

DOI:10.19651/j.cnki.emt.2415507

基于树莓派 4B 的无人机动态追踪平台设计*

陈浩安¹ 李晖² 黄瑞¹ 符平博¹ 张见¹

(1.南京信息工程大学电子与信息工程学院 南京 210044; 2.中国航空研究院研究生院 扬州 225006)

摘要: 针对无人机领域中的监管问题,基于 YOLOv5-Lite 的改进模型,提出了一种随着训练过程为模型动态地分配损失权重的指数移动样本加权函数。通过模型运算,控制二自由度云台对无人机实时跟踪,且视频采集、模型计算和二轴云台控制均在树莓派 4B 本地进行。优化过的模型在保持原模型参数数量的同时,在 mAP@.5:.95 指标中达到了 70.2%,相较于原模型提高了 1.5%。在树莓派上的实时推理平均速度为 2.1 FPS,处理效率更高。树莓派在模型推理的同时,通过 I2C 协议控制舵机平台对无人机目标进行追踪,保持对无人机的实时动态监测,提高了系统的可靠性,具有更好的实用价值。

关键词: 无人机;追踪;树莓派;YOLOv5-Lite;目标检测

中图分类号: TN915.81 **文献标识码:** A **国家标准学科分类代码:** 510.5030

Design of dynamic tracking platform for unmanned aerial vehicle based on Raspberry Pi 4B

Chen Haoan¹ Li Hui² Huang Rui¹ Fu Pingbo¹ Zhang Jian¹

(1. College of Electronics and Information Engineering, Nanjing University of Information Science and Technology, Nanjing 210044, China; 2. Graduate School of Chinese Aeronautical Establishment, Yangzhou 225006, China)

Abstract: Facing the challenges of regulating unmanned aerial vehicles (UAV), and based on an YOLOv5-Lite improved model, this paper incorporates an exponential moving sample weight function that dynamically allocates loss function weights to the model during the training iteration. Through model computations, we achieve real-time UAV tracking using a two-degree-of-freedom servo platform. Furthermore, video capture, model calculations, and servo control are all performed locally on a Raspberry Pi 4B. The optimized model maintains the original model's parameter count while achieving a mAP@.5:.95 score of 70.2%, representing a 1.5% improvement over the baseline model. Real-time inference on the Raspberry Pi yields an average speed of 2.1 frames per second (FPS), demonstrating increased processing efficiency. Simultaneously, the Raspberry Pi controls a servo platform via the I2C protocol to track UAV targets, ensuring real-time dynamic monitoring of UAVs. This optimization enhances system reliability and offers superior practical value.

Keywords: UAV; tracking; Raspberry Pi; YOLOv5-Lite; target detection

0 引言

近年来,无人机(unmanned aerial vehicles, UAV)在各种^[1-3]领域的关注度越来越高,尤其在消费级市场。无论协助空中作业还是辅助监管,UAV 都体现出巨大商业价值和前景。同时也出现了利用无人机运载危险物品、传播非法信息等违法行为,加上无人机种类复杂多样,限制条例错综复杂,如何管控好无人机是一项严峻的挑战。

针对 UAV 现有的地面警戒设备成本高、不易部署、对

目标的追踪特性差等缺点^[4-5],本文设计了一种易安装、低成本、易拓展、高灵活性的无人机动态追踪平台,用以协助管理方更好地对禁飞区域进行管控。尤其是针对那些提前设定好航线,采用静默飞行^[6]来躲避无线电探测手段的 UAV,本平台补齐了这方面的短板,提高了监管方的管理能力。

为了实现 UAV 的实时跟踪,本文分析和设计了以树莓派 4B^[7]为载体的图像处理与控制系统。树莓派的算力相对薄弱,对研究人员在选取模型方面有较大的限制,李

收稿日期:2024-02-20

* 基金项目:国家自然科学基金(61661018)项目资助

学宾等运用 Movidius 神经计算棒成功在树莓派上部署了 MobileNet-SSD 模型,调整了卷积层宽度并且替换了激活函数后,均值平均精度(mean average precision, mAP)指标达到 71.06% 的同时,将检测速度维持在了 8.24 每秒传输帧数(frames per second, FPS)^[8]。冷晨^[9]设计了一种基于树莓派的交通标志智能识别系统,其在 SC-YOLOv5s 中添加了 Coordinate Attention 注意力机制和 Stem block 结构,优化后模型参数量大小为 7.7 MB,虽然识别准确率达到 83.3%,但是推理延时达到了 2.9 s,不能满足对延迟有较高要求的实时任务。陈佳林等^[10]基于边缘计算的概念,设计了搭载英特尔二代神经计算棒的树莓派多线程处理系统,检测视频流中的目标速度达到了 21.6 FPS,有效提高了检测速度和图像利用率,但是其硬件成本过高,其高 FPS 依赖于英特尔的神经计算棒^[11]加速而非树莓派本身。上述研究还有一个共同的缺点,都没有运用到树莓派丰富的通用输入/输出接口(general purpose input/output ports, GPIO),导致了硬件资源的浪费。本文将 GPIO 接口中的 SDA 和 SCL 引脚与舵机控制模块 PCA9685 相连接,树莓派的 3.3 V 引脚为之供电。在模型识别目标无人机的同时,通过集成电路总线(inter-integrated circuit, I2C)^[12]向舵机传递偏转信息,控制舵机云台上的摄像头进行实时追踪。

本文的主要工作内容和研究贡献如下:

