

文章编号:1673-2022(2022)03-0012-05

基于云边端协同垃圾分类检测系统的设计和实现

周 原

(闽江师范高等专科学校 数字信息工程学院,福州 350018)

摘要:针对人工分拣垃圾成本过高影响垃圾分类推广的问题,将计算机视觉中的物体检测应用于垃圾分类领域,实现“云、边、端”协同的垃圾分类检测系统开发。从垃圾物体数据采集和预处理,到配置 YOLO v3 神经网络模型,并将模型部署在硬件开发板上,通过摄像头识别垃圾物体并完成垃圾分类检测任务,实现理论研究到生产实践落地的整个过程。

关键词:垃圾分类;物体检测;云边端协同;Caffe 框架;YOLO v3

中图分类号:TP391

文献标志码:A

0 引言

随着人们消费能力的日益增强和生活节奏的不断加快,消费物品种类和数量呈急剧上升趋势,产生了越来越多的生活垃圾和工业垃圾,城市生活的垃圾量、清理量不断加大,给城市垃圾的管理和处置带来了巨大的压力^[1]。垃圾分类在助力环保、资源回收以及降低人工分拣成本等方面起到了重要作用。但是垃圾分类的落地并不理想,错误投放垃圾,导致垃圾分类的难度增大,人力分拣垃圾成本高,影响了垃圾分类的全面推广。

随着计算机视觉和深度学习的发展,计算机系统已经能够高质量地对物体进行正确识别。将计算机视觉中目标检测技术应用于垃圾分类领域能够极大地解决垃圾分类成本高的问题。从垃圾生产端来看,垃圾分类系统能够帮助识别垃圾种类,指导垃圾的正确投放。从处理垃圾的工厂端来看,自动化垃圾识别分类能提高自动化水平,降低人工分拣成本。智能垃圾分类助力减少垃圾对环境的污染,将垃圾变废为宝。由此可见,智能垃圾分类的应用普及是必然趋势。

目前,智能垃圾分类的应用集成了计算机视

觉、深度学习和机器人技术等。来自加拿大的分类垃圾桶 Oscar,通过人工智能摄像头和显示屏,帮助识别不同垃圾应该投入哪个对应垃圾桶里。垃圾分拣系统 Zen Robotics Recycler(ZRR)机器人以平均每小时 3000 件的速度全自动分离各种垃圾,可以识别纸板、金属、木材、石膏、石块、混凝土等 20 余种可回收垃圾。FANUC 分拣机器人 Waste Robot 能够对物品的化学成分以及形状进行实时扫描和分析,利用视觉分析系统对物品进行跟踪和判断,驱动机械臂实时抓取物品进行分类。

基于深度学习和计算机视觉技术,将人工智能中的物体检测应用于垃圾分类领域,设计并实现基于云边端协同的垃圾检测系统,从垃圾数据采集和预处理,到配置神经网络模型,并将模型部署在硬件开发板上,实现理论研究到生产实践落地的整个过程。

1 总体结构设计

基于视觉的垃圾检测系统包含三部分:数据准备与预处理;模型训练与评估;模型转换与边缘部署。“云、边、端”协同开发流程如图 1 所示。

数据的准备和预处理包括:数据采集和标注,划分训练数据集和测试数据集,配置标签文件,将训练数据集转换为符合 Caffe 框架训练要求的 lmdb 格式的数据库文件。对模型进行训练和评估包括:环境和框架配置、Caffe solver 超参数设置、模型训练和模型评估。对模型进行转换和边缘部署主要是为了实现在边缘端的应用,通过摄像头采集垃圾物体数据,对采集数据预处理,进行模型推理得到结果,并将结果通过驱动 QT 界面反馈,将分类结果显示在平板的屏幕上。通过基于云边端协同垃圾分类检测系统实现智能垃圾分类,助力降低人工分拣成本。

