

# 基于 PP-PicoDet 技术的智能垃圾分类

倪吴广, 汪朵拉, 张卓

(河海大学 物联网工程学院, 江苏 常州 213002)

**摘要:** 目前在垃圾分类目标检测上大多采用 YOLOv5 系列算法, 该算法在相同参数数量的情况下, 检测精度和检测速度都对不太高, 难以满足实际应用需求; 对基于 PP-PicoDet 技术的垃圾分类目标检测应用进行了研究, 并将其与几种常见的垃圾分类目标检测算法进行实验对比分析; 结果表明, PP-PicoDet 算法能够在使用更少的参数数量的情况下, 实现较高的检测精度和速度, 能够满足移动端部署需求。

**关键词:** 垃圾分类; 深度学习; 目标检测; YOLOv5; PP-PicoDet 轻量级模型

## Intelligent Garbage Classification Based on PP-PicoDet Technology

NI Wuguang, WANG Duola, ZHANG Zhuo

(College of Internet of Things Engineering, Hohai University, Changzhou 213002, China)

**Abstract:** Currently, YOLOv5 series algorithm is often used for garbage classification target detection, the algorithm has the features of low detection accuracy and detection speed for same parameters, and difficult to meet practical application requirements. A garbage classification target detection application based on PP-PicoDet technology is researched, and this method is compared and analyzed with traditional garbage classification target detection algorithms by experiments; The results show that the PP-PicoDet algorithm can achieve higher detection accuracy and speed with less parameters, and it can meet the requirements of deployment on mobile devices.

**Keywords:** garbage classification; deep learning; target detection; YOLOv5; PP-PicoDet model

### 0 引言

不正确的垃圾分类不仅会增加垃圾处理的成本和工作量, 还会导致可回收资源的浪费, 影响资源的回收利用, 甚至带来环境污染等问题。如何引导人们正确地进行垃圾分类, 从源头上减轻垃圾分类的工作负担, 提高垃圾分类的智能化程度, 是解决垃圾分类问题的关键。

近年来, 随着深度学习技术的高速发展, 相关技术被应用到了垃圾分类领域中, 如文献 [1] 设计了一种基于视觉感知的智能扫地机器人, 通过 YOLOv2 目标检测算法, 引导扫地机器人对垃圾进行自动识别与按类处理; 文献 [2] 设计了一种基于深度学习的智能垃圾桶, 通过语音识别技术与图像识别技术, 帮助用户正确分类垃圾; 文献 [3] 设计了一种可回收垃圾的视觉检测系统, 采用 YOLOv5 目标检测算法, 检测和识别垃圾的类别。这些应用不仅提高了垃圾分类的智能化程度, 同时也表明深度学习技术在垃圾分类任务中是一种有效的方法<sup>[4]</sup>。

目前深度学习技术应用在垃圾分类领域, 主要有图像分类与目标检测技术。PP-PicoDet 是百度飞桨提出的一种轻量级目标检测技术<sup>[5]</sup>, 本文主要研究基于 PP-PicoDet 技术的智能垃圾分类算法。该目标检测算法模型相对于其他

深度学习网络, 不仅计算量小、延迟低、精度高, 同时支持多种硬件环境部署, 在垃圾智能分类中有着较强的实用性。

### 1 目标检测在垃圾分类上的研究现状

作为计算机视觉领域中的一个重要研究方向, 目标检测又叫目标提取, 就是将待处理图片或视频中所需要识别的对象有效地提取分割出来, 得出目标的类别、位置、匹配率等相关信息<sup>[6]</sup>。近年来, 随着计算机性能的不断提升, 深度学习技术高速发展, 新的目标检测算法层出不穷。按照完成目标检测任务的步骤划分, 主流的目标检测算法可以大致分为两个类别: Two-stage 与 One-stage<sup>[7]</sup>。

Two-stage 目标检测算法可以简单地理解为两步走算法: 首先需要找出候选区域, 然后对其进行调整分类。代表算法为 R-CNN 系列算法, 包括 Mask R-CNN、Fast R-CNN 等。

One-stage 目标检测算法则无需先生成候选区域, 而是直接在网络中提取相关特征来预测目标的类别和位置。代表算法为 YOLO 系列算法, 如: YOLOv4、YOLOv5、YOLOX、YOLOv6 等。

目前在垃圾目标检测实际应用中, 大多采用了 YOLO

收稿日期: 2023-04-14; 修回日期: 2023-05-20。

基金项目: 国家自然科学基金项目(12274113)。

作者简介: 倪吴广(2000-), 男, 大学本科。

通讯作者: 张卓(1978-), 女, 硕士, 高级实验师。

引用格式: 倪吴广, 汪朵拉, 张卓. 基于 PP-PicoDet 技术的智能垃圾分类[J]. 计算机测量与控制, 2023, 31(10): 291-298.

系列算法,如文献 [1] 设计了一种基于 YOLOv2 的智能扫地机器人;如文献 [8] 设计了一种基于 YOLOv3 的地面垃圾检测与清洁度评定方法;如文献 [9] 设计了基于 YOLOv5s 网络的垃圾分类和检测模型。

文献 [10] 对几种 YOLO 系列算法的数据对比,如表 1 所示。

表 1 几种 YOLO 系列算法的数据对比

Model	Size	mAP <sup>val</sup> (0.5:0.95)	mAP <sup>val</sup> (0.5)	Params	Latency (bs=1)
YOLOv5n	640	28.0	45.7	1.9 M	1.7 ms
YOLOv5s	640	37.4	56.8	7.2 M	2.7 ms
YOLOX-Tiny	416	32.8	50.3	5.1 M	1.4 ms
YOLOX-s	640	40.5	59.3	9.0 M	3.0 ms
YOLOv6-n	640	35.9	36.3	4.3 M	1.2 ms

表 1 中所有模型均在 COCO 2017 数据集上进行训练与测试,采用的 GPU 均为 Tesla T4,并且所有的模型都没有经过预训练,都在数据集上训练了 300 个 epoch。