1)首先,针对 YOLOv5-Lite 损失函数进行优化,提出了一种全新的指数移动样本加权函数,不但可以加强模型在训练初期对简单样本学习,而且随着迭代次数的增加,还可以提高对困难样本的关注力度。这种动态的指数移动样本加权函数能覆盖模型训练的全过程,而且由于指数移动样本加权函数是针对训练过程的优化,对于进行检测的终端,不会产生额外的计算量,保持了模型的轻量化、大大提升了识别速度。

2)其次,利用 NCNN 框架摸索出了一套方法简单、调用方便的部署流程,让中小型深度学习模型在树莓派 Raspbian 操作系统中得以流畅运行,生成的可执行文件能被多种语言调用。

3)最后,在树莓派(Raspberry Pi 4B)平台上搭建了系统环境(OpenCV^[13]、PyTorch 和 Python)和模型依赖(CMake),从摄像头平台采集实时无人机位置,经过模型运算后,通过 I2C 接口向由双舵机组成的自由云台传输角度参数,使摄像头平台能一直追踪无人机位置并进行转动。

1 总体设计

针对这些问题,本文首先采用更为轻便的 YOLOv5-Lite^[14-15]为基础,并在 YOLO-FaceV2^[16]的基础上提出一种新的样本加权函数,能为训练过程中的样本分配更合理的权重,该方法还融合指数滑动平均^[17]的思想,使计算检测框的重叠度(intersection over union, IoU)均值时更为平

滑。然后通过 NCNN(注:NCNN 是腾讯优图实验室首个开源项目,是为手机端极致优化的高性能数据网络前向计算框架)计算框架将优化过后的模型部署在树莓派 4B 上,编写 C++ 程序生成可执行文件,通过该方法部署后的模型在树莓派中运算时,不需要任何额外的硬件支持。最后,树莓派通过摄像头采集视频进行计算,得出 UAV 目标框在画面中的位置后,通过编写好的舵机控制代码,用 GPIO 接口输出电信号给 PCA9685 控制板,控制二自由度舵机旋转,以达到镜头能一直追踪 UAV 飞行的动态效果。动态追踪系统总体结构示意图如图 1 所示。

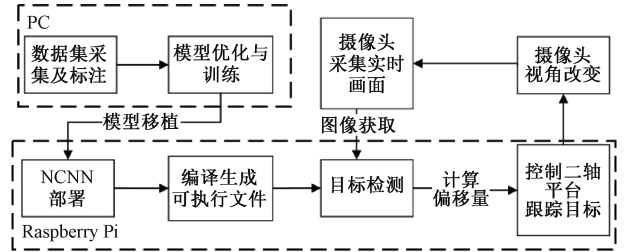


图1 动态追踪系统总体结构示意图

2 模型选取

2.1 YOLO v5-Lite 算法原理及其改进

考虑到对网络轻量化程度要求很高,本文选用 YOLO v5-Lite 为基础并在此之上进行改进。YOLO v5-Lite 使用一种优化过后的主干网络,针对 ShuffleNetV1 输入与输出纬度不一致且分组卷积数过多的问题^[18],构造了能平衡输入和输出的 ShuffleNetV2 作为主干网络^[19],如图 2 所示。

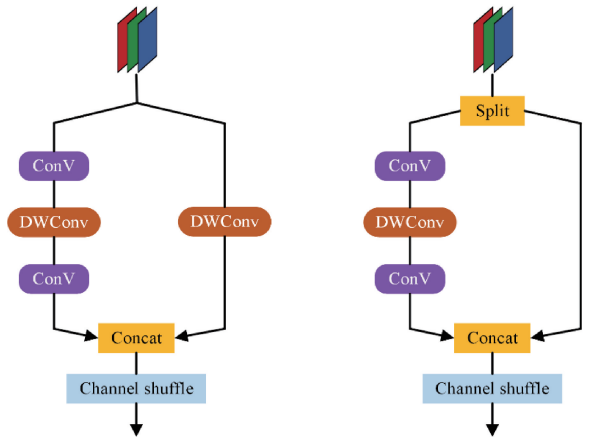


图2 ShuffleNetV2 网络结构

图 2 左边是 ShuffleNetV2 的空间下采样模块,相对于 ShuffleNetV1,移除了其中的通道分离操作,使输出通道增加了一倍。右边是 ShuffleNetV2 的基础模块,其中 Split 是引入的新操作,在一开始就将特征图在通道维度分成两个分支:位于左侧的分支采用同等映射,位于右侧的分支连续使用了三次卷积,并保持输入输出通道数不变。其中的 1×1 卷积取消了分组卷积,避免了信息的屏蔽和阻塞。

在左右分支结合时,采用 Concat 连接而不是 TensorAdd 连接,减少了元素级运算。为了更好的提取融合特征,YOLO v5-Lite 在主干网络和输出层之间采用了特征金字塔网络(feature pyramid network,FPN)与路径聚合网络(path aggregation network,PAN)^[20]结构来提高检测性能,使 YOLO v5-Lite 可以同时检测不同尺寸的对象,其整体结构图如图 3 所示。

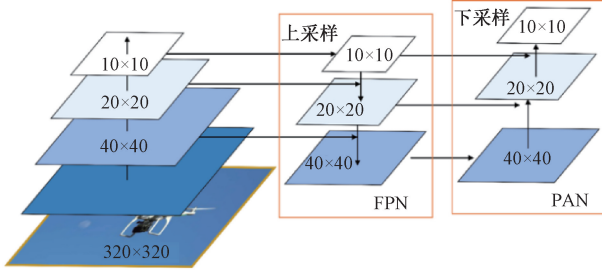


图 3 Neck 层结构

由于过于小的模型会让检测效果大大下降,而大模型在树莓派 4B 上进行运算时,计算一张图片耗时过长,实时目标检测的帧率过低。综合考虑,本文采用 YOLO v5-Lite 的 v5-Lite-s 版本,其大小是所有版本中除了 v5-Lite-e 以外最小的,其结构示意图如图 4 所示。

