Computer Science and Application 计算机科学与应用, 2022, 12(4), 847-857
Published Online April 2022 in Hans. http://www.hanspub.org/journal/csa
https://doi.org/10.12677/csa.2022.124086

Hans汉斯

基于TensorRT加速推理的电梯内电动车检测系统

高胥智[1]，魏伟波[1]，王　静[2]

[1]青岛大学，计算机科学技术学院，山东　青岛
[2]青岛市崂山区育才学校，山东　青岛

收稿日期：2022年3月6日；录用日期：2022年4月5日；发布日期：2022年4月12日

摘　要

电动车入户充电存在安全隐患，容易发生火灾，针对高层住户电动车入户充电的问题，本文提出了基于TensorRT加速推理的电梯内电动车检测系统，通过在电梯内部署电动车检测设备来有效遏制高层住户电动车入户的问题。该系统使用YOLOX网络训练目标检测模型，数据类型分为电动摩托车、电动自行车与自行车三类，通过模型迁移将训练好的目标检测模型部署到JetsonNano设备上，该设备通过Jetson-GPIO来做到对电梯的控制。实验结果表明，基于TensorRT加速推理的电梯内电动车检测系统，在性能与识别准确率上均优于传统方法，该设备能够有效遏制高层住户电动车入户问题。

关键词

YOLOX，TensorRT，JetsonNano，目标检测，深度学习

Electric Two-Wheelers Detection System in Elevator Based on TensorRT Accelerated Inference

Xuzhi Gao[1], Weibo Wei[1], Jing Wang[2]

[1]College of Computer Science and Technology, Qingdao University, Qingdao Shandong
[2]Qingdao Laoshan Yucai School, Qingdao Shandong

Received: Mar. 6th, 2022; accepted: Apr. 5th, 2022; published: Apr. 12th, 2022

Abstract

Electric two-wheelers charging at home has potential safety hazards and is prone to fires. In order

文章引用: 高胥智, 魏伟波, 王静. 基于 TensorRT 加速推理的电梯内电动车检测系统[J]. 计算机科学与应用, 2022, 12(4): 847-857. DOI: 10.12677/csa.2022.124086

to solve the problem of charging electric two-wheelers at home by high-rise residents, this paper proposes an electric two-wheelers detection system in elevators based on TensorRT accelerated inference. By deploying electric two-wheelers detection equipment in the elevator, this situation can be effectively contained. The system uses YOLOX to train the target detection model to distinguish electric motorcycles, electric bicycles and bicycles, and then deploys the trained target detection model to JetsonNano devices through model of migration. The device uses Jetson-GPIO to control the elevator. Experimental results show that the electric two-wheelers target detection system in elevators based on TensorRT accelerated inference is superior to traditional methods in performance and accuracy. The equipment can effectively curb the problem of electric two-wheelers entering the homes of high-rise residents.

Keywords

YOLOX, TensorRT, JetsonNano, Object Detection, Deep Learning

1. 引言

近年来，城市拥堵愈发严重，给人们出行通勤都带来了不便，而电动车出行更加方便环保，价格低廉，符合一直以来大众所倡导的绿色生活理念。随着社会对电动车需求的不断增加，国内电动车保有量持续增长[1]，但随着电动车数量的增加，充电成为了新的问题。根据应急救援部公布的数据，自 2021 年 1 月至 2021 年 7 月，我国共发生 6462 起电动车引发的燃烧事故，而其中多数发生在其充电过程中，因此，如果事故发生时，电动车放置在室内、楼道等相对封闭的环境中，就容易引发重大伤亡事故[2] [3]。目前，市面上针对电动车入户问题较为成熟的解决方案，即通过在电梯轿厢内部署电动车检测系统[4]来阻止电动车入户。市面上现有的产品中，选用的目标检测模型大多基于 SSDMobileNet 或 YOLOv3 进行训练[5]，使用树莓派或第三方定制的硬件设备进行部署。SSDMobileNet 虽然解决了小型开发设备性能低的问题，但其即便是进行了优化，训练后的模型在识别率上仍不及 YOLO-tiny 系列。数据采集上，由于采用专用数据集和开源数据集进行训练，没有考虑到倒车及遮挡等问题，实际应用中的产品达不到其官方宣称的识别率标准。此外，由于硬件设备的问题，训练后的模型仅做了简单的部署应用，本质上仍在特定的深度学习框架下运行，并没有进行高性能推理。本文从数据集的选取入手，使用 YOLOX-Nano 训练模型，用支持 TensorRT 加速推理的 JetsonNano 设备。经实验，本文的系统设备能够更高效的识别各角度及各种遮盖情形下的电动车，通过该设备控制电梯，限制电动车进入，从而解决高层住户电动车入户充电的问题。

