

# 基于嵌入式平台的航拍目标智能识别

田祥瑞<sup>1</sup>, 贾茆钧<sup>1</sup>, 罗欣<sup>1</sup>, 尹婕<sup>1</sup>, 徐鹏<sup>2</sup>

(1. 南京航空航天大学 自动化学院, 南京 210016;

2. 中国电子科技南湖研究院, 浙江嘉兴 314001)

**摘要:** 基于多旋翼无人机实现目标识别具有成本低、灵活性高的优点, 能够对近地低空目标进行高强度监测, 在国防军事领域和民用领域具有巨大的应用前景; 但无人机机载计算机常使用功耗小、重量轻、可靠性高的嵌入式设备, 该类设备算力有限, 难以实时运行现有深度学习目标识别算法, 因此研究深度学习航拍小目标识别技术在嵌入式设备中实时运行有重要意义; 基于YOLOv4设计了适用于无人机俯视小目标的轻量化网络, 并基于BN层 $\gamma$ 系数对网络进行剪枝, 采用了TensorRT对算法进行硬件加速; 同时, 制作了小型军用目标数据集, 基于该数据集, 在机载嵌入式运算平台上对原始YOLOv4算法和改进的算法分别进行了测试, 改进算法与原YOLOv4相比, 准确率提升了2.3%, 速度提升了3.3倍。

**关键词:** 目标识别; YOLOv4; 深度学习; 无人机; 网络轻量化

## Target Intelligent Recognition of Aerial Photography Images Based on Embedded Platform

TIAN Xiangrui<sup>1</sup>, JIA Yinjun<sup>1</sup>, LUO Xin<sup>1</sup>, YIN Jie<sup>1</sup>, XU Peng<sup>2</sup>

(1. School of Automation Engineering, Nanjing University of Aeronautics and Astronautics, Nanjing 210016, China;

2. China Academy of Electronic and Information Technology, Jiaxing 314001, China)

**Abstract:** Target recognition and tracking based on multi-rotor UAVs have the advantages of low cost and high flexibility, It can carry out the high-intensity monitoring of low-ground and low-altitude targets. It has a huge application prospect in the national defense and military field and civilian field. Embedded devices with low power consumption, light weight, and high reliability are often used in UAV airborne computers. The computing power of such devices is limited, and the existing target recognition algorithms based on deep learning are difficult to run on the devices in real time. Therefore, it is of great significance to research on the technology of aerial small targets recognition based on deep learning and run in real time in embedded equipment. On the basis of YOLOv4, a lightweight backbone network which is suitable for overlooking small targets is improved and designed, and the network is pruned based on the coefficient  $\gamma$  of the BN layer, and the TensorRT is used to accelerate the algorithm of hardware. Besides, an image dataset of small military target is produced, and it is used to test the original YOLOv4 and improved algorithms on the airborne embedded computing platform. Compared with the original YOLOv4, the accuracy of the improved algorithm is increased by 2.3%, and the speed by 3.3 times.

**Keywords:** target recognition; YOLOv4; deep learning; UAV; network lightweight

## 0 引言

无人机具有成本低、灵活性高等优点, 通过无人机航拍, 可以弥补卫星和载人航空遥感技术对近地低空小目标进行侦测监控的不足<sup>[1]</sup>。因此无人机在民用领域和军事领域得到了广泛应用。在军事领域, 无人机智能识别可以用于对地面军事目标进行侦察跟踪, 也可以负载武器对目标进行自主打击。无人机通过识别算法发现目标后, 保持安全距离对目标进行隐蔽跟踪侦察, 和传统的使用车辆或人员进行跟踪的方式相比, 无人机可以实现低成本、零伤亡的智能识别。在民用领域, 融合目标识别算法后, 无人机

可以用于监测农作物生长情况, 抢险救灾, 对被困人员进行搜救, 加快救援速度, 在罪犯逃逸时也可以使用无人机进行追踪搜捕等, 具有非常好的应用前景。深度学习作为机器学习、人工智能的拓展方向, 通过模仿人类思想, 自适应处理数据, 探寻数据信息的内在规律、表述形式等, 为系统运行提供数据支撑。在此背景下, 目标识别技术的实现可依靠深度学习技术, 对目标进行精细化辨别, 提高目标识别与处理的精确性, 为系统操作提供可靠性依据。基于深度学习的目标识别技术已经成为当下研究热点, 但基于深度学习的方法对硬件计算能力要求高, 由于无人机带负载能力有限, 无人机机载计算机通常采用功耗低、可

收稿日期: 2022-08-11; 修回日期: 2022-08-25。

基金项目: 国家自然科学基金(61973160, 62073161); 江苏省自然科学基金(BK20210298)。

作者简介: 田祥瑞(1986-), 男, 山东成武人, 博士, 讲师, 主要从事机器人智能感知、无人机协同方向的研究。

引用格式: 田祥瑞, 贾茆钧, 罗欣, 等. 基于嵌入式平台的航拍目标智能识别[J]. 计算机测量与控制, 2022, 30(11): 153-160.

靠性高的嵌入式设备<sup>[2]</sup>。常见的无人机机载计算机如树莓派、NVIDIA Jetson Nano、NVIDIA Jetson TX2 等设备。其中，性能最强的 NVIDIA Jetson TX2 包含两个 CPU，配有 256 个 NVIDIA CUDA 核心的 GPU，主频最高 1.12 GHz，而搭载 NVIDIA GeForce GTX 1080 GPU 的台式服务器，就拥有 256 个 CUDA 核心，主频最高 2.8 GHz。因此，嵌入式平台性能远远低于实验室中的服务器，对深度学习算法推理速度有直接的影响<sup>[3]</sup>。现有算法大都在实验室中以达到最高准确率为目的，而在嵌入式平台上直接使用现有深度学习网络进行快速识别实时性较差，需要对网络进行轻量化改进以提高推理速度。

