦相似度非参数化的计算避免了在少样本条件下模型陷入过拟合, 整体的网络结构如图 3 所示.

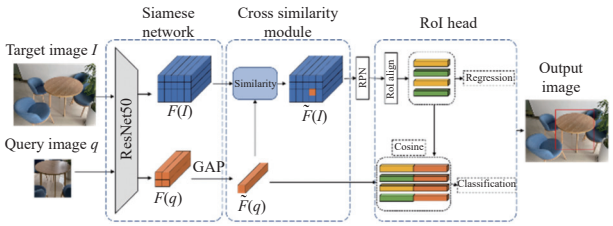


图 3 改进的 SimDet+结构图

SimDet+的网络结构主要包含 4 部分: 孪生网络特征提取模块、交叉相似度特征增强模块、RPN 候选框提取模块和 RoI 检测头.

SimDet+中最重要的部分就是交叉相似度特征增强模块, 结构如图 4 所示, 输入分别包含 3 个部分: 目标特征$F(I)$, 查询特征$F(q)$和真实的相似度图 IoU map. 目标特征$F(I) \in \mathbb{R}^{C \times W_I \times h_I}$是目标图片经过孪生网络提取之后的特征, 查询特征$F(q) \in \mathbb{R}^{C \times W_q \times h_q}$是查询图片经过孪生网络提取之后的特征, 两者在通道维度上是一致的, 目的是为了后续的相似度计算. 首先对查询特征$F(q)$做全局平均池化 (global average pooling, GAP)[21] 处理, 得到$\tilde{F}(q)$. 使用 GAP 代替传统的全连接层, 一方面可以减少参数量, 因为 GAP 是没有参数的, 同时也不容易使模型陷入过拟合. 第 2 步是将$\tilde{F}(q)$复制$w_I \times h_I$份, 使其和$F(I)$具有相同的形状. 第 3 步对于两个向量在特征维度上计算余弦相似度$S$. 第 4 步, 将相似度乘以原始特征$F(I)$, 即得到了特征增强之后的目标特征$\tilde{F}(I)$, 其中包含了查询图片的信息, 有助于缓解后续 RPN 阶段针对可见类别的偏向性.

本文对于 SimDet 的改进在于 RoI 检测头部分, 原本的计算分类部分是将 RoI 之后的特征和查询特征 $\tilde{F}(q)$拼接之后, 通过两层全连接层计算类别概率. 在少样本条件下容易陷入过拟合, 泛化性较差. 因此本文使用查询特征$\tilde{F}(q)$和 RoI 之后的特征做相似度计算, 直接作为类别概率. 对应的也将原先的交叉熵损失函数更换为了均方差损失函数.

$$L_{\text{cls}} = \frac{1}{N}\sum_{i=1}^{N}(y_i - \hat{y}_i)^2 \tag{3}$$

其中, $y_i$为查询特征向量$\tilde{F}(q)$, $\hat{y}_i$为 RoI 之后的特征向量.

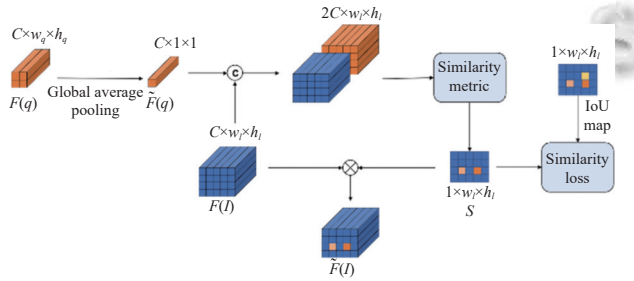


图 4 交叉相似度特征增强模块

最终整个网络全部的 loss 为:

$$L = L_{\text{Sim}} + L_{\text{MR}} + L_{\text{reg}} + L_{\text{cls}} + L^R_{\text{reg}} + L^R_{\text{cls}} \tag{4}$$

$L_{\text{Sim}}$是 SimDet 中提出的交叉相似度损失函数, $L_{\text{MR}}$为 CoAE 中提出的边缘排序损失函数, 目的是通过 RoI Align 对 RPN 产生的候选框进行排序, 来提高 RPN 候选框的质量, $L^R_{\text{reg}}$, $L^R_{\text{cls}}$分别为 Faster R-CNN 中 RPN 的回归和分类损失函数, $L_{\text{reg}}$为 Faster R-CNN 中的回归损失函数, $L_{\text{cls}}$为上述均方差损失函数.