2. 相关技术

2.1. YOLOX 目标检测网络

YOLOX 目标检测网络于 2021 年由旷视科技发布。其 BaseDetection 团队总结近两年来目标检测领域各角度的优秀进展，将 YOLO 系列检测器调整为了 Anchor-Free 的形式并集成了如解耦头、标签分配、数据增广等技术，使得其性能取得了大幅地提升。研发团队基于 YOLOv3-SPP，即 DarkNet53 骨干和空间金字塔池化(Spatial Pyramid Pooling, SPP)，增加了指数加权平均、余弦退火学习率，IoU Loss 和

IoU-aware 分支。另外，YOLOX 使用 BCE Loss 来训练 Cls.和 Obj.分支，其中 Reg.分支使用了 IoU Loss。YOLOX 的数据增强使用了马赛克(Mosaic)、随机水平翻转(Random Horizontal Flip)、颜色抖动(ColorJitter)和多尺度，随机大小裁剪(Random Resized Crop)因为同马赛克增强有部分重复所以被去除[6]。

YOLOX 网络相比于 YOLO 系列网络最大的改进点是解耦头(Decoupled head)。一直以来，分类与回归任务的冲突都是在目标检测中常见的问题，YOLO 系列的工作也都是使用一个 branch 就同时完成 Obj.、Cls.及 Reg.三部分的预测。研发团队在实验中发现检测头耦合会对模型性能产生负面影响，所以 YOLOX 中采用了轻量解耦头替换 YOLO 系列惯用的检测头，从而显著改善模型的收敛速度，同时轻量的解耦头可以带来更短的推理耗时。YOLOX 解耦头与 YOLOv3-v5 的头部区别如图 1 所示。

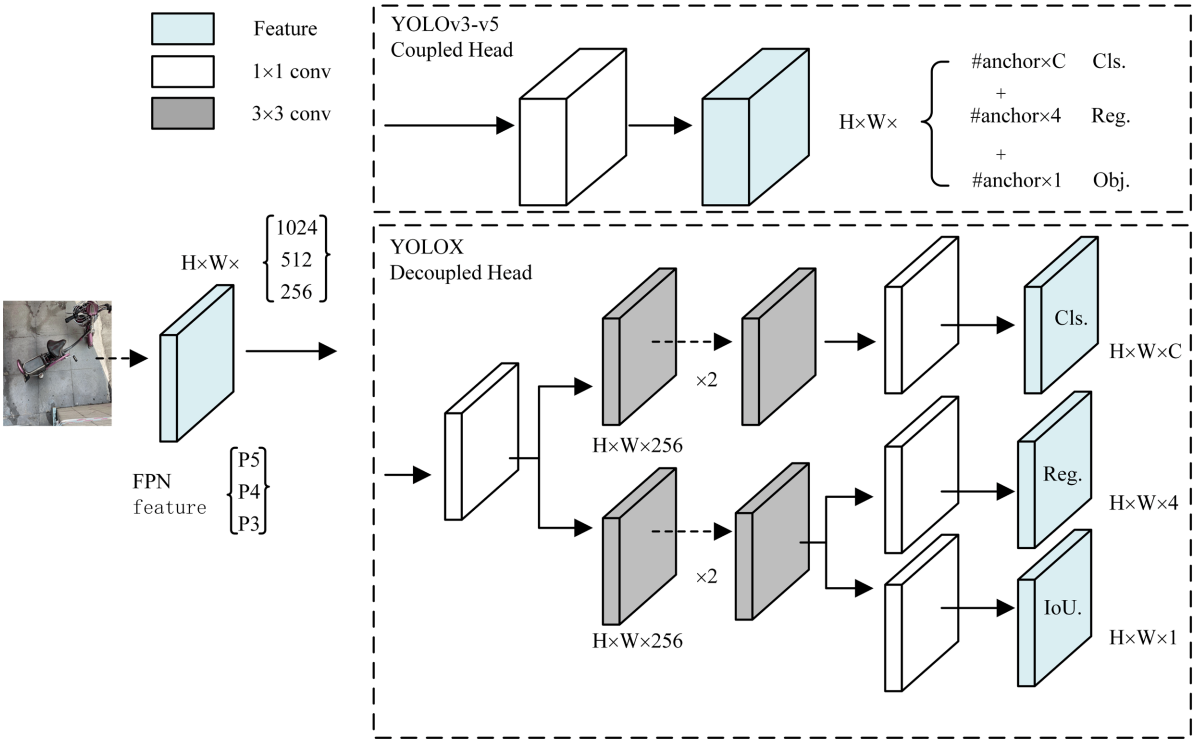

Figure 1. The difference between decoupled head and YOLO head
图 1. YOLOX 解耦头与传统 YOLO 头部的区别