虽然传统的目标检测算法运算速度快，如 SIFT、SURF 和 HOG 等，但需要人工设计特征，将设计的特征在图像上以滑动窗口的方式，进行特征提取，然后对每类物体单独训练浅层分类器，完成目标的识别，这样的方式导致传统算法检测精度和适应性较差<sup>[4]</sup>。2012 年 Hinton 教授所带领的团队利用卷积神经网络 (convolutional neural networks, CNN) 设计了 AlexNet 网络<sup>[5]</sup>，在 2012 年的 ImageNet 竞赛中取得冠军，CNN 遂成为热点研究方向。目前，CNN 在图像分类、目标识别、目标跟踪等计算机视觉领域取得了重大应用<sup>[6]</sup>，其基本原理是利用图像卷积进行深层次特征的提取，进而实现目标分类与定位。基于深度学习的目标检测算法主要分为两大类，一类是先通过 Selective Search、Edge Boxes<sup>[7]</sup>等算法进行候选区域的搜索，再进行特征提取与目标检测，代表算法有 R-CNN、Fast R-CNN 及 Faster R-CNN<sup>[8-10]</sup>等，该类算法精度较高，但速度慢，难以应用于无人机嵌入式计算平台进行实时目标检测。另一类是直接进行特征提取来实现目标的分类与边界框回归，代表算法有 YOLO (you only look once, YOLO)<sup>[11]</sup>系列算法及 SSD<sup>[12]</sup>算法等，该类算法速度相比两阶段算法得到了提升，与对实时性要求较高的无人机目标检测较为契合。目前的目标识别在人脸识别、行人检测这些面向自然场景图像的任务中，应用已经非常成熟。但由于近地低空的无人机成像视角不同，无人机机载计算机算力低，无人机图像中的目标尺度变化大，且小目标的比例远高于自然场景图像，直接将现有算法应用于无人机航拍目标识别效果较差。同时现有算法应用无人机进行俯视小目标进行侦测时，存在精度不高、在嵌入式平台上运行速度慢等缺点。因此，研究适用于无人机的小目标识别算法有着重要意义。

本文首先介绍了 YOLO 系列算法基本原理<sup>[13]</sup>，然后针对在嵌入式平台上的实时运行，对 YOLOv4 进行了网络优化分析与改进，在机载嵌入式平台上进行了改进算法的验证。

## 1 YOLO 系列算法原理

两阶段算法由于需要先生成候选区域，使得算法实时性差，为改善该问题，基于回归理念，对图像目标位置和类别直接进行回归预测。YOLOv1 直接将输入图片分为  $S \times$

$S$  的小块，每个小块预测  $B$  个边界框 (bounding box, BBox)，采用 BBox 的置信度 *confidence* 反映模型对该 BBox 的预测的准确率，计算公式如式 (1)，其中， $Pr (Object)$  代表是否存在目标，若存在，则 *confidence* 为真实值 (*truth*) 和预测值 (*pred*) 的交并比 (IOU)，若不存在，则置信度为 0。

$$confidence = Pr(Object) * IOU_{pred}^{truth} \quad (1)$$

在推理阶段，每个 BBox 都要预测目标位置、置信度和  $C$  个类别信息，输出  $S \times S (5 * B + C)$  的张量。类别信息与置信度相乘，按式 (2) 计算得分 *csc*，将得分低于设置阈值的结果直接滤除，对剩余的预测结果进行非极大值抑制 (NMS) 处理，得到识别结果。

$$csc = Pr(Class_i) * IOU_{pred}^{truth} \quad (2)$$

在深度学习中，最重要工作之一是对损失函数的设计，为方便优化，YOLOv1 采用误差平方和作为损失函数，主要含有坐标损失和分类损失两大部分，如式 (3) 所示。其中， $\lambda_{coord}$ 、 $\lambda_{noobj}$  是超参数，用于控制不同项权重，YOLOv1 中  $\lambda_{coord} = 5$ ， $\lambda_{noobj} = 0.5$ ， $I_{ij}^{obj}$  表示网格  $i$  中第  $j$  个对象， $x_i$ 、 $y_i$ 、 $w_i$ 、 $h_i$ 、 $C_i$ ，分表代表了预测的中心宽高及类别。

$$Loss = \lambda_{coord} \sum_{i=0}^S \sum_{j=0}^B I_{ij}^{obj} [(x_i - \hat{x}_i)^2 + (y_i - \hat{y}_i)^2] + \lambda_{coord} \sum_{i=0}^S \sum_{j=0}^B I_{ij}^{obj} [(\sqrt{w_i} - \sqrt{\hat{w}_i})^2 + (\sqrt{h_i} - \sqrt{\hat{h}_i})^2] + \sum_{i=0}^S \sum_{j=0}^B I_{ij}^{obj} (C_i - \hat{C}_i)^2 + \lambda_{noobj} \sum_{i=0}^S \sum_{j=0}^B I_{ij}^{noobj} (C_i - \hat{C}_i)^2 + \sum_{j=0}^S I_i^{obj} \sum_{c \in classes} (p_i(c) - \hat{p}_i(c))^2 \quad (3)$$

$$\begin{cases} b_x = \sigma(t_x) + c_x \\ b_y = \sigma(t_y) + c_y \\ b_w = p_w e^{t_w} \\ b_h = p_h e^{t_h} \end{cases} \quad (4)$$

YOLOv2 设计了一个新的骨干网络结构 Darknet-19，通过前后特征融合，增强了网络对小目标的识别能力<sup>[14]</sup>。借鉴 Faster R-CNN 的 anchor 机制，使用 Kmeans 聚类算法得到大小合适的 anchor。YOLOv2 将输入图像提取为  $13 \times 13$  大小的特征图，对特征图的每个网格进行预测 BBox，如图 1 所示，每个 BBox 预测边界框的位置信息 ( $t_x$ ,  $t_y$ ,  $t_w$ ,  $t_h$ ) 和置信度  $t_o$ ， $t_x$ ,  $t_y$  经 Sigmoid 函数映射到 0 到 1 之间，将预测中心固定在 cell 内，使得模型更加稳定， $c_x$ ,  $c_y$  表示该 cell 距离特征图左上角坐标的距离， $p_w$ ,  $p_h$  代表 anchor 的宽高， $b_x$ ,  $b_y$ ,  $b_w$ ,  $b_h$  代表最后基于 anchor 得到的预测结果。计算公式如 (4) 所示，式中  $\sigma$  代表 Sigmoid 函数。

