

基于 Jetson TX2 及轻量级深度学习的自主型无人机 机载处理系统设计

钱林钧¹ 夏凡凡² 肖志峰^{1*} 陈钊² 慕杨²

¹ 武汉大学测绘遥感信息工程国家重点实验室 湖北省 武汉市 430079

² 武汉大学遥感信息工程学院 湖北省 武汉市 430079

摘要

具备自主环境感知和机载决策的自主型无人机是无人机的发展趋势,长期以来,由于机载处理能力和算法发展水平的限制,无人机机载实时准确的目标识别一直是自主型无人机面临的一个重要挑战。本文针对此问题,提出一种利用机载 NVIDIA JETSON TX2 嵌入式计算机实现以轻量级深度学习模型为基础的目标检测方法:获取无人机在不同姿态下的目标图片,学习目标区域特征,在部署了深度学习模型 YOLO (You Look Only Once) 的 TX2 上使用 Darknet 框架进行目标检测训练,得到针对不同目标的检测分类器,获取目标在图像中的位置。为验证该方法的可行性,在“2018 年高分无人飞行器智能感知技术大赛”的线上仿真场景、线下比赛环境及实地飞行实验中进行了数字和圈的检测及车辆、行人检测,实验表明该方法在实时性和准确性上可适用于现场环境无人机机载实时目标检测,对实现自主型无人机机载智能决策有重要意义。

关键词 无人机 Jetson TX2 轻量级深度学习 目标检测

中图分类号 TP751

Design of Autonomous UAV Airborne Processing System Based On Jetson TX2 And Lightweight Deep Learning

Qian, Linjun Xia, Fanfan Xiao, Zhifeng

State Key Laboratory of Information Engineering in Surveying, Mapping and Remote Sensing, Wuhan University, Wuhan 430079, China

注:本文受高分青年创新基金 GFZX04061502 支持

ABSTRACT

Aiming at the problem how the UAV can accomplish the task to recognize targets like various numbers and circles, this paper proposes a method to use the UAV with a computer—NVIDIA JETSON TX2 and use this computer to do the lightweight deep learning to achieve the function of detecting targets. First, pictures that containing targets under different flight attitudes of the UAV are needed. Then, the features of each target are to be selected. Next, classified pre-training is done on the TX2 which is deployed a deep learning model named YOLO to get a proper classifier, so we can get the right position of targets in pictures. In order to verify the feasibility of the method, we test the simulation results of the number and circle detection effect of the offline and online environment in *High-resolution UAV intelligent sensing technology contest in 2018* and the detection effect of the vehicle and people in the captured vehicle and people video, indicating that the method has high value of application to detect targets in site environment.

Keywords: UAV; TX2; lightweight deep learning; target detection

1 引言

随着信息技术的飞跃发展, 无人机系统正在向感知外界环境以实现适应环境并自我控制的自主飞行无人机方向发展。在自主飞行的无人机领域, 对无人机下达起飞命令后, 在不对其进行进一步操控的情况下, 机载传感器可以帮助无人机精确地定位周边障碍, 感知周围景物分布状况, 以达到无人机自主控制的目的。目标检测技术作为计算机视觉领域的重要组成部分, 是实现上述场景的关键技术。[1]。但是, 由于无人机机载处理能力的限制, 机载目标检测长期依赖准确率不高, 无法进行可靠的目标识别。深度学习技术作为近年来计算机视觉领域的突破口, 可以产生准确率高的模型, 在各个领域都得到了广泛关注。在无人飞行器领

域,采用深度学习,使得机器能够对相机拍到的图片或视频进行理解,从而确定下一步行动,然后对自身下达指令。

在深度学习领域中,如何将准确率高的深度学习神经网络模型方便地部署到机载处理环境是一大难点。传统的深度学习算法参数多、计算量大、模型大,使得在固定性能的处理器的处理器上能够执行的图形运算较少。轻量级深度学习能改进这一缺点,快速构建视觉系统,提高深度学习的预算效率。论文从自主无人机和深度学习的研究现状、为实现自主无人机智能感知功能的无人机硬件架构、软件架构、基于轻量级深度学习的目标检测以及集成实验五个方面介绍基于 Jetson TX2 及轻量级深度学习的自主型无人机机载处理系统设计。

2 研究现状

无人机自主控制系统,主要包括“无人机综合环境感知技术做出信息处理,自主完成控制策略”。在此方面,欧美国家取得了较为良好的成绩,比如,美国的全球鹰以及捕食者无人机,可以在低空以及复杂的环境下进行工作并完成任务。相较而言,我国的无人机技术仍处于比较初级的发展阶段,在无人机感知处理、精确目标打击、飞行规划等方面还存在很多问题,并不算完全意义上的自主控制系统[2]。

精准的目标检测是实现无人机感知处理、自主飞行的前提,随着深度学习的发展,计算机视觉领域的目标检测技术取得了长足的进步。[1]

深度学习的概念由 Hinton 等人于 2006 年提出[3、5],其团队于 2009 年设计出基于深度学习的神经网络 AlexNet,使得深度学习在图像识别领域的精度显著提升,在此之后,googleNet、ResNet 等卷积神经网络相继被设计出来[1];2016 年,Deep Mind 公司所开发的 Alpha Go 击败围棋世界冠军李世石,见证了深度学习发展的一次飞跃[5]。深度学习发展的过程中,神经网络逐渐加深,深度学习对数据特征提取的细致程度以及数据处理的效果逐渐得到认可,以深度学习为基础的目标检测技术也得以发展。

