

基于深度学习的智能交通车流监测与预测研究

孙志娟, 李景景, 冯玉涛

(郑州工业应用技术学院信息工程学院, 河南 郑州 450064)

✉ sunzj123vip@sina.com; lijingjing1314@sina.com; fengyt@sina.com



摘要: 为了方便交通部门改善交通拥堵问题,使用旭日 X3 嵌入式开发板作为硬件平台,通过 YOLOv8 深度学习网络识别道路上通行的车辆及其车辆类型。使用开放神经网络交换 (Open Neural Network Exchange, ONNX) 格式可视化编辑工具去掉了模型的输出头,将网络中的激活函数由 SiLU 函数更换为 ReLU 函数,将模型输出由 80 个检测类别更改为 4 个检测类别,在 Small 版本中,使用非极大值抑制算法 (Non-Maximum Suppression, NMS) 将最合适的检测框筛选出来,然后用 SORT (Simple Online and Realtime Tracking) 多目标追踪算法和匹配算法将独立帧检测到的车辆关联起来,实现车辆自动计数。在服务器上配置好 YOLOv8 的训练环境,训练 3 个周期,测试模型的 mAP 指标为 0.635,推理速度提升至 20 fps 左右,目标检测系统的计数精度达到 98%,可以准确获取到路口的交通流数据,帮助改善交通拥堵问题。

关键词: YOLOv8 深度学习网络; NMS 算法; SORT 多目标追踪算法

中图分类号: TP311.1 **文献标志码:** A

Research on Intelligent Traffic Flow Monitoring and Prediction Based on Deep Learning

SUN Zhijuan, LI Jingjing, FENG Yutao

(School of Information Engineering, Zhengzhou University of Industrial Application Technology, Zhengzhou 450064, China)

✉ sunzj123vip@sina.com; lijingjing1314@sina.com; fengyt@sina.com

Abstract: In order to facilitate the transportation department to improve traffic congestion, this paper proposes to use the Horizon Sunrise X3 embedded development board as a hardware platform to identify vehicles and their models on the road through the YOLOv8 deep learning network. The Open Neural Network Exchange (ONNX) format visual editing tool is used to remove the output header of the model, replace the activation function SiLU in the network with ReLU function, and change the model output from 80 detection categories to 4. In version Small, Non-Maximum Suppression (NMS) algorithm is used to filter out the most suitable detection boxes. Then, the vehicles detected by independent frames are associated with the multi-target tracking algorithm and matching algorithm using SORT (Simple Online and Realtime Tracking) to realize automatic vehicle counting. The training environment of YOLOv8 is configured on the server with a training period of 3 cycles. With the mAP index of the test model being 0.635 and the reasoning speed increasing to about 20 fps, the counting accuracy of the target detection system reaches 98%, which can accurately obtain the traffic flow data at the intersection and help improve the traffic congestion problem.

Key words: YOLOv8 deep learning network; NMS algorithm; SORT multi-target tracking algorithm

0 引言 (Introduction)

随着城市化进程的加快,城市交通问题日益严峻。精确掌握交通车流量信息是开展城市交通管理的基础,获得准确且及时的车流量数据可以帮助交通管理部门科学调配交通资源^[1]。目前,城市道路已广泛布置监控摄像头,利用摄像头视频和图

像处理算法自动监测和统计车流量。使用这种方法采集的数据精度较高,并且能够实现全天候、全自动的监测。然而,图像处理算法计算量巨大,在边缘设备上实现实时车流量统计一直面临困难。本文设计利用“地平线”推出的旭日 X3 嵌入式开发板部署 YOLOv8 深度学习网络模型,实现了实时监测和统计交通车流量,对推动该领域的发展具有重要的意义。

1 系统整体设计方案(System overall design scheme)

本设计使用旭日 X3 嵌入式开发板作为核心硬件平台, Linux 作为操作系统。采用摄像头模块采集道路车流视频数据,通过 4G 无线模块和远程服务器实现通信。本设计使用 YOLOv8 目标检测器和多目标追踪算法 SORT 构建高精度的实时跟踪系统,该系统可以获取道路上行驶车辆的视频,并从视频中识别其中的车辆及其车辆类型,对识别到的车辆进行计数,然后将识别到的车辆信息通过 4G 通信模块实时上传到服务器。系统构成模块框图如图 1 所示。