YOLOv3 借鉴深度残差网络和特征图金字塔的思想，使用了 FPN<sup>[15]</sup>，用不同尺度的特征图来检测目标，设计了主干网络 Darknet-53，Darknet-53 比 ResNet-101<sup>[16]</sup> 准确率提高了 0.1%，速度提高了 1.47 倍<sup>[17]</sup>，在理论上，YOLOv3 没有太多的改进。

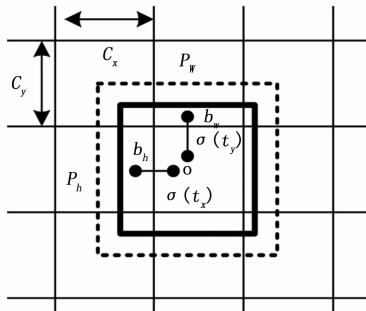


图 1 基于 anchor 的位置预测示意图

YOLOv4 由 Alexey Bochkovskiy 在 2020 年改进完成, 是一个端到端的一阶段网络, 其网络结构如图 2 所示。YOLOv4 设计了新的骨干网络 CSPDarknet53, 该骨干网络与 Darknet53 相比, 在每个 Resblock 中, 增加了新的跨层连接, 提高了网络准确率。与 YOLOv3 一样, 融合了前后特征图的信息, 并用多个尺度特征图来识别目标, 如图 2 中 PANet 部分所示, 最后, 增加了空间金字塔 (spatial pyramid pooling, SPP) 模块, 扩展了模型的感受野<sup>[18]</sup>, 进一步提高识别准确率。在数据增强方面采用了 CutMix, 提出了 Mosaic 等数据增强方法, CutMix 通过将两张图按随机比例进行混合, 可有效改善密集目标识别中的目标重叠导致识别率低的问题<sup>[19]</sup>; Mosaic 则通过对 4 张图片按照随机缩放、随机裁剪和随机排布的方式进行拼接, 增强小目标识别效。此外, 融合了目标检测领域最新的定位损失函数 CloU loss, 有效指导网络训练<sup>[20]</sup>, 采用了 Mish 激活函数, 增强了网络的非线性表达能力<sup>[21]</sup>。融合多种方法的 YOLOv4 在精度上得到了极大的提高, 但由于网络模型更大, 对硬件的要求也进一步提高, 在嵌入式平台上存在推理速度慢, 计算资源占用率高的问题。

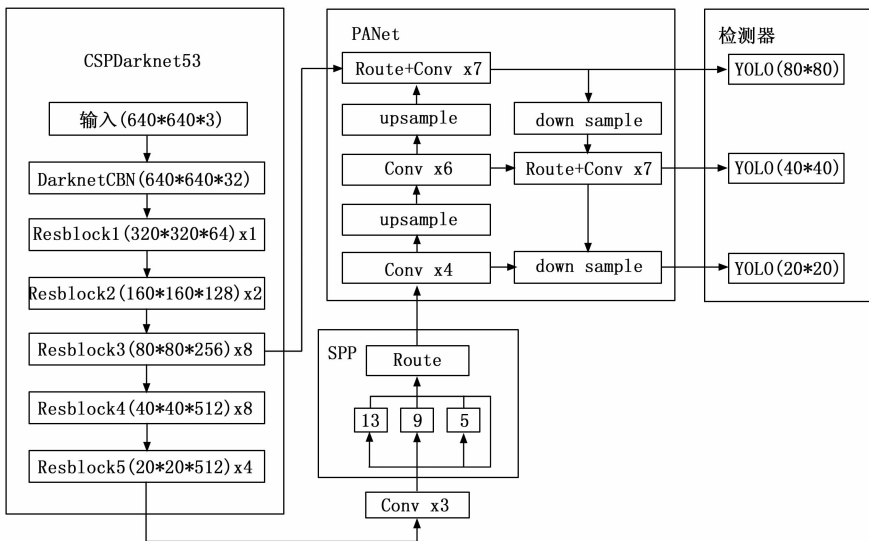


图 2 YOLOv4 网络结构图

## 2 YOLOv4 骨干网络优化设计

YOLOv4 主要应用于平视视角下常规尺寸的目标识别, 无人机俯视目标较小, 使用 YOLOv4 直接检测, 存在浪费计算资源用于检测大目标、容易丢失小目标细节信息的问题。分别从网络深度和网络宽度两个方面, 改进设计适用于小目标识别的 YOLOv4 骨干网络, 以提高网络准确率和速度。

### 2.1 网络轻量化分析

卷积神经网络通过使用卷积核对图像特征进行提取, 在深度学习中, 卷积核的参数通过训练得到, 将得到合适参数的卷积核对输入图像进行处理。如图 3 所示, 对相同输入, 不同参数的卷积核会产生不同的特征图 (feature map), 有的卷积核类似高通滤波器, 通过锐化后可以增强细节信息的图像; 有的卷积核类似低通滤波器, 通过模糊图像, 减少图像细节突出重要信息, 有的对水平或竖直的纹理信息敏感, 而有的对颜色信息敏感。

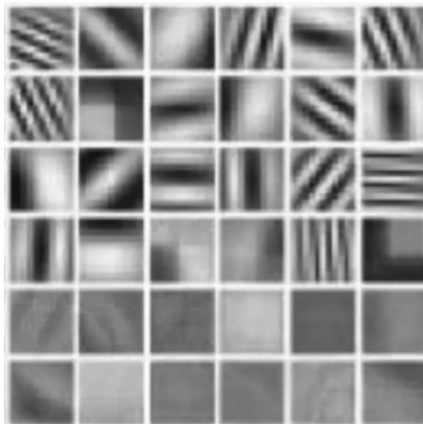


图 3 不同卷积核提取的特征图

单层卷积核的数量称为卷积层的通道数, 也叫网络宽度, 网络的宽度越宽可以提取到更多的特征, 帮助算法更好的进行决策。任何勒贝格可积函数不能被宽度小于  $n$  的 ReLU (rectifiedLinear units, ReLU) 网络逼近<sup>[22]</sup>, 如式 (5) 所示:

$$\int_R |f(x)| dx < \infty \quad (5)$$

因此, 网络需要保证一定的宽度, 才能实现对复杂问题的求解, 但随着网络宽度增加计算量呈指数增加, 使得算法实时性变差。所以, 需要对特定问题, 设计与其复杂度相匹配的网络, 以实现速度与精度的平衡。