2.2. TensorRT 加速推理

训练对于深度学习来说是为了获得一个高性能的模型，而深度学习模型在训练阶段，为了能够保证前后向的传播，每次梯度都会进行极小的更新，此时则需要相对较高的进度，像如 FP32 这样的 Float 类型。但在推断时，由于没有了训练过程中的反向迭代，因此只需要向前计算即可。由此，在推理阶段即便使用了像如 FP16 或 INT8 甚至更低精度的技术仍不会对其推理结果有大的精度损失[7] [8]，也就是说，推理更关注的是低响应时间、低资源消耗、高吞吐量以及方便的部署流程[9]。TensorRT 就是为了解决推理及部署问题提出的，它是由 C++、CUDA、Python 三种语言编写的一个库，能够促进对英伟达 GPU 的高性能推理。TensorRT 的原理就是通过优化内核选择和组合层来优化网络，以改善延迟、电源效率、吞吐量和内存消耗等问题。此外，为了能够让模型在英伟达 GPU 上更好的加速运行，英伟达还对 TensorRT 还做了相应的提速优化：如算子融合、量化、内核自动调整、动态张量显存和多流执行。TensorRT 加速推理示意图如图 2 所示：

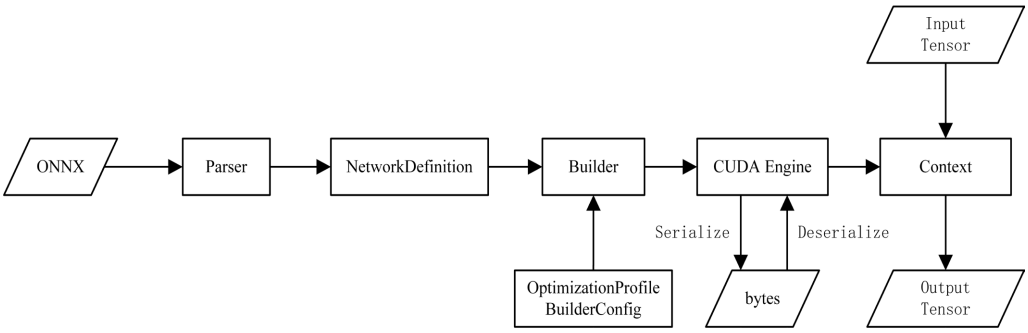

Figure 2. The diagram of TensorRT accelerated inference
图 2. TensorRT 加速推理流程图

此外，TensorRT 在英伟达的支持下不断更新，目前已经支持开放式神经网络交换(Open Neural Network Exchange, ONNX)、Caffe、Uff 等类型的解析模型，其中对 ONNX 路线的支持最好[10]。ONNX 是微软和 Facebook 合作开发的开放式神经网络交换工具，能够更好的对模型移植部署，在其他框架上进行推理。TensorRT 提供了 C++和 Python 接口、深度定制的 ONNX 解析器和低耦合封装技术，这些都能更好的结合 YOLOX 等模型进行部署。

2.3. DeepStream SDK

DeepStream SDK 是基于英伟达硬件运行的工具，它参考了 GStreamer 来构建，由一系列插件构成，主要应用于整个视觉流程的解决方案，同 OpenCV 等视觉库不同，DeepStream 建立了一套完整的端到端支持方案，包含视频编解码、TensorRT 推理引擎、OpenCV 跟踪器以及显示渲染插件等[11] [12]。DeepStream 代码框架示意图如图 3 所示：