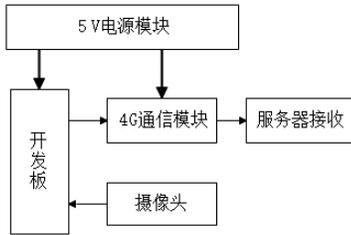


图 1 系统构成模块框图

Fig. 1 Block diagram of the system composition module

2 系统硬件设计(System hardware design)

本文设计的硬件系统由旭日 X3 嵌入式开发板、ATK-M750C 4G 无线通信模块、摄像头和 5V 开关电源组成。相机使用移动设备和嵌入式系统的摄像头串行接口协议 MIPI CSI 总线 and 开发板进行通信。

ATK-M750C 4G 无线通信模块使用串口与主机进行通信,在安装 Linux 操作系统的主机中,串口设备会被抽象为 `/dev/ttyS*` 设备文件。这种方式只需要对串口设备文件进行文件读写操作,就能实现对串口设备收发数据。在系统中,首先设置 ATK-750 模块和服务器通信的 IP 地址和端口号,其次将模块设置为 TCP(Transmission Control Protocol)穿透模式。连接开发板后,将开发板串口 3 的波特率设置为与模块的波特率一致,然后打开系统文件路径 `/dev/ttyS3` 的设备文件,发送数据时通过 `write` 系统调用就可以将数据传送到 ATK-M750 模块。系统配置了一个具有公网 IP(Internet Protocol)的服务器,用于接收系统发送的数据,并在服务器上运行了一个服务程序,该程序使用 TCP 协议通信,打开了本机的 7788 端口用于接收 4G 模块发送的数据。一旦和 4G 模块建立 TCP 连接,就循环接收 4G 模块发送的计数信息,然后把信息保存在本地文件中。系统的硬件电路原理图如图 2 所示。

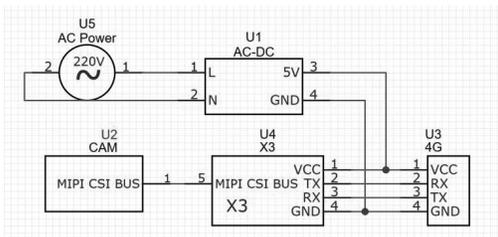


图 2 系统的硬件电路原理图

Fig. 2 The schematic diagram of the system's hardware circuit

3 目标识别系统设计(Design of target recognition system)

3.1 YOLOv8 深度学习网络模型微调

YOLOv8 官方发布的模型在 COCO2017 数据集上做了

300 轮预训练,可以识别日常生活中 80 种常见物体的类别,但本系统主要识别的对象是车辆类别,不需要具备识别其他物体的能力,因此需要对 YOLOv8 深度学习网络模型进行微调。

UA-DETRAC 数据集包含在中国北京市和天津市两个城市的 24 个不同的地点使用 Canon EOS 550D 相机拍摄的 10 h 的道路交通视频,约有 14 万帧图像数据,手动注释了 8 250 辆车辆,总共有 121 万个标记的对象边界框,本研究要在该数据集上训练 YOLOv8 深度学习网络模型对车辆位置和车辆类别的检测能力,因此需要对模型做两项微调,一是需要将网络中的激活函数由 SiLU 函数更换为 ReLU 函数,然后将 UA-DETRAC 的数据标注转换成符合 YOLOv8 规定的格式。二是将模型输出由 80 个检测类别更改为 4 个检测类别。在服务器上配置 YOLOv8 的训练环境,训练 3 个周期的模型,模型的 mAP 指标达到 0.6 以上,计算速度快,精确度高。

3.2 YOLOv8 深度学习网络模型在地平线旭日 X3 上部署

将训练好的深度学习网络模型部署到终端的流程通常包括导出 ONNX 模型、模型量化、模型推理 3 个步骤。ONNX (Open Neural Network Exchange) 是一种表示神经网络模型的开放格式,模型部署工作中的第一步通常是将不同的神经网络框架训练好的模型转换成 ONNX 通用的表达形式^[2]。在本文的设计中,调用 YOLOv8 提供的模型导出脚本,可以将训练得到的 PyTorch 格式的模型转换为 ONNX 模型。模型直接导出的 ONNX 模型数据都是使用单精度浮点数进行计算,运算速度慢,因此需要把单精度浮点数转换为 8 位的整数进行计算,该过程称为模型量化^[3]。在旭日 X3 嵌入式开发板上,还需要将 ONNX 模型转换为“地平线”产品专用的模型,用于调用专用计算单元 BPU 加速计算。在“地平线”的 AI 工具链中,这一步转换过程和模型量化是同时进行的。

