

文章编号: 2095-2163(2021)07-0138-06

中图分类号: TN911.73

文献标志码: A

# 嵌入式口罩佩戴检测系统研究与实现

柯 鑫, 张荣芬, 刘宇红

(贵州大学 大数据与信息工程学院, 贵阳 550025)

**摘要:** 常态化疫情防控形势下, 公共场合佩戴口罩可以有效降低交叉感染风险, 针对口罩佩戴检测中的小目标检测困难以及实时性较差的问题, 提出了基于嵌入式平台 Jetson nano 的口罩佩戴检测系统, 通过增加 YOLOv3-tiny 的主干网络层深度, 引入注意力机制以及 TensorRT 模块, 提升了嵌入式系统口罩佩戴检测任务的精度和实时性, 改进后的 YOLOv3-tiny 算法 *mAP* 值达到了 87.5%, *FPS* 为 20.4, 相较于改进前精度提升 12.3%, 帧率提升 10.4 fps。

**关键词:** 疫情防控; 口罩佩戴检测; Jetson nano; YOLOv3-tiny

## Research and implementation of embedded mask wearing detection system

KE Xin, ZHANG Rongfen, LIU Yuhong

(College of Big Data and Information Engineering, Guizhou University, Guiyang 550025, China)

**[Abstract]** Under the situation of normalized epidemic prevention and control, wearing a mask in public can effectively reduce the risk of cross infection. In view of the difficulty in detecting small targets and poor real-time performance in mask wearing detection, a mask wearing detection system based on the embedded platform Jetson nano is proposed. By increasing the depth of the backbone network layer of YOLOv3-tiny, introducing the attention mechanism and the TensorRT module, the accuracy and real-time performance of the mask wearing detection task of the embedded system are improved. The improved YOLOv3-tiny algorithm has a *mAP* value of 87.5% and an *FPS* of 20.4. Compared with the previous improvement, the accuracy has increased by 12.3% and the frame rate has increased by 10.4 fps.

**[Key words]** epidemic prevention and control; mask wearing detection; Jetson nano; YOLOv3-tiny

## 0 引言

自 2019 年 12 月开始, 新型冠状病毒肺炎 (COVID-19) 在全国范围内进行快速的传播与流动, 新型冠状病毒的传染性很强, 而其主要的传播媒介为人, 传播途径为空气中的飞沫以及气溶胶。空气中带有病毒的飞沫和气溶胶极有可能通过人的呼吸道进行传播, 因此口罩作为呼吸系统的屏障, 佩戴口罩可以有效阻隔病毒的传播。虽然到目前为止还未提出一种针对口罩佩戴的检测算法, 但是关于人脸识别方面的对应算法已经存在, 并且能够有效地对人脸进行识别<sup>[1]</sup>。本文以针对人脸的目标检测算法为思路, 通过自制口罩数据集来进行针对性的训练, 提升对佩戴口罩的检出率<sup>[2]</sup>。采用的深度学习模型为 YOLO, 相比 R-CNN、FAST R-CNN、FASTER R-CNN 等模型, YOLO 的性能更加突出。

和前文提到的几种深度学习模型相比, YOLO 最大的优势是速度快, 实时性好。YOLO 算法的优秀性能来源于其复杂的模型, 模型越复杂, 计算量相应地也就越大, 因此在计算资源和内存都有限的嵌入式平台难以实现, 随着算法模型的优化、算力的增强, 边缘计算开始崭露头角, 在嵌入式设备中部署此类算法成为热门研究对象, 此时就出现了 Tiny-YOLO 算法以及其新版本的 YOLOv3-tiny, 相比较于 YOLO 算法而言更加精简, 占用更少的计算资源及内存。Jetson nano 支持深度学习框架, 且计算能力出众, 因此本文尝试 YOLOv3-tiny 移植到 Jetson nano 平台中, 并进行优化加速实现计算资源局限平台的口罩佩戴的检测。

## 1 目标检测算法

随着 GPU 并行计算技术和深度学习的发展, 卷

**基金项目:** 贵州省科技计划项目(黔科合平台人才[2016]5707)。

**作者简介:** 柯 鑫(1995-), 男, 硕士研究生, 主要研究方向: 机器学习、机器视觉、目标检测; 张荣芬(1997-), 女, 博士, 教授, 主要研究方向: 嵌入式系统、机器视觉、大数据与计算应用; 刘宇红(1963-), 男, 硕士, 教授, 主要研究方向: 嵌入式系统、大数据应用、机器视觉与机器学习。