从表 1 中可以看出,如果要满足较高的精度值,就需要较多的参数,从而使得模型的推理速度变慢。因此,模型在实际应用中往往精度可以达到需求,但推理速度较慢,难以实际应用。

文献 [11-12] 对人工智能在垃圾分类中实际的应用展开了相关研究,结果表明:大多数智能垃圾分类硬件设备,存在着技术研发阶段投入资金多,实际应用与后期维护时成本高等问题。因此,大多数居民区难以对这些智能化设备进行采购和使用,导致垃圾分类硬件设施的智能化程度较难在短时间内提升。

智能手机的普及,使得在移动端部署智能垃圾分类算法模型,拥有以下优点:

- 1) 技术研发成本相对较低,投入应用后便于管理;
- 2) 便于推广和宣传垃圾分类相关知识,提高居民垃圾分类的自主意识;
- 3) 便于溯源不文明投放垃圾行为,实施相关处罚;
- 4) 易于根据居民正确投放垃圾的数量给与居民相应的

奖励;

- 5) 可以在短时间内大面积使用,提升正确垃圾分类的效率。

从这些优点可以看出,推进移动端智能垃圾分类应用的部署有着重要的意义。

常见的 YOLOv5 系列算法在移动端实际部署时的推理结果帧率较低,较难满足用户对低时延的需求,所以在移动端部署方面仍面临较大困难。

在移动端部署目标检测相关模型,面临轻量化、速度快、精度高等需求。百度飞桨设计的轻量级目标检测算法 PP-PicoDet<sup>[5]</sup>,对模型的速度、精度、部署友好性进行了优化,可以较好地解决这些问题。百度飞桨在 Github 上给出的部分测试数据如表 2 所示。

表 2 中时延测试环境:英特尔酷睿 i7 10750H CPU (4 线程,FP16 预测,采用 OpenVINO)。PP-PicoDet 模型是在 COCO train2017 上训练,并且在 COCO val2017 上进行验证。

表 2 部分 PP-PicoDet 算法的测试数据

Model	Size	mAP <sup>val</sup> (0.5:0.95)	mAP <sup>val</sup> (0.5)	Params	Latency CPU/ ms
PP-PicoDet-s	320 * 320	29.1	43.4	1.18	4.8
PP-PicoDet-m	320 * 320	34.4	50.0	3.46	8.2
PP-PicoDet-m	416 * 416	37.5	53.4	3.46	12.7
PP-PicoDet-l	640 * 640	42.6	59.2	5.80	62.5

上述数据中,PP-PicoDet 在较少参数量的情况下可以实现较高精度和较高速度,基本可以满足在移动端部署的需求。

本文研究了基于 PP-PicoDet 模型的智能垃圾分类,并将其与常用的 YOLOv5 算法做一个详细的对比测试。

## 2 算法结构

### 2.1 YOLOv5 整体网络结构

如图 1 所示,以 YOLOv5s 为例,讲解 YOLOv5 的整体网络结构。YOLOv5 是在 YOLOv4<sup>[13]</sup> 提出后一个多月内就

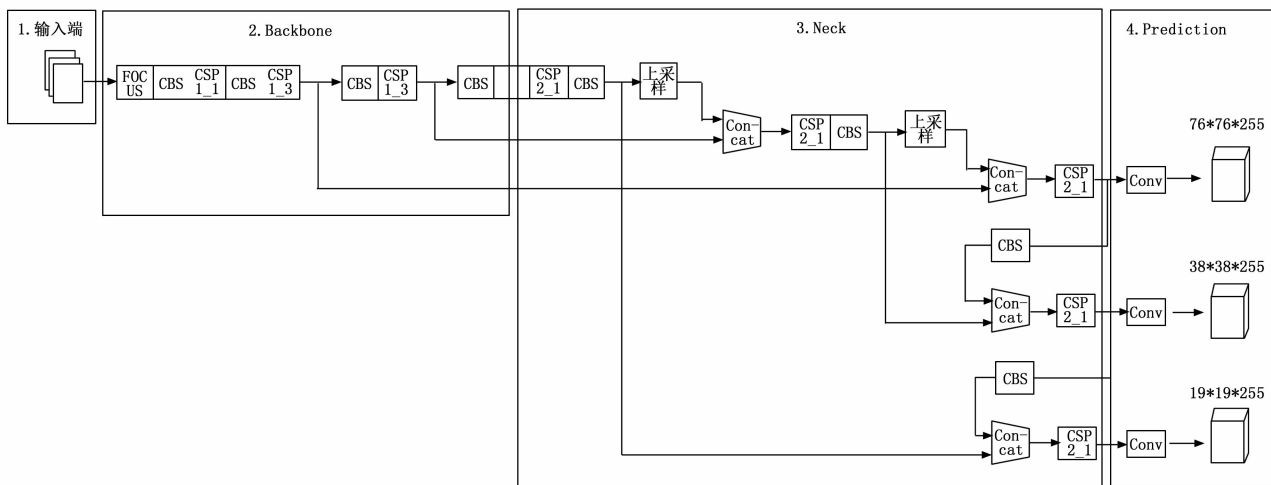


图 1 YOLOv5s 整体网络结构图

提出的, 所以模型的网络结构上与 YOLOv4 十分相似, 仍分为 4 个部分, 分别为 input、Backbone、Neck 和 Prediction。

### 2.1.1 Input

Input 端由 Mosaic 数据增强、自适应图片缩放、自适应锚框计算 3 部分组成。

将 4 张图片随机排布、随机裁剪、随机缩放进行拼接, 得到新的图片, 实现 Mosaic 数据增强。通过数据增强, 不仅可以对原有的数据集进行扩充, 还能提高对小目标的检测能力, 使得该模型的鲁棒性得到提高。

自适应图片缩放是指通过计算得出图片所需要添加最少的黑边。因为不同大小的图片统一到规定的尺寸后, 将会出现需要增加黑框的情况, 如果增加的黑边较多则会增加信息的冗余, 增加所需要的推理时间, 所以要通过自适应计算的方式以获得最少的黑边添加量。