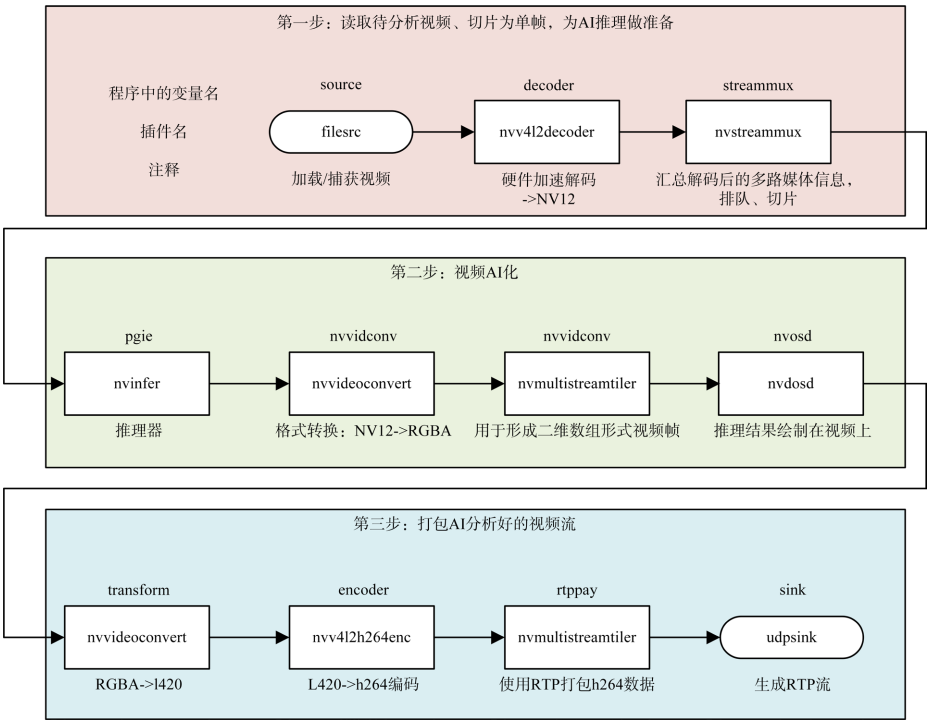

Figure 3. The diagram of DeepStream code framework
图 3. DeepStream 代码框架示意图

其中 nvinfer 是核心部分，其内部自带 TensorRT INT8/FP16 加速功能，是 TensorRT 的推理插件，理论上只要硬件支持就不需要进行额外的操作，具有更好的兼容性。

2.4. JetsonNano 2GB 开发套件

JetsonNano 2GB Developer Kit 是英伟达公司在 2020 年 GPU 技术大会(GPU Technology Conference, GTC)上发布的全新产品，区别于之前标准版的 JetsonNano，内存由标准版的 4 GB 缩小为 2 GB，从 B01 版本的双 CSI 相机支持变为单 CSI 相机接口，优化了散热布局和接口类型，缩小了硬件尺寸。售价上降低了 40%，仅为 59 美元，这也间接降低了产品成本。同时，其核心的 GPU 没有缩水，仍为 128 核心麦克斯韦架构，计算能力达到 5.3，支持 CUDA 和 TensorRT [13]，这颗 GPU 内核融合了 FP16 从而提高了性能和能效。低成本高性能也使得它更适合用于小型人工智能设备的研发。

软件支持上，基于 Ubuntu 18.04 定制的 JetPack 4.5.1 自带 TensorRT 7.1.3，包括对混合精度推理的支持、量化模型的支持、布尔值的张量和空张量等。