**通讯作者:** 刘宇红 Email: yhliu2@gzu.edu.cn

**收稿日期:** 2021-04-21

积神经网络在目标检测算法中得到了越来越多的应用<sup>[3]</sup>。从目前来看,目标检测算法主要分成两大类,一类是如 R-CNN<sup>[4]</sup>、faster R-CNN<sup>[5]</sup> 基于区域提议的检测算法<sup>[6]</sup>、Josepy 等人<sup>[7]</sup> 在 2016 年提出了基于回归的算法 YOLO(You Only Look Once),该算法将目标检测理解成为一种回归问题,很大程度上提高了检测的速度。Wei 等人<sup>[8]</sup> 提出了单阶段多尺度检测模型、即 SSD (Single Shot MultiBox Detector),引入了锚框机制,直接基于锚框回归出检测框,在检测速度和检测精度上均有很好的效果,2017~2018 年期间,Josepy 等人分别提出了改进的 YOLOv2<sup>[9]</sup> 和 YOLOv3<sup>[10]</sup> 两个版本的算法,在 SSD 模型的基础上进一步提升了检测精度和速度。同时为了适应边缘计算的趋势,以及更好地在嵌入式平台上运行该算法,提出一种轻量版的 YOLO 模型,即 YOLOv3-tiny。该模型通过减少特征层以及独立预测分支达到了速度的提升,初步达到了嵌入式系统的实时检测性能要求<sup>[11]</sup>。

## 2 Jetson nano 嵌入式平台

Jetson nano 的 CPU 为 ARM Cortex-A57 64-bit @ 1.42 Ghz, GPU 为 NVIDIA Maxwell w/128 CUDA cores@ 921 Mhz, 配备了 4 GB 的 LPDDR4。Jetson nano 具有 128 个 CUDA 核心的 GPU 功能更强大,性能优异,因此 Jetson nano 适合于本文边缘计算场景<sup>[12]</sup>。Jetson nano 具有 2 种电源模式,分别为 5 W (低功耗模式) 和 10 W (高功耗模式),当运行 YOLOv3-tiny 模型的时候需要调至高功耗模式,并且供电的电源也必须是 5 V、2 A 的标准电源,否则 Jetson nano 将会出现掉电现象,无法正常运行模型。

## 3 YOLOv3-tiny 算法及改进

### 3.1 YOLOv3-tiny 模型

YOLOv3-tiny 是 YOLOv3 的简化版本,主要区别为主干网络采用一个 7 层 conv+max 网络提取特征(和 darknet19 类似),嫁接网络采用的是 13 \* 13、26 \* 26 的探测网络,YOLOv3-tiny 的网络结构如图 1 所示。YOLO v3-tiny 的优点主要是:网络简单,计算量较小,可以实现边缘计算,在 Jetson nano 上运行 YOLOv3-tiny 模型可以达到 10 fps/s 的检测速度,一定程度上满足嵌入式平台下的实时要求。YOLOv3-tiny 检测速度的提升很大,但是随之而来的是检测精度下降,由于去掉了残差模块,减少了卷积层和多尺度特征融合层的个数,对于深层特征图

中目标细节信息表达能力不佳,虽然模型中用到了多尺度融合的方法,但是中小尺寸目标漏检现象仍然存在。

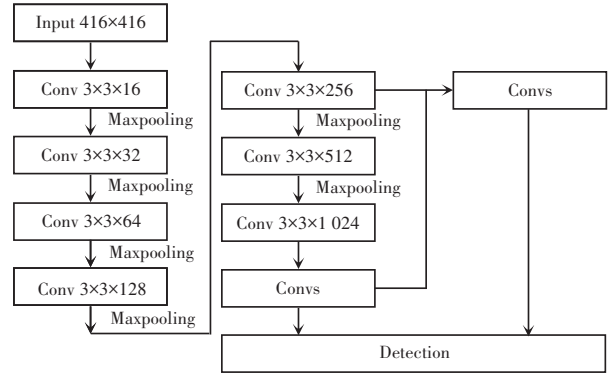


图 1 YOLOv3-tiny 网络结构图

Fig. 1 Network structure of YOLOv3-tiny

### 3.2 改进 YOLOv3-tiny 的网络结构

YOLOv3 采用 darknet53 作为 backbone 网络,深度为 107 层,而裁剪后 YOLOv3-tiny 的网络深度为 24 层,原来的 3 层 YOLO 层变为 2 层,每层 YOLO 层有 3 个 anchors,一共 6 个 anchors 值,网络模型层次架构如图 2 所示。