自适应锚框是指在网络训练过程中通过计算输出的预测框与初始给定的锚框之间的距离, 进而反向更新初始锚框, 优化调整网络模型的参数。

### 2.1.2 Backbone

YOLOv5s 的 Backbone 部分由两个部分组成, 分别是 Focus 结构和 CSP 结构。首先是 Focus 结构, Focus 结构在 YOLOv5 中被首次提出, 其中最为重要的部分为切片操作, 即将特征图划分为不同区域, 进行分割, 然后将相同的区域重新拼接, 就可以得到新的特征图。然后是 CSP 结构, YOLOv5 在 Backbone 和 Neck 模块上分别设计了一种 CSP 结构, 应用在 Backbone 上的是 CSP1\_X 结构, 应用在 Neck 部分上的是 CSP2\_X 结构。

### 2.1.3 Neck

YOLOv5 刚出来时, Neck 部分仅有 FPN 结构<sup>[14]</sup>, 在后面又增加了 PAN 结构, 其中 FPN 采用的方式是自上而下的上采样以实现特征信息的传递融合, PAN 采用的方式是自下往上的下采样以形成特征金字塔。FPN+PAN 结构如图 2 所示。

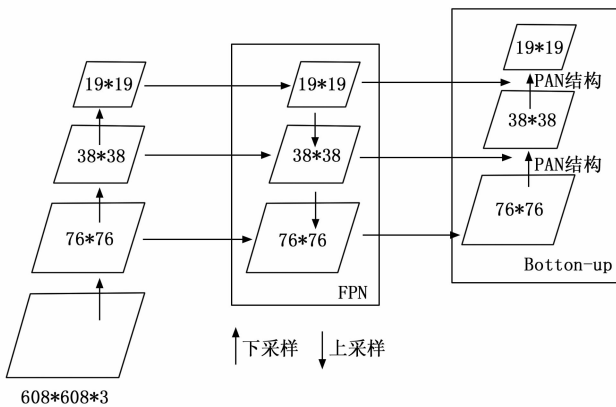


图 2 FPN+PAN 结构示意图

### 2.1.4 Prediction

Prediction 部分, 一般由非极大值的抑制操作 NMS 和

Bounding box 损失函数所构成。关于边界框无法重合的情况, YOLOv5 建议选用 GIOU\_Loss 作为损失函数的解决方案。NMS 操作实现了局部极大值的搜索, 与此同时抑制了非极大值元素, 从而消除冗余边界框, 保留最优目标框, 并检测出较多的目标。

## 2.2 PP-Picodet 整体网络结构图

如图 3 所示, PP-PicoDet 算法模型整体上可分为: 骨干网络模块、特征融合模块、网络输出模块。

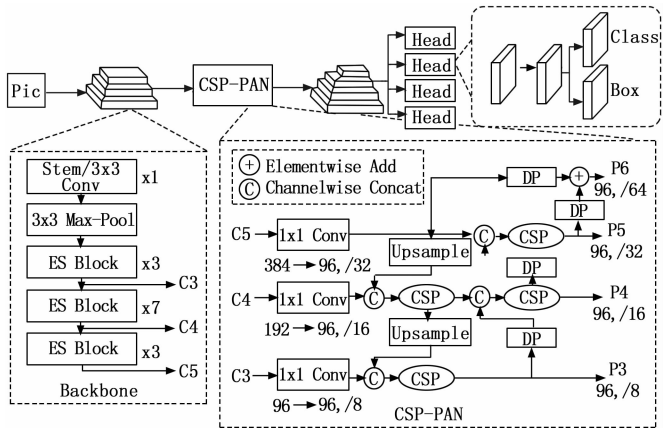


图 3 PP-Picodet 整体网络结构图

### 2.2.1 骨干网络模块

PP-PicoDet 的骨干网络模块, 是在原 SHuffleNetV2<sup>[15]</sup> 的基础上, 在 MobileNetV3 添加的 SE 功能块, 该模块内包含了 ReLU 和 H-Sigmoid 两层激活函数, 经过改进后所产生的一种对移动端友好的羽量级的 ESnet。

ESnet 的两个基本模块如图 4 所示, 由基础的深度可分离卷积 (pw conv, dw conv)、Ghost block、SE block 和 channel shuffle 模块组成<sup>[5]</sup>。

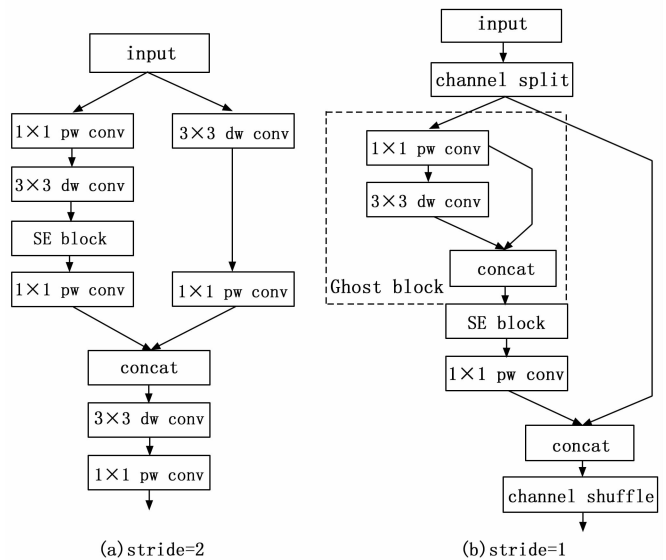


图 4 PP-Picodet 骨干网络结构图

### 2.2.2 特征融合模块

为从总体上大幅度减少神经网络的参量, 并增加可观

性能, PP-PicoDet 改变了一直被广泛应用的 CSP-PAN<sup>[5]</sup>, 进行了自上而下和自下而上二路特征的整合, 通过采用  $1 \times 1$  卷积的方法统一每层的宽度、增加特征尺度, 提高了在大目标情况下系统的检测能力。