3. 电梯内电动车检测系统框架

结合上述技术，本文设计的基于 TensorRT 加速推理的电梯内电动车检测系统设计流程如图 4 所示：

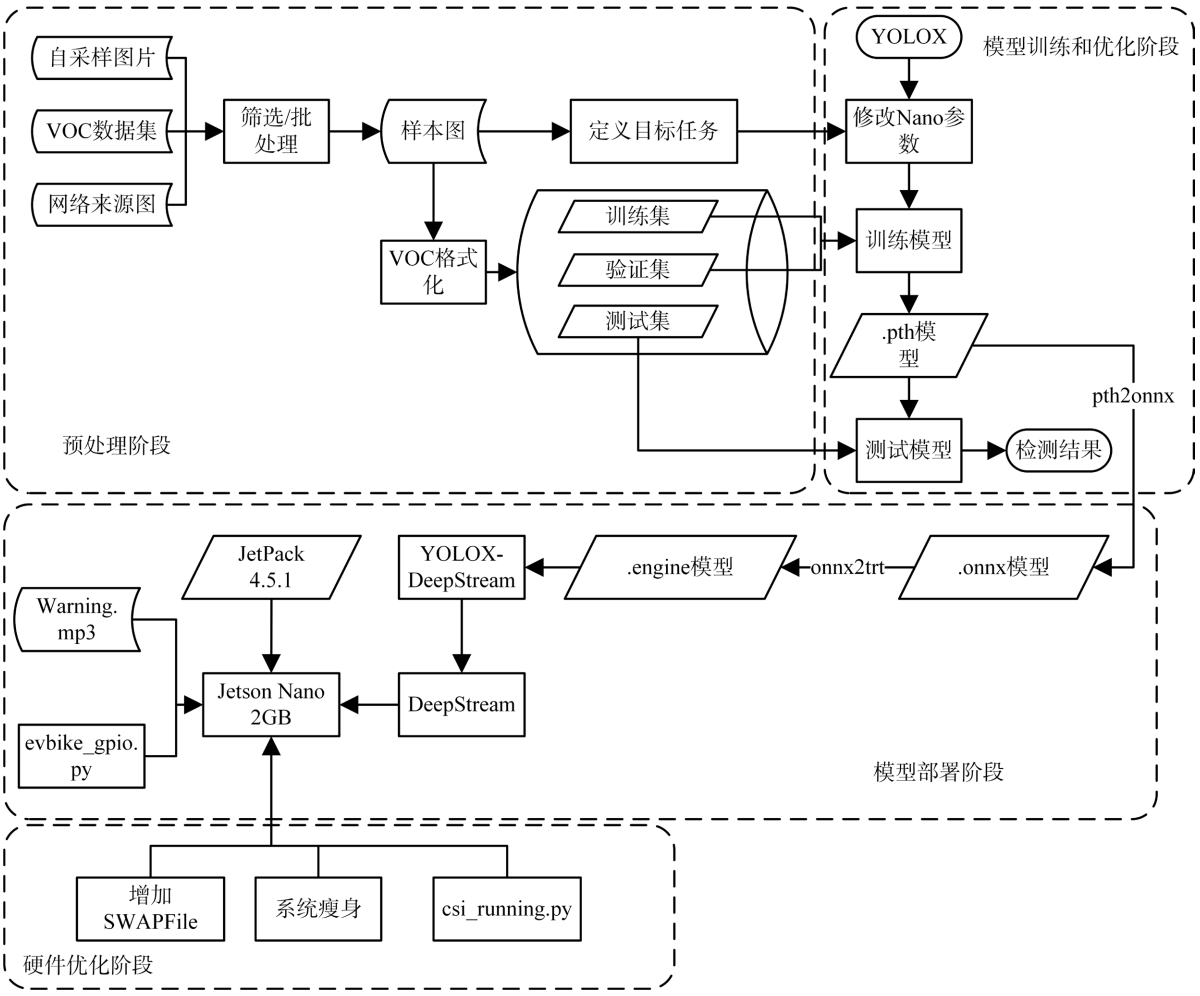

Figure 4. The diagram of DeepStream code framework
图 4. DeepStream 代码框架示意图

该设计分为四个阶段：预处理阶段，模型的训练阶段，模型部署阶段以及硬件优化阶段。

3.1. 预处理阶段

选定 YOLOX 网络结构，定义目标任务为电动自行车、电动摩托车和自行车，准备样本图像并按需求对样本图像进行预处理，然后对样本图像进行 VOC 格式化，得到 VOC 数据集。然后，将数据按照 7:1:2 的比例划分为训练集、验证集和测试集三部分。

关于数据集的制作，目标样本图像有三个来源：自行采集、VOC2012 数据集(自行车)和网络图片。

3.1.1. 样本图像自行采集

针对自行采集的样本，选定的采集地点分为室内和室外场景，室内场景包括停车场、楼道、电梯以及模拟电梯场景；室外场景选定城市街道及露天停车场。在拍摄光线的选择上，室内场景选择白天及有昏暗灯光的夜晚，室外场景选择天气晴朗光线较弱的下午及路灯开启的夜晚。

在横向对比了市面上数款产品后，本文总结了其在目标检测中错误识别的情况(倒车进入电梯轿厢或衣物等其他物体遮挡大部分目标)，为了更好的应用于训练及电梯内场景的应用，本文分别模拟了上述不同情况的目标状态，即包裹有雨衣的非机动车、装有防风棉衣的电动车以及载物状态下的电动车。同时，根据电梯内住户与电动车同框的情形，本文还分别采集了住户骑在非机动车上与住户站在非机动车旁造成遮挡情况的目标图像。此外，为了能够更充分地训练，本文还进行了针对目标的多角度拍摄及多数量拍摄(考虑到电梯内的空间，多数量情况下图像内的有效目标最多不超过 3 个)。整理完成后的样本如图 5 所示：

(a)电动自行车　(b)电动摩托车　(c)自行车

Figure 5. Sample images
图 5. 样本图片

3.1.2. 样本图像自行采集

下载 VOC2012 开源数据集，对数据集中的图像进行筛选，选出所有包含自行车目标的样本图像，利

用 python 脚本筛选与样本图像同名的标签文件。使用 labelme 标注工具将筛选后样本图片中的其他类型标签删除。利用 python 的 xml 树工具编写脚本，批量修改标签文件的路径，方便后续使用。

3.1.3. 图像样本网络爬取

电商平台的买家评论区跟商品详情中有训练所需的图片，通过爬虫技术爬取各大电商平台的图片，其中搜索关键词为电动车、电动自行车、电动摩托车、自行车。此部分的样本图像来源不可控，存在图片重复、不相关内容和图片质量差等问题，需要按照训练所需的样本需求进行人工筛选。