3.3 提升模型推理速度

未经过任何修改的 YOLOv8 官方模型,其 Small 版本经量化后,在旭日 X3 嵌入式开发板上部署之后,模型的推理速度仅为每秒 3~4 帧,远远没有达到实时检测的效果,所以必须对其推理性能进行优化,提高其推理速度。

在本文的设计中,对模型一共做了 3 处修改。首先通过观察模型发现,其检测头涉及大量的广播运算,并导致了大量的量化/反量化操作,造成无谓的计算资源损耗,为了解决此问题,系统使用 ONNX-Modify 工具去掉模型的输出头。

其次将模型中所有激活函数更换为 ReLU 函数^[4]。YOLOv8 深度学习网络模型中原有的激活函数为 SiLU 函数, SiLU 函数的表达式为 $SiLU(x) = x / (1 + e^{-x})$,网络训练过程中, SiLU 函数因为导数连续,收敛速度比 ReLU 函数更快,在同样数量的训练周期,模型的性能通常也会比使用 ReLU 的版本好。但是,在模型部署中可以以牺牲一部分模型准确度为代价来提高模型的推理速度。

再次将模型的输入图像分辨率从 640×640 修改为 512×512 。YOLOv8 深度学习网络模型中的主要运算为卷积运算,其计算复杂度与图像的长宽之积成正比,所以降低图像的分辨率能够非常有效地降低模型的推理时间。

3.4 非极大值抑制算法设计

YOLOv8 深度学习网络模型使用 anchor box 的方式进行目标检测,对同一个目标会有多个 anchor box 进行预测,那么

这么多个预测结果可能会重叠,在 Small 版本中,输入图像的分辨率为 512×512 时, YOLOv8 深度学习网络模型将会输出约 20 000 个预测框。为了将最合适的检测框筛选出来,可以使用 NMS 算法^[5]。

NMS 算法的主要步骤如下:

第一步是将置信度过低的框过滤掉。通过选取不同的过滤阈值,可以过滤几十到几百个框。第二步将输出框按置信度从高到低排序,从置信度最高的框开始,后边每一个框都和该框计算其交并比(IoU)值。IoU 值越大,表示两个框的重合度越高,将大于 IoU 阈值的框标记为弃用;若 IoU 值小于阈值,认为该框和第一个框不是同一个物体,则保留该框。再循环以上步骤,从保留下来的置信度第二高的框开始,重复第二步,直到所有框都计算完毕。循环步骤结束后仍然保留下的框,就是经过 NMS 算法之后的输出框。

4 计数算法设计(Counting algorithm design)

4.1 SORT 多目标追踪算法设计

YOLOv8 是一种单帧目标检测算法,它会独立地检测每一帧图像,但是检测并提取车辆图像后,相邻两帧图像中属于同一车辆的检测框还不具备联系。要想实现系统对检测到的车辆进行自动计数,就要设计一种方法能够将独立帧中检测到的物体关联起来,此时可以采用 SORT 多目标追踪算法^[6]。

SORT 多目标追踪算法是基于 Kalman 滤波和匈牙利算法进行跟踪^[7]。SORT 多目标追踪算法的流程如下:首先对每一帧图像使用目标检测算法(如 YOLO)生成目标候选框,其次计算各个框与现有追踪框(初始化帧设为检测框)的 IoU 值。得到当前帧检测目标和追踪目标的 IoU 矩阵,使用线性分配算法将当前帧中的检测框和追踪框关联起来。未匹配到的追踪框将被标记为失效,匹配到的检测框将被添加到追踪列表中。最后使用 Kalman 滤波更新匹配上的追踪框的状态。SORT 多目标追踪算法工作流程如图 3 所示。

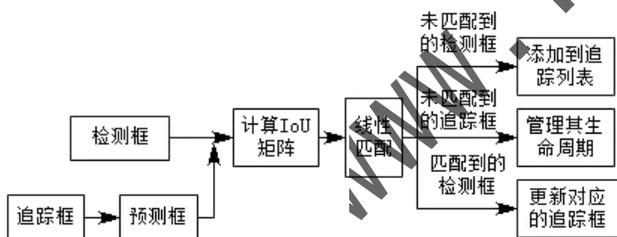


图 3 SORT 多目标追踪算法工作流程

Fig. 3 SORT multi-target tracking algorithm workflow

当获得当前帧的目标检测结果后,就可以开始进入目标追踪阶段。SORT 多目标追踪算法首先预测追踪列表中所有追踪框在当前帧的位置,其次计算预测框和当前帧的检测框的 IoU 值,获取 IoU 矩阵。将 IoU 矩阵取负值,或使用 1 减去矩阵中的所有元素,得到的新的矩阵将作为 $\cos t$ 矩阵输入线性匹配算法。线性匹配算法会给出匹配结果。得到匹配结果后,对于匹配到的追踪框,使用其对应的检测框更新追踪框的位置;对于未匹配到的追踪框,根据算法配置参数决定是否将其从追踪列表中移除;对于未匹配到的检测框,将其添加到追踪列表中。该算法的核心操作是卡尔曼滤波算法和线性匹配算法。