收稿日期:2022-04-08

作者简介:周原(1990-),女,湖北荆门人,讲师,博士,主要研究领域为自然语言处理、物联网应用、软件形式化方法。

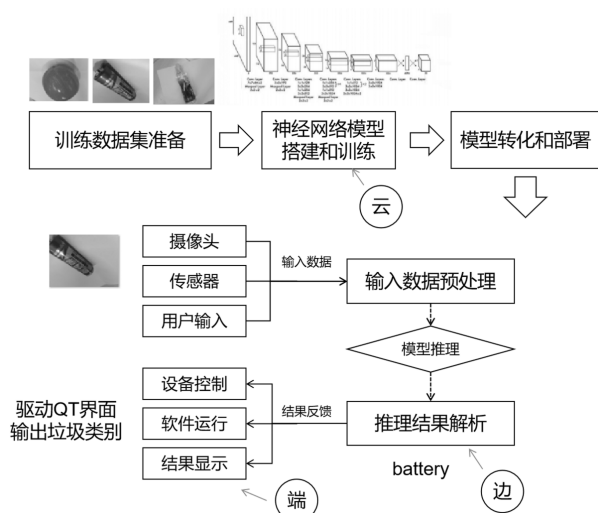


图1 “云、边、端”协同开发流程

2 数据准备与预处理

为了成功模仿或超越人类的视觉功能,计算机视觉在对目标设备进行研发的过程中,需要通过对大量模型训练,实现对图像的标注。图像标注是一个将标签添加到图像的过程。数据采集和数据标注是整个开发过程中工作量最大、最基础的一个环节。项目选用精灵标注工具,对1391个垃圾物体图片进行标注,从而得到相应的xml格式的垃圾物体标注文件。标注后的图片样例见图2。

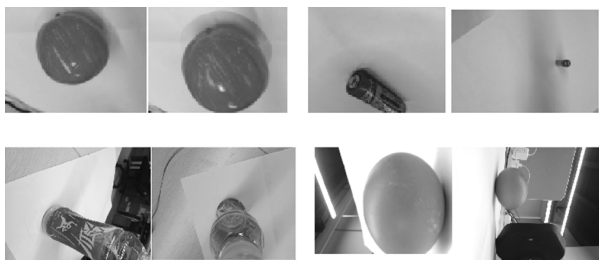


图2 自建数据集

模型训练过程是训练、测试不断迭代的过程,图片标注之后,按照设定比例(8:2)将标注的垃圾图片数据集随机划分为训练数据集和测试集,并生成图片和标注文件地址对应的训练集以及测试集关系清单的txt文件。

在训练模型之前,模型需要知道目标检测物体的类别标签。模型及其优化以纯文本的形式定义,而非代码模式,使用Caffe框架可以在配置中定义模型和优化,不需要硬编码,更方便高效^[2]。学术和工业领域,运算速度对模型和海量数据的处理至关重要,Caffe结合cuDNN,速度更快。Caffe工

作流程如图3所示。Caffe框架中的layer是模型和计算的基本单元,用以实现前向和反向传播。再通过blob结构来存储、交换和处理网络中间正向和反向迭代时的数据和导数信息,实现数据传递。solver定义针对net网络模型的求解方法,记录神经网络训练过程,保存神经网络模型参数。