3.1.4. 数据集制作

将人工采集的样本图像和网络图像进行整合、筛选，结合 VOC 数据集中的自行车样本图，各类目标样本的数量按 1:1:1 的比例进行整合。

本文使用 ACDSee2018 Pro 工具将上述筛选后的样本中人工采集图像和网络爬取的样本图像按照 VOC2007 数据集的标准，即宽、高最大 500 像素，水平和垂直分辨率 96 dpi，位深 24 bit 进行等比例压缩，随后打乱顺序重新编号，使用 labelme 工具进行标注，制成 VOC 数据集。

最终本文制作的样本数据集包括：12,000 张 jpg 图像，12,000 个 xml 标签，18,536 个回归框，其中包括电动自行车、电动摩托车、自行车样本各 4000 张。三类目标样本数据集的分配情况如表 1 所示：

Table 1. Dataset allocation table
表 1. 数据集分配表

目标类	总数据集	训练集	测试集	验证集
bicycle	4000	2800	800	400
ebike	4000	2800	800	400
emotobike	4000	2800	800	400
合计	12,000	8400	2400	1200

3.2. 模型训练阶段

本文主要针对电动自行车、电动摩托车和自行车三类目标进行分类检测，对于目标模型训练在 CPU 为 AMD (R) Ryzen(R) R5-3600 CPU @ 3.60 GHz，内存为 32.0 GB DDR4-3600 MHz，显卡为 NVIDIA TITAN Xp 的电脑上进行，机器配置了 Pytorch 框架，通过 GPU 进行训练。由于训练好的目标模型最终需要部署在 JetsonNano 2 GB 这样的小算力设备上，因此选用 YOLOX 系列最小结构的 YOLOX-Nano，其网络参数量只有 0.91 M 和 1.08 G FLOPs，但却有着比 YOLOv3-Tiny、YOLOv4-Tiny 等更加优秀的性能。学习率上，使用 SGD、nesterov 和 momentum 的优化器及学习率调节器，学习率如公式(1)所示。

$$\mathrm{lr} = \mathrm{lr} * \frac{\mathrm{batch\_size}}{64} \tag{1}$$

其中 lr 为学习率，batch_size 为一次训练抓取的样本数。学习率初始值为 0.01，之后采用带有 warmup 策略的余弦退火学习率，权重衰减系数为 0.0005，并且在最后的 15 个 epoch 采用固定的最小学习率以配合数据增强。训练的 batchsize 为 16，epcoh 为 320。

3.3. 模型部署阶段

上一阶段中，在 Pytorch 框架下训练得到了.pth 格式的目标检测模型，在部署阶段，需要将该模型转化为能被 TensorRT 使用的.engine 模型。

首先为 JetsonNano 烧入 2 GB 版的 JetPack 4.5.1 系统，第一次进入系统后需更新系统组件，安装并更新 pip3 等相应的功能以及 protobuf 等支持 DeepStream 和 TensorRT 运行的工具库，然后激活系统内自带的 GPIO 文件和 TensorRT 文件。录制一段提示音频用于提示用户电动车不允许进入电梯，音频存为.mp3 格式。JetPack 4.5.1 内置了 DeepStream 开发工具，本文参考 DeepStream 文件中自带的 YOLOv3-Tiny 文件，增加了可用于部署 YOLOX 的 YOLOX-DeepStream。将转换为.engine 格式的模型命名为 evbike.engine，将该模型文件放入 YOLOX-DeepStream 中。

本文通过一个 Python 脚本调用 GPIO 的 11 引脚，在 CSI 相机检测到电动摩托车或电动自行车后上拉一个高电平控制电梯，使得用户不能使用电梯按钮进行关门操作，此时播放提示音频提示用户将电动车驶离电梯，当相机检测不到电梯内存在电动车后，下拉低电平恢复电梯正常运行。基于 TensorRT 加速的电梯内电动车检测设备工作流程如图 6 所示：