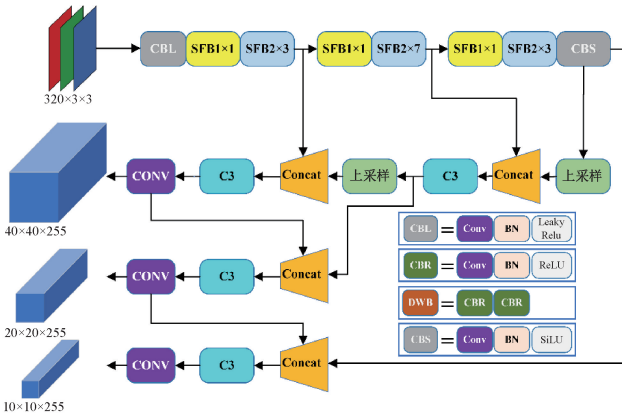


图 4 YOLO v5-Lite-s 结构

结构图中,第一行为模型的 Backbone 部分,从原始图片中提取输入特征,包括低级别的纹理、中级别的形状和高级别的语义信息。模型最后输出的三个检测头是 Head 部分,负责从提取的特征中预测目标的位置与类别。而连接 Backbone 和 Head 的中间模块即为 Neck 部分,通过使用上文提到的 FPN 和 PAN 结构,在特征传递的同时,引入上下文信息并进行了尺度变换,有助于提高目标检测的性能。

YOLO v5-Lite 采用轻量级的网络架构,适用于资源受限的应用场景。在保持较小模型尺寸的同时提供了良好的检测性能,能够灵活地用各种框架部署到嵌入式终端设备树莓派 4B 中。

2.2 模型优化

在模型训练的过程中,样本的不均衡问题一直存在。

容易样本数量庞大而困难样本较为稀疏,如何给予样本更合理的权重引起了许多人的关注。YOLO-FaceV2^[16]团队设计了一种样本加权函数来解决这一问题,但其设计的样本加权函数并未给稀疏的困难样本添加权重,导致模型对困难样本没有充分的学习。本文对其加权函数进行优化,并在划分样本难易程度的 IoU 均值计算中引入指数滑动平均的思想,使每次迭代中的 IoU 均值更新得更为平滑,不易受到异常值的影响。

假设训练迭代次数为 n ,记录每次迭代计算的检测框 IoU 均值 $[\theta_0, \theta_1, \theta_2, \dots, \theta_n]$,此时若直接将第 t 次迭代的 IoU 均值 θ_t 作为当次衡量样本困难程度的阈值,会有两个缺点:对于目前普遍的 mini-batch 训练方式,参数的更新可能存在抖动,该 θ_t 由于当此计算出现明显的异常值,可能导致难易样本划分出现紊乱;且当迭代次数 n 较大时,记录数据占用的内存是比较大的,计算效率也比较低。鉴于以上原因,本文采用指数滑动平均模型对每次迭代产生的 IoU 均值进行更新,每获取一个数据就将其和历史数据加权平均:

$$\mu_t = \beta \times \mu_{t-1} + (1 - \beta) \times \theta_t \quad (1)$$

其中, μ_t 为前 t 次迭代中 θ_t 的滑动加权均值, $\beta \in [0, 1)$ 为衰减因子。将式(1)展开得:

$$\begin{aligned} \mu_t &= \beta \times \mu_{t-1} + (1 - \beta) \times \theta_t = (1 - \beta) \times \theta_t + \beta \times (\beta \times \mu_{t-2} + (1 - \beta) \times \theta_{t-1}) \\ &= (1 - \beta) \times \theta_t + \beta \times (1 - \beta) \times \theta_{t-1} + \dots + \beta^{t-2} \times (1 - \beta) \times \theta_2 + \beta^{t-1} \times (\beta \times \mu_0 + (1 - \beta) \times \theta_1) \end{aligned} \quad (2)$$

从式(2)可以看到,第 t 次迭代会给历史中的第 $k(k < t)$ 次迭代分配一个 $\beta^{t-k} \times (1 - \beta)$ 的权值,也就是给与近期数据更高的权重,迭代初期的权重会被衰减因子 β 加以稀释,这可使模型在测试数据集上更具鲁棒。

在此基础上,为了进一步优化样本加权函数,着重为样本分配更合理的权重,提出最终采用的样本加权函数为:

$$f(x) = \begin{cases} \frac{1 - e^{-\mu_t}}{e^{-\mu_t} - 0.1} x & x < e^{-\mu_t} - 0.1 \\ e^{1 - \mu_t} & e^{-\mu_t} - 0.1 < x < e^{-\mu_t} \\ e^{1 - x} & x \geq e^{-\mu_t} \end{cases} \quad (3)$$

其中,将 $e^{-\mu_t}$ 设置为分类边界,IoU 小于 $e^{-\mu_t}$ 的取负样本,大于 $e^{-\mu_t}$ 的取正样本。这样设计加权函数还有一种好处:权重分配会按照 μ_t 值的大小而自动调节。由于巧妙地设置了两个单调性相反的指数函数,当 μ_t 较小时,说明模型检测的效果并不好,同时划分难易程度的阈值较低,更多的样本反而被划分为容易样本,而高权值是悬停在困难样本处的,导致在需要重视简单样本时模型却还在关注困难样本。而当以 $e^{-\mu_t}$ 为阈值时,随着 μ_t 变小,阈值会对应的右移,优先考虑将简单样本处理好。