典型的目标检测网络可以分为进行特征提取的前端和用于输入前端特征并输出图像目标定位的后端。其中,后端定位模块主要发展为“基于区域选择的深

度学习网络”和“基于卷积层的回归预测”两种定位技术。[1]“基于区域选择的深度学习网络”以 R-CNN 和 Fast R-CNN 为代表。R-CNN 的网络规模大、训练过程繁琐、占用空间大；Fast R-CNN 采用了 POIpooling 方式对 R-CNN 进行改进[6]，这使得 Fast R-CNN 在准确度和速度上较 R-CNN 有较大的提升，但未实现端到端的深度学习神经网络[4]。“基于卷积层的回归预测”的代表模型有 YOLO(You Only Look Once)系列。YOLO 不需要在定位网络中使用候选框卷积的方法，直接在卷积层中抽取相应的特征，将目标区域预测及目标类别预测整合于单个神经网络，对目标的类别和位置做一个坐标图的回归预测。YOLO 把目标检测转化为回归问题，实现了端到端的深度学习神经网络，提升了检测速度，但在精准定位方面不够有理想。

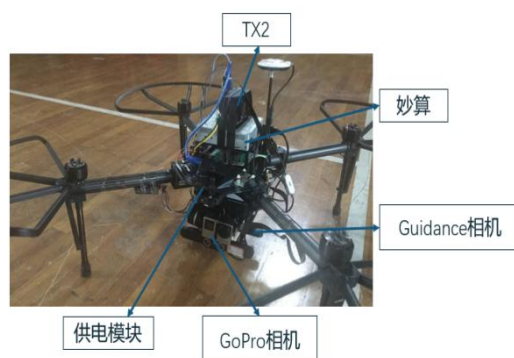
由此可见，如何权衡网络模型规模与训练速度精度的关系是深度学习领域需要进一步探索的问题[3]。网络的轻量化意味着通过更少的储存量达到或者超过现有网络的效果，是解决上述问题的研究方向。在网络轻量级化的研究中，主要的成果包括：squeezeNet 减少模型参数，以达到 AlexNet 的准确率但模型效果却只有 AlexNet 的百分之一的效果[1]；YOLOv3 采用多个尺度融合的方式做预测，并将网络扩充为 53 层，改进了 YOLO 系列不擅长检测小物体等缺点等。

3 硬件构架

机载硬件由飞行器子系统、感知子系统以及任务执行与决策子系统、机载模块总成以及其他飞行模块组成。下图所示为无人机总体硬件架构

图 1 无人机总体硬件架构

Fig.1 the overall hardware architecture of UAV



3.1 飞行器子系统

飞行器子系统由飞行平台 DJI M100 四旋翼无人机构成。DJI M100 四旋翼无人机提供飞行平台，以实现飞行功能。

3.2 感知子系统

感知子系统由云台相机以及视觉传感导航系统 DJI GuidanceSDK 组成。飞宇 mini 3d pro 云台为三轴增稳平台；GoPro Hero4 相机提供 HDMI 接口，可实时采集视频数据。DJI GuidanceSDK 可通过 USB 或者 UART 接口与 DJI M100 或其他飞行机器人系统；实时获取机体坐标系下的飞机三位速度和坐标、图像深度信息和超声波信息；实时输出机体坐标系下的 IMU 数据等传感器数据；实时输出双目摄像头的 8 比特、16 比特灰度图；通过连接 USB 接口高速率传输所有数据，并对摄像头参数进行精准控制；通过连接 UART 接口传输除图像外的所有数据，可方便地适配所有带串口的系统。

3.3 任务执行子系统

任务执行子系统由妙算以及 NVIDIA JETSON TX2 微型计算机两台计算机组成。

妙算拥有 PC 独立显卡级别的绘图能力，可让机器人实时进行复杂的图像处理，广泛应用于计算机视觉、深度学习等人工智能领域；支持 DJI M100 开发，实现订阅 Guidance 并发布数据图像的功能。下表为妙算的配置及参数。

表 1 妙算配置及参数

Table.1 the Configuration and parameters of Manifold

序号	配置	型号及参数
1	处理器	NVIDIA 4-Plus-1™ 四核心 ARM® Cortex-A15 CPU 低功耗 NVIDIA Kepler™ Geforce® 图像处理器 图像信号处理器 超低功耗音频处理器
2	内存	2 GB DDR3L RAM 16 GB eMMC 4.51 存储空间
3	网络	10/100/1000BASE-T 局域网端口

4	USB	USB 3.0 Type-A Host 端口×2 USB 2.0 Type-A Host 端口×2 Micro-B USB 接口(Host/Slave 模式) 基于 DJI 经纬 M100 飞行平台的 USB 扩展接口 (CAM_IN, CAM_OUT)
5	输入电压	14V - 26V

Jetson TX2 是一台嵌入式人工智能超级计算平台。具有质量小、性能高的特点，适合机器人、无人机、智能摄像机和便携医疗设备等智能终端设备。其“嵌入式”性质可以很好地解决这些终端要具备人工智能计算能力所要面临的带宽不足、延迟等问题。TX2 支持终端运行神经网络和深度学习过程，在执行图像分类、导航和语音识别等任务时，可在更短的时间内达到更高的精度。这使得在无人机上实现以轻量级深度学习来进行目标检测 and 智能感知成为可能。下表所示为 TX2 的配置及参数。

表 2 TX2 配置及参数

Table.2 the configuration and parameters of TX2

序号	配置	型号及参数
1	处理器	NVIDIA® Pascal™ , ARMv8 (64-bit) HMP CPU Complex
2	内存	8GB LPDDR4
3	接口	1 x USB 3.0; 1 x USB 2.0; 1 x HDMI; 1 x RJ45; 1x Micro SD; 1 x RJ-45 8P8C;
4	存储	32GB eMMC 5.1
5	电源	19V 直流电压 3A