单层网络虽然可以提取到丰富的特征, 但无法完成信息的整合, 并且由于单层神经网络不具备处理非线性问题的

能力, 因此需要采用深度神经网络。理论上, 一个两层的网络可以拟合任意有界连续函数, 但该变换过于复杂, 使得网络难以收敛, 且需要非常宽的网络, 所以, 需要通过加深网络深度, 将复杂问题逐层分解, 降低每一层的变换难度, 使网络更加容易训练。研究发现深度网络中的浅层卷积神经网络先对图像简单的边缘等信息进行学习, 中间层则学习到了简单形状, 高层学习目标的具体形状, 深度越深, 可以提取到更高级的语义信息。但由于通过卷积层不断的对信息的提炼, 会造成位置等细节信息的丢失, 尤其在针对小目标时, 网络过深会导致目标定位信息丢失, 从而难以进行目标识别。所以, 网络并不是越深越好, 同样需要针对特定问题设计网络的深度。

深度和宽度会影响最终的模型大小和推理速度, 可根据公式 (6) 进行测算:

$$\begin{cases} \text{params} = C_o \times (k^2 \times C_i + 1) \\ \text{FLOPs} = 2 \times C_i \times k^2 \times C_o \times W \times H \end{cases} \quad (6)$$

其中:  $C_o$ 、 $C_i$  分别为输出输入特征图通道数,  $k$  为卷积核大小,  $W$ 、 $H$  为特征图宽与高, 各卷积层的参数量 ( $\text{params}$ ) 决定了模型的大小, 推理速度则主要由各卷积层的计算量 ( $\text{FLOPs}$ ) 决定。因此, 针对特定应用场景, 通过对网络深度与宽度进行优化设计, 可在确保精度的前提下提升网络的运算速度。

### 2.2 针对小目标识别的网络深度改进

无人机通过俯视拍摄目标, 由于距离较远, 使得目标在图像中所占像素较少, 若采用较深的卷积神经网络虽然可提取到更高级的语义信息, 但会造成小目标空间信息的丢失。YOLOv4 在提取图像的过程中, 先将  $640 \times 640$  的原始图像提取为  $20 \times 20$  的特征图, 即缩小了 32 倍, 网络难以识别  $32 \times 32$  像素及以下尺寸的目标, 对近似尺寸的小目标即使能进行类别预测, 也不能对目标位置进行准确回归。

针对目标细节信息丢失导致算法准确率降低的问题, 首先, 对图 2 中原始骨干网络进行分析, 图 2 中原始骨干网络最后两个 Resblock 主要用于中型尺寸目标的特征提取, 这是造成小目标信息丢失的主要原因, 所以将残差块 Resblock4 舍去。其次, 由于原网络中路径聚合网络 (PANnet) 中特征图经三次降采样再上采样, 该过程虽最终将特征图恢复为原大小, 但在恢复过程中采用线性插值完成, 使得特征图变模糊, 影响网络检测小目标, 因此将检测器适当提前。然后, 为提高识别精度, 在网络中加入 SPP 模块, 并在之后使用 10 个卷积层进行特征的提取。最后, 通过三个尺度的检测器完成检测, 改进后的网络如图 4 所示。

将改进后的网络在公开数据集 VisDrone<sup>[23]</sup> 上进行训练, VisDrone 由天津大学的 AISKYE 团队进行搜集制作, 是一个包含了目标识别、目标跟踪、人群计数等多个方向的数据集。该数据集标注了超过 260 万个边界框, 其中目标识别方向共有 10209 张图像, 包含了 10 个类别, 分别为: 行人、人、汽车、面包车、公共汽车、卡车、摩托车、自

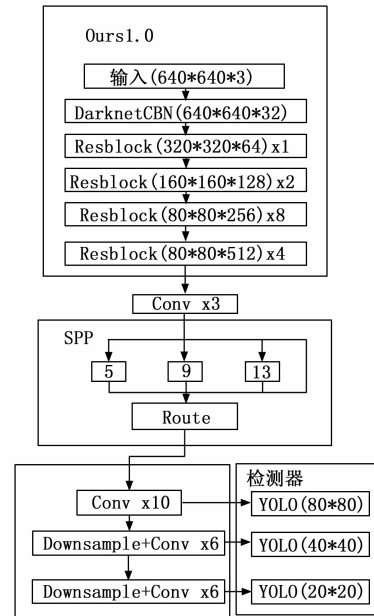


图 4 深度改进后的网络结构

行车、遮阳篷三轮车和三轮车。采用的训练平台为 Inter Core i7-8700CPU, NVIDIA GeForce GT 1080, Ubuntu18.04 TS。网络框架基于 Darknet。采用 CUDA、cuDNN 对计算进行加速, 安装 OpenCV 环境实现数据增强。搭建好软件环境后, 对神经网络超参数进行设置。首先, 有关权重更新方式, 批量梯度下降每次更新都需要把所有样本输入, 得到误差后对网络权重进行一次更新, 兼顾了所有样本的情况, 可以得到全局最优解, 但当训练样本很多的时候, 由于每一次迭代都需要对所有样本进行计算, 训练会很慢。小批量梯度下降对多个样本, 只更新一次权重, 这样的更新方式可以使网络大致朝着全局最优的方向迭代, 也可以解决批量梯度下降训练慢的问题。本文选择了小批量算法作为权重更新方式, 批量 (batch) 设置为 64, 分解系数 (subdivision) 设置为 64。采用动量法, 将动量法中的 momentum 设置为 0.9。其次, 对训练样本进行数据增强, 数据增强可以增加训练样本, 得到更好的训练效果, 同时还能防止模型过拟合问题。主要包含把图片随机旋转  $-90$  度到  $+90$  来增加训练样本的数量, 将 saturation 设置为 1.5, exposure 设置为 1.5, hue 设置为 0.1, 分别对原始图像做饱和度、曝光度和色调进行随机调整, 采用了 mosaic 数据增强, 该方法通过将多张图片进行裁剪再拼接到一张图上作为训练数据, 有利于丰富目标背景和小目标检测。最后, 使用预训练权重对网络进行训练, 学习率设置为 0.001 3, 并将最大训练轮数设置为 20 000, 学习率分别在第 12 000、18 000 轮时降低为原来的十分之一。训练完成后, 在大疆机载计算机 Manifold-2G 上对算法进行测试, Manifold-2G 的处理器是英伟达研发的 Jetson TX2, 拥有 256 个 CUDA 计算核心的 GPU, CPU 复杂部分由两个