尤其需要注意的是,当 μ_t 变大时,说明模型的检测水

平正在提升,所以逐渐需要将权重转移到困难样本上。随着 μ_t 变大,模型处理能力提升,关注点需要从简单样本转向困难样本。而 $e^{-\mu_t}$ 左移的同时,最高权重 $e^{1-e^{-\mu_t}}$ 进一步升高,大力提高对低 IoU 检测框的权重分配,如图 5 所示。

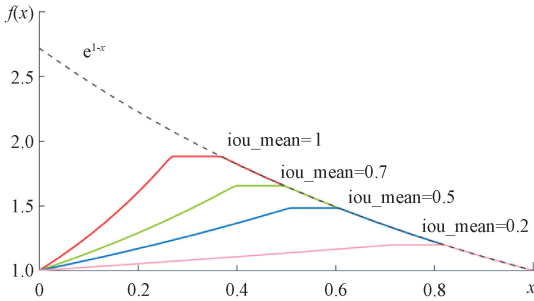


图 5 引入指数移动样本加权函数

随着模型识别的效果越来越好,模型的 IoU 均值逐渐趋近于 1,权重函数随虚线部分不断向左上方的最高点滑动,使得模型越来越专注于困难样本,这种动态的指数移动样本加权函数能覆盖模型训练的全过程,提高模型整体的检测水平。

表 2 多模型对比下的消融实验

模型	指数移动	模型大小	FLOPs	mAP@.5,.95	延时
YOLOfastestV2	否	1.09 MB	0.9 G	56.3%	357 ms
YOLO v5-Lite	否	3.17 MB	3.7 G	68.7%	485 ms
YOLO v5-Lite	是	3.17 MB	3.7 G	70.2%	485 ms
YOLO v5-s	否	14.4 MB	15.8 G	78.4%	2 875 ms
YOLO v5-s	是	14.4 MB	15.8 G	80.3%	2 875 ms

可以看到经过指数移动样本加权函数优化后,作为 YOLOv5 原模型中参数量最少的版本,YOLO v5-s 的检测精确度比 YOLO v5-Lite 高 10.1%,但由于 YOLO v5-s 推理速度花费时间过长,在需要实时检测的实际场景中帧率过低,很难达到追踪效果;YOLO-fastestV2 虽然平均每帧的推理速度比 YOLO v5-Lite 快了 128 ms,但准确率低了 13.9%,这会造成大面积的误识别。权衡准确率与推理速度,YOLO v5-Lite 最适合作为快速无人机检测模型部署到树莓派 4B 中。

在模型的训练过程中,还有两个比较重要的参数:查准率(Precision)和查全率(Recall)^[22]。查准率是模型检测出的所有 UAV 检测框中,其本身真实为 UAV 的比例,而查全率可以理解为被模型正确检测出来的无人机数量占样本中所有真实无人机数量的比例。为了更好的分析这两个概念,引入混淆矩阵,如表 3 所示。

表 3 中,真正例(true positive, TP)表示预测结果为正例,真实结果也为正例;假正例(false positive, FP)表示预测结果为正例,真实结果为反例;假负例(false negative, FN)表示预测结果为反例,真实结果为正例;真负例(true

2.3 模型性能评估

为检验优化模型的性能,本文使用表 1 中的多种评价指标对模型进行性能评估。

表 1 评价指标

评价指标	参数说明
mAP	反应模型的检测精度
模型大小	反映模型占用内存,是评估模型轻量化程度的重要指标
推理速度	用推理一张图片所用的平均时长来反映模型的整体性能
计算量	FLOPs,即模型所需浮点运算次数,代表计算时间复杂度

模型训练次数设定为 300 轮,在数据集上对模型进行了消融实验。考虑到树莓派对模型的轻量化需求,表 2 给出了和 YOLO v5-Lite 参数量最为接近的 YOLO-fastestV2^[21]与 YOLO v5-s 在添加指数移动样本加权函数后的对比结果。

表 3 二分类混淆矩阵

混淆矩阵	无人机	背景
真值	无人机	预测结果为正例, 预测结果为反例, 真实结果也为正例/TP 真实结果也为正例/FN
	背景	预测结果为正例, 预测结果为反例, 真实结果为反例/FP 真实结果为反例/TN

negative, TN)表示预测结果为反例,真实结果为反例。按照本文检测无人机目标的需求,既需要模型尽可能准确的识别出无人机,又需要尽可能多的将所有 UAV 样本检测出来,对模型的查全率与查准率都有较高的要求,如式(4)和(5)所示。

$$Precision = \frac{TP}{TP + FP} \quad (4)$$

$$Recall = \frac{TP}{TP + FN} \quad (5)$$

根据不同的置信度排序所有的预测框,分别计算出对应的一组 Precision 和 Recall 的值,就能绘制出 P-R 曲线。

曲线与坐标轴所围面积越大,模型性能就更好。图 6 给出了优化模型训练过程中的 Precision-Recall 曲线图,当 Precision 和 Recall 的值相等时,达到平衡点,也就是在图 6 中的红点处,模型的查准率和查全率二者最为均衡,经优化过后,最终的 Precision 为 97.8%,Recall 为 96.3%。