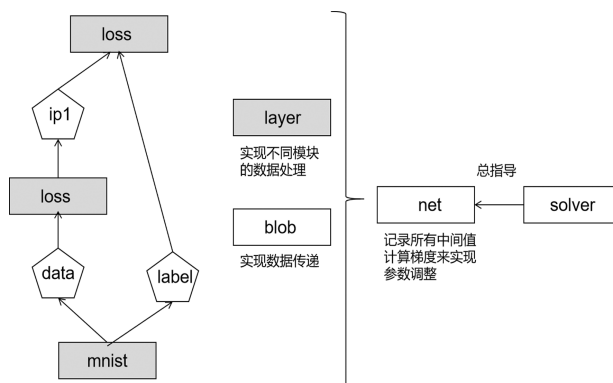


图3 Caffe 工作流程

在Caffe框架下,通过输入标签参数来自动生成模型所要使用到的标签配置文件和标签文件。其中,labelmap.prototxt是Caffe框架使用的标准配置文件,用以描述模型文件具有的标签清单,得到垃圾图片的标签及其格式如图4所示。

```

1 item {
2   name: "background"
3   label: 0
4   display_name: "background"
5 }
6 item {
7   name: "apple"
8   label: 1
9   display_name: "apple"
10 }
11 item {
12   name: "battery"
13   label: 2
14   display_name: "battery"
15 }
16 item {
17   name: "bottle"
18   label: 3
19   display_name: "bottle"
20 }
21 item {
22   name: "egg"
23   label: 4
24   display_name: "egg"
25 }

```

图4 Caffe 框架下标签清单格式

使用Caffe框架下自带的convert_annotset工具,将训练数据集转换成复合Caffe训练要求的lmdb格式数据文件,这是一种小型的键值对数据库^[3]。其中,每个键值对都是数据集中的样本。使用lmdb数据库来存放垃圾物体图像数据,是因为其具有极高的存储速度,在系统访问开发中使用大量图片、小文件时,用以减少磁盘IO时间开销。得到lmdb格式数据需要满足以下要求:(1)垃圾物体的图片地址与其对应标注文件的txt清单文件;(2)转换工具convert_annotset。使用转换工具

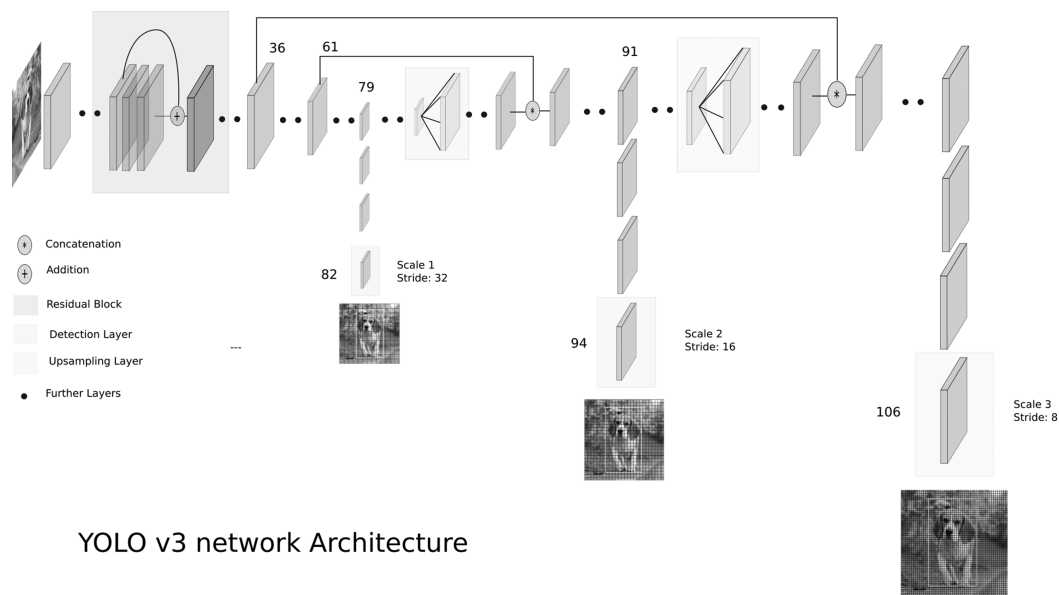
为普通图片数据源转换为 lmdb 数据库提供便捷方法,配置关键参数快速找到 lmdb 数据文件。执行完脚本后,根目录下出现 Caffe 训练数据文件和测试数据库文件。