3.4 机载模块总成和其他记载模块

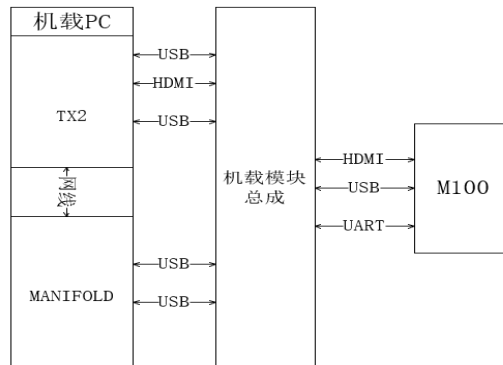
机载模块总成是机载硬件系统的中间执行处理单元，是软件系统联通交流的平台。

其他飞行模块包括供电模块等。

下图为机载硬件总体框图。

图 2 机载硬件总体框图

Fig.2 overall block diagram of onboard hardware



4 软件架构

4.1 ROS (Robot Operating System) 平台简介

整个软件架构基于 ROS 系统搭建，ROS 是一个适用于机器人的开源的元操作系统。目前 ROS 主要支持 Ubuntu 系统。它提供一些标准操作系统服务，如硬件抽象、底层设备控制、常用功能实现、进程间消息以及数据包管理等[8]。它也提供用于获取、编译、编写、和跨计算机运行代码所需的工具和库函数。

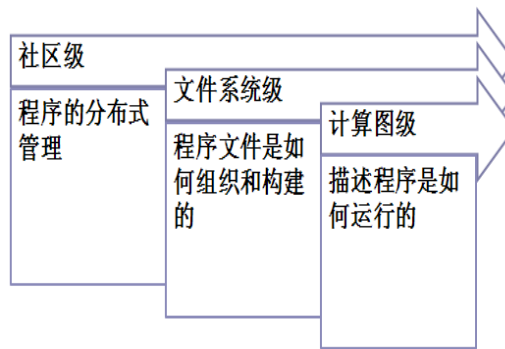
为机器人开发者提供模块化和复用性高的代码和功能上的支持是 ROS 平台的主要开发宗旨[7]。ROS 是一个分布式的进程（也就是“节点”）框架，这些进程被封装在易于被分享和发布的程序包和功能包中。ROS 也支持一种类似于代码储存库的联合系统，这个系统也可以实现工程的协作及发布。这个设计可以使一个工程的开发和实现从文件系统到用户接口完全独立决策（不受 ROS 限制）。同时，所有的工程都可以被 ROS 的基础工具整合在一起。

根据 ROS 系统代码的维护者和分布来标示，ROS 总体结构只要分为两大部分：提供分布式计算的基本工具的核心部分（main）和全球范围内的代码（universe），包括库的代码、功能代码以及应用级代码。

一般是从另一个角度对 ROS 分级的，主要分为三个级别：计算图级、文件系统级、社区级。

图 3 ROS 分级

Fig.3 the rating of ROS



4.2 大疆 DJI SDK 简介

SDK (Software Development Kit), 即软件开发工具包, 用来开发适用于某个产品或某个平台的软件, 通常由产品或平台的厂商提供给开发者使用。在这个工具包里通常包括各种各样的调试工具、API 文档、开发指南和模拟器。大疆 SDK 开发分为三种: Mobile SDK, Onboard SDK, Guidance SDK。

1. **Mobile SDK** 以大疆遥控器作为通信设备, 以高数据速率传输所有数据并精确控制摄像机参数, 可进行移动端 (安卓手机, 苹果手机) 应用开发, 连接框图及设备支持如下图, 通常都为大疆整机, 也可为组装机。
2. **Onboard SDK** 使用飞控上的串口接口, 通过数传无线或有线进行通信, 传输除图像以外的所有数据, 适用于所有使用串行通信的系统, 支持单片机、移动端或 PC 端地面站开发。
3. **Guidance SDK** 为大疆无人机 M100 搭载 Guidance 视觉系统, 该系统配备五组视觉超声波组合传感器, 采用高精度立体视觉算法, 近地面定位精度可达厘米级, 在复杂地形和高速飞行条件下均可提供定位信息, 视觉定位系统的有效高度高达 20 米。

4.3 消息体系设计

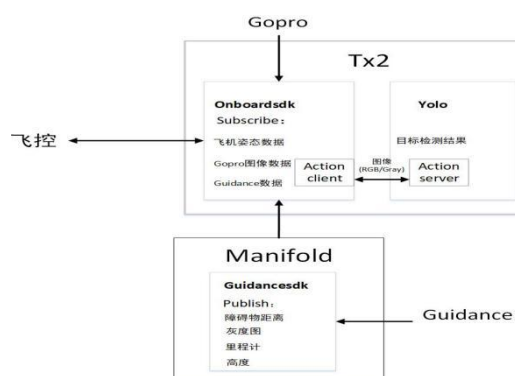
由于目标识别与深度学习, 定位和路径规划需要大量的数据处理, 采用单一卡片式电脑难以胜任, 本系统充分利用 ROS 的分布式处理框架, 首先使服务器

和大疆 M100 主控制器连接到同一个 WIFI 网络下，然后核心计算机 TX2 与妙算对相机图传消息，无人机姿态消息，Guidance 相机深度消息等进行订阅并将识别结果，图像处理结果等消息发布，来实现所有节点在节点管理器中注册并由节点管理器统一管理。达到不同主机上的节点进行有效通信的目的[15]。