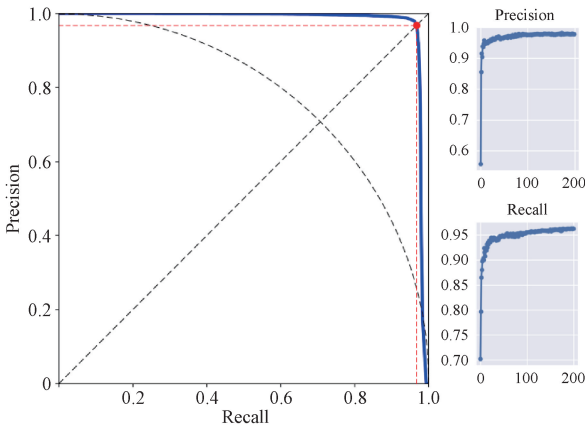


图 6 P-R 曲线

3 嵌入式平台实现与测试

3.1 硬件设备介绍

本文用于进行模型部署和最终边缘计算的设备为树莓派(Raspberry Pi) 4B。内存为 8 GB,中央处理器(central processing unit, CPU)为 64 位,选用官方的 Raspbian 操作系统,小巧轻便、搭配了丰富的 GPIO 接口,支持多种深度学习框架的配置与安装。除树莓派以外所用到的硬件设备如表 4 所示,平台总体连接如图 7 所示。

表 4 硬件设备及参数

硬件设备	参数说明
PCA9685	舵机扩展芯片,可以通过 I2C 控制 16 路舵机
SG90 舵机	堵转扭矩为 1.3 kg·cm,精度为 180°±3°的舵机
摄像头	分辨率为 480P,USB2.0 直插
供电模块	支持多路 5 V、12 V 供电
大疆 mini2	测试系统追踪效果的无人机

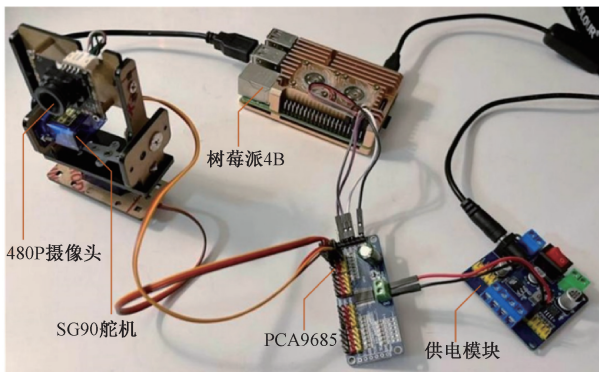


图 7 平台硬件连接

相较于李学宾^[8]和陈佳林等^[10]采用 Intel 神经计算棒辅助计算的方案,本研究更希望模型的计算更依赖于树莓派本身。在不增加任何硬件用于加速计算的情况下,本文设计的硬件平台单帧检测速度仅需 485 ms,比同样不使用硬件加速设备的方案^[9]快了 5.9 倍,且本平台整体的硬件价格只有使用硬件加速设备方案的一半。硬件通讯通过杜邦线和 USB 数据线连接,为后续研究增加多组舵机控制和传感器留足了拓展接口,真正做到了易安装、低成本、易拓展、高灵活性的部署优势。

3.2 模型移植

在搭建硬件环境时,树莓派 Raspbian 系统对环境的安装方式与 Linux 系统相似,且只需要安装 OpenCV 和 NCNN,无其他任何依赖、简单方便。其中,NCNN 是一个为手机端极致制优化的高性能神经网络前向计算框架,其设计初衷是为了深度神经网络在手机端的部署和应用,在手机端 CPU 的运行速度比目前所有的开源框架都快。本文经过试验后发现,NCNN 在树莓派中进行神经网络的前向推理速度也远快于 Pytorch 框架,选择 NCNN 进行模型的部署。

在电脑端训练好的 YOLO v5-Lite 优化模型为 Pytorch 框架,先将其用导出工具转换为开放神经网络交换(open neural network exchange, ONNX)格式^[23],用于在各种深度学习训练和推理框架中转换。再进入 NCNN 工具包,调整锚框大小以及缩放因子,利用工具包中的 ONNX2NCNN(ONNX to NCNN)把模型转换成 NCNN 的格式文件。利用 NCNN 团队提供的 C++源文件,编译完成即移植成功。

3.3 二轴云台效果实现

由于需要实现对无人机的动态追踪,本文所采用的方法是:每识别到一个 UAV,就去计算检测框的中心点在摄像头画面中心点 X 轴和 Y 轴上的偏差,以控制舵机偏转的速度和角度,如图 8 所示。

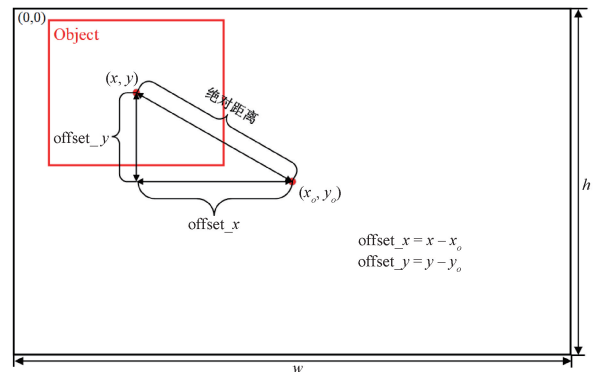


图 8 偏移量归一化

将偏移量进行归一化,分别除以图像宽度和高度:

$$\bar{x} = \frac{x - x_0}{2x_0}, \bar{y} = \frac{y - y_0}{2y_0} \quad (6)$$

其中, (x_0, y_0) 是画面中心点坐标, (x, y) 是 UAV 检测框的中心点坐标, 经式(6)归一化后就获得了 UAV 在画面中心 x 、 y 轴上的归一化偏移量 \bar{x} 和 \bar{y} , 这里引入归一化的原因是为了将变量控制在 $[-1, 1]$ 之间。因为在实验中发现, 如果命令舵机一次性需要旋转的角度过大, 容易出现舵机在 0 到 180° 之间来快速乱摆的现象, 非常难以控制, 须对偏移量做归一化处理。