3 模型训练与评估

神经网络训练过程包括模型训练和模型测试,虽然两者的模型主体架构完全相同,但其数据输入方式不同,且对应不同数据集,需要对数据层部分参数进行手动配置。本项目的神经网络模型是基于 YOLO v3^[4]的深度学习框架,优点是速度快,充分发挥多核处理器和 GPU 并行运算的功能。并且,在尺寸中等偏小的物体上有超高准确率,适合垃圾物体的识别。

YOLO v3 网络结构如图 5 所示。它只有卷积层,通过调节卷积步长控制输出特征图的尺寸,对于输入图片尺寸没有特别限制;它采用金字塔特

征图,借鉴 ResNet,将输入的特征图与输出特征图对应维度进行相加;同时也借鉴 DenseNet 将特征图按照通道维度直接进行拼接。YOLO 输出特征图解码是它的关键点之一。根据不同的输入尺寸,会得到不同大小的输出特征图。每个特征图的每个格子中,都配置有 3 个不同的先验框。检测框位置(4 维)、检测置信度(1 维)、类别(80 维)都在其中,共计 85 维。v3 沿用 v2 中关于先验框的技巧,使用 k-means 对数据集中的标签框进行聚类,得到类别中心点的 9 个框,作为先验框。有了先验框与输出特征图,就可以解码检测框。置信度在输出 85 维中占固定一维,由 sigmoid 函数解码,解码之后数值区间在 $[0,1]$ 中。COCO 数据集有 80 个类别,类别数在 85 维输出中占了 80 维,每一个维度代表一个类别的置信度,使用 sigmoid 函数替代了 v2 中的 softmax,取消了类别之间的互斥,可以使网络更加灵活。



YOLO v3 network Architecture

图5 YOLO v3 模型结构

Caffe 框架下,神经网络各功能层高度模块化,以“layer{”为标志形成块的形式堆叠,最后组成完整的神经网络。配置神经网络训练模型和测试模型文件之后,进行 solver 超参数配置。Caffe solver 定义整个模型如何运转,它的配置信息如下:(1)网络参数配置,包括 train_net 训练神经网络模型文件地址, test_net 测试神经网络模型文件地址, test_iter 网络测试迭代次数为图片总数量 N 除以批次大小, test_interval 测试间隔设置为 500, display 设置为 0;(2)学习率参数配置,包括 base_lr

基础学习率设置为 0.01, lr_policy 学习率调整策略设置为 multistep, max_iter 最大迭代次数设置为 100 000;(3)模型优化参数配置, type 迭代优化算法设置 SGD;(4)模型保存快照参数配置, snapshot 快照输出率设置为 5000。

调用 Caffe 框架自带命令 caffe train 执行模型训练,包括从头训练和基于预训练模型进行训练两种训练模式,得到不同的 log 日志文件。输出日志文件如图 6 所示,可以看到 no_obj 期望值越来越小, obj 置信度越来越接近 1, iou 预测出和实际


```

I0927 08:27:23.409741 284 sgd_solver.cpp:121] Iteration 600, lr = 0.001
I0927 08:27:23.794435 284 region_loss_layer.cpp:384] noobj: 0.00923472 obj: 0.624716 iou: 0.736336 cat: 0.958055 recall: 0.958333 recall
75: 0.53373 count: 8
I0927 08:27:23.799358 284 region_loss_layer.cpp:384] noobj: 0.00639577 obj: 0.503852 iou: 0.731104 cat: 0.965832 recall: 0.972222 recall
75: 0.490079 count: 8
I0927 08:27:25.798764 284 region_loss_layer.cpp:384] noobj: 0.00664614 obj: 0.635714 iou: 0.785602 cat: 0.970403 recall: 0.972222 recall
75: 0.720238 count: 8
I0927 08:27:25.810533 284 region_loss_layer.cpp:384] noobj: 0.0054146 obj: 0.592158 iou: 0.769333 cat: 0.968601 recall: 0.972222 recall
75: 0.646825 count: 8
I0927 08:27:27.639855 284 region_loss_layer.cpp:384] noobj: 0.00866301 obj: 0.585169 iou: 0.73213 cat: 0.973961 recall: 0.958333 recall
75: 0.527778 count: 8
I0927 08:27:27.645052 284 region_loss_layer.cpp:384] noobj: 0.00618069 obj: 0.426669 iou: 0.721397 cat: 0.970257 recall: 0.958333 recall