ARM v8 64 位 CPU 集群组成。软件系统为 Ubuntu 16.04, 安装了 CUDA9、OpenCV3.0。本文测试都是在 Manifold-2G 上进行, 后续不再加以说明。对算法基于默认参数进行测试, 深度改进测试结果如表 1 所示。

表 1 深度改进实验结果

| 骨干网络         | 计算量/BFLOPs | 参数量/MB | mAP   | FPS |
|--------------|------------|--------|-------|-----|
| CSPDarknet53 | 141        | 256.3  | 34.1% | 2.1 |
| Ours1.0      | 105        | 37.1   | 37.3% | 2.6 |

测试结果表明在针对俯视角小目标进行网络改进后, 准确率提高了 3.2%, 网络参数量减少为原网络的 15%, 表明了算法改进的有效性。但其计算量仅降低 36 BFLOPs, 改进算法运行速度提升依然较小, 需要在网络宽度方面继续改进。

### 2.3 网络宽度改进

在 2.2 节对网络深度的改进中, 计算量仅降低 36 BFLOPs, 改进算法运行速度提升较小。通过分析, 这是由于将检测器关联层提前, 在降低模型大小的同时, 使得通道数较多的卷积层用于处理较大的特征图。根据公式 (7) 可知这将使模型的计算量大大增加, 使得网络加速并不明显。由于俯视角目标特征相对一致, 在保证一定精度的同时, 可通过适当减少网络宽度, 实现网络的加速。

宽度改进后的网络结构如图 5 所示, 首先将图 4 中最后的 Resblock 通道数由 512 降低为 256, 可降低该 Resblock 的 4 倍计算量; 其次, 因为 SPP 模块使通道数增大四倍, 导致 SPP 模块后的卷积层计算量和参数量增大。因此, 在 SPP 模块之前先通过一次下采样 (downsample), 将特征图大小由原来的 80 \* 80 变为 40 \* 40, 使得 downsample 到上采样 (upsample) 之间的计算量降低 4 倍; 最后, 通过一次上采样将特征图恢复为原大小, 由于下采样操作仅将特征图大小变为 40 \* 40, 并不影响小目标识别, 且有效减少了计算量和参数量。

将改进后的网络再次在数据集 VisDrone 上进行了相关测试, 测试结果如表 2 所示。从测试结果看出, 改进后的骨干网络在无人机视角小目标识别任务中, 参数量减少为原网络的 10.4%, 精度提高了 2.7%, 推理速度是原网络的 1.8 倍。但由于网络改进时仅通过对整个 Resblock 进行通道数设计, 网络存在冗余, 仍需对单个卷积层通道数进行进一步精简。

表 2 宽度改进实验结果

| 骨干网络         | 计算量/BFLOPs | 参数量/MB | mAP   | FPS |
|--------------|------------|--------|-------|-----|
| CSPDarknet53 | 141        | 256.3  | 34.1% | 2.1 |
| Ours2.0      | 65         | 26.8   | 36.8% | 3.8 |

### 3 网络剪枝与量化加速

无人机带负载能力有限, 且由于无人机带由多种传感器, 使得无人机周围电磁环境复杂, 另一方面, 为保证系

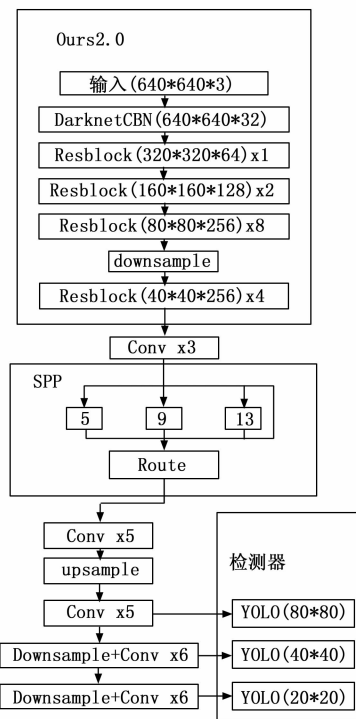


图 5 宽度改进后网络结构

统的精简, 通常使用无人机的电池对机载计算机进行供电, 所以, 无人机机载计算机通常采用功耗低、可靠性高的嵌入式设备, 该类设备算力较低。本文采用了 DJI Manifold-2G 作为机载计算平台, 该设备搭载了 NVIDIA jetson TX2 核心, 与搭载 NVIDIA GeForce GT 1080 的台式服务器相比, 1080 拥有 2560 个 CUDA 核心, CUDA 核心数反映了硬件的算力, 对深度学习算法推理速度有直接的影响, TX2 仅仅拥有 256 个 CUDA 核心, 因此, 直接在 TX2 等嵌入式平台上进行使用深度学习网络是不切实际的, 需要对网络进行剪枝量化, 以提高推理速度。采用对网络的剪枝与量化以提高网络实时性进行了相关工作。

#### 3.1 网络剪枝

为提高算法在嵌入式设备上运行的实时性, 通过对模型进行剪枝可进一步提高推理速度。基于 BN 层  $\gamma$  系数对冗余通道进行剪枝<sup>[24]</sup>, BN 层将网络的每一层的输入做标准化处理, 使数据分布向整体数据的均值方差方向转移, 从而使得网络更容易初始化, 并可以加速网络训练<sup>[25]</sup>。

$$\begin{cases}
 \mu_{\beta} = \frac{1}{m} \sum_{i=1}^m x_i \cdot y_i \\
 \sigma_{\beta}^2 = \frac{1}{m} \sum_{i=1}^m (x_i - \mu_{\beta})^2 \\
 \hat{x}_i = \frac{x_i - \mu_{\beta}}{\sqrt{\sigma_{\beta}^2 + \epsilon}} \\
 y_i = \gamma \hat{x}_i + \beta = BN_{\gamma, \beta}(x_i)
 \end{cases} \quad (7)$$