在舵机控制上, 采用比例、积分和微分 (proportion integral derivative, PID) 控制算法^[24] 中的比例控制机制。当偏移量越大, 显然需要舵机转动的角度增量也就越大, 采用简单实用的线性关系去描述二者之间的关系, 将偏移量转化为角度增量, 原理如图 9 所示。

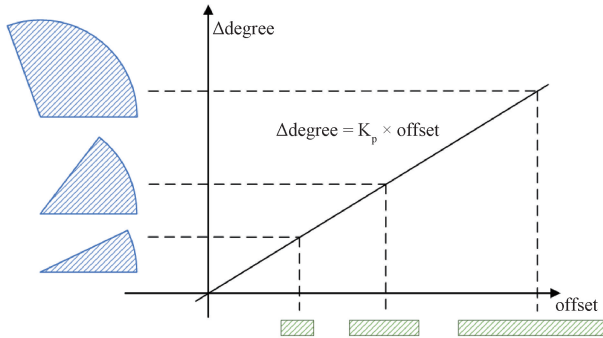


图9 偏移量映射角度增量示意图

按照映射关系, 只需调整斜率 K_p , 就能调整每次舵机摆动幅度的大小。如果舵机转动幅度过大, 一下子丢失视野中的无人机目标, 说明 K_p 太大, 需要调小; 如果舵机转动缓慢, 完全跟不上 UAV 的速度, 就说明 K_p 太小, 需要增大; 若舵机转动方向与无人机方向相反, 则需要调整斜率 K_p 为负数。通过不断的实验与尝试, 最终获得适合的 K_p 。

找到合适的 K_p 后, 还可能会碰到舵机已经能较好地追踪目标但是在中心点小幅度的来回摆动。这是因为在目标值附近, 由于实时视频的检测不可能每一帧都能来得及处理, 模型是每处理好当前帧后再获取最新的一帧, 那么在处理过程中造成的短暂的延迟就会导致舵机已经转动过了中心点而检测框并未识别做出响应, 这样在中心点往往会出现小幅度来回摆动的问题。

本文添加“死区”, 在靠近中心点附近人为设定一个小范围, 当 UAV 检测框的中心点落在这个死区中的时候, 就不给舵机发送转动信号。“死区”的范围设置太大会导致只有在画面边界, 舵机才会转动; 反之太小, 则导致很难抑制抖动效果。经过反复的实验, 将死区反映到偏移量上, 最适合的范围是 $[-0.1, 0.1]$, 也就是偏移量在这个范围内时就不转动舵机, 超出这个范围就让舵机追踪角度增量转动。

3.4 实验与测试

为了最真实地测试系统的追踪效果, 采用携带 UAV

实飞进行检验。在操纵 UAV 飞行的同时, 打开树莓派 4B 运行优化模型, 模型操纵舵机镜头自动追踪 UAV 移动^[25]。与此同时用拍摄舵机的追踪情况, 由于本文设计的嵌入式算法在检测到 UAV 目标框时, 会通过舵机转动将画面中心点偏移至检测框中心点, 所以理论上 UAV 应该始终保持在画面镜头中心区域。当画面镜头中丢失目标或检测不到 UAV 时, 判定追踪失效。本次测试采用大疆 mini 2 UAV 为追踪目标^[26], 其具体参数如表 5 所示。

表5 无人机参量

	数值
尺寸	245 mm×289 mm×56 mm (长×宽×高, 含桨叶)
最大续航里程	15.7 km
最大上升速度	5 m/s(运动档)
最大下降速度	3.5 m/s(运动档)
水平飞行速度	16 m/s(运动档)
最长飞行时间	31 min(无风环境 17 km/h 匀速飞行)
最大旋转角速度	130°/s(运动档)

接下来分别给出 UAV 在蓝天、高楼建筑、植物密集三种场景下不同高度、不同距离和不同光线环境等多种情况下的检测结果。

图 10 为在无遮挡的天空中, 当 UAV 垂直高度处于 25 m 以下、水平距离 8 m 以内时, 识别效果良好, 且置信度都维持在 82% 以上; 而当 UAV 高度超出 25 m 后, 识别准确率就会稍有下降; 检测的临界值在水平距离 8 m, 高度 36 m 时出现, 此时 UAV 在摄像头中极小, 模型已经几乎识别不到 UAV 的存在, 此时检测置信度为 62%。

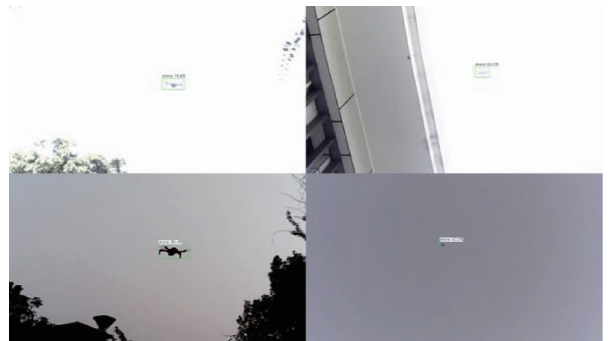


图10 天空环境(无遮挡)

而 UAV 在黄昏或者阴天等光线环境较暗的情况中飞行时, 识别效果并未受到影响, 如图 10 所示。相对于肉眼, 数码镜头的对比度更高, 此时天空背景与无人机反差较大, 对识别结果有积极的影响。当亮度进一步降低, 尤其是 UAV 在进行夜间飞行的时候, 此时若无外部光源, 本文的动态追踪系统就暂时失去作用, 后续需增加夜视功能的摄像头和传感器。