基于 ROS 系统的消息订阅与发布机制如下图：

图 4 基于 ROS 系统的消息订阅与发布机制

Fig.4 message subscription and release mechanism based on ROS system



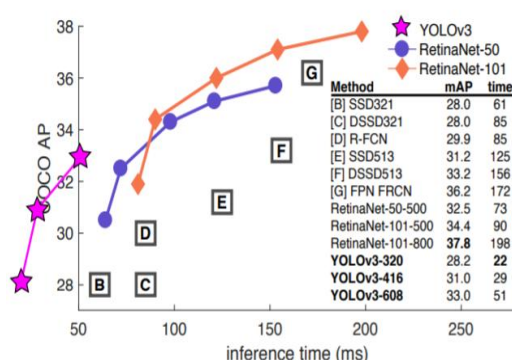
- (1) GoPro 为前视相机，帧率为 25 帧每秒，视野为宽视野。在设计中，GoPro 负责主要的目标识别数据采集工作，将图传信号发布出来并由 TX2 计算机进行订阅。
- (2) Guidance 为大疆无人机自带的视觉定位系统，负责发布无人机距离障碍物的距离，里程计消息，高度消息。
- (3) Yolo 为本系统选择的目标识别与深度学习系统，订阅 GoPro 与 Guidance 的图传消息，发布目标识别结果。

5 基于轻量级深度学习的目标检测

传统的深度学习模型具有参数很多、模型很大的缺点，轻量级深度学习在确保准确性的基础上，能同时拥有稳定性和高效性。Darknet 是轻量级深度学习模型 YOLO(You Only Look Once)网络的框架，适用于快速构建视觉系统，完成目标检测。在 titan x GPU 上，在保证检测准确率的前提下 (63.4% mAP, VOC 2007 test set)，可以达到 45fps 的检测速度[9]。

图 5 YOLO 相较于其他检测方法速度更快

Fig. 5 YOLO runs significantly faster than other detection methods with comparable performance.



论文采用轻量级深度学习模型 YOLO 方法完成目标检测功能。首先，前视 GoPro 相机获取 RGB 图像，通过 YOLO 方法，检测出图像中的数字，获取目标数字的大体方位，当无人机移动至下视 Guidance 相机视野可以覆盖所识别数字的矩形标示牌时，下视相机获取当前灰度图像，将灰度图像转化为彩色图像（其彩色图像的 RGB 三通道赋予同样的值），再采用 YOLO 方法识别目标数字，并通过标定数字的矩形框确定目标数字中心的位置。

5.1 YOLO 简介

YOLO 是一种将目标区域预测及目标类别预测整合于单个神经网络的算法，它采用分格思想，将全图划分为 $S \times S$ 个格子，每个格子负责中心在该格子的目标的检测，规定如果一个栅格内包含某个目标 ground truth 的中心点，那么在训练过程中，该栅格就仅对该目标的边界框进行回归预测[10]。这样可以一次性预检测所有格子所含目标的框格大小、定位置信度以及所有类别概率向量。

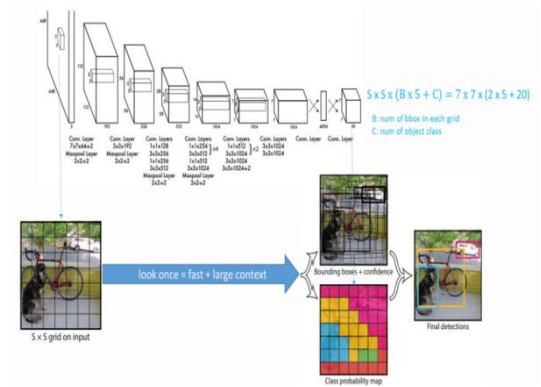
YOLO 网络结构由 24 个卷积层与 2 个全连接层构成，其中，卷积层提供图像特征，全连接层预测输出概率。网络入口为 416×416 (YOLOv2)。图片进入网络经过重定义大小(resize)，网络的输出结果为一个张量，维度为： $S \times S \times (B \times 5 + C)$ 。其中， S 为划分网格数， B 为每个网格负责目标个数， C 为类别个数。该表达式说明：

- (1) 每个小格会对应 B 个边界框，边界框的宽高范围为全图，表示以该小格为中心寻找物体的边界框位置；
- (2) 每个边界框对应一个分值，代表该处是否有物体及定位准确度；

(3) 每个小格会对应 C 个概率值，先验概率由 K-Means 算法给出，找出最大概率对应的类别 $P(\text{Class}|\text{object})$ ，并认为小格中包含该物体或者该物体的一部分。YOLO 网络示意图如下图所示：

图 6 YOLO 网络示意图

Fig.6 the picture of YOLO network



5.2 在 Jetson TX2 上部署 YOLO

Jetson TX2 是基于 NVIDIA Pascal™ 架构的 AI 单模块超级计算机，性能强大（1 TFLOPS），外形小巧，节能高效，适合无人机等智能终端设备，支持终端运行庞大的神经网络，以实现短时间、高精度的目标检测功能。在 TX2 上运行 YOLO 方法需要首先在 TX2 上部署 YOLO：

(1) 在 TX2 上部署 Darknet。

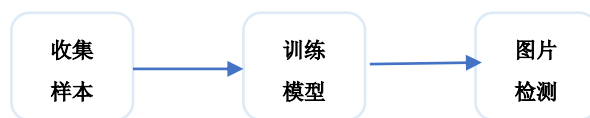
Darknet 易于安装，只有两个可选的依赖项，及支持各种图像类型的 OpenCV 和提供 GPU 计算的 CUDA。在安装 Darknet 基础系统后，可选择安装 OpenCV 和 CUDA。