## 3 实验

### 3.1 数据集

本文在室内场景下采集制作了消毒场景数据集. 消毒场景数据集包括: 用于自监督训练的视频数据, 以及用于少样本目标检测的数据集. 本文自监督预训练的视频数据集由机械臂上的 RealSense 深度相机采集, 仅使用 RGB 信息. 视频时长为 22 min, 训练过程前将视频数据抽帧存储为图片格式, 抽帧间隔为 1, 6, 30, 分别对应数据集 video-1, video-6, video-30.

消毒场景数据集的检测数据总共包含 8 类办公楼常见的物品: 桌子、椅子、柜子、门、电梯面板、洗手台、扶手、窗户, 如图 5 所示. 此数据集总共包含 107 张图片, 将 89 张划分为训练集, 将 18 张划分为测试集. 图片的像素为 4032×3024, 每张图片大约有 2 个物体. 因为数据量较少, 本文使用了马赛克数据增强对训练集进行数据增强, 数据统计如表 1 所示.


图 5 办公楼场景消毒物品展示

表 1 少样本目标检测数据集统计

| 数量 | 类别 | 训练集 | 增强之后 | 测试集 |
|---|---|---|---|---|
| 目标数量 | 桌子 | 12 | 267 | 3 |
| | 椅子 | 39 | 731 | 14 |
| | 门 | 31 | 734 | 6 |
| | 柜子 | 15 | 368 | 4 |
| | 电梯面板 | 17 | 357 | 5 |
| | 洗手台 | 10 | 266 | 2 |
| | 扶手 | 26 | 561 | 3 |
| | 窗户 | 15 | 338 | 5 |
| 图片数量 | — | 89 | 890 | 18 |

### 3.2 自监督学习预训练的影响实验

MoCo 模型使用 PyTorch 框架[22] 实现, 数据增强部分主要使用了 Torchvision 中自带的实现方式, 具体的数据增强方法是随机裁剪和调整尺寸、色彩抖动、丢失颜色色彩、增加高斯模糊、水平翻转. 模型的骨干网络使用的是 ResNet50, 批量大小为 32, 总共训练了 200 个 epoch. 使用 SGD 优化器, 初始学习率为 0.03, 其中的动量参数为 0.9, 使用的是余弦学习率调整策略. 特征向量的维度是 128, 队列的长度为 8 192, 温度调节参数$\tau$为 0.07, 动量编码器的参数为 0.999.

表 2 展示了使用不同数据集进行预训练的模型在下游任务的 mAP 值, 这里选择全部的类别作为可见类别, 训练数据为数据增强之后的训练集, 验证数据为测试集数据. 通过实验数据可以发现: (1) 当视频抽帧的间隔为 30 帧时效果最好, 也即 video-30 数据集, 并且此时数据量最少, 不同图片之间具有较好的差异性. (2) 预训练和非预训练效果差异非常明显, 几乎所有的

模型使用预训练之后几乎都有一倍的提升, 由此可见预训练对于消毒场景中少样本目标检测的重要性. (3) 相比较于 CoAE 和 SimDet 模型, SimDet+提升非常明显, 由此可见使用非参数化余弦相似度分类的重要性.

表 2 自监督预训练对少样本目标检测的影响

| 数据集 | mAP | | |
|---|---|---|---|
| | CoAE | SimDet | SimDet+ |
| video-1 | 0.2962 | 0.2870 | 0.4239 |
| video-6 | 0.3263 | 0.3097 | 0.3890 |
| video-30 | 0.2867 | 0.3775 | **0.5302** |
| without pretrain | 0.1856 | 0.2046 | 0.2162 |

### 3.3 SimDet+有效性实验

首先针对消毒场景的少样本目标检测任务做了实验. 根据少样本目标检测任务常见的设置, 先对于数据集进行划分, 总共分了 4 组, 每一组选择 2 类数据作为未见类别的数据, 其他的类别用于训练, 模型训练使用的是马赛克数据增强之后的训练集数据.