### 2.2.3 网络输出模块

PP-PicoDet 将 CSP-PAN 层的输出进行两次 DW-PW 卷积, 再将结果进行分类。在标签分配时, 没有采用在全局训练过程中是不可改变的固定标签分配策略, 而是采用了随着训练过程不断变化的标签分配策略 SimOTA<sup>[16]</sup>。同时, 为了与 SimOTA 中的代价矩阵和目标函数一致, 采用了 Varifocal Loss (VFL)<sup>[17]</sup> 和 GIoU loss<sup>[18]</sup> 的加权和作为代价矩阵, 公式如下:

$$\text{cost} = \text{loss}_{\text{vfl}} + \lambda \cdot \text{loss}_{\text{giou}}$$

### 2.2.4 PP-PicoDet 轻量化网络的创新点与优势

1) 采用 CSP 结构构建 CSP-PAN 作为特征融合模块。CSP-PAN 将所有分支的输入通道统一为  $1 \times 1$  卷积, 显著提高了特征提取能力, 降低了网络参数。将  $3 \times 3$  深度可分离卷积扩大到  $3 \times 3$  深度可分离卷积来扩展感受野。

2) 采用 SimOTA 动态标签分配策略, 并对一些计算细节优化。用 Varifocal Loss (VFL) 和 GIoU loss 的加权和作为代价矩阵, 在不损害效率的情况下提高精度。

3) 进一步增强了网络结构, 并提出了一种新的主干网, 即增强 ShuffleNet (ESNet), 它的性能优于 ShuffleNetV2。

4) 改进了 NAS 模型, 可以自动查找最佳体系结构以进行对象检测, 实现了更好的效率和准确性的权衡。

## 3 实验设置与结果分析

### 3.1 实验设置

#### 3.1.1 实验数据

通过搜集网上各种垃圾分类公开数据集, 以及采用网络爬虫等技术爬取图片, 再结合相关程序与人工对获得的数据进行高质量清洗和筛选, 最终得到本次实验所用的数据集。该数据集共 14 964 张图片, 包含 44 种类别垃圾, 完全覆盖可回收垃圾、厨余垃圾、有害垃圾、其他垃圾这四种常见的垃圾分类类别。在数据集划分上, 采用 8: 1: 1 随机划分的方式, 即: 训练集 11 972 张, 测试集 1 496 张, 验证集 1 496 张。

数据集标注方法为: 先使用 LabelImg 工具对数据集图像进行目标位置及类别标注, 以 PASCAL VOC 格式保存为 XML 文件, 然后通过自主设计的相关程序将其转换成 COCO 数据集格式以及 YOLO 数据集格式。标注文件中包含了最小外接矩形框的长和宽、垃圾中心坐标及其所属类别等相关信息。

数据集中的部分内容如表 3 所示, 部分图片如图 5 所示。

#### 3.1.2 对比算法

YOLOv5 系列算法中, YOLOv5s 是目前被用在垃圾分

表 3 实验数据集部分内容展示

垃圾名称	分类名称	样本数量	标注数量
书籍纸张	可回收垃圾	232	284
旧衣服	可回收垃圾	165	227
易拉罐	可回收垃圾	569	670
饮料瓶	可回收垃圾	346	458
蛋壳	厨余垃圾	415	525
剩饭剩菜	厨余垃圾	978	1 237
干电池	有害垃圾	679	852
过期药物	有害垃圾	332	400
烟蒂	其他垃圾	735	834
大骨头	其他垃圾	441	463



图 5 部分数据图片展示

类方面较多的模型, 而 YOLOv5l 在目标检测任务中具有更高的精度和更强的鲁棒性, 因此本次实验选择 YOLOv5\_s\_640 和 YOLOv5\_l\_640 这两种具有代表性的 YOLOv5 系列算法与 PP-PicoDet 系列算法进行对比分析。具体的算法介绍如下:

1) YOLOv5\_s\_640: 其主要特点是网络结构比较轻量级, 适合在计算资源有限的情况下进行目标检测任务。该模型使用了一系列卷积层、池化层、上采样层等基本的神经网络组件, 并且采用了 PaddleYOLO 在 COCO train2017 训练集上训练得到的 YOLOv5\_s\_640 模型作为网络预训练模型, 能够高效地实现对图片或视频中的物体进行实时检测和识别。其中, 640 表示输入图片的大小为  $640 \times 640$  像素。

2) YOLOv5\_l\_640: 采用了 PaddleYOLO 在 COCO train2017 训练集上训练得到的 YOLOv5\_l\_640 模型作为网络预训练模型。和 YOLOv5\_s\_640 相比, YOLOv5\_l\_640 模型的网络结构更加深层, 拥有更多的卷积层和更大的感受野, 因此在目标检测任务中具有更高的精度和更强的鲁棒性。YOLOv5\_l\_640 需要更多的计算资源, 但是在一些对精度要求比较高的场景下, YOLOv5\_l\_640 能够取

得更好的检测效果。其中, 640 表示输入图片的大小为 640 640 像素。

3) PP-PicoDet\_s\_320: 该模型基于 anchor-free 的目标检测算法, 使用了十分轻量的基于网络预训练模型 PPLCNet\_x0\_75\_pretrained, 利用了一系列轻量级的卷积层和池化层等基本神经网络组件来实现目标检测和识别任务。其中, 320 表示输入图片的大小为 320 320 像素。与 YOLOv5\_s\_640 相比, PP-PicoDet\_s\_320 模型的网络结构更加轻量级, 但是在一些对计算资源要求比较苛刻的场景下, PP-PicoDet\_s\_320 能够取得更好的检测效果。