```

图6 训练输出结果

标注的交集相关,值越大效果越好,loss 参数随着训练次数增加而降低。

最后,选择合适的评估方法,迭代地对模型进行优化。将评价指标数据可视化能够快速且直观地判断模型训练情况,提取日志文件中的参数信息,处理 loss 参数信息和 iou 参数信息,如图 7 所

示。loss 图中的折线总体呈下降趋势,说明随着不断迭代训练次数的增加,损失在不断减少。由 iou 图可见,总体趋势向上,随着训练次数增大,预测框和物体标注框的重合程度正当上升,但值并不高,说明仍需继续训练达到模型收敛。

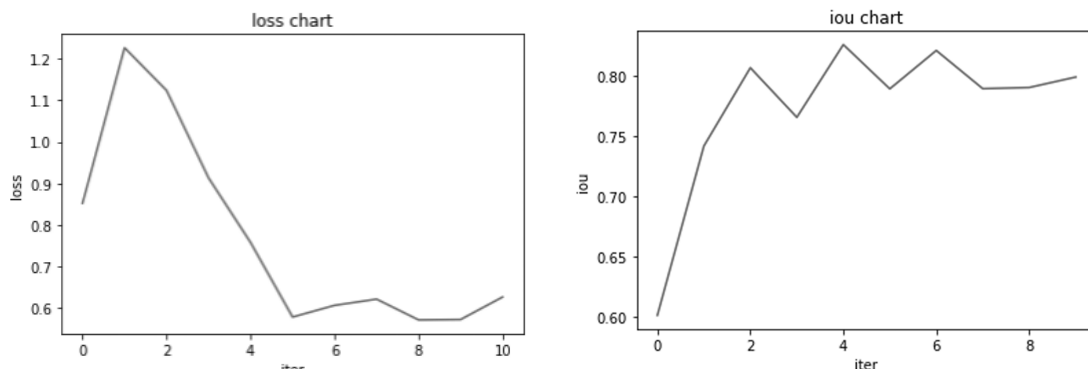


图7 模型评估图

4 模型转换与边缘部署

完成数据集划分,通过神经网络学习训练集中的数据得到模型参数,通过测试集来模拟测试模型的泛用性,这些都是基于一种数据源的图片文件进行的,且输出结果是对图片预测程度正确与否的系列评价指标。

边缘计算是一种分布式处理和存储的体系结构,是利用靠近数据源的边缘地带来完成运算程序,可在大中小型运算设备、本地端网络内完成^[5]。将训练好的神经网络模型部署到边缘端设备,虽然部署的神经网络模型主体结构 with 训练时的神经网络结构保持相同,但与训练和测试过程存在两大区别:(1)边缘端的神经网络模型需要处理摄像头采集到的未标注垃圾物体图片,所以神经网络模型的数据层不必再定义数据集的来源,但需要定义输入数据的大小格式;(2)边缘端的神经网络模型处理图片后,输出不再是对图片预测准确率

评价指标,而是对实时采集到的物体图片预测和分类的反馈结果。

ResNet-152 模型论文中提出了越宽越深越大的模型往往比越窄越小越浅的模型精度要高的证明^[6]。但越宽越深越大的模型对计算机资源要求更高,现在的模型应用越来越倾向于从云端部署到边缘侧,而受限于边缘侧设备的计算资源,需要考虑设备存储空间、内存大小、运行功耗和时延性等问题。特别是在移动端和嵌入设备等资源受限的边缘侧应用场景中,更需要模型量化,以较低的推理精度损失将连续取值的浮点型模型权重等数据定点近似为有限多个离散值的过程。模型量化增加了操作复杂度,在量化时需要做一些特殊处理。