SimDet+模型同样使用 PyTorch 框架实现, 使用的是上文 video-30 预训练的骨干网络, 迁移到检测任务之后, 将 ResNet50 的 Layer3 之前的模块全部冻结, 从 Layer3 开始继续训练. 批量大小为 8, 总共训练了 100 个 epoch. 同样使用了 SGD 优化器, 初始学习率为 0.01, 每隔 40 个 epoch, 学习率调整减小到原来的 1/10.

通过表 3 可以看出, 在测试集上的评估中, 改进的 SimDet+相比较 SimDet 检测结果有所提升, 相比 CoAE 提升非常明显. 少样本目标检测中未见类别的设置, 主要是验证模型的泛化性, 但是实验结果显示, 对于未见类别的检测, 3 个模型效果都比较差, 因为消毒场景本身数据量较少, 训练导致了对于可见类别过拟合. 在实际应用的过程中, 是将全部类别都作为可见类别的, 并不影响模型的实际部署. 实验中扶手和窗户的检测效果相对较差, 因为数据集中的扶手基本上都是长条状的, 因此较难检测, 窗口很多都是透明的, 差异性比较大, 因此检测效果也较差.

表 3 测试集检测效果评估

| 数据划分 | 模型 | 模型可见类别 AP | | | | | | mAP | 模型未见类别 AP | | mAP | 总mAP |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | | 桌子 | 椅子 | 门 | 电梯面板 | 洗手台 | 柜子 | | 扶手 | 窗户 | | |
| split1 | CoAE | 0.1818 | 0.0455 | 0.0000 | 0.4545 | 1.0000 | 0.5455 | 0.37121 | 0.0000 | 0.0000 | 0.0000 | 0.2784 |
| | SimDet | 0.6364 | 0.1970 | 0.1636 | 0.0000 | 1.0000 | 1.0000 | 0.49949 | 0.0000 | 0.0000 | 0.0000 | 0.3746 |
| | SimDet+ | 0.2380 | 0.2619 | 0.2836 | 0.4799 | 1.0000 | 0.9454 | **0.5348** | 0.0000 | 0.0000 | 0.0000 | **0.4011** |
| 数据划分 | 模型 | 模型可见类别 AP | | | | | | mAP | 模型未见类别 AP | | mAP | 总mAP |
| | | 桌子 | 椅子 | 门 | 电梯面板 | 扶手 | 窗户 | | 洗手台 | 柜子 | | |
| split2 | CoAE | 0.0606 | 0.1036 | 0.1033 | 0.7013 | 0.0280 | 0.0545 | 0.1752 | 0.0165 | 0.0000 | 0.0083 | 0.1335 |
| | SimDet | 0.1818 | 0.1818 | 0.3367 | 0.8182 | 0.3636 | 0.0000 | 0.3137 | 0.0000 | 0.0000 | 0.0000 | 0.2353 |
| | SimDet+ | 0.6364 | 0.2445 | 0.2745 | 0.8182 | 0.0000 | 0.0119 | **0.3309** | 0.0109 | 0.0188 | **0.0149** | **0.2519** |
| 数据划分 | 模型 | 模型可见类别 AP | | | | | | mAP | 模型未见类别 AP | | mAP | 总mAP |
| | | 桌子 | 椅子 | 洗手台 | 柜子 | 扶手 | 窗户 | | 门 | 电梯面板 | | |
| split3 | CoAE | 0.0843 | 0.1979 | 1.0000 | 0.5657 | 0.0130 | 0.0000 | 0.3102 | 0.0121 | 0.0000 | 0.0061 | 0.2341 |
| | SimDet | 0.0145 | 0.2187 | 1.0000 | 1.0000 | 0.0000 | 0.2727 | 0.4177 | 0.0114 | 0.0000 | 0.0057 | 0.3147 |
| | SimDet+ | 0.3636 | 0.1994 | 1.0000 | 0.8636 | 0.3636 | 0.0000 | **0.4651** | 0.0107 | 0.0178 | **0.0142** | **0.3523** |
| 数据划分 | 模型 | 模型可见类别 AP | | | | | | mAP | 模型未见类别 AP | | mAP | 总mAP |
| | | 门 | 电梯面板 | 洗手台 | 柜子 | 扶手 | 窗户 | | 桌子 | 椅子 | | |
| split4 | CoAE | 0.0965 | 0.8182 | 1.0000 | 0.5455 | 0.0280 | 0.0000 | 0.4147 | 0.0152 | 0.0472 | 0.0312 | 0.3188 |
| | SimDet | 0.4586 | 0.7273 | 1.0000 | 0.6136 | 0.0000 | 0.0000 | 0.4666 | 0.0000 | 0.0160 | 0.0080 | 0.3519 |
| | SimDet+ | 0.2955 | 0.7455 | 1.0000 | 1. 0000 | 0.1212 | 0. 0000 | **0.5270** | 0. 0000 | 0.0222 | **0.0111** | **0.3980** |