4) PP-PicoDet\_m\_320: 和 PP-PicoDet\_s\_320 相比, PP-PicoDet\_m\_320 模型网络结构更加深层, 并且拥有更多的卷积层和更大的感受野, 因此在目标检测任务中具有更高的精度和更强的鲁棒性。该模型同样基于 anchor-free 的目标检测算法, 网络预训练模型为 PaddleDetection 官方发布的预训练模型 PP-PicoDet\_m\_320\_coco\_lcnnet<sup>[19]</sup>, 采用了一系列卷积层、池化层等基本的神经网络组件来实现目标检测和识别任务。其中, 320 表示输入图片的大小为 320 320 像素。PP-PicoDet\_m\_320 相比 PP-PicoDet\_s\_320 需要更多的计算资源, 但是在一些对精度要求比较高的场景下, PP-PicoDet\_m\_320 能够取得更好的检测效果。

5) PP-PicoDet\_m\_416: 网络预训练模型为 PaddleDetection 官方发布的预训练模型 PP-PicoDet\_m\_416\_coco\_lcnnet<sup>[19]</sup>。和 PP-PicoDet\_m\_320 相比, PP-PicoDet\_m\_416 模型的输入图片大小更大, 为 416 416 像素。这意味着该模型能够检测到更多的细节和目标, 从而在一些对检测精度要求较高的任务中表现更好。PP-PicoDet\_m\_416 模型采用了一系列卷积层、池化层等基本的神经网络组件, 并且基于 anchor-free 的目标检测算法, 能够快速、准确地检测和识别图片或视频中的目标物体。相比较 PP-PicoDet\_m\_320, PP-PicoDet\_m\_416 模型需要更多的计算资源, 但是能够取得更好的检测效果。

6) PP-PicoDet\_l\_640: 使用了较大的基于网络预训练模型 PPLCNet\_x2\_0\_pretrained。相比于 PP-PicoDet\_m\_416, PP-PicoDet\_l\_640 模型的输入图片大小更大, 为 640 640 像素。这意味着该模型能够检测到更多的细节和目标, 从而在一些对检测精度要求较高的任务中表现更好。PP-PicoDet\_l\_640 模型采用了一系列卷积层、池化层等基本的神经网络组件, 并且基于 anchor-free 的目标检测算法, 能够快速、准确地检测和识别图片或视频中的目标物体。相比较 PP-PicoDet\_m\_416, PP-PicoDet\_l\_640 模型需要更多的计算资源, 但是能够取得更好的检测效果。PP-PicoDet\_l\_640 还拥有更深的网络结构和更大的感受野, 因此在一些复杂场景下, 会具有更强的鲁棒性和精度。

通过采用相同的数据集、训练环境、训练方式与评估指标, 比较以上这些算法, 最终筛选出最适合用于移动端部署垃圾目标检测的算法模型。

### 3.1.3 部署环境

实验训练的环境为远程连接云计算服务器搭建。服务器配置: CPU 处理器为 16 核 Intel (R) Xeon (R) Platinum 8350C CPU @ 2.60 GHz, 显卡为 NVIDIA RTX 3090, 显卡内存为 24 GB, 操作系统为 Ubuntu18.04, CUDA 版本为 11.2, Python 版本为 3.8, PaddlePaddle 版本为 2.2。

### 3.1.4 评估指标

本次算法实验使用训练过程中的  $loss$  和全部目标的  $mAP$  作为评估指标。

训练时  $loss$  值的计算公式如下:

$$loss = loss_{\text{bbox}} + loss_{\text{dfl}} + loss_{\text{vfl}}$$

$loss$  值计算公式中,  $loss_{\text{bbox}}$  为预测值与检测目标真实位置之间的损失, 其中  $loss_{\text{vfl}}$  为 Varifocal Loss,  $loss_{\text{dfl}}$  为 Focal Loss。

在对目标检测算法的性能进行评估过程中, 有如下相关概念: True Positive (TP) 是 IoU 超过预先给定的阈值的检测框数量 (同一个真实框只统计一次); False Positive (FP) 是 IoU 小于等于预先给定的阈值的检测框总量, 简单点说就是检测到对于同一个真实框的剩余检查框的总的数量; False Negative (FN) 是指检测过程中无法检查到的真实框的总量。于是有:

$$Precision = \frac{TP}{TP + FP} = \frac{TP}{\text{all detections}}$$

$$Recall = \frac{TP}{TP + FN} = \frac{TP}{\text{all ground truths}}$$

式中, all detections 和 all ground truths, 分别表示了全部预测框和全部真实框的总量。而  $PR$  曲线一般指的是用  $Precision$  和  $Recall$  绘制的曲线。在特定类别目标的  $PR$  曲线上, 该类别的  $AP$  值得计算公式如下:

$$AP = \sum_{i=1}^{n-1} (r_{i+1} - r_i) P_{\text{inter}}(r_i + 1)$$

上述计算公式中,  $r_1, \dots, r_n$  是按升序进行排列的  $Precision$  插值段的第一个插值处对应的  $Recall$  值, 在本次实验中, 我们所选取的是 0, 0.1,  $\dots$ , 1 这 44 个  $Recall$  值。各类别  $AP$  的平均数就是  $mAP$ :

$$mAP = \frac{\sum_{i=1}^k AP_i}{k}$$

式中,  $k$  为目标类别数<sup>[20]</sup>。

## 3.2 实验过程

将各模型依次在划分好的数据集上进行训练, 分析模型训练过程中相关参数的变化情况, 并综合比较各模型训练效果、导出模型的性能。

在模型训练策略方面, 所有算法都训练 100 个 epoch, 每 5 个 epoch 进行一次验证, 训练时小模型设置 batch\_size 为 32, 大模型设置 batch\_size 为 16, 验证时 batch\_size 为 8; 测试时 batch\_size 为 1; 初始 LearningRate 设置为 0.002, 在 steps 为 300 时达到最大值 0.08, 然后随着训练步数增加慢慢递减至趋于 0。训练完成后保存在验证集上表现最好的模型, 并在测试集上进行最终评估。

### 3.3 实验结果

#### 3.3.1 训练过程曲线图对比

各模型训练完成后的 mAP 曲线对比如图 6 所示。从图中可以看出，各个模型训练过程中，在验证集上的 mAP (0.5 : 0.95) 值变化曲线未出现明显的过拟合现象。