(2) 部署 YOLO。

(3) 测试 YOLO-Darknet。

5.3 样本收集及模型训练

使用 YOLO 网络框架 Darknet 进行分类预训练，可以得到适合自己的分类器，此处欲得到的是区别圆圈和数字 1-9 的分类器。利用 YOLO 实现目标检测的步骤由以下流程图给出：



具体步骤如下：

(1) 收集样本

为使得训练数据与实际场景尽可能一致，我们采用前视相机对数字进行不同距离与角度的拍摄。

选取样本时，只有可能出现水平放置的数字（线上场景中为 1，2，3，10；线下场景中为 1，2，9）的样本图片数量为 200 张左右，既有可能出现水平放置的数字又有可能竖直放置的数字（线上场景中为 4，5，6，7，8，9；线下场景中为 3，4，5，6，7，8）的样本数量为水平和竖直摆放的数字图片各 150 张左右。

图 7 线上仿真场景选取样本

Fig.7 the selected samples of online environment

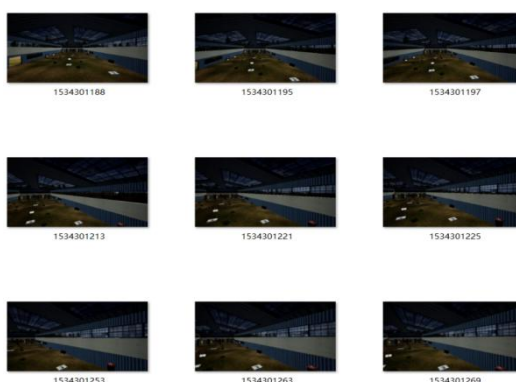


图 8 线下实地场景选取样本

Fig.8 the selected samples of offline scene



再利用 LabelImg 软件进行数据采集。首先获得与 PASCAL VOC[11]数据集文件 Annotations 中相似的 xml 标注文件，即标注文件包含了图片的名称、宽度、高度、通道数以及该图片中包含目标名称、目标在图像中的位置等信息[10]。再选取含有<name>number</name>或者<name>circle</name>的 xml 文件，将 GT 边界框转换为(xc, yc, w, h)的表示方式，使得 xc, yc, w, h 被归一化到 0~1 之间。其中，(xc, yc) 表示相关格网中心，(w, h) 表示与全图信息相关的格网的宽和高。

(2) 训练模型

数字和圆圈的数据集制作完成后，需要对 YOLO 模型进行训练。训练主机配置为：内存 12GB GDDR5X，显卡为 NVIDIA TITAN XP 台式机显卡。为了发挥主机的性能，需要开启 GPU 加速。

在训练神经网络时，学习率（Learning Rate）等参数深刻影响着神经网络的性能。学习率是一个重要的超参数，它控制着基于损失梯度调整神经网络权值的速度，是 YOLO 模型训练中重要的参数。权值与学习率的关系可以表示为：

$$\theta_1' := \theta_1 - \alpha \frac{\partial}{\partial \theta_1} J(\theta_1)$$

其中， θ_1' 为新权值， θ_1 为当前权值， α 为学习率， $\frac{\partial}{\partial \theta_1} J(\theta_1)$ 为梯度。由上式可见，学习率越大，模型收敛到局部最小（即达到最高精度）的速度越快，如果学习率过大，则有可能错过局部最优解；如果学习率过小，训练速度会变慢，找到局部最优解的时间加长，并且有可能陷入找局部最优解的循环。

关于如何确定较好的学习率，Leslie N. Smith 在 Cyclical Learning Rates for Training Neural Networks 中指出，如果先设置一个较低的学习率，然后随着训练迭代逐渐增大这个值，最终可以获得一个较好的学习率。YOLOv2 模型中，初始学习率为 0.001，迭代次数达到 40000 和 60000 时，学习率按照×0.1 的比率发生变化。学习率以及影响学习率的参数意义情况如下：

表 3 学习率以及影响学习率的参数意义

Table.3 Learning and parameters affecting learning rate

参数	意义
learning_rate=0.001	初始学习率

burn_in=1000	可能与学习率的动态变化有关
max_batches = 80200	训练的最高次数，达到该值训练停止
policy=steps	学习率更新的方法，有 CONSTANT, STEP, EXP, POLY, STEPS, SIG, RANDOM 几种，STEPS 表示随步数进行衰减
steps=40000,60000	在训练次数达到 40000 和 60000 时进行学习率调整
scales=.1,.1	每次更改的学习率为上一个学习率 \times scales， 为当前步的学习率

调整好训练模型的参数后，便可开始训练。训练过程中，loss 值随着迭代次数的增加而改变，整体呈下降趋势。下图分别是迭代 0-45000 次、1-100 次和 100-45000 次的过程中 loss 值得变化：