BN 层通过两个可学习的参数保留特征和提高网络的灵活性, 其算法过程如公式 (7) 所示, 式中  $x_i$ 、 $y_i$  为输入输

出样本,  $\mu_B$ 、 $\sigma_B^2$  为批处理数据均值与方差,  $\gamma$ 、 $\beta$  为可学习的缩放因子和移位因子。由公式 (7) 可知, 当  $\gamma$  较小时, 该通道对后续模块只有很小的输入, 即该模块重要性较低。因此可通过稀疏化后的系数对通道的重要性进行评估, 从而去除冗余通道。据此原理对设计的网络进行剪枝, 首先, 对网络模型 BN 层参数使用 L1 正则化进行稀疏, 对稀疏化后模型的值进行排序, 采用参数量全局剪去 54%、单层保留至少 10% 的策略, 对网络进行通道剪枝, 剪枝后的网络测试结果如表 3 所示。

表 3 网络剪枝实验结果

| 网络  | 计算量/BFLOPs | 参数量/MB | mAP   | FPS |
|-----|------------|--------|-------|-----|
| 剪枝前 | 65         | 26.8   | 36.8% | 3.8 |
| 剪枝后 | 28         | 8.6    | 35.9% | 7.3 |

通过剪除冗余通道, 网络参数量降低为原来的 32.1%, 计算量降低为原本的 4.3%, 网络仅在损失 0.9% 的准确率的情况下, 网络推理速度提高到剪枝前的 1.9 倍。

### 3.2 使用 TensorRT 量化加速

为充分利用硬件平台的计算资源, 使用 TensorRT 对网络进行硬件加速。首先, 因为模型推理时主要由 GPU 启动 CUDA 进行运算操作, CUDA 核心的启动对每一层输入输出张量的读写操作造成大量的时间和计算资源浪费, 带来了内存带宽的瓶颈问题。因此, 通过层间融合将横向的卷积层、偏置层、激活函数层融合为一个 CBN 层, 可加速推理过程。另一方面, 由于推理过程不需要反向传播, 在推理过程中, 相比使用 32 位浮点数的存储的权重文件, 使用 16 位浮点数存储的权重文件对模型精度可以达到近似的精度, 所以将权重文件中参数量化为 16 位浮点数, 减小计算复杂度从而提高推理速度。

模型测试结果如表 4 所示, 对使用 TensorRT 加速后, 由于权重文件格式与原 YOLOv4 权重文件格式不同, 不对计算量与参数量进行对比。采用 TensorRT 加速后, 网络精度仅降低 0.3%, 但推理速度提高了 26%, 极大地增强了算法在嵌入式平台上的实时性能。

表 4 硬件加速实验结果

| 网络  | 计算量/BFLOPs | 参数量/MB | mAP   | FPS |
|-----|------------|--------|-------|-----|
| 加速前 | 16         | 3.4    | 35.9% | 7.3 |
| 加速后 |            |        | 35.6% | 9.2 |

## 4 军事目标数据集构建与算法测试

### 4.1 军事目标数据集的构建

与传统算法人为设计特征不同, 深度学习算法可以自主学习特征, 自主学习一般需要大量的监督数据。对同一个深度学习算法, 用于指导学习的数据越多、数据质量越好, 训练出来的网络性能越好, 所以数据集在目标检测中具有非常重要的作用。另一方面, 军事目标由于安全性和

保密性, 鲜有开源数据集, 因此为了验证改进算法对军用目标的有效性, 解决军用目标数据集缺失的问题, 制作了一个小型的军用目标数据集。

通过网上搜索符合俯视军事目标的图片, 主要来源为各大搜索引擎, 由于无人机视角下的军用目标图片在角度、目标大小以及目标种类方面都有着限制, 军用目标的特殊性和保密性要求, 使得在国内外搜索引擎上直接检索得到的符合要求的图片较少, 远远满足不了数据集的要求。为此, 从公开实战演习的视频中, 截取符合要求的图片。选取士兵、坦克、装甲车、军舰、导弹发射车五类目标作为数据集制作对象, 最终收集到符合要求的图片 3582 张。对搜集到的图片进行重命名、格式修改等初始化处理, 然后, 为方便 YOLOv4 的训练, 采用 PASCAL VOC 标准制作数据集, PASCAL VOC 在目标识别检测领域是一套标准化数据集格式, 为数据集的制作提供了标准的文件格式。按照 VOC 标准使用 Labelimg 完成数据集的标注, 标注结果如图 6 所示。

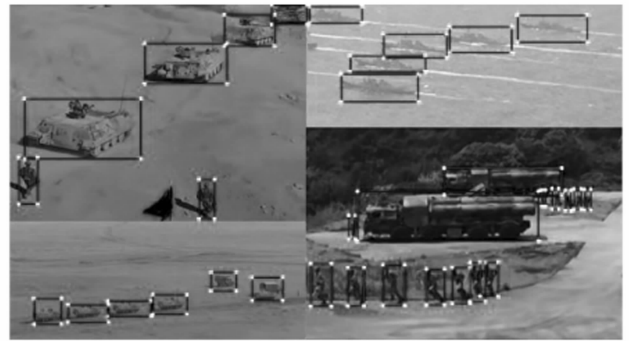


图 6 军事数据集标注

最后, 对标注好的数据集进行训练集、验证集和测试集之间的比例分配。由于数据集规模不是太大, 测试集不宜分配太多数据, 将大部分数据分配到训练和验证集中, 最终分配结果为训练集: 验证集: 测试集=7.2: 1.8: 1。

### 4.2 军用目标数据集测试分析

采用标注的数据集分别对原 YOLOv4 和改进算法进行训练, 训练采用的电脑配置为 Inter Core i7-8700CPU, NVIDIA GeForce GT 1080, 系统版本 Ubuntu18.04, 深度学习框架采用 Darknet, 并将其移植到嵌入式系统 Manifold-2G 上进行测试, 实验效果如图 7 所示。