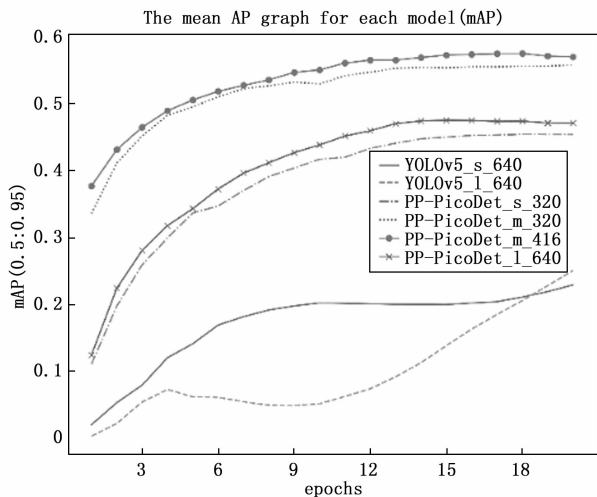


图 6 各模型 mAP 曲线对比图

总体上来看，在相同训练轮数的情况下，PP-PicoDet 系列算法的 mAP (0.5 : 0.95) 值远高于 YOLOv5 系列算法。其中，较小的模型 PP-PicoDet\_s\_320 和较大的模型 PP-PicoDet\_l\_640 的验证结果虽然相对较低，但也都接近 0.5，而 PP-PicoDet\_m\_320 和 PP-PicoDet\_m\_416 得出的结果大致相同，且表现较好。

各个算法模型在训练过程中的 loss 曲线如图 7 所示。

从图中可以看出，模型 PP-PicoDet\_s\_320 和模型 PP-PicoDet\_m\_320 训练时收敛速度相对较快，而较大的模型 PP-PicoDet\_l\_640 和 YOLOv5 系列模型在训练过程中收敛速度较慢，且最终计算出的 loss 值的效果较差。在所有模型中，PP-PicoDet\_m\_416 的 loss 值下降的最低，并且训练时间也相对较快，是综合表现较好的模型。

#### 3.3.2 模型评估效果对比

我们对上述各个算法在 100 个 epoch 训练过程中保存下来的最好的模型，在验证集上进行评估，评估结果如表 4 所示。

表 4 模型训练结果评估

Model	Training Time/h	mAP <sup>val</sup> (0.5:0.95)	mAP <sup>val</sup> (0.5)	FPS
YOLOv5_s_640	5	0.229	0.366	66
YOLOv5_l_640	9	0.251	0.362	38
PP-PicoDet_s_320	2.9	0.454	0.573	120
PP-PicoDet_m_320	3.2	0.557	0.661	126
PP-PicoDet_m_416	4.2	0.575	0.681	105
PP-PicoDet_l_640	10.4	0.475	0.578	43

从评估表格中可以看出，除了模型 PP-PicoDet\_l\_640

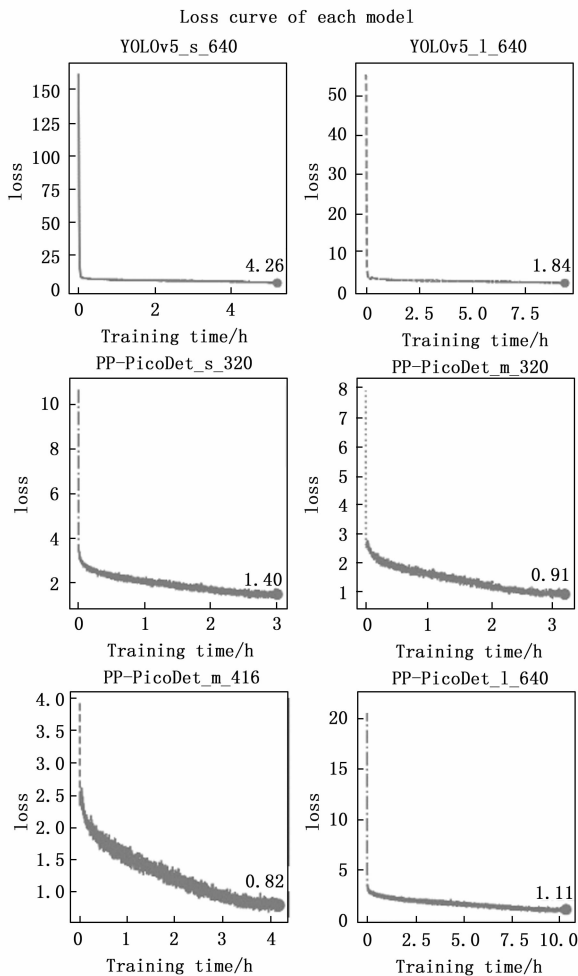


图 7 各模型 loss 曲线对比图

之外，其他模型的训练时间都较快，并且预测速度达到了 100 FPS 以上，相比于同类型的其他算法，其检测速度和精度均显著提升。相比之下，YOLOv5 系列算法在精度和速度上都处于明显的劣势。因此，可以看出在训练相同轮数的情况下，PP-PicoDet 系列算法可以更快地达到较高的精度，而且训练的速度较快，能够节约计算成本，更加适合移动端应用的部署。

#### 3.3.3 模型测试

与模型评估相同，在测试集上进行测试，测试结果如表 5 所示。

表 5 模型训练结果测试

Model	mAP <sup>val</sup> (0.5:0.95)	mAP <sup>val</sup> (0.5)	FPS
YOLOv5_s_640	0.246	0.374	65
YOLOv5_l_640	0.268	0.372	39
PP-PicoDet_s_320	0.492	0.606	129
PP-PicoDet_m_320	0.588	0.684	117
PP-PicoDet_m_416	0.604	0.705	111
PP-PicoDet_l_640	0.505	0.606	45