layer	filters	size	input	output	FLOPs
0 conv	3x3x3	1	416x416x3	416x416x3	0.156 GFLOPs
1 conv	2x2x2	2	208x208x3	208x208x3	0.156 GFLOPs
2 conv	3x3x3	2	104x104x3	104x104x3	0.156 GFLOPs
3 conv	2x2x2	2	52x52x3	52x52x3	0.156 GFLOPs
4 conv	3x3x3	2	26x26x3	26x26x3	0.156 GFLOPs
5 conv	2x2x2	2	13x13x3	13x13x3	0.156 GFLOPs
6 conv	3x3x3	2	13x13x3	13x13x3	0.156 GFLOPs
7 conv	2x2x2	2	13x13x3	13x13x3	0.156 GFLOPs
8 conv	3x3x3	2	13x13x3	13x13x3	0.156 GFLOPs
9 conv	2x2x2	2	13x13x3	13x13x3	0.156 GFLOPs
10 conv	3x3x3	2	13x13x3	13x13x3	0.156 GFLOPs
11 conv	2x2x2	2	13x13x3	13x13x3	0.156 GFLOPs
12 conv	3x3x3	2	13x13x3	13x13x3	0.156 GFLOPs
13 conv	2x2x2	2	13x13x3	13x13x3	0.156 GFLOPs
14 conv	3x3x3	2	13x13x3	13x13x3	0.156 GFLOPs
15 conv	2x2x2	2	13x13x3	13x13x3	0.156 GFLOPs
16 conv	3x3x3	2	13x13x3	13x13x3	0.156 GFLOPs
17 route	15				
18 conv	2x2x2	2	13x13x3	13x13x3	0.156 GFLOPs
19 conv	2x2x2	2	13x13x3	13x13x3	0.156 GFLOPs
20 route	15				
21 conv	2x2x2	2	13x13x3	13x13x3	0.156 GFLOPs
22 conv	2x2x2	2	13x13x3	13x13x3	0.156 GFLOPs
23 route	15				

图 2 YOLOv3-tiny 网络模型层次架构

Fig. 2 Network model hierarchy of YOLOv3-tiny

卷积层数较浅可以更好地表征小目标,而较深的卷积层数对大尺度目标具有较好的表征能力<sup>[13-14]</sup>,随着层数的加深,网络结构对于特征提取的效果也越好,但是过深的网络结构会导致产生梯度爆炸或者梯度消失等问题,因此不能一味地添加网络层数来增加特征提取能力。针对 YOLOv3-tiny 卷积层少,检测精度不高的问题,由于本文是以人脸为检测对象,所以为了获得更好分辨率的信息,添加的卷积层采用 3x3 卷积核的改进方法,在保证效率的同时提升了精度,以增加在使用场景的实用性和

准确性<sup>[15]</sup>。随着添加的网络层数增多,检测精度在逐步上升,与此同时由于增加了卷积层数,计算量在增大,层数的增加与精度的提升并非线性关系,当添加的层数大于 4 层时,模型的推理速度下降得比较多,但是提升的精度却不明显,因此本文在综合精度和效率的前提下,增加了 4 个 3×3 的卷积层起到了加深网络层数的效果,为了提高模型的学习能力,同时减少模型的参数,本文在添加的 4 个 3×3 的卷积层中添加了对应的 1×1 卷积层,较好地平衡了精度与速度。改进的 YOLOv3-tiny 网络结构如图 3 所示。