图 9 迭代 0-45000 次过程中 loss 值得变化

Fig. 9 the change in the value of loss in first to 45000th iteration

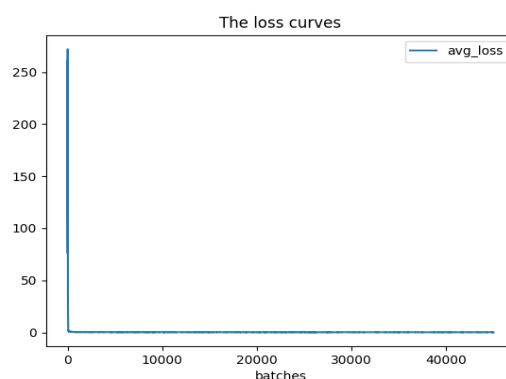


图 10 迭代 0-100 次过程中 loss 值得变化

Fig. 9 the change in the value of loss in first to 100th iteration

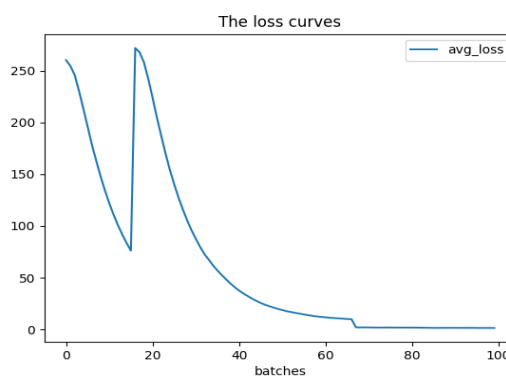
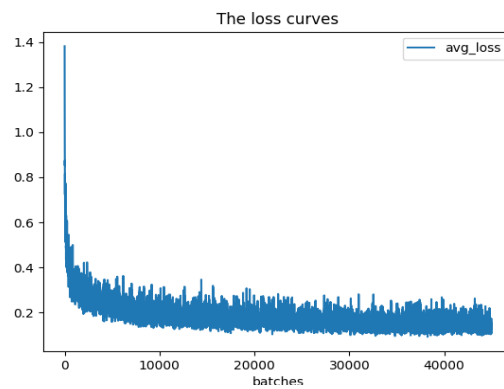


图 11 迭代 100–45000 次过程中 loss 值得变化

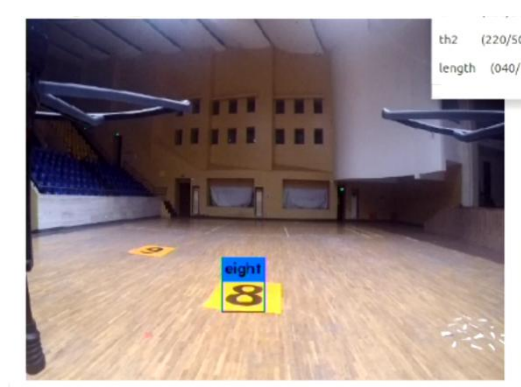
Fig. 9 the change in the value of loss in 100th to 45000th iteration



训练完成后的测试效果如下：

图 12 测试效果

Fig.12 the effect of testing



5.4 实验分析

经过线上仿真环境和线下现场环境的测试，分别得到以下结果。

(1) 线上仿真环境

图 13 线上仿真场景多目标检测结果 1
Fig.13 the result of multi-targets detection-1

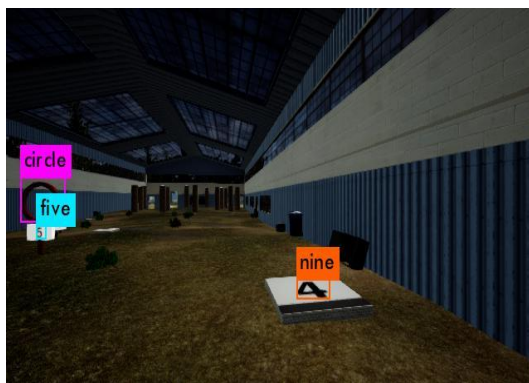


图 14 线上仿真场景多目标检测结果 4
Fig.14 the result of multi-targets detection-4



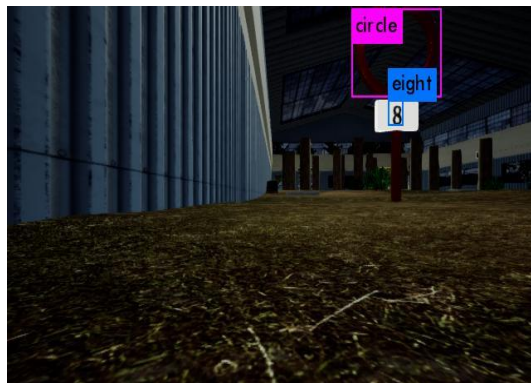
图 12 能识别出图中较为明显的多目标且精度较高，体现了 YOLO 方法的精准度，也为后续确定数字中心提供了条件。图 13 中数字“8”出现漏检情况，出现此情况的可能原因有：

1. YOLO 模型中，每个小格预测边界框数为 2，而 YOLO 默认同格子里所有边界框为同种类物体，因而 YOLO 对小物体及邻近特征检测效果差；
2. 训练样本的选取未完全按照“多角度、多高度”的准则进行，导致提取的某一数字特征不完全，在测试图片中该数字变换角度后无法正确识别。

在重新选取 100 张不同角度和高度的数字“8”作为样本并训练后，测试效果如下：

图 15 线上仿真场景多目标检测结果 6

Fig.15 the result of multi-targets detection-6



(2) 线下现场环境

图 16 线下环境目标检测结果 1

Fig.16 the result of offline target detection-1



相对于线上仿真场景，线下环境的目标检测更偏向于单一目标检测。由于无人机在寻找目标数字时，飞行策略确定为左右扫视寻找目标数字，即无人机保持水平姿态搜索目标，所以在选取样本时，根据场地大小确定数字间的大致距离，推断出相机视野可以覆盖数字标识牌的波动范围，从而确定为检测目标数字，无人机寻找目标时的飞行高度范围，然后模拟无人机在此高度范围下的搜寻状态，用相机拍摄此状态下的目标数字图片作为训练样本。在以上条件下，边缘与无人机航线接近水平的标识牌上的数字和距离无人机较近的数字更易被识别到，而标识牌与无人机航线所成角度大（如图中数字“3”）、相机视野内较小的数字（如图中数字“6”、“7”）不易被识别出。