从测试表格中可以看出, YOLOv5 系列算法的推理速度较慢, 并且准确率也相对较低。相比之下, PP-PicoDet 系列各个模型的泛化性较好、准确率较高, 测试效果都和验证集上的验证效果相差不多。其中, 模型 PP-PicoDet\_m\_416 的 mAP (0.5; 0.95) 值和 mAP (0.5) 都到了 0.6 以上, 属于较高水平, 并且在检测速度上达到了 111 FPS, 在总体上和 PP-PicoDet\_m\_320 表现的都较好。

### 3.3.4 实验结果分析

经过实验验证, 可以看出相比于目前常见的用于垃圾分类目标检测的 YOLOv5 系列算法, 基于 PP-PicoDet 的垃圾分类目标检测算法能够在较少参数量的情况下实现较高的检测速度和检测精度, 并且减少了训练时间。总体上基本能够满足在移动端部署垃圾分类目标检测算法模型的需求, 对于提高垃圾分类目标检测的智能化程度有较大的帮助。

## 4 软件测试

### 4.1 软件系统设计框图

如图 8 所示, 软件系统主要由四大部分组成: MySQL 数据库、目标检测算法、Python Flask 框架、前端页面。其中, MySQL 数据库采用的是 MySQL8.0 版本, 目标检测算法由上述实验训练所得, Flask 框架是 Python 中一种常用的小型后端框架, 前端页面采用 HTML、CSS、JavaScript 三种网页设计基本语言进行设计, 并使用了开源的前端 UI 组件库 Layui。

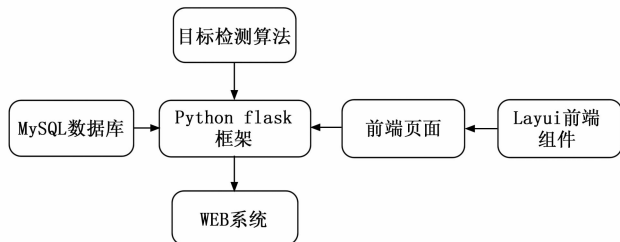


图 8 软件系统设计框图

### 4.2 前后端框架介绍

首先是前端框架, 由 HTML、CSS、JavaScript 三种网页基本语言、开源的前端 UI 组件库 Layui 组成。其中, HTML 语言用于创建网页的结构和内容, 是网页制作的基础。CSS 语言用于表示组件的样式, 通过 CSS 语言可以美化网页, 增加用户的体验感。JavaScript 语言为控制语言, 语法上富有逻辑性, 通过 JavaScript 语言可以在网页中添加控制行为, 增强网页的动态性和交互性。Layui 组件库是一个基于浏览器端原生态模式, 面向全层次的前后端开发者, 易上手且开源免费的 Web 界面组件库, 里面有许多常用的 Web 组件, 基本满足简单的 Web 开放需求。

其次是后端框架, 采用的是 Python 中的 Flask 框架, 一种轻巧方便的小型后端框架。它的 WSGI 工具箱选用的是 Werkzeug, 模板引擎选用的是 Jinja2。Werkzeug 库十分强大, 其功能相对完备, 不仅支持对 URL 路由请求的集

成, 而且还支持 Cookie、会话管理、交互式 Javascript 调试等服务。Jinja2 库不仅可以较好地防御外部黑客的脚本攻击, 还可以提高系统的运行速度。相较于其他同类型框架而言, Flask 框架能够较好地结合 MVC 开发模式, 更加轻便、快捷、安全且易于掌握。使用 Flask 框架能够在短时间内实现功能丰富的 Web 系统设计。

Flask 框架工作原理如图 9 所示。

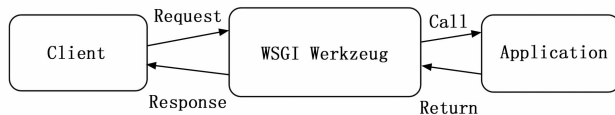


图 9 Flask 框架工作原理图

设计出的算法测试软件系统运行页面, 如图 10 所示。



图 10 软件系统运行页面

### 4.3 垃圾分类目标检测算法测试

通过选择 100 张图片对各个训练出的算法模型进行测试, 平均速度测试结果如表 6 所示。

表 6 模型训练结果测试

Model	Average detection time/ms
YOLOv5_s_640	421
YOLOv5_l_640	625
PP-PicoDet_s_320	52
PP-PicoDet_m_320	61
PP-PicoDet_m_416	83
PP-PicoDet_l_640	268

通过对软件系统的多次测试, 可以大致发现, 在仅使用 4 核 Intel (R) Core (TM) i5-10300H CPU @ 2.50 GHz 2.50 GHz 处理器的情况下, PP-PicoDet 系列算法的速度远高于 YOLOv5 系列, 最快的 PP-PicoDet\_s\_320 的速度几乎是 YOLOv5s 的 8 倍, 表现较好的 PP-PicoDet\_m\_416 也大约是 YOLOv5s 的 5 倍, 速度提升较为明显。测试结果表明, PP-PicoDet 系列目标检测算法在实际应用场景下, 具有较高的实用性, 有助于推动垃圾分类向智能化方向发展。

## 5 结束语

通过实验对比 PP-PicoDet 目标检测算法和 YOLOv5 系

列目标检测算法的性能，并测试 PP-PicoDet 目标检测算法在垃圾分类目标检测上的具体应用，根据实验结果可以看出基于 PP-PicoDet 技术的智能垃圾分类，相比于目前常用的同类型算法，具有以下优点：

- 1) 模型参数量小，训练速度较快，且泛化能力较强；
- 2) 在相同参数量情况下可以实现较高的检测精度和检测速度；
- 3) 具有较强的实际应用意义，能够助力提高我国垃圾分类智能化程度，改善垃圾处理效率。

通过人工智能技术，能够给人们带来一定的便利，但也会让人们逐渐淡化对垃圾分类的主动意识。如何建立健全垃圾分类智能化管理机制，对正确分类给予奖励，对错误投放能够做到溯源，并进行合理惩罚，还是需要不断地去研究和完善，这样才能够真正地解决垃圾处理难题<sup>[21]</sup>。

参考文献：

[1] 宁 凯, 张东波, 印 峰, 等. 基于视觉感知的智能扫地机器人的垃圾检测与分类 [J]. 中国图象图形学报, 2019, 24 (8): 1358 - 1368.