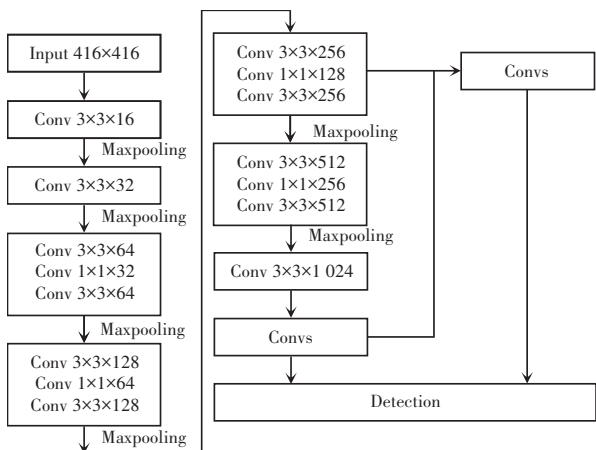


图 3 改进的 YOLOv3-tiny 网络结构

Fig. 3 Improved network structure of YOLOv3-tiny

### 3.3 引入 TensorRT 的模块

在 IEEE 754 标准中定义了一种半精度浮点类型<sup>[16]</sup>,在 CUDA 中被称作 Half 类型,在相同的时间周期内完成两个半精度浮点类型运算,相对于单精度的数据类型,半精度浮点类型的运算速度更快且效率更高。TensorRT 主要采用了层间融合或张量融合、精度校准 2 种优化方法。对此可做研究详述如下。

(1) 层间融合或张量融合 (Layer & Tensor Fusion)。如图 4 左侧是 GoogLeNetInception 模块的计算图。这个结构中有很多层,在部署模型推理时,每一层的运算操作都是由 GPU 完成的, TensorRT 通过对层间的横向或纵向合并(合并后的结构称为 CBR,意指 convolution, bias, and ReLU layers are fused to form a single layer),横向合并可以把卷积、偏置和激活层合并成一个 CBR 结构,只占用一个 CUDA 核心。纵向合并可以把结构相同,但是权值不同的层合并成一个更宽的层,也只占用一个 CUDA 核心。合并之后的计算图(图 4 右侧)的层次更少了,占用的 CUDA 核心数也少了,因此整个模

型结构会更小、更快、更高效。

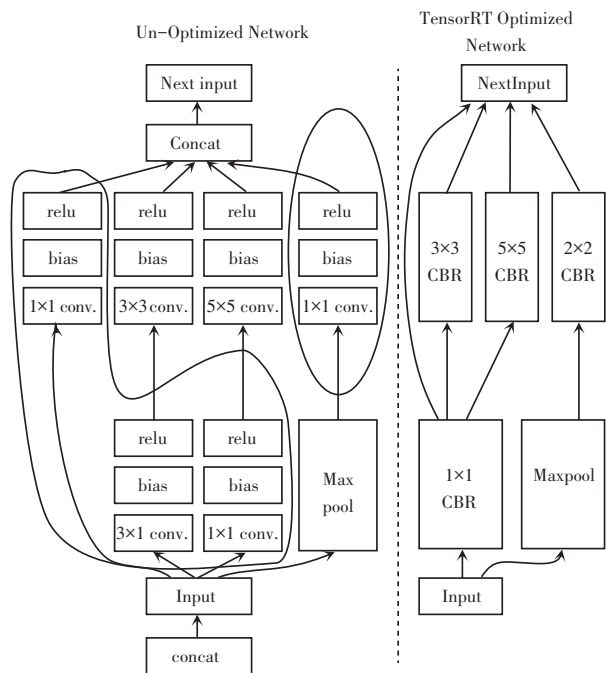


图 4 GoogLeNetInception 模块的计算图以及 TensorRT 简化和合并后计算图

Fig. 4 Calculation diagram of the GoogLeNetInception module, simplified and combined calculation diagram of TensorRT

(2) 数据精度校准 (Weight & Activation Precision Calibration)。通常,深度学习框架在训练时网络中的张量(Tensor)都是 32 位浮点数的精度(Full 32-bit precision, FP32),在网络训练完成后,反向传播这一步骤不存在了,因此可以适当降低数据精度,比如降为 INT8 或 FP16 的精度。更低的数据精度将会使得内存占用和延迟更低,模型体积更小。基于 TensorRT 库函数实现适用于本文设计的口罩佩戴检测系统在 Jetson nano 平台上的加速框架,程序整体框架流程图如图 5 所示。

