

doi: 10.7690/bgzd.2023.01.003

基于级联式 Snappy-CenterNet 的锥套目标检测算法

杨乐¹, 张怡², 王国屹², 李华²

(1. 中国航空工业集团公司金城南京机电液压工程研究中心液压与作动系统部, 南京 211106;

2. 南京航空航天大学导航研究中心, 南京 211106)

摘要: 针对空中加油因场景光照变化、环境遮挡等情况造成的锥套目标识别精度低、实时性差的问题, 提出一种基于级联式 Snappy-CenterNet 深度网络的锥套目标检测算法。在 CenterNet 网络的基础上, 以 HourglassNet 为主干网络, 改进其 bottleneck 结构并引入中心池化的方法, 对整体的网络结构进行优化, 通过级联式的网络提升整体检测精度。实验结果表明: 该算法可实现在多种复杂场景下对锥套目标的可靠检测, 检测结果的精确率与召回率均可达 99%, 位置精度与区域精度分别可达 99% 与 96%, 更新率可达 33.68 Hz, 满足空中加油近距视觉导航阶段对于锥套识别的指标要求。

关键词: 空中加油; 锥套识别; 深度学习; 目标检测; 级联网络

中图分类号: TP391.41 **文献标志码:** A

Drogue Target Detection Algorithm Based on Cascaded Snappy-CenterNet

Yang Le¹, Zhang Yi², Wang Guoyi², Li Hua²

(1. Hydraulic and Actuation System Department, AVIC Jincheng Nanjing Engineering Institute of Aircraft Systems, Nanjing 211106, China;

2. Navigation Research Center, Nanjing University of Aeronautics and Astronautics, Nanjing 211106, China)

Abstract: Aiming at the problems of low accuracy and poor real-time performance of drogue target recognition in aerial refueling caused by scene illumination changes and environmental occlusion, a drogue target detection algorithm based on cascaded Snappy-CenterNet deep network is proposed. On the basis of CenterNet network, HourglassNet is used as the backbone network, the bottleneck structure is improved and the central pooling method is introduced to optimize the overall network structure, and the overall detection accuracy is improved through the cascaded network. The experimental results show that the proposed algorithm can reliably detect drogue targets in a variety of complex scenes, and the precision and recall of the detection results can reach 99%, the position accuracy and region accuracy can reach 99% and 96%, respectively, and the update rate can reach 33.68 Hz, which meets the requirements of aerial refueling near vision navigation for drogue recognition.

Keywords: aerial refueling; drogue recognition; deep learning; target detection; cascaded net

0 引言

随着制空权在现代战争中对于战场态势的影响愈加明显, 作为可以有效提升飞行器滞空时间与飞行半径的关键技术, 空中加油已经逐渐成为当今航空军事领域的主要研究方向之一^[1]。在实际加油过程中, 对接环节是决定任务能否成功实现的核心步骤, 在该环节内准确、快速地获取锥套的所在位置可以有效提升空中加油对接过程的可靠性与安全性。视觉导航具备自主性强、环节鲁棒性高等特点, 借助视觉图像捕获与识别锥套区域信息已成为对接环节中获取锥套位置的主要手段^[2]。对于利用视觉图像对锥套目标进行检测识别的问题, 国内外多家研究单位已开展了大量的研究^[3-4]。

西班牙马德里理工大学于 2012 年提出了一种基于图像匹配的空中加油视觉导航方法, 通过模板匹配与区域分割方法实现对锥套目标的检测与跟踪, 进而解算锥套位置^[5]。美国空军大学于 2020 年提出了一种基于卷积神经网络的加油锥套立体视觉引导方法, 利用图像生成的 3 维点云信息来解算锥套的空间位姿^[6]。空军工程大学于 2015 年提出了一种基于外接关键特征的锥套检测与跟踪的方法, 通过识别布置在锥套端面的红色标识环实现锥套的检测与跟踪^[7]。南京航空航天大学于 2017 年提出了一种基于自适应核窗口的锥套跟踪方法, 利用对中值滤波跟踪算法的改进, 实现锥套目标的快速跟踪^[8]。北京理工大学于 2018 年提出了一种基于颜色分割的锥套检测与定位方法, 借助目标的颜色与形状特

收稿日期: 2022-09-01; 修回日期: 2022-10-12

作者简介: 杨乐(1989—), 男, 江苏人, 硕士, 高级工程师, 从事航空液压与作动、无人机空中自主控制研究。

E-mail: 570689154@qq.com.