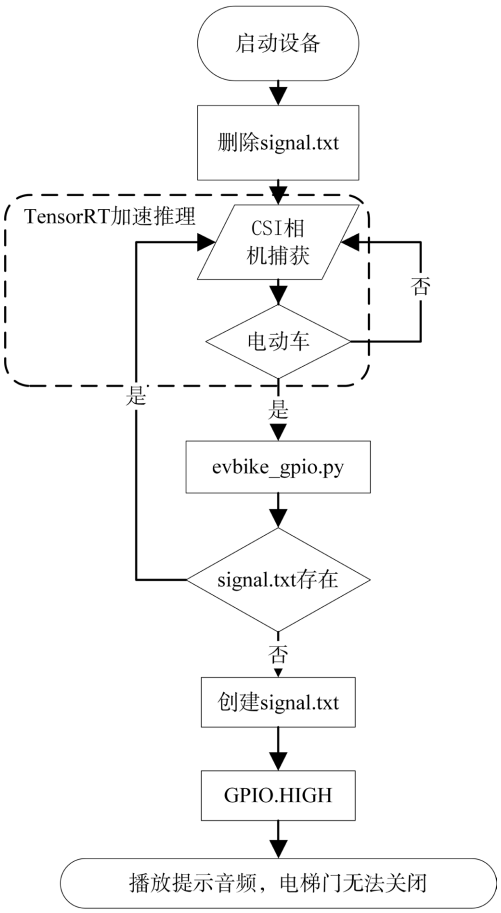

Figure 6. Device works flow chart
图 6. 设备工作流程图

3.4. 硬件优化阶段

由于 Jetson Nano 2 GB 版本仅有 2 GB 内存，故需要增加 SWAPFile 虚拟内存以获得更好性能。本文针对项目实际的使用情况，对 JetPack 4.5.1 系统进行瘦身，移除了包括 libre-office 在内的软件及其他多余的相关依赖。此外，本文修改了开机启动选项，使检测程序能够在开机后自动运行，同时增加一个监测脚本在后台监测目标检测程序的运行状态，一旦发生崩溃或关闭现象，立即重启系统。

4. 实验仿真及结果分析

4.1. 测试数据及实验环境

本文为了实际工程应用的需要，进行了两部分测试，分别为部分样本图片的测试和电梯轿厢内采集的视频影像测试。本文重新选取测试样本，针对经过 TensorRT 加速推理的 YOLOX-Nano 模型、未经过 TensorRT 加速推理的 YOLOX-Nano 模型、未经过 TensorRT 加速推理的 YOLOv4-Tiny 模型和未经过 TensorRT 加速推理的 SSDMobileNet 模型进行了横向对比测试。

本次实验测试在 CPU 为四核 ARM® A57 @ 1.43 GHz，GPU 为 128 核 NVIDIA Maxwell 架构，算力为 5.3，内存为 2 GB 共用显存的 JetsonNano 2 GB 设备上，系统环境为基于 Ubuntu 18.04 深度定制的 JetPack 4.5.1。样本一选取了测试集样本中的 600 张图片，样本二则是选取了电梯轿厢内的三段监控视频录像，录像规格为 720 p 30 fps，视频码率为 2500 kbps，视频编码为 h.264，视频时长 50 秒。

4.2. 实验结果评估及结论

针对本章节描述的两部分样本测试的效果图如图 7 所示：

Figure 7. Test renderings
图 7. 测试效果图

本文使用 mAP (mean Average Precision)对结果进行评估，mAP 是目标检测常用的精度评估指标，通过计算每个分类的 AP (Average Precision)再求平均得到，AP 是各类目标的检测精度，由 P-R (Precision-Recall)曲线积分得到，Precision 计算如公式(2)所示：

$$\text{precision} = \frac{\text{TP}}{\text{TP} + \text{FP}} \tag{2}$$

式中 TP (True Positive)为模型预测为正类的正样本，FP (False Positive)为模型预测为正类的负样本。Recall

计算如公式(3)所示：

$$\text{recall} = \frac{\text{TP}}{\text{TP} + \text{FN}} \tag{3}$$

式中 FN (False Negative)为模型预测为负类的正样本。根据上述计算得到测试结果如表 2 所示：

Table 2. Object detection results of the four models
表 2. 四种模型的目标检测结果对比

样本	网络模型	检测时间/秒	mAP
样本 1	TensorRT 加速推理的 YOLOX-Nano 模型	23.256	0.951
	YOLOX-Nano 模型	35.827	0.959
	YOLOv4-Tiny 模型	43.092	0.889
	SSDMobileNet 模型	29.945	0.813
样本 2	TensorRT 加速推理的 YOLOX-Nano 模型	54.227	0.939
	YOLOX-Nano 模型	82.019	0.954
	YOLOv4-Tiny 模型	100.732	0.875
	SSDMobileNet 模型	71.364	0.801

分析本次测试结果，可以得出，在针对小型开发设备性能优秀的 YOLOX-Nano 模型基础上，经过 TensorRT 加速推理得到了更快的推理速度，模型性能进一步提升。实验证明，本文采用的基于 TensorRT 加速推理的电梯内电动车识别系统有着高效的目标检测效率，且相比于市面上的其他产品，识别速度更快，准确率更高。此外，经过 TensorRT 加速推理的 YOLOX 模型，虽然在部署阶段不可避免地需要进行格式转换，且在由.pth 到.onnx 再到.engine 转换的过程中，精度有所损失，但在精度损失较小的情况下换来更快的推理速度是现阶段的最优选择。