4.2 卡尔曼滤波算法设计

卡尔曼滤波是一种利用线性系统的状态方程,结合系统的观测值,对系统状态做出最优估计的算法。卡尔曼滤波假设系统是带噪声的线性系统,并假设系统的过程噪声和观测噪声都服从高斯分布,因此系统的状态方程可以写成公式(1)^[8]:

$$\begin{cases} X_k = AX_{k-1} + BU_{k-1} + w_k \\ Z_k = HX_k + v_k \end{cases} \quad (1)$$

卡尔曼滤波的工作流程分为以下 3 个步骤。

第一步是对系统的初值做初始化:

$$\begin{cases} X_0^+ = E[X_0] \\ P_0^+ = E[(X_0 - \hat{X}_0)(X_0 - \hat{X}_0)^T] \end{cases} \quad (2)$$

第二步是利用系统方程对系统做出先验估计:

$$\begin{cases} \hat{X}_k^- = A\hat{X}_{k-1}^- + BU_{k-1} \\ P_k^- = AP_{k-1}^- A^T + Q \\ K_k = P_k^- H^T (HP_k^- H^T + R)^{-1} \end{cases} \quad (3)$$

第三步是结合系统状态的观测值对先验估计值进行校正:

$$\begin{cases} \hat{X}_k^+ = \hat{X}_k^- + K_k(Z_k - H\hat{X}_k^-) \\ P_k^+ = (1 - K_k H)P_k^- (1 - K_k H)^T + K_k R K_k^T \end{cases} \quad (4)$$

4.3 线性匹配算法设计

线性匹配算法用于确定追踪框和当前帧的检测框的对应关系,可以计算所有追踪框和检测框的 IoU 值[这一步的计算复杂度为 $O(m \times n)$, 假设有 m 个追踪框和 n 个检测框],然后把两个框之间的 IoU 值当作其关联权重,从而把问题转化为最小权值匹配问题,即在二分图中求出一组点对的匹配,使得这组点对的权值和最小。

在本文设计的系统中,首先根据追踪框得到当前帧的预测框,其次计算预测框和当前帧检测框的 IoU 值,得到 IoU 值的矩阵,调用 scipy(Python 中一个用于科学计算的库)中的 linear_sum_assignment 函数求解匹配结果。

4.4 计数算法设计

当系统追踪到一个新的目标时,给该目标赋一个 ID 用于标识该目标,赋值 ID 的方法如下:对于第 1 个目标,对其赋值 ID 为 1;以后每追踪到一个新的目标,就把上一个目标的 ID 值加 1 作为新目标的 ID。通过这样的 ID 赋值方法,可以实现对车辆的计数。对于多种车型,对每一种车型都建立一个追踪器,从而实现对车辆的分类追踪。为了提升计数的准确度,对系统的算法做了如下改动。

(1)为了应对检测器检测质量差和经常出现未检测出物体类型的情况,系统延长了将追踪物体移出追踪列表的判定时间,从丢失 2 帧就将其移出改为丢失 15 帧再移出。此项改动提升了追踪的稳定性。

(2)对于远处的车辆,其在图像中对应一个很小的目标,检测器给出的检测结果准确度较差,可能导致系统出现检测不稳定、类别分类错误等问题。通过忽略检测器给出的检测框面积小于一定阈值的检测框,可以提升追踪的稳定性,从而提升计数精度。

5 系统调试(System debugging)

5.1 目标检测系统调试

目标检测系统主要关注检测精度和检测速度两个参数,当对 YOLOv8 深度学习网络模型进行 3 轮微调训练之后,测试

模型的 mAP 指标为 0.635。经微调后的模型针对小型轿车、卡车、公交车及其他车辆的测试结果如表 1 所示。

表 1 经微调后的模型针对小型轿车、卡车、公交车及其他车辆的测试结果

Tab.1 The test results of the fine-tuned model for small cars, trucks, buses, and other vehicles