征分割出目标候选区域，并对锥套的相对位姿进行估计^[9]。

随着硬件设备算力的大幅提升，利用神经网络提升目标检测的精度与实时性也逐渐得到了广泛的关注。南京航空航天大学于 2019 年提出了一种基于区域建议网络的加油锥套识别方法，采用 Faster-RCNN 网络准确提取视觉图像内的锥套所在区域^[10]。西北工业大学于 2020 年提出了一种基于双目视觉的深度学习目标检测与跟踪方法，采用神经网络提取图像中锥套的特征点并进行匹配^[11]。

为改善场景光照变化、环境遮挡等情况对锥套识别的精度与实时性产生的严重影响，笔者通过改进 CenterNet 深度网络提升多场景下锥套目标检测的准确性与快速性，设计一种级联式的网络结构改善小目标情况下锥套的检测效果。

1 基于中心池化的 Hourglass 主干网络设计

卷积神经网络的特征提取模块通常由主干网络 (backbone) 负责。CenterNet 网络模型的 backbone 是目前较为前沿的一种主干网络——堆叠沙漏网络 (stacked hourglass networks)^[12]，然而其体量较大，难以满足空中加油的实时性需求；因此，需对其结构进行针对性改进，以同时满足检测精度与速度的要求。

1.1 Hourglass bottleneck 结构改进

该网络的原始 bottleneck 结构包含 2 条前向通路，如图 1(a)所示，主通路包含 2 次 3×3 的卷积，

副通路包含一次 1×1 的卷积，均用于提取目标图像的特征。很显然，该结构可获得足够的特征图关键信息，但是通过式(1)、(2)可知结构总体的参数量与运算量较大，造成整体网络的训练速度缓慢。

$$W_{\text{para}} = C_i \times k_w \times k_h \times C_o; \quad (1)$$

$$W_{\text{freq}} = C_i \times k_w \times k_h \times C_o \times W \times H. \quad (2)$$

式中： W_{para} 、 W_{freq} 为结构的参数量与计算量 (乘加次数)； C_i 、 C_o 为输入与输出特征图的通道数； k_w 、 k_h 为卷积核的宽高尺寸； W 、 H 分别为特征图的宽与高。

笔者提出一种改进的 Hourglass bottleneck 结构设计，在保证网络模型精度的基础上提升网络效率。由于主要的运算量集中在卷积运算部分，因此引入尺寸为 1×1 的卷积核来减少结构的参数量，并在输出前进行一次 1×1 的卷积，提升特征图维度。由式(1)可知改进前后结构的参数量，其输入输出通道数均为 256，改进前结构参数量计算过程为 $256 \times 3 \times 3 \times 256 \times 2 + 256 \times 1 \times 1 \times 256$ ，结果为 1 245 184，改进后结构参数量计算过程为 $256 \times 1^2 \times 64 + 64 \times 3^2 \times 64 + 64 \times 1^2 \times 256 + 256 \times 1^2 \times 256$ ，结果为 135 168，约下降了 89.1%。此外，为了提升模型对于非线性元素的表达能力，将所有的卷积层 (Conv) 移动至批归一化层 (batch normalization, BN) 之前，使得输入首先经过 Conv 模块进行特征提取，然后经过 BN 模块平衡数据分布，再经过 ReLU 激活函数引入非线性环节，改进后的结构如图 1(b)所示。