实验结果如表 5 所示, 通过对 YOLOv4 进行了适用于小目标的改进, 并采用了剪枝量化等方法, 改进算法在精度上提高 2.3% 的同时, 推理速度提高了 3.3 倍。展现了改进算法对军事航拍小目标识别的有效性, 提升了航拍目标识别的精度与速度。单类目标识别精度如表 6 所示, 在单类目标识别测试结果中, 改进算法与原算法相比, 对坦克、装甲车、军舰、导弹发射车、士兵 5 类目标的识别精度分别提高了 4%、2%、1%、2%、6%, 对士兵识别精度提升



图 7 改进算法对军用目标的识别效果

最大, 而士兵相对其他类别尺寸较小, 表明针对小目标识别的算法改进是有效的。

表 5 算法对比结果

| 算法     | 参数量/MB | mAP   | FPS |
|--------|--------|-------|-----|
| YOLOv4 | 256.1  | 74.5% | 2.1 |
| 改进算法   |        | 76.8% | 9.1 |

表 6 单类目标识别精度

| 精度<br>算法 | 精度  |     |     |       |     |
|----------|-----|-----|-----|-------|-----|
|          | 坦克  | 装甲车 | 军舰  | 导弹发射车 | 士兵  |
| YOLOv4   | 76% | 79% | 69% | 88%   | 58% |
| 改进算法     | 80% | 81% | 70% | 90%   | 64% |

## 5 结束语

嵌入式平台算力资源有限, 难以运行大型的深度网络, 且现有目标识别网络并不是针对航拍小目标的实时识别设计的。本文基于现有 YOLOv4 算法, 对其在网络深度、宽度方面进行了骨干网络的优化设计, 并采用剪枝和硬件加速技术进一步提升网络的推理速度。在制作的军用目标数据集上, 改进算法比原算法准确率提高了 2.3%, 推理速度提高了 4.3 倍, 验证了采用的网络优化改进的有效性。虽然在嵌入式平台上提高了算法的推理速度, 但依然达不到实时性, 仍需硬件和算法方面进行改进。由于 YOLOv4 采用锚框对目标进行粗略的预测, 大多数锚框都是负样本, 造成训练中正负样本不均衡, 未来可以基于无锚框的方式对小目标进行识别, 提高算法效率。此外, 通过引入自注意力机制, 对可能存在目标的位置进行重点处理, 从而进一步提升算法的准确率和实时性。

### 参考文献:

[1] 朱俊辉. 基于机器视觉的无人机跟踪系统研究 [D]. 长春: 吉林大学, 2020.  
 [2] 刘文, 陆小锋, 毛建华, 等. 基于机载计算机的无人机智能

巡检方案 [J]. 计算机测量与控制, 2022, 30 (7): 181-186.  
 [3] 赵文政. 面向无人机平台与低功耗嵌入式设备的对地目标检测与跟踪算法研究 [D]. 天津: 天津理工大学, 2020.  
 [4] 杨剑锋, 王润民, 何璇, 等. 基于 FCN 的多方向自然场景文字检测方法 [J]. 计算机工程与应用, 2020, 56 (2): 164-170.  
 [5] KRIZHEVSKY A, SUTSKEVER I, HINTON G E. ImageNet classification with deep convolutional neural networks [J]. Communications of the ACM, 2017, 60 (6): 84-90.  
 [6] 江波, 屈若崑, 李彦冬, 等. 基于深度学习的无人机航拍目标检测研究综述 [J]. 航空学报, 2021, 42 (4): 137-151.  
 [7] 王春哲. 基于候选区域目标检测的关键技术研究 [D]. 北京: 中国科学院大学 (中国科学院国家空间科学中心), 2020.  
 [8] GIRSHICK R, DONAHUE J, DARRELL T, et al. Rich feature hierarchies for accurate object detection and semantic segmentation [C] // Proceedings of the IEEE conference on computer vision and pattern recognition, IEEE, 2014: 580-587.  
 [9] GIRSHICK R. Fast r-cnn [C] // Proceedings of the IEEE international conference on Computer Vision, IEEE, 2015: 1440-1448.  
 [10] REN S, HE K, GIRSHICK R, et al. Faster R-CNN: Towards Real-Time Object Detection with Region Proposal Networks [J]. IEEE Transactions on Pattern Analysis and Machine Intelligence, 2017, 39 (6): 1137-1149.  
 [11] REDMON J, DIVVALA S, GIRSHICK R, et al. You only look once: Unified, real-time object detection [C] // Proceedings of the IEEE conference on Computer Vision and Pattern Recognition, IEEE, 2016: 779-788.  
 [12] LIU W, ANGUÉLOV D, ERHAN D, et al. SSD: Single shot multibox detector [C] // European Conference on Computer Vision, Springer, Cham, 2016: 21-37.  
 [13] BOCHKOVSKIY A, WANG C Y, LIAO H Y M. Yolov4: Optimal speed and accuracy of object detection [C] // IEEE Conference on Computer Vision and Pattern Recognition, IEEE, 2020, 57 (5): 9-12.  
 [15] LIN T Y, DOLLÁR P, GIRSHICK R, et al. Feature pyramid networks for object detection [C] // Proceedings of the IEEE conference on Computer Vision and Pattern Recognition, IEEE, 2017: 2117-2125.  
 [16] HE K, ZHANG X, REN S, et al. Deep residual learning for image recognition [C] // Proceedings of the IEEE conference on Computer Vision and Pattern Recognition, IEEE, 2016: 770-778.  
 [17] REDMON J, FARHADI A. YOLOv3: An Incremental Improvement [C] // IEEE Conference on Computer Vision and Pattern Recognition, IEEE, 2018, 87 (8): 101-104.  
 [18] HE K, ZHANG X, REN S, et al. Spatial Pyramid Pooling in Deep Convolutional Networks for Visual Recognition [J]. IEEE Transactions on Pattern Analysis and Machine Intelligence, 2014, 37 (9): 1904-1916.  
 [19] YUN S, HAN D, OH S J, et al. Cutmix: Regularization

strategy to train strong classifiers with localizable features [C] // Proceedings of the IEEE/CVF International Conference on Computer Vision, 2019: 6023 - 6032.