类别	图片数/个	目标数/个	P	R	mAP50	mAP50-95
小型轿车	56 340	548 555	0.642	0.805	0.791	0.562
卡车	56 340	38 519	0.496	0.595	0.561	0.419
公交车	56 340	71 785	0.698	0.850	0.811	0.569
其他车辆	56 340	16 915	0.687	0.226	0.379	0.244

直接使用 ONNX 模型在 CPU 上推理,获得的推理速度约为 3.5 fps,经几次改进,推理速度提升为 20 fps 左右。最终经过量化的模型在旭日 X3 嵌入式开发板上的运行结果如图 4 所示。微调后的 YOLOv8 深度学习网络模型在数据集上表现良好,具有较高的检测精度,可以在实际场景中应用。



图 4 目标检测识别结果

Fig. 4 Target detection and recognition results

5.2 计数系统调试

计数系统的核心是计数精度,在测试场景下统计系统的计数精度,结果为 98%。其中,计数结果可视化图如图 5 所示,系统实物图如图 6 所示。



图 5 计数结果可视化图

Fig. 5 Visualization of counting results

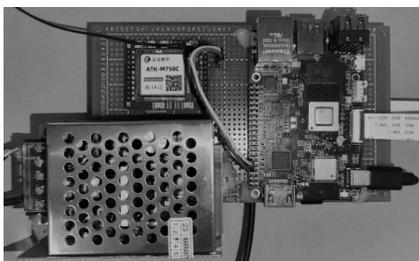


图 6 系统实物图

Fig. 6 System physical drawing

5.3 系统运行结果

系统接收到的数据格式为时间戳、小型轿车的计数、卡车的计数、公交车的计数和其他车辆的计数,系统运行结果图如图 7 所示。

```

root@www:~# python3 server.py
接收到的数据为: Sat May 6 18:19:30 2023 car:0, van:0, bus:0, others:0
接收到的数据为: Sat May 6 18:19:30 2023 car:1, van:0, bus:0, others:0
接收到的数据为: Sat May 6 18:19:30 2023 car:1, van:1, bus:0, others:0
接收到的数据为: Sat May 6 18:19:30 2023 car:2, van:1, bus:0, others:0
接收到的数据为: Sat May 6 18:19:30 2023 car:2, van:1, bus:0, others:0

```

图 7 系统运行结果图

Fig. 7 Diagram of system operation results

6 结论(Conclusion)

本文设计的系统使用旭日 X3 作为硬件平台,结合高清摄像头捕获图像的信息,在服务器上部署了 YOLOv8 深度学习网络模型,详细介绍了该模型如何进行微调,以适应具体的应用场景。使用 NMS 算法和 SORT 多目标追踪算法对模型输出进行筛选和目标追踪,并对模型进行微调训练之后,目标检测系统在识别速度和识别精度上均达到系统设计的要求,可以准确获取到路口的交通流数据。最后将本地的检测信息通过 4G 无线通信模块实时传输到服务器,实现数据实时上报。

参考文献(References)

- [1] 曹长玉. 基于深度学习的车流监测与预测算法研究[D]. 厦门:集美大学,2020.
- [2] 贺凯,田聪,顾斌. TensorFlow 与 ONNX 模型转换研究与实现[D]. 西安:西安电子科技大学,2022.
- [3] 过铭涛. 基于改进 YOLOv3 的目标检测模型研究与应用[D]. 南京:南京邮电大学,2022.
- [4] 刘磊. 基于 YOLO 网络的智能交通车流量统计研究[D]. 西安:西安科技大学,2019.
- [5] DING J J, ZHANG J H, ZHAN Z Q, et al. A precision efficient method for collapsed building detection in post-earthquake UAV images based on the improved NMS algorithm and faster R-CNN[J]. Remote sensing, 2022, 14(3):663.
- [6] 李广. 基于视频图像的车流量的检测研究[D]. 鞍山:辽宁科技大学,2022.
- [7] KAVITHA N, CHANDRAPPA D N. Performance analysis towards GUI-based vehicle detection and tracking using YOLOv3 and SORT algorithm[J]. International journal of intelligent systems technologies and applications, 2022, 20(5):436-451.
- [8] 刘剑,龚志恒,林璐瑶,等. 基于局部图块目标匹配的交通车流跟踪与统计[J]. 控制工程,2014,21(3):436-440,445.

作者简介:

孙志娟(1993-),女,硕士,助教。研究领域:信息技术,电子信息。

李景景(1991-),女,硕士,讲师。研究领域:信息技术,电子信息。

冯玉涛(2001-),男,本科,助教。研究领域:信息技术,电子信息。