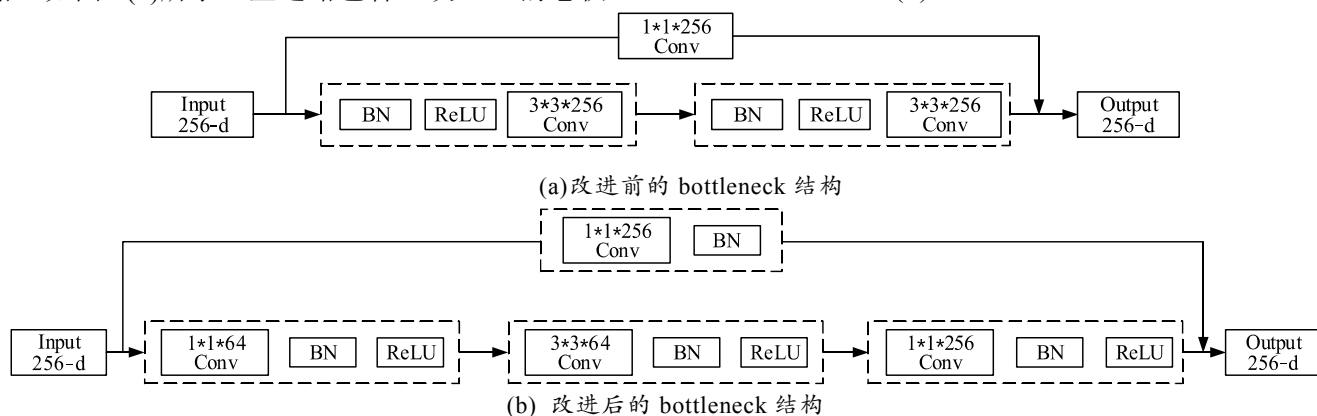


图 1 改进前后的 bottleneck 结构

1.2 基于中心池化的 Hourglass 模块设计

如图 2 所示，空中加油过程中使用的锥套一般呈类圆形目标，可以近似认为是中心对称图形，有效利用锥套成像中心点可帮助提高检测与识别的准确率。

如图 3 所示，在 CenterNet 的主干网络中，降采样依靠最大池化进行，然而对于锥套这种类圆形目标，最大池化只能保留局部区域内特征的最大值属性，无法突出其中心点的重要性；因此，引入中心池化的方法，将目标成像中心点的增益引入到池

化层中。

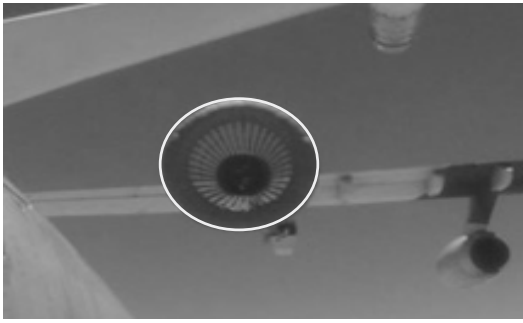


图 2 锥套

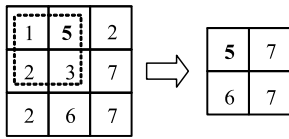


图 3 最大池化过程

中心池化的具体操作流程如下：以一个 3×3 的特征图为例，如图 4 所示，假设对第 1 个像素点进行中心池化，分别从左到右、从上到下、从下到上、从右到左按顺序逐行逐列进行搜索，在搜索过程中，寻找前向通路中的最大值，用于替换当前点值，4 个方向均搜索完成后，求和得到的值即为对该像素点进行中心池化的结果。

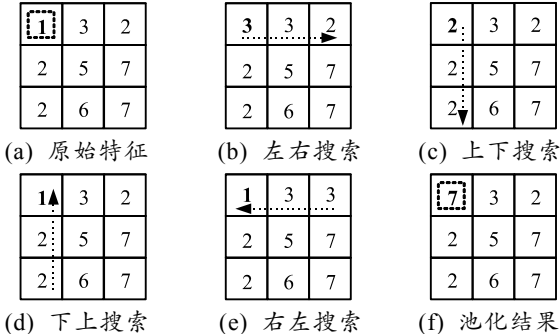


图 4 中心池化过程

2 基于改进 Hourglass 的 CenterNet 检测器

2.1 改进的 CenterNet 网络结构设计

为了提升精度，CenterNet 网络中包含 2 组 Hourglass 模块，如图 5 所示。虽然上文对 Hourglass 结构的优化可在一定程度上提升网络效率与精度，但是 2 组 Hourglass 模块带来的计算量导致模型仍然无法满足空中加油对于实时性的要求；因此，笔者通过缩减 Hourglass 结构的个数解决运算速度缓慢的问题，并对 CenterNet 网络结构进行一定的优化调整来平衡在网络提速过程中可能出现的精度下降问题，为方便表述，将改进 Hourglass 的 CenterNet 检测器命名为 Snappy-CenterNet。

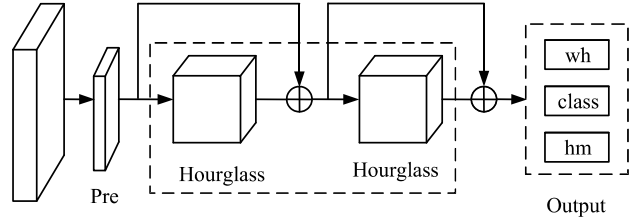


图 5 改进前的 CenterNet 网络结构

首先，在图像输入后引入预处理模块，通过卷积层与上文改进过的 Hourglass bottleneck 结构进行特征提取，将该特征图作为后续改进的 Hourglass 主干网络的输入，如图 6 所示。该网络结构通过缩减 Hourglass 结构个数并将原始的残差模块替换为改进的 Hourglass bottleneck 模块来降低网络运行耗时，再利用优化过的 Hourglass 主干结构平衡运算速度与精度。