5. 结论

本文总结市面上其他产品的检测缺陷，分类上将电动车细分为电动自行车与电动摩托车，细化了两个目标之间的特征以及两目标与自行车之间的特征。在图像预处理阶段，本文分析实际应用中遭遇的特殊情况，优化了数据集的采集和配比。在神经网络的选择上，本文选用 YOLOX 网络训练 YOLOX-Nano 模型，模型性能高于市面其他产品所使用的网络模型。在硬件选择上，本文选用了支持加速推理的 JetsonNano 开发套件，其 2 GB 版本在优化和支持完善的情况下也降低了成本。同时，本文针对 JetsonNano 2 GB 设备进行系统瘦身及优化，增加了虚拟内存，删除了除支持目标检测系统运行及后期系统维护的所有软件依赖。在 DeepStream 5.1 的基础上编写 YOLOX-DeepStream，成功将目标模型、CSI 摄像头、GPIO 和 TensorRT 关联起来，经过 TensorRT 技术加速推理后的目标检测模型以极小的精度损失换来了更快的推理速度。

本文设计的检测设备局限于 JetsonNano 2 GB 设备的性能，在检测到电动车目标后通过 Python 脚本调用 GPIO 接口时，由于相继调用 Python 脚本与 GPIO 带来的瞬时系统负载，稳定在 30 fps 的帧率会出现 1~2 s 的丢帧现象，因此，在后续的产品优化上需要使用 C++代替 Python 脚本，针对 YOLOX-DeepStream 进行进一步修改，将 GPIO 与音频的调用统一封装至 DeepStream 中。此外，设备在加密与防破解上仍有一定的漏洞，后续优化需为设备部署更好的软硬件加密方式。

参考文献

[1] 刘纯银. 治理电动车“上楼入户”要疏堵结合[N]. 钦州日报, 2021-07-13(002).

[2] 史一棋, 孙立极. 解决电动自行车停放和充电问题需综合施策[N]. 人民日报, 2021-07-05(007).

[3] 吴文诩. 禁止电动车“上楼入户”须下“狠手” [N]. 新华每日电讯, 2021-05-12(011).

[4] 静子. 电动车出电梯, 安全意识进人心[J]. 人民周刊, 2021(9): 18.

[5] 张文韬. 基于边缘计算的电动车入户充电检测方法研究[D]: [硕士学位论文]. 合肥: 安徽建筑大学, 2021.

[6] Ge, Z., Liu, S.T., Wang, F., Li, Z.M. and Sun, J. (2021) YOLOX: Exceeding YOLO Series in 2021. https://doi.org/10.48550/arXiv.2107.08430

[7] 周立君, 刘宇, 白璐, 刘飞, 王亚伟. 使用 TensorRT 进行深度学习推理[J]. 应用光学, 2020, 41(2): 337-341.

[8] 施一飞. 对使用 TensorRT 加速 AI 深度学习推断效率的探索[J]. 科技视界, 2017(31): 26-27.

[9] Jeong, E., Kim, J., Tan, S., Lee, J. and Ha, S. (2021) Deep Learning Inference Parallelization on Heterogeneous Processors with TensorRT. *IEEE Embedded Systems Letters*, **14**, 15-18.

[10] 陈欢. 基于 YOLO 的移动端视频目标跟踪的设计与实现[D]: [硕士学位论文]. 哈尔滨: 黑龙江大学, 2021.

[11] Chen, Y.-C., Fathoni, H. and Yang, C.-T. (2020) Implementation of Fire and Smoke Detection Using DeepStream and Edge Computing Approachs. 2020 *International Conference on Pervasive Artificial Intelligence* (*ICPAI*), Taipei, 3-5 December 2020, 272-275. https://doi.org/10.1109/ICPAI51961.2020.00058

[12] Uddin, M.I., Alamgir, M.S., Rahman, M.M., Bhuiyan, M.S. and Moral, M.A. (2021) AI Traffic Control System Based on DeepStream and IoT Using NVIDIA Jetson Nano. 2021 2*nd International Conference on Robotics, Electrical and Signal Processing Techniques* (*ICREST*), DHAKA, 5-7 January 2021, 115-119. https://doi.org/10.1109/ICREST51555.2021.9331256

[13] Süzen, A., Duman, B. and Şen, B. (2020) Benchmark Analysis of Jetson TX2. Jetson Nano and Raspberry PI Using Deep-CNN. 2020 *International Congress on Human-Computer Interaction, Optimization and Robotic Applications* (*HORA*), Ankara, 26-28 June 2020, 1-5. https://doi.org/10.1109/HORA49412.2020.9152915