6 集成实验

6.1 仿真实验

线上仿真的虚拟环境为基于 AirSim 平台修改的虚拟环境。AirSim 是微软发布的、用于给开发人员提供训练和测试机器人、无人机的区分和预计的能力的模拟平台[12]。

如图，运行环境为 Windows 平台，程序通过 RPC 机制与虚拟环境通信。

图 17 线上虚拟环境

Fig.17 online environment



考虑软件使用便捷性，运行平台选用 ubuntu16.04 ROS 环境。

线上仿真的应用程序编程接口（API）包含：

获取图片 API——simGetImages();

获取气压信息 API——get BarometerData(float period);

获取磁力计 API——getMagagnetometerdata(float period)等。

执行线上仿真过程中，利用 YOLO 进行目标检测的结果图如下：

图 18 仿真场景目标检测结果

Fig.18 the result of targets detection

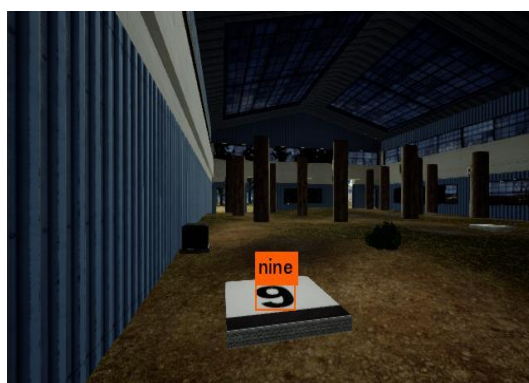


图 19 仿真场景目标检测结果
Fig.19 the result of targets detection



图 20 仿真场景目标检测结果
Fig.20 the result of targets detection

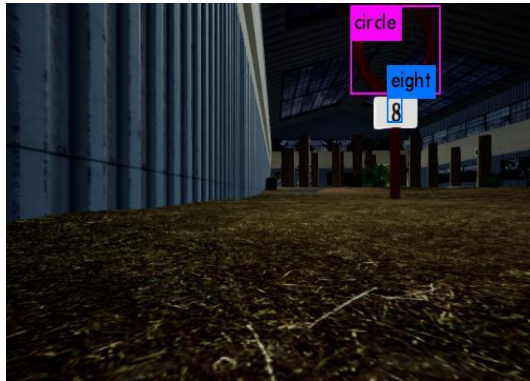


图 21 仿真场景目标检测结果
Fig.21 the result of targets detection

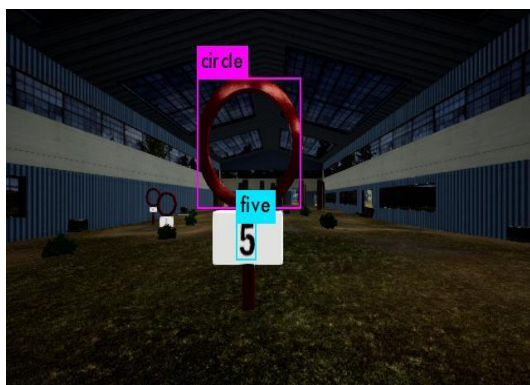


图 22 仿真场景目标检测结果
Fig.22 the result of targets detection



6.2 现场实验

6.2.1 现场数字检测

线下环境搭建地点为上海交通大学霍英东体育馆主馆。现场环境布置如下图所示：

图 23 现场环境
Fig.23 site environment



无人机启动后进入自主飞行模式，通过机载传感器，开始智能感知算法，自动识别地面停机坪和障碍圈，并按照编号完成降落停机坪和穿越障碍圈功能。无人机完成上述任务的过程中，需分别对前视和下视所获取的图像进行目标检测，目标检测帧率为 10FPS，检测效果如下：

图 24 前视目标检测效果

Fig.24 the front-view result of target detection



图 25 前视目标检测效果

Fig.25 the front-view result of target detection



图 26 前视目标检测效果

Fig.26 the front-view result of target detection



图 27 下视目标检测效果 1

Fig.27 the down-view result of target detection-1

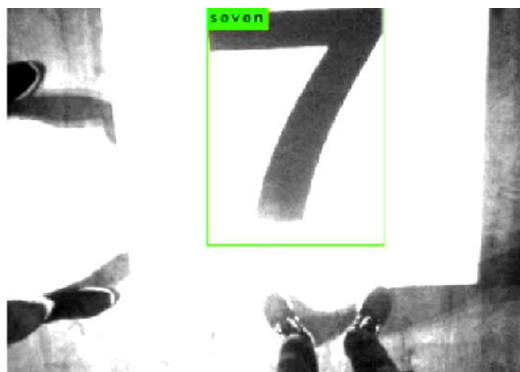


图 28 下视目标检测效果 2

Fig.28 the down-view result of target detection-2



现场环境与线下仿真环境最大的区别之一在于仿真环境中光照条件相对固定，现场环境受天气、灯光等不可控因素的影响，使得光照条件的不稳定对识别效果产生了很大的影响，特别是对下视 Guidance 相机所获得的灰度图像。文献 [13] 对光照条件使得图像退化的机理做出了解释，并提出根据机理采用逆操作进行图像恢复，最后进行图像增强的方法改善图像视觉效果。

6.2.2 真实场景下的车辆及行人检测

在实现数字和圈检测功能后，为探索 YOLO 在现实生活中的功能应用，进行了基于 YOLO 模型的车辆和行人实时监测实验。

基于图片的目标检测虽然能保证准确率，但检测速度达不到视频检测的要求 [14]，因此采用实时性高的 YOLO 模型完成基于视频的车辆和行人检测。实验平

台的主要硬件配置为 Pascal GPU 和 12GB G5X 内存。使用 YOLO 框架 Darknet, 进行检测的视频通过无人机航拍获得, 检测效果图如下:

图 29 车辆检测效果图

Fig.29 the result of vehicle detection

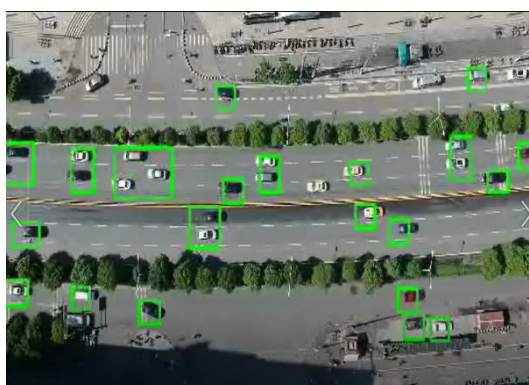


图 30 行人检测效果图

Fig.30 the result of people detection



7 结论

基于机载 TX2 电脑以及轻量级深度学习模型 YOLO 的目标检测方法具有训练方便、实现简单等优点, 实现了识别率高、误检率低的检测效果。

但从测试的结果上来看, 仍存在误检和漏检的情况。适当地增加样本容量, 可以减弱光线对样本质量的影响、拍摄角度对实际识别状况的影响。但由于 YOLO 网络自身特点, 即每个小格预测边界框数为 2, 而 YOLO 默认为同格子里所有边界框为同种类物体, 因而 YOLO 对小物体及邻近特征检测效果差。

YOLO 模型对 RGB 图像的检测效果优于灰度图像。灰度图像的识别效果受光照条件影响大。可以尝试采用先将图像灰度值恢复到光照合适时候的水平,然后用自适应直方图均衡算法增大图像的动态范围 and 对比度的方法对灰度图像进行处理,再进行目标检测[13]。

参考文献:

[1] Zhang Yi-fan. Research and Implementation of Light Weight Real Time Multiobjective Detection System Based on Deep learning. Beuin University of Posts and Telecommunications.2018

张逸凡. 基于深度学习的轻量级实时多目标检测系统的研究与实现[D]. 北京邮电大学, 2018.

[2] Ma Xiao. Analysis on the Status Quo and Development Trend of Autonomous Control Technology of UAV System[J]. SCIENCE & TECHNOLOGY INFORMATION. 2016

马晓. 浅析无人机系统自主控制技术研究现状及发展趋势[J]. 科技资讯, 2016, 14(17):1+3.

[3] BENGIO Y . Learning deep architectures for AI[J]. Foundations and Trends in Machine Learning , 2009 , 2 (1) : 1-127 .

[4] Yin Bao-cai, Wang Tongwen, Wang Lichun. Review of Deep Learning[J]. JOURNAL OF BEIJING UNIVERSITY OF TECHNOLOGY. 2015.

尹宝才, 王文通, 王立春. 深度学习研究综述[J]. 北京工业大学学报, 2015

[5] Wang Fei-fei. Analysis of the status quo of deep learning research[J]. Electronic Technology and Software Engineering.2018

王菲斐. 深度学习研究现状分析[J]. 电子技术与软件工程, 2018(10):152-153.

[6] Girshick R. Fast r-cnn[C]Proceedings of the IEEE international conference on computer vision. 2015: 1440-1448.

[7] Chen We-ican. Design and Research of Robotic 3D Simulation Platform Based on ROS[D]. Northeastern University. 2014

陈炜灿. 基于 ROS 的机器人三维仿真平台设计与研究[D]. 东北大学, 2014

[8] Chen Chang-sheng. ARM9-based IIC serial communication[J]. Computer knowledge and Technology. 2010

陈长生. 基于 ARM9 的 IIC 串行通信[J]. 电脑知识与技术, 2010, 6(06):1490-1493.

[9] YOLOv3: An Incremental Improvement Joseph Redmon, Ali Farhadi University of Washington

[10] Zhang Ning. Development of Dynamic Target Detection System Based on UAV Platform[D]. ZheJiang University. 2018

张宁. 基于无人机平台的动态目标检测系统开发[D]. 浙江大学, 2018.

- [11] Everingham M, Eslami S M A, Van Gool L, et al. The Pascal Visual Object Classes Challenge:A Retrospective[J]. International Journal of Computer Vision, 2015,111(1):98-136.
- [12] Deng Xiao-lei. Microsoft Open Source System AirSim A real flight simulator for drones[N]. China's Computer News. 2017
邓晓蕾. 微软开源系统 AirSim 一个无人机的真实飞行模拟器[N]. 中国计算机报, 2017-02-27 (005).
- [13] Tao Zhi-feng. A Method of Improving the Image Quality of Deterioration Influenced by Illumination[J]. GUIDANCE & FUZE. 2009
陶志锋. 一种改善受光照影响而恶化的图像质量的方法[J]. 制导与引信, 2009, 30 (03) :52-55.
- [14] Wang Yu-ning, Pang Zhi-heng, Yuan De-ming. Vehicle Detection Based on POLO in Real Time[J]. JOURNAL OF WUHAN UNIVERSITY OF TECHNOLOGY. 2016
王宇宁, 庞智恒, 袁德明. 基于 YOLO 算法的车辆实时检测[J]. 武汉理工大学学报, 2016, 38 (10) :41-46.
- [15] Li Jiang-yong, Liu Xue-mei, Li xue-xia, Du Bo-yang. Design of Open Source Mobile Robot System Based on ROS[J]. Mechanical and electrical engineering. 2017
李建勇, 刘雪梅, 李雪霞, 杜博阳. 基于 ROS 的开源移动机器人系统设计[J]. 机电工程, 2017, 34 (02) :205-208.