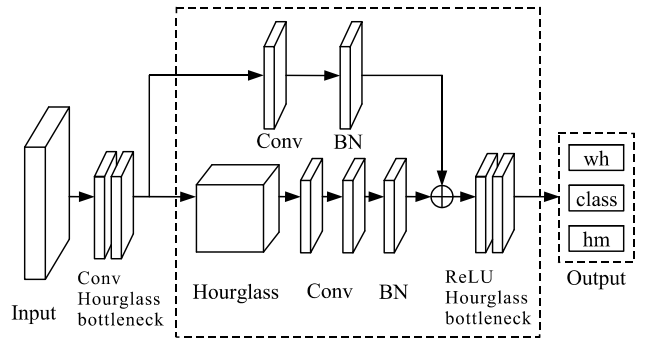


图 6 改进后的 Snappy-CenterNet 网络结构

2.2 基于级联网络的小目标检测优化

在空中加油近距对阶段的前期过程中，锥套目标在图像中成像较小，会出现检测精度下降的情况。针对这一情况，笔者设计一种级联式检测网络结构，用于提升小目标的检测精度。

该级联网络由两级 CenterNet 网络串联而成，第一级网络用于提取目标的初始检测框，将该检测框放大 3 倍作为二级网络的输入，级联网络检测过程如图 7 所示。



(a) 一级网络检测结果

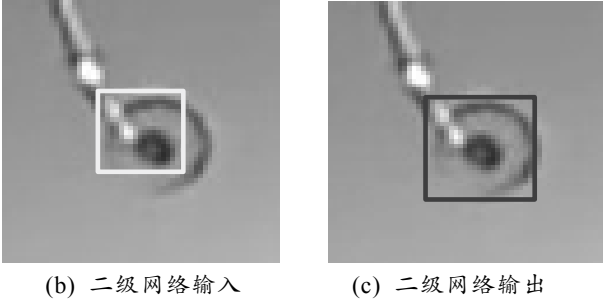


图 7 级联 CenterNet 网络检测过程

上图中：(a)为一级网络检测结果；(b)为对一级网络检测框放大 3 倍截取后的局部区域按比例放大的结果，比例系数为 α ；(c)为二级网络的检测结果。假设一级网络的检测框参数为 $B_1(x_1, y_1, w_1, h_1)$ ，放大 3 倍后局部区域的参数为 $B_2(x_2, y_2, w_2, h_2)$ ，按比例放大后得到的二级网络输入图像的参数为 $B_3(w_3, h_3)$ ， B_1 、 B_2 两者之间的表示关系为：

$$\left. \begin{aligned} x_2 &= x_1 - w_1 \\ y_2 &= y_1 - h_1 \\ w_2 &= 3w_1 \\ h_2 &= 3h_1 \end{aligned} \right\}。 \quad (3)$$

式中： (x_1, y_1) 、 (x_2, y_2) 分别为一级输出检测框与放大 3 倍后局部图像的左上角点坐标， (w_1, h_1) 、 (w_2, h_2) 分别为一级输出检测框与放大 3 倍后局部图像的宽高。 B_2 、 B_3 两者之间的表示关系为：

$$\left. \begin{aligned} w_3 &= \alpha w_2 \\ h_3 &= \alpha h_2 \\ \alpha &= H/2h_2 \end{aligned} \right\}。 \quad (4)$$

式中： (w_3, h_3) 为二级输入图像的宽高； α 为预设的比例系数； H 为全局图像的高。

二级网络输出的检测框参数为 $B_4(x_4, y_4, w_4, h_4)$ ，将该检测框归一化至全局图像中，可得 $B(x, y, w, h)$ 。其中各参数求解公式如下：

$$\left. \begin{aligned} x &= x_1 - w_1 + \frac{6h_1}{H} \cdot \left(x_4 + \frac{w_4}{2}\right) \\ y &= y_1 - h_1 + \frac{6h_1}{H} \cdot \left(y_4 + \frac{h_4}{2}\right) \\ w &= \frac{6h_1 \cdot w_4}{H} \\ h &= \frac{6h_1 \cdot h_4}{H} \end{aligned} \right\}。 \quad (5)$$