其次本文也针对交叉相似度特征增强模块、边缘排序损失函数、全连接层分类、余弦相似度分类进行了消融实验. 通过表 4 可以得出结论: (1) 在其他部分相同的情况下, 使用余弦相似度分类要优于全连接层分类; (2) 交叉相似度特征增强模块和边缘排序损失函数具有重要的作用, 作用都是提升 RPN 阶段生成候选

框的质量; (3) 余弦相似度分类和其他模块相结合可以达到最优的效果, 因为此模块重点在于提升 RoI 检测头分类的效果, 与其他模块是不冲突的.

表 4 消融实验

| Cosine similarity module | Margin loss | Fc cls | Cosine cls | mAP |
|---|---|---|---|---|
| — | — | √ | — | 0.3742 |
| — | √ | √ | — | 0.4044 |
| — | — | — | √ | 0.3935 |
| — | √ | — | √ | 0.4186 |
| √ | — | √ | — | 0.3647 |
| √ | √ | √ | — | 0.4827 |
| √ | — | — | √ | 0.4141 |
| √ | √ | — | √ | **0.5302** |

### 3.4 消毒机器人部署

为了验证本文算法的有效性, 本文在课题组自主研发的消毒机器人上进行真实场景测试. 消毒机器人的主要功能就是对于办公场景的自动化喷雾消毒和擦拭消毒, 少样本目标检测任务的主要功能是粗定位消毒物体, 然后调用机械臂控制程序通过深度相机进行精确定位执行擦拭或者喷雾任务.

消毒机器人如图 6 所示, 主要由移动 AGV, 消毒机器人主体和六轴机械臂 3 部分组成. 在应用的过程中, 因为所有检测类别都是固定的, 所以本文应用的模型是通过将 8 类物体看作可见类别训练得到的. 在模型的部署过程中也进行了部分优化, 第一, 在使用的过程中, 本文将目标图片的特征向量直接存储到程序中, 减少了目标图片提取特征的过程, 节省了一部分计算时间. 第二, 通过上文的模型讲解可以知道, 检测过程是根据查询图片进行单类别目标检测的, 为了加速计算过程, 在部署时将多个类别查询图片在 batch 维度拼接, batch 上的每一个维度负责检测一类物体, 因此通过并行计算加速了模型的推理速度. 图 7 展示了一组真实场景的检测案例.

## 4 结论

本文针对少样本条件下消毒场景中可以轻易获取同源视频数据, 并且无需人工标注的特点, 从自监督预训练和少样本目标检测微调两个阶段入手, 延续了 SimDet 中相似度计算的思想, 针对少样本目标检测模型 SimDet 中的 RoI 检测头进行了改进, 提出了 SimDet+模型. 针对办公楼消毒场景, 本文制作了用于自监督预训练和目标检测标注的相关数据集, 并进行了验证, 实验结果表明: 第一, 自监督预训练对于消毒场景是非常有必要的; 第二, 提出的 SimDet+模型有效提高了少样本目标检测的效果. 最后本文还将模型在消毒机器人上进行了部署和应用.