- [20] ZHENG Z, WANG P, LIU W, et al. Distance-IoU loss: Faster and better learning for bounding box regression [C] // Proceedings of the AAAI Conference on Artificial Intelligence, 2020, 34 (07): 12993 - 13000.
- [21] MISRA D. Mish: a self regularized non-monotonic activation function [J/OL]. arXiv preprint, 2019: 1 - 14. [2020 - 8 - 13]. <https://arxiv.org/pdf/1908.08681v3.pdf>.
- [22] DETTMERS T. Understanding convolution in deep learning [J]. Retrieved March, 2015, 25: 1451 - 1460.

- [23] ZHU P, DU D, Wen L, et al. VisDrone-VID2019: The vision meets drone object detection in video challenge results [C] // Proceedings of the IEEE/CVF International Conference on Computer Vision Workshops, 2019: 227 - 235.
- [24] LIU Z, LI J, SHEN Z, et al. Learning efficient convolutional networks through network slimming [C] // Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, IEEE, 2017: 2736 - 2744.
- [25] IOFFE S, SZEGEDY C. Batch Normalization: Accelerating Deep Network Training by Reducing Internal Covariate Shift [C] // Proceedings of the International Conference on Machine Learning, 2015: 448 - 456.

(上接第 132 页)

标位置,都能够自行计算出减速点。在不同的工况下,该系统具有一定的自适应能力。

## 5 结束语

本文通过对步进电机原有的梯形加减速算法分析,对其进行改进,改进后的梯形加减速算法通过时间来确定减速点。实验结果表明,改进后的梯形加减速算法在加速阶段和减速阶段理想曲线和实际曲线误差较小。改进后的步进电机梯形加减速控制能够保证精度和可靠性,实时性高,代码实现简单。在步进电机的位置改变后,能够自动计算出减速点,具有一定的自适应能力,大大提高了其灵活性,提高了加工效率,对于优化裁切机运动控制系统的轨迹,具有一定的意义。现阶段该系统是开环控制,后续可以将编码器的数据反馈给 FPGA 做成闭环控制的系统,能够提高加减速控制的精度,同时可以利用 FPGA 并行处理的优点,把单轴拓展到多轴上实现裁切机的加减速控制。

## 参考文献:

- [1] 王立红. 基于 PLC 的步进电机升降速优化算法研究 [J]. 微电机, 2018, 51 (4): 73 - 75.
- [2] 闫吉. 基于 STM32 的 3D 打印步进电机控制方法研究 [J]. 制造业自动化, 2021, 43 (11): 129 - 131, 136.
- [3] 刘鹏, 罗衍涛, 陈成军, 等. 基于 FPGA 的运动控制系统加减速算法设计 [J]. 仪表技术与传感器, 2017 (6): 145 - 149.
- [4] 胡呈祖. 基于 FPGA 的梯形加减速运动控制算法 [J]. 轻工机械, 2020, 38 (6): 60 - 63.
- [5] 邱靖超. 基于 FPGA 的步进电机多轴运动控制系统设计 [D]. 太原: 中北大学, 2021.
- [6] 桑鹏, 吕建工, 彭吉龙, 等. 步进电机变速控制系统设计 [J]. 电子设计工程, 2013, 21 (22): 155 - 158.
- [7] 孙孔政, 徐抒岩, 曹小涛, 等. 基于矩频曲线的步进电机加减速控制 [J]. 微电机, 2014, 47 (1): 81 - 85.
- [8] 王邦继, 刘庆想, 周磊, 等. FPGA 在多轴步进电机控制器中的应用 [J]. 电机与控制学报, 2012, 16 (3): 78 - 82,

- 89.
- [9] QIAN F, LEI W. FPGA-based acceleration and deceleration control for CNC machine tools [C] // International Conference on Mechatronic Sciences, IEEE, 2014.
- [10] (日)坂本文著;王自强,译. 步进电机应用技术 [M]. 北京: 科学出版社, 2010.
- [11] 冯涛, 李擎, 潘月斗, 等. 步进电机梯形加减速曲线规划控制实验系统设计 [J]. 煤矿机械, 2020, 41 (7): 23 - 25.
- [12] AMRITHA R, ANIL A, ANJU S S, et al. FPGA Implementation of PWM Based Sinusoidal Drive Generation for Stepper Motor [C] // 2020 IEEE International Conference for Innovation in Technology (INOCON), IEEE, 2020.
- [13] 熊金, 刘伟. 基于指数曲线加减速控制中减速点的判断与计算 [J]. 自动化技术与应用, 2013, 32 (1): 7 - 9, 24.
- [14] 冯涛, 李擎, 潘月斗, 等. 步进电机梯形加减速曲线规划控制实验系统设计 [J]. 煤矿机械, 2020, 41 (7): 23 - 25.
- [15] YU J, WANG Y, KANG H, et al. A control system of three-axis stepper motor based on the FPGA [C] // International Conference on Mechatronic Sciences, IEEE, 2014.
- [16] 陈汝倬. 基于 FPGA 的步进电机控制器设计 [J]. 科技信息, 2009 (8): 437 - 439.
- [17] 于晶, 杨晓慧, 黄勇. 基于 FPGA 的按键消抖电路设计方法的研究 [J]. 电子设计工程, 2011, 19 (22): 1 - 3.
- [18] 邱靖超, 刘新妹, 殷俊龄, 等. 一种基于 FPGA 的多通道步进电机控制系统的设计 [J]. 国外电子测量技术, 2021, 40 (4): 72 - 77.
- [19] 李立珺. 基于 FPGA 的除法器算法研究 [J]. 科技信息, 2013 (5): 82 - 104.
- [20] 鲁伟, 刘士兴, 孙操, 等. 基于 STM32 的增量式编码器测速设计及实验验证 [J]. 计算机测量与控制, 2019, 27 (12): 259 - 263.
- [21] 夏斯泰. 基于 FPGA 的测速模块的设计与实现 [D]. 南京航空航天大学, 2013.