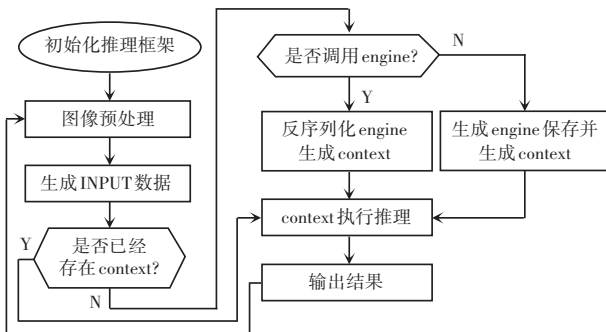


图 5 目标检测网络推理加速框架工作流程

Fig. 5 Reasoning acceleration framework of target detection network

在计算资源有限的嵌入式平台上生成序列化引擎这一步骤比较消耗时间,但是在同样的计算平台

和相同的参数下,对于序列化引擎可以进行复用,因此只需要提前生成一次序列化引擎便可以多次重复使用,减少了对计算资源的使用,节省了时间。TensorRT 加速推理框架主要部分为 context 数据、从不同框架模型中导入具有统一解析协议的序列化引擎文件,并利用序列化引擎进行反序列化得到 context 进行模型推理。

### 3.4 注意力机制模块

本文添加了一种新的体系结构单元,称之为 SE 模块(Squeeze-and-Excitation)<sup>[17]</sup>。SE 模块是通过重新学习卷积特征通道之间的相互依赖关系,筛选出了针对通道的注意力,以改变权重的方式来对原网络的语义信息进行调整。该单元通过对卷积得到的 feature map 进行处理,得到一个和通道数一样的一维向量作为每个通道的评价分数,然后将该分数施加到对应的通道上,得到其结果,实现过程如图 6 所示。

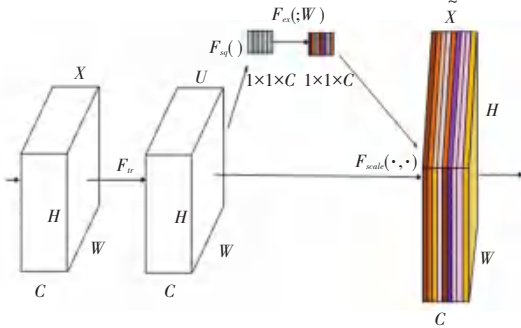


图 6 SE 注意力模块

Fig. 6 SE attention module

在此基础上,研发得到的 SE 模块的代码具体如下。

```
class SELayer(nn.Module);
def __init__(self, channel, reduction = 16);
    super(SELayer, self), _init_()
    self.avg_pool = nn.AdaptiveAvgPool2d(1)
    self.fc = nn.Sequential(
        nn.Linear(channel, channel // reduction,
bias = False),
        nn.ReLU(inplace = True),
        nn.Linear(channel // reduction,
channel, bias = False),
        nn.Sigmoid()
    )
def forward(self, x);
    b, c, _, _x.size()
```

```
y = self.avg_pool(x).view(b, c)
y = self.fc(y).view(b, c, 1, 1)
return x * y.expand_as(x)
```

YOLO v3 采用类似 FPN 上采样(Upsample)和融合做法,融合了 3 个尺度(13 \* 13、26 \* 26 和 52 \* 52),在多个尺度的融合特征图上分别独立做检测,但是在 YOLOv3-tiny 中缩减为 2 个尺度(13 \* 13 和 26 \* 26)。由于本文研究的对象为人脸,在大部分情况下的检测对象较小,因此为了增强对小目标检测信息能力,本文将 SE 模块融入到尺度为 26 \* 26 输出部分,对信息进行 refine,从而优化学习到的内容,尤其是加强了对于小目标的检测能力,添加了 4 层卷积层并且融入 SE 模块后的网络,本文称之为 SE-YOLOv3-tiny,网络结构图如图 7 所示。