图 6 消毒机器人

图 7 消毒场景检测结果案例

## 参考文献

1 LeCun Y, Misra I. Self-supervised learning: The dark matter of intelligence. Meta AI, 2021: 23.

2 Sermanet P, Eigen D, Zhang X, *et al*. Overfeat: Integrated recognition, localization and detection using convolutional networks. arXiv:1312.6229v4, 2014.

3 Liu W, Anguelov D, Erhan D, *et al*. SSD: Single shot MultiBox detector. Proceedings of the 14th European Conference on Computer Vision. Amsterdam: Springer, 2016. 21–37.

4 Redmon J, Divvala S, Girshick R, *et al*. You only look once: Unified, real-time object detection. Proceedings of 2016 IEEE Conference on Computer Vision and Pattern Recognition. Las Vegas: IEEE, 2016. 779–788.
5 Redmon J, Farhadi A. YOLO9000: Better, faster, stronger. Proceedings of 2017 IEEE Conference on Computer Vision and Pattern Recognition. Honolulu: IEEE, 2017. 6517–6525.
6 Redmon J, Farhadi A. YOLOv3: An incremental improvement. arXiv:1804.02767, 2018.
7 Bochkovskiy A, Wang CY, Liao HYM. YOLOv4: Optimal speed and accuracy of object detection. arXiv:2004.10934, 2020.
8 Lin TY, Goyal P, Girshick R, *et al*. Focal loss for dense object detection. Proceedings of 2017 IEEE International Conference on Computer Vision. Venice: IEEE, 2017. 2999–3007.
9 Girshick R, Donahue J, Darrell T, *et al*. Rich feature hierarchies for accurate object detection and semantic segmentation. Proceedings of 2014 IEEE Conference on Computer Vision and Pattern Recognition. Columbus: IEEE, 2014. 580–587.
10 Girshick R. Fast R-CNN. Proceedings of 2015 IEEE International Conference on Computer Vision. Santiago: IEEE, 2015. 1440–1448.
11 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]
12 Hsieh TI, Lo YC, Chen HT, *et al*. One-shot object detection with co-attention and co-excitation. Advances in Neural Information Processing Systems. 2019, 32: 1–10.
13 Cai RJ, Qin YJ, Qi LZ, *et al*. SimDet: Cross similarity attention for one-shot object detection. Proceedings of 2021 International Joint Conference on Neural Networks (IJCNN). Shenzhen: IEEE, 2021. 1–7.
14 van den Oord A, Li YZ, Vinyals O. Representation learning with contrastive predictive coding. arXiv:1807.03748, 2018.
15 He KM, Chen XL, Xie SN, *et al*. Masked autoencoders are scalable vision learners. arXiv:2111.06377, 2021.
16 Bao HB, Dong L, Wei FR. BEiT: BERT pre-training of image transformers. arXiv:2106.08254, 2021.
17 Wu ZR, Xiong YJ, Yu SX, *et al*. Unsupervised feature learning via non-parametric instance discrimination. Proceedings of 2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition. Salt Lake City: IEEE, 2018. 3733–3742.
18 Ye M, Zhang X, Yuen PC, *et al*. Unsupervised embedding learning via invariant and spreading instance feature. Proceedings of 2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition. Long Beach: IEEE, 2019. 6203–6212.
19 Chen T, Kornblith S, Norouzi M, *et al*. A simple framework for contrastive learning of visual representations. Proceedings of the 37th International Conference on Machine Learning. Vienna: PMLR, 2020. 1597–1607.
20 He KM, Fan HQ, Wu YX, *et al*. Momentum contrast for unsupervised visual representation learning. Proceedings of 2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition. Seattle: IEEE, 2020. 9726–9735.
21 Lin M, Chen Q, Yan SC. Network in network. arXiv: 1312.4400, 2013.
22 Paszke A, Gross S, Massa F, *et al*. PyTorch: An imperative style, high-performance deep learning library. Advances in Neural Information Processing Systems. Vancouver. 2019, 32: 1–12.

(校对责编: 孙君艳)