[2] 潘仲勋, 葛宇童, 董京伟, 等. 基于深度学习的智能垃圾桶设计 [J]. 科技与创新, 2022 (1): 66 - 68.

[3] 陈亮宇, 赵涵迪, 李浩正, 等. 可回收垃圾的视觉检测系统设计 [J]. 信息技术与信息化, 2022 (1): 23 - 25.

[4] 魏 晋. 双碳愿景下的生活垃圾分类数字化管理发展研究 [J]. 建设科技, 2021 (17): 53 - 57, 60.

[5] YU G, CHANG Q, LV W, et al. PP-PicoDet: A better real-time object detector on mobile devices [EB/OL]. (2021-11-10) [2022-11-10]. <https://arxiv.org/abs/2111.00902>.

[6] HE K, ZHANG X, REN S, et al. Spatial pyramid pooling in deep convolutional networks for visual recognition [C] //EC-CV. 2014: 12 - 23.

[7] 侯学良, 单腾飞, 薛靖国. 深度学习的目标检测典型算法及其应用现状分析 [J]. 国外电子测量技术, 2022, 41 (6): 165 - 174.

[8] 王一田, 唐开强, 留沧海, 等. 基于 YOLO v3 的地面垃圾检测与清洁度评定方法 [J]. 传感器与微系统, 2022, 41 (4): 129 - 133.

[9] 王 莉, 何牧天, 徐 硕, 等. 基于 YOLOv5s 网络的垃圾分类 (上接第 290 页)

[19] PAPALKAR R R, ALVI A S. Analysis of defense techniques for Ddos attack in Iot-review [J]. ECS Transactions, 2022, 107 (1): 3061 - 3068.

[20] JEESHITHA J, RAMAKOTESWARARAO G. Extensive study on DDoS BotNet attacks in multiple environments using deep learning and machine learning techniques [J]. ECS transactions, 2022, 107 (1): 15181 - 15193.

[21] 赵志强, 易秀双, 李 婕, 等. 基于 GR-AD-KNN 算法的 IPv6 网络 DoS 入侵检测技术研究 [J]. 计算机科学, 2021, 48 (S1): 524 - 528.

[22] 文伟平, 胡叶舟, 赵国梁, 等. 基于流量特征分类的异常 IP

类和检测 [J]. 包装工程, 2021, 42 (8): 50 - 56.

[10] LI C, LI L, JIANG H, et al. YOLOv6: A single-stage object detection framework for industrial applications [EB/OL]. (2022-09-07) [2022-11-15]. <https://arxiv.org/abs/2209.02976>.

[11] 徐 锐, 陈敏谊, 刘美红, 等. 人工智能在垃圾分类中的应用 [J]. 品牌与标准化, 2022 (1): 107 - 109.

[12] 宋淑苇, 杜欢政, 陆 莎. 阶梯式推进生活垃圾分类数字化改革的实践与思考 [J]. 资源再生, 2021 (11): 28 - 31.

[13] 李 庆, 龚远强, 张 玮, 等. 用于智能垃圾分拣的注意力 YOLOv4 算法 [J]. 计算机工程与应用, 2022, 58 (11): 260 - 268.

[14] LIN T Y, DOLLÁR P, GIRSHICK R, et al. Feature pyramid networks for object detection [C] //Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2017: 2117 - 2125.

[15] MA N, ZHANG X, ZHENG H T, et al. Shufflenet v2: Practical guidelines for efficient CNN architecture design [C] // Proceedings of the European Conference on Computer Vision (ECCV), 2018: 116 - 131.

[16] GE Z, LIU S T, WANG F, et al. YOLOX: exceeding YOLO series in 2021 [EB/OL]. (2021-08-06) [2022-12-13]. <https://arxiv.org/abs/2107.08430>.

[17] ZHANG H, WANG Y, DAYOUB F, et al. Varifocalnet: an IOU-aware dense object detector [C] //Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2021: 8514 - 8523.

[18] ZHENG Z, WANG P, LIU W, et al. Distance-IoU loss: faster and better learning for bounding box regression [C] // Proceedings of the AAAI Conference on Artificial Intelligence, 2020, 34 (7): 12993 - 13000.

[19] MA Y, YU D, WU T, et al. PaddlePaddle: An open-source deep learning platform from industrial practice [J]. Frontiers of Data and Computing, 2019, 1 (1): 105 - 115.

[20] 陈永祺, 顾 茜, 林 郁. 基于 PP-PicoDet 的半自动标注烟丝异物检测研究 [J]. 中国烟草学报, 2023, 29 (2): 11 - 21.

[21] 王泗通. 人工智能应用的社会风险及其治理——基于垃圾分类智能化实践的思考 [J]. 江苏社会科学, 2022 (5): 108 - 116.

识别系统的设计与实现 [J]. 信息安全, 2021, 21 (8): 1 - 9.

[23] 刘永定, 阳爱民, 邓 河. 基于机器学习的 P2P 流量分类系统设计及实现 [J]. 微计算机信息, 2009, 25 (33): 129 - 131.

[24] 贾万祥, 张平华. 基于多特征识别的非线性网络流量异常检测方法 [J]. 湖北科技学院学报, 2023, 43 (2): 145 - 150.

[25] 周家恺, 蔡方中. 网络流量时延特征数据的识别方法仿真 [J]. 计算机仿真, 2022, 39 (5): 398 - 401, 460.

[26] 卢宛芝, 丁要军. 基于半监督多视图特征协同训练的网络恶意流量识别方法 [J]. 通信技术, 2022, 55 (4): 513 - 518.

[27] 阳 洋, 王明森, 沈 为. 基于 DNS 和流量特征的业务识别系统设计 [J]. 工业控制计算机, 2017, 30 (7): 65 - 67.