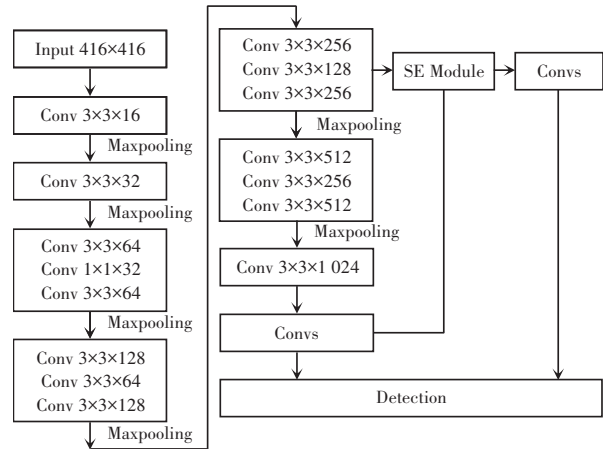


图 7 SE-YOLOv3-tiny 网络结构

Fig. 7 Network structure of SE-YOLOv3-tiny

## 4 实验

### 4.1 实验数据集

目前还没有针对口罩佩戴的公开的自然场景数据集,所以本文使用了 WIDER FACE 人脸数据集并从中随机抽取 2 000 张人脸图片,自行在网络上搜集佩戴口罩的图片 1 000 张,共收集训练图片 3 000 张,本文按照 7 : 3 的比例将自制的口罩数据集划分为 2 组,即训练集和测试集,其中 2 100 张图片作为训练样本,另外的 900 张图片作为测试样本。然后按照 VOC 数据集格式使用 LabelImage 对训练集和测试集进行统一的标注,标注信息包括了 2 种佩戴口罩的状态,分别用 1 和 0 表示。其中,1 对应为 masking,0 对应为 unmasking。

### 4.2 实验结果

本文分别使用 YOLOv3-tiny 算法及本文的不同改进优化算法对自建的口罩数据集进行训练和测

试,在迭代 50 000 次后得到最终的权重文件,图 8 给出了  $loss$  值变化曲线, $loss$  值反映了预测值与实际值的偏差。当  $loss$  值越接近 0 就代表模型的性能越好,可以看出在迭代 50 000 次后趋于稳定,不再明显下降。

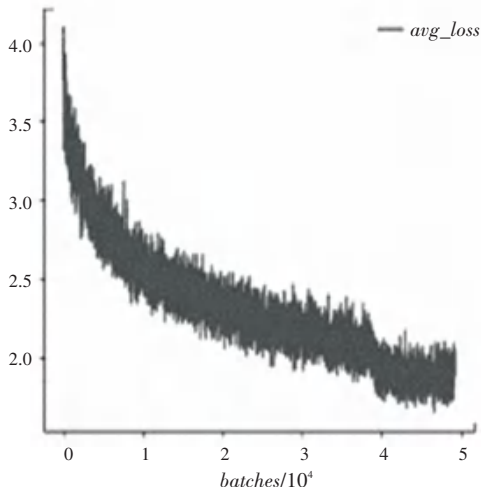


图 8 损失曲线

Fig. 8 Loss curve

为了更好地展示优化后的结果,本文针对精度提升和速度提升分别做了 2 个对比试验,分析论述具体如下。

(1)实验一。本文提出了 2 种 YOLOv3-tiny 的网络结构优化方法,该实验通过口罩数据集对原始的 YOLOv3-tiny 模型、增加 SE 模块以及添加网络层的优化方法进行了对比实验,得到结果见表 1。

表 1 口罩数据集对比检测结果

Tab. 1 Detection result comparison of mask data set

Detection method	$mAP$ / %	Masking	Unmasking
YOLOv3-tiny	75.2	74.9	75.5
Improved YOLOv3-tiny	87.4	87.0	87.8
SE YOLOv3-tiny	88.3	88.1	88.5

由表 1 可知,通过引入了 4 个  $3 \times 3$  卷积层,同时每个  $3 \times 3$  卷积层引入  $1 \times 1$  卷积层使得  $mAP$  值提升为 87.4%,相比 YOLOv3-tiny 提升了 11.2%,引入了 SE 模块 SE YOLOv3-tiny 的  $mAP$  值提升了 0.9%。此时,可以看到佩戴口罩与未佩戴口罩的  $AP$  存在差距,因为佩戴口罩的人脸照片和普通人脸的照片比例为 1:2,由于佩戴口罩的图片较少,所以  $AP$  值略低于未佩戴口罩的  $AP$  值。另外,为了直观地体现优化网络结构后的效果,选取了部分测试样本图片作为对比,如图 9 所示。