图 11 表示在建筑背景下,无论背景中建筑的远近,系统对 UAV 的识别效果都保持在不错的水准。然而需要注意的是,当 UAV 与建筑背景颜色相似时(如图 11 右下角所示),容易丢失目标,不过在实际使用中,只要 UAV 飞出颜色相同的背景,就能被模型重新捕获,所以在建筑背景下,本文的动态追踪系统适应性较好,在大部分情况下都能准确识别出 UAV。



图 11 城市建筑群密集环境

图 12 中的模型在较为杂乱的植物背景下,性能出现较大的下降。首先,在跟踪快速 UAV 轨迹时,容易将树木的枝杈误识别成 UAV,这是由于树木边缘的枝杈结构与 UAV 的结构相似,且团簇的树叶也与旋转的 UAV 桨叶相似,对模型的检测结果影响很大。其次,在高亮度时,UAV 在植物的枝杈处也难以被识别到,这是由于在高亮环境下,UAV 容易被看成像是枝杈中空缺的部分。因此,植物背景对模型检测的影响最大,是最容易误检的情况。



图 12 植物密集环境

测试的过程中,舵机云台响应迅速,在不出现误检的情况下,能一直保持对无人机的追踪转向,当 UAV 水平飞行速度不超过 10 m/s 时(UAV 处于 Normal 档位),系统始终能维持 UAV 处于画面中心区域。通过对动态追踪系统的反复实测发现,相较于水平飞行,本文的系统对垂直飞行的 UAV 有着更快的反应速度和更高的识别准确率。图 13 为舵机追踪效果图,可以看到当无人机以 8 m/s 的速度近距离飞行(左图)和 10 m/s 的速度远距离飞行(右图)时都能取得不错的跟踪效果。

表 6 汇总了所有情况下的 UAV 识别结果,涵盖了白

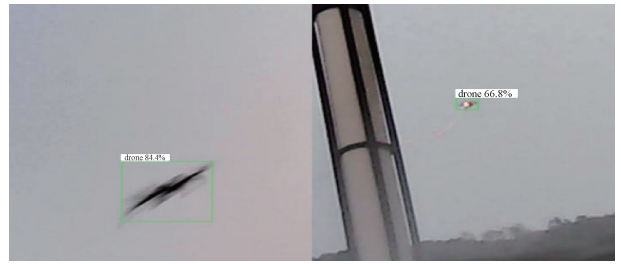


图 13 无人机动态追踪效果

天 UAV 进行飞行作业的所有情况,证明模型在不同情况下的普适性。

表 6 不同情况识别准确率

场景	远距离	暗光	建筑	植物	阴影	整体
样本数量	26	20	18	23	17	104
正确识别数	17	15	13	15	10	70
识别准确率/%	65.4	75.0	72.2	65.2	58.9	65.4

综上,本文的 UAV 动态追踪系统在中高空和高楼建筑背景中的识别效果良好,系统能做到实时追踪检测,且对光照要求不高,昏暗和高亮的环境对系统的影响微乎其微,实际测试中的整体部署准确率为 65.4%。

而在三种常见环境中,暗光环境下的植物背景对系统的影响最大,性能下降最多,检测效果下降了 7%,最主要的是容易将树木的边缘枝杈识别成 UAV,导致准确率下降。本文研发的 UAV 动态追踪平台在所有场景中均能将推理速度保持在 2.1 FPS 以上,比起凌晨团队^[9]和 Yu 团队^[15]的模型,更符合实际应用背景,在不借助任何外部硬件加强算力的同时,部署轻便、效率更高。

4 结 论

本文在理论基础和实际测试的基础上,结合边缘计算和深度学习的框架。在 YOLO v5-Lite 的损失函数中引入了样本加权函数,辅以指数滑动平均的思想,提高困难样本的权重比例,合理规划模型训练损失的流向,提高模型的检测水平。结合嵌入式应用,将模型移植到嵌入式终端设备上,解决了部署过程的繁琐、检测速度较低的难题,通过模型识别传达参数控制舵机和摄像头完成对 UAV 的追踪。

经过训练优化过后的 YOLOv5-Lite 模型内存占用 3.17 MB,mAP@.5:.95 识别精度为 70.2%,在树莓派上的平均推理延时为 485 ms,对比经典的轻量化目标检测模型 YOLO v5s,模型压缩 77.9%,在仅损失 8.2%的精度下,速度提升了 5.9 倍,在实时场景下检测效率更高;对比 YOLO-fastest V2,在精度方面提高 13.9%的同时,单帧检测仅多用时 128 ms,优势明显。

在 UAV 通用场景中(如天空、建筑和植被背景)进行多种距离、方位和亮度下的测试,模型的平均识别准确率

达到了 65.4%。在快速动态追踪时,对速度在 8 m/s 以内的无人机,模型均能很好的捕捉并且追踪无人机控制舵机偏转,保持无人机在视频画面的中央位置。在树莓派 4B 这样算力受限的设备上完成了深度学习目标检测任务,比其他方案极大地节省了硬件成本,具有易安装、低成本、易拓展和高灵活性的特点,完善了无人机监管措施,具有良好的应用前景。

实际的 UAV 飞行环境更为复杂,需要综合考虑更多的因素。因此在本文研究的基础上,未来的改进方向为:考虑 UAV 夜间执行航拍任务的场景越来越多,考虑在模型处理前引入图像处理,增加夜视或者红外模块,完善在更多场景中对 UAV 的检测与识别。考虑到成本和部署的问题,本文中选用的舵机精度和旋转角度略有遗憾,负责采集的摄像头分辨率也有待提高,在未来的工作中,考虑加入能全角度方位旋转的舵机来控制云台的转向,并在舵机上电后保持匀速缓慢水平旋转,加入高分辨率摄像头让系统保持警戒,一经监测到 UAV 非法飞行,立刻向远程控制端人员发送警报并提供方位信息。为了模型的展示效果,本文未对平台进行封装处理。后续的使用中可以用亚克力面板将平台构造为多层叠加的结构,便于携带的同时对拓展和部署更加友好。