制作边缘部署使用的神经网络模型文件,从获取训练时使用的神经网络模型文件中的所有层,删除没有 bottom 字段的用于数据输入的数据层,并设置新的数据层,修改输出层。在训练时一般需要输出 loss,但边缘部署对未知数据进行识

别,输出内容为检测垃圾物体的分类概率,需要 softmax 层,并将该输出层封装在 API 内。

对应开发使用到的硬件资源包括开发板、高清 USB 摄像头和触摸显示屏。开发板选用 NNIE,这是海思媒体 SoC 针对神经网络特别是深度学习卷积神经网络进行加速处理的硬件单元,芯片为 Hi3559A。使用该开发板之前,需要先配置转换工具 nnie_mapper 的配置文件。

使用 nnie_mapper 转换工具来实现 caffe model 转换为 wk 文件。针对开发板芯片选择对应的版本命令 nnie_mapper_11,执行转换命令生成 wk 文件,并下载标签文件,将它们置于 model 文件夹下,用于后续部署。电脑端与开发板连接,使用同一网段,设置配置 ip 地址。使用 MobaXterm 连接到开发板,将项目文件夹上传到开发板;运行项目文件,将目标物体放入摄像头下方,测试识别效果如图 8 所示。测试苹果置于摄像头下,识别出物体是苹果,且分类属于厨余垃圾。



图 8 垃圾分类识别效果

经过训练后的模型,在复杂背景下识别垃圾的能力大大提升,经测试准确率可达 90%,有一定的实用价值。

5 结语

本文研究了在垃圾分类中运用机器视觉和物联网应用技术实现云边端协同垃圾物体分类检测应用开发。通过对整个云边端协同开发流程进行梳理,从对垃圾物体数据进行采集、预处理,到配置神经网络模型,并在该模型上进行训练和评估,最终将模型分别部署在云端和边缘端,边缘端通过开发板控制摄像头实时采集物体数据,并将推理计算后的分类结果显示在触控屏上。通过训练

策略得到性能好的模型,并赋能到实际垃圾分类检测场景之中,发挥 AI 分类检测的能力。

参考文献:

- [1]葛鑫,蒋玲,陈慧蓉,等.基于物联网的垃圾回收站环境检测系统设计[J].湖北民族大学学报:自然科学版,2020,38(1):98-101.
- [2]许淑扬.深度学习框架 Caffe 在图像分类中的应用探析[J].信息与电脑:理论版,2019,31(23):95-96.
- [3]毕林,赵辉,贾明涛.面向数据库特征的基于 LMDB 与线性八叉树海量块段模型存储技术[J].中国有色金属学报:英文版,2016,26(9):2462-2468.
- [4]王铭杰.基于 YOLO V3 的垃圾自动定位及分类方法[J].无线互联科技,2019,16(20):110-112.
- [5]周国强,王宇健,周子涵.基于云边协同的新一代机载预测及健康管理架构设计[J].测控技术,2022(4):1-7.
- [6]He K,Zhang X,Ren S et al.Deep Residual Learning for Image Recognition[C].IEEE Conference on Computer Vision and Pattern Recognition (CVPR),2016:770-778.

Research and Implementation of Garbage Classification and Detection System Based on Cloud-Edge-End Collaboration

ZHOU Yuan

(School of Digital Information Engineering, Minjiang Teachers College, Fuzhou 350018, China)

Abstract: Aiming at solving the problem that the cost of manual sorting waste affects the promotion of waste classification, the object detection in computer vision is applied to the field of waste classification to realize the development of "cloud-edge-end" collaborative waste classification and detection system. From garbage object data collection and pre-processing, to configuring the Yolo-V3 neural network model, and deploying the model on the hardware development board, identifying garbage objects through the camera and completing the task of garbage classification and detection. The whole process from theory research to production practice is realized.

Key words: garbage classification; object detection; cloud-edge-end collaboration; Caffe framework; YOLO-v3