由上述内容可知在增加了卷积层以及添加了 SE 模块后,检出的准确率以及对小目标的检出的效

果都有较大的提升,在人群密集的场景下也有很好地检出能力,并且能正确地进行分类,证明了本文的优化方法可行性。

(2)实验二。实验一针对提升精度进行了优化方案,并进行对比试验。本实验将针对模型的速度优化做出对比试验。该实验通过 TensorRT 模块对模型进行处理,本文将原始的 YOLOv3-tiny 以及改进的 YOLOv3-tiny 模型转换得到 trt 文件(经 TensorRT 加速后得到的模型),在研究中统一使用 trt 后缀,表示加速后的模型,并进行测试,得到的检测结果见表 2。



(a) 优化前

(b) 优化后

图 9 口罩数据集对比检测结果

Fig. 9 Detection result contrast of mask data set

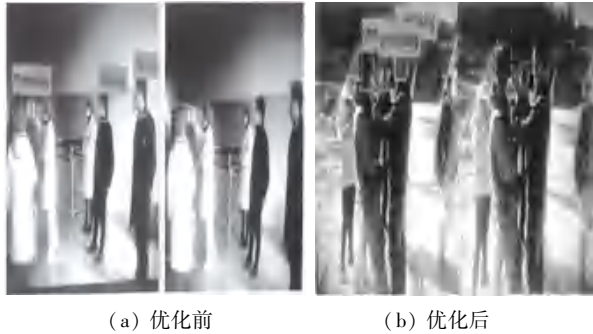
表 2 优化后口罩数据集测试对比结果

Tab. 2 Detection result comparison of mask data set after acceleration

Detection method	$mAP$ / %	Time-consuming/ ms	Frame rate/ fps
YOLOv3-tiny	75.2	98	10.0
YOLOv3-tiny_trt	73.9	45	22.2
Improved YOLOv3-tiny_trt	86.3	48	20.8
SE YOLOv3-tiny_trt	87.5	49	20.4

由表 2 可知,通过引入 TensorRT 模块,各个网络模型均得到了较大幅度的速度提升,改进 YOLOv3-tiny\_trt 相比于原始的 YOLOv3-tiny 网络,帧率提升至 20.8 fps,相比较未加速之前约有 2 倍的速度提升,与此同时  $mAP$  值提升了 11.1%,检测精度明显优于原始的 YOLOv3-tiny,SE YOLOv3-tiny\_trt 相对于改进 YOLOv3-tiny\_trt 算法而言,检测的速度几乎相同没有明显的下降,但是在检测精度上提升了 1.2%,这说明了引入 SE 模块的可行性,本文提出的优化方法使 YOLO 算法在嵌入式设备中得到了较大幅度的提升,已经具备实时检测的效果,可以应用于实际的项目工程中。为了直观地体现出优化之

后的实时检测效果,本文使用 YOLOv3-tiny、SE YOLOv3-tiny\_trt 两种模型分别进行测试,在视频中进行试验,由于场景比较复杂可以更好地检验模型的性能。图 10(a)、(b)均是对视频检测结果截图以及对比。



(a) 优化前 (b) 优化后

图 10 非正脸场景模型检测结果对比

Fig. 10 The detection comparison of different model serve as profile scene

可以看出,在总体上本文的 SE YOLOv3-tiny\_trt 算法能够有效地识别口罩佩戴情况,优化后提升了检测的准确率以及小目标的检测优化后的模型对于非正脸的检出率有着较好的提升,但可以看到的是对于侧脸的小目标的检测会存在漏检情况,同时存在着将颜色形状相接近的帽子误判为口罩情况如图 10(b)所示,这点在以后的工作中需要深入研究和进一步优化。图 10(a)表现出了对于未正确佩戴口罩识别的情况,体现出了一定的抗干扰能力。以上结果表明通过优化后的模型更加适应实际场景的需求,例如地铁站、火车站等人流密集、人脸目标较小的情况,具有一定的实际应用价值。