参考文献

- XINGGUO T, GAOMING Z. Research on surface defect detection technology of wind turbine blade based on UAV image[J]. *Instrumentation*, 2022, 9(1): 41-48.
- 樊博,高玮玮,单明陶,等.融合注意力机制与重影特征映射的无人机交通场景目标轻量级语义分割[J]. *电子测量与仪器学报*, 2023, 37(3): 21-28.
- KAINZ O, DOPIRIAK M, MICHALKO M, et al. Traffic monitoring from the perspective of an unmanned aerial vehicle[J]. *Applied Sciences*, 2022, 12(16): 7966.
- SCHULZKE M. Drone proliferation and the challenge of regulating dual-use technologies[J]. *International Studies Review*, 2019, 21(3): 497-517.
- NAKAMURA H, KAJIKAWA Y. Regulation and innovation: How should small unmanned aerial vehicles be regulated[J]. *Technological Forecasting and Social Change*, 2018, 128: 262-274.
- XIA Y, BATTAR R, NAGI R. Controlling a fleet of unmanned aerial vehicles to collect uncertain information in a threat environment[J]. *Operations Research*, 2017, 65(3): 674-692.
- 赵纯,杨志家,谢闯,等.基于树莓派的 EtherCAT 主站设计与性能分析[J]. *电子测量技术*, 2022, 45(20): 37-42.
- 李学宾.基于树莓派的实时目标检测[D].武汉:华中科技大学,2020.
- 冷晨.基于树莓派嵌入式交通标志智能识别系统[D].上海:东华大学,2023.
- 陈佳林,和青,李云波,等.基于边缘计算和深度学习的园区目标检测方法[J]. *电子技术与软件工程*, 2020(16): 146-148.
- BERNABÉ S, GONZÁLEZ C, FERNÁNDEZ A, et al. Portability and acceleration of deep learning inferences to detect rapid earthquake damage from VHR remote sensing images using Intel OpenVINO toolkit[J]. *IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing*, 2021, 14: 6906-6915.
- 章挺,余飞鸿.基于 FPGA 的多视频接口的紫外成像系统设计[J]. *电子测量技术*, 2021, 44(1): 103-109.
- 李成勇,王莎,陈成瑞.基于 OpenCV 的人脸识别系统设计与实现[J]. *国外电子测量技术*, 2021, 40(11): 168-172.
- CHENG Z, HUANG R, QIAN R, et al. A lightweight crop pest detection method based on convolutional neural networks[J]. *Applied Sciences*, 2022, 12(15): 7378.
- 高嘉,刘涛,王显峰,等. TWRD-Net: 一种用于曳引钢丝绳缺陷的实时检测网络算法[J]. *仪器仪表学报*, 2023, 44(6): 223-235.
- YU Z P, HUANG H B, CHEN W, et al. YOLO-face v2: A scale and occlusion aware face detector[J]. *ArXiv Preprint*, 2022, ArXiv:2208.02019.
- SUKPARUNGSEE S, AREEPONG Y, TABORAN R. Exponentially weighted moving average-moving average charts for monitoring the process mean[J]. *PLOS One*, 2020, 15(2): e0228208.
- AL HAJJA Q A. Leveraging ShuffleNet transfer learning to enhance handwritten character recognition[J]. *Gene Expression Patterns*, 2022, 45: 119263.
- CHEN Z, YANG J, CHEN L, et al. Garbage classification system based on improved ShuffleNet v2[J]. *Resources, Conservation and Recycling*, 2022, 178: 106090.
- LIAO D, CUI Z, ZHANG X, et al. Surface defect detection and classification of Si3N4 turbine blades based on convolutional neural network and YOLO v5[J]. *Advances in Mechanical Engineering*, 2022, 14(2): 16878132221081580.
- ZHANG H, XU D, CHENG D, et al. An improved lightweight YOLO-Fastest V2 for engineering vehicle recognition fusing location enhancement and adaptive label assignment[J]. *IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing*, 2023, 16: 2450-2461.
- WILLIAMS C K I. The effect of class imbalance on precision-recall curves[J]. *Neural Computation*, 2021, 33(4): 853-857.
- CHAUDHARI P, CHAUDHARI S, CHUDASAMA R, et al. End-to-end deep learning inference with CMSSW via ONNX using docker[J]. *ArXiv Preprint*, 2023, ArXiv:2309.14254.
- ZHANG H, YUAN X. An improved particle swarm algorithm to optimize PID neural network for pressure control strategy of managed pressure drilling[J]. *Neural Computing and Applications*, 2020, 32: 1581-1592.
- CHANG B, YANG R, GUO C, et al. A new application of optimized random forest algorithms in intelligent fault location of rudders[J]. *IEEE Access*, 2019, 7: 94276-94283.
- HAHN F, VALLE S, RENDÓN R, et al. Mango fruit fly trap detection using different wireless communications[J]. *Agronomy*, 2023, 13(7): 1736.

作者简介

陈浩安,硕士研究生,主要研究方向为深度学习和航空智能优化。

E-mail:202212490571@nuist.edu.cn

李晖(通信作者)教授,主要研究方向为星地互联网和 6G 移动通信。

E-mail:hitlihui1112@163.com