## 5 结束语

本文基于 YOLOv3-tiny 算法提出了一种基于嵌入式平台实时口罩检测算法,该方法通过添加卷积层的数量增加网络模型的深度,提升了网络模型的特征提取能力。引入注意力机制,添加了 SE 模块来优化所学习到的内容,提升检测准确率以及使用半精度推理模块 TensorRT 来提升检测的速度,最终达到了精度与速度的平衡,能较好地应用于实际场景。本文通过自建的 3 000 张图片的数据集进行训练的结果表示,本文提出的优化方法可以有效地检测实际场景下是否佩戴口罩,平均精度达到了 87.5%,每秒帧率达到了 20.8 fps,由此证明了本文优化方法的合理性。在以后的研究中将会进一步优化网络结构以提升检测精度,同时保证检测的速度,增大数据集的样本量,针对性解决手部遮挡等遮挡

面部的情况识别,提高口罩佩戴检测能力和效率。

## 参考文献

- [1] 王远大. UCloud 开放人脸口罩检测服务借助 AI 算法加快疫情防控[J]. 通信世界, 2020(5):33-34.
- [2] XU Tang, DU D K, HE Zeqiang, et al. PyramidBox: A context-assisted single shot face detector [M]//FERRARI V, HEBERT M, SMINCHISESCU C, et al. Computer Vision-Eccv 2018. Eccv 2018. Lecture Notes In Computer Science. Cham:Springer, 2018, 11213:812-828.
- [3] 陈超,齐峰. 卷积神经网络的发展及其在计算机视觉领域中的应用综述[J]. 计算机科学, 2019,46(3): 63-73.
- [4] GIRSHICK R, DONAHUE J, DARRELL T, et al. Rich feature hierarchies for accurate object detection and semantic segmentation [C]//IEEE Conference on Computer Vision and Pattern Recognition (CVPR). Columbus, USA:IEEE,2014: 580-587.
- [5] REN S, HE K, GIRSHICK R, et al. Faster R-CNN: Towards real-time object detection with region proposal networks [C]//Proceedings of International Conference on Neural Information Processing Systems. Cambridge:MIT Press,2015: 91-99.
- [6] ZITNICK C L, DOLL R P. Edge boxes: Locating object proposals from edges[M]. Cham: Springer International Publishing, 2014.
- [7] JOSEPY R, DIVVALA S, GIRSHICK R, et al. You only look once: Unified, real-time object detection [C]//2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR). Las Vegas, NV, USA:IEEE, 2016:21-37.
- [8] WEI L, ANGUELOV D, ERHAN D, et al. SSD: Single shot multiBox detector [C]// European Conference on Computer Vision(ECCV). Berlin:Springer,2016:21-37.
- [9] JOSEPY R, FARHADI A. YOLO9000: Better, faster, stronger [C]// 2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR). Honolulu, HI, USA:IEEE, 2017: 6517-6525.
- [10] JOSEPY R, FARHADI A. YOLOv3: An incremental improvement [C]//Proceedings of 2018 IEEE Conference on Computer Vision and Pattern Recognition. Washington D.C., USA: IEEE Press,2018:1-6.
- [11] MA Jing, CHEN Li, GAO Zhiyong. Hardware implementation and optimization of tiny-YOLO network [M]//ZHAI G, ZHOU J, YANG X. Digital TV and Wireless Multimedia Communication. IFTC 2017. Communications in Computer and Information Science. Singapore :Springer,2018,815:224-234.
- [12] 齐健. NVIDIA Jetson TX2 平台:加速发展小型化人工智能终端 [J]. 智能制造, 2017(5):20-21.
- [13] 吕俊奇, 邱卫根, 张立臣, 等. 多层卷积特征融合的行人检测 [J]. 计算机工程与设计, 2018,39(11):3481-3485.
- [14] 刘辉, 彭力, 闻继伟. 基于改进全卷积网络的多尺度感知行人检测算法[J]. 激光与光电子学进展, 2018, 55(9):318-324.
- [15] 陈聪, 杨忠, 宋佳蓉, 等. 一种改进的卷积神经网络行人识别方法[J]. 应用科技, 2019,46(3):51-57.
- [16] IEEE. IEEE standard for floating-point arithmetic:754-2008[S]. Washington D.C., USA:IEEE Press,2008.
- [17] HU Jie, LI Shen, SUN Gang. Squeeze-and-excitation [C]//2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition. Salt Lake City, UT, USA: IEEE, 2017:7132-7141.