式中： (x_4, y_4) 、 (x, y) 为二级网络输出检测框的左上角点坐标与最终结果检测框的中心点坐标； (w_4, h_4) 、 (w, h) 为二级网络输出的检测框与结果检测框的宽与高。

3 实验设计与结果分析

为验证本文中算法的效果，设计相关实验系统进行测试，然后对实验结果进行分析，进而给出算法可行性结论。

3.1 实验系统设计

算法实验系统主要从硬件平台、软件平台与实验数据集 3 部分进行设计。其中，硬件平台为包含高性能显卡模块的桌面端计算机，具体参数如表 1 所示。

表 1 实验系统硬件平台相关参数

参数名称	具体内容
CPU	Intel Core i7-6800K
CPU 核数	6
CPU 主频	3.40 GHz
GPU	Nvidia Geforce GTX1080 Ti
GPU 个数	2
内存	32 GB

软件平台包括操作系统、编程语言、显卡驱动、GPU 加速模块、网络框架函数库、图像处理函数库等部分，具体参数如表 2 所示。

表 2 实验系统软件平台相关参数

部件名称	具体内容
操作系统	Ubuntu 16.04
编程语言	Python 3.5.2
显卡驱动	Nvidia Driver 384.130
GPU 加速模块	CUDA 9.0 + cuDNN 7.0.5
网络框架函数库	Pytorch 0.4.1
图像处理函数库	Opencv 4.2.0

鉴于空中加油场景的特殊性，从网络获取的真实空中加油场景视频的数量不足以用于神经网络模型的训练；因此，为丰富锥套样本数据集，所处实验室根据实际锥套的尺寸、外形等特点，一比一制作了相应的锥套模型，如图 8 所示。

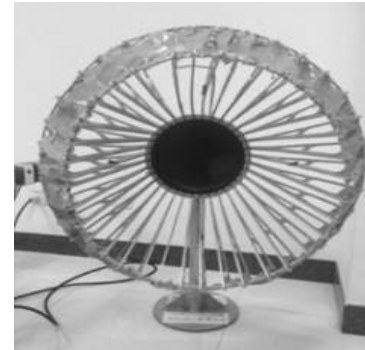


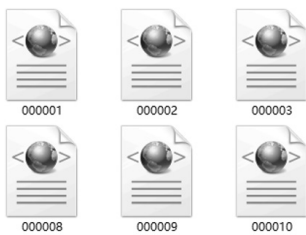
图 8 自制锥套等比例模型

分别对 2 类加油视频中出现的锥套目标进行位置标定，作为标称值存储在标签文件中，共 7 000 组样本集，包含了不同光照情况、背景环境、遮挡

情况等多种情况,将其中 70%的样本用于训练网络,30%的样本作为测试集供算法模型的验证,其中小目标样本约有 400 组,样本集内容如图 9 所示。



(a) 样本图像



(b) 标签文件

```
[{"segmentation": [[264, 277, 264, 326, 313, 326, 313, 277]], "id": 1}, {"segmentation": [[1379, 146, 279, 256, 479, 256, 479, 146, 1379]], "category_id": 1, "id": 2}, {"segmentation": [[226, 88, 226, 53, 53, 55], "category_id": 0, "id": 3}, {"segmentation": [[105, 105, 6, 103, 96], "category_id": 1, "id": 4}, {"segmentation": 5, "bbox": [196, 141, 89, 78], "category_id": 0, "id": 5}, {"image_id": 6, "bbox": [551, 113, 22, 34], "category_id": 1, "ignore": 0, "image_id": 7, "bbox": [212, 144, 23, 24], "category_id": 0, "image_id": 8, "bbox": [335, 119, 14874, "iscrowd": 0, "ignore": 0, "image_id": 9, "bbox": [103, 372, 621], "area": 1558, "iscrowd": 0, "ignore": 0, "image_id": 10, "bbox": [360, 142, 395, 142, 395, 103], "area": 1365, "iscrowd": 0, "segmentation": [[456, 238, 456, 370, 576, 370, 576, 238]], "category_id": 1, "id": 12}, {"segmentation": [[103, 96, 103, 96, 21, 32], "category_id": 1, "id": 13}, {"segmentation": 11, "bbox": [116, 279, 27, 27], "category_id": 1, "id": 14}, {"image_id": 15, "bbox": [170, 151, 41, 40], "category_id": 0, "ignore": 0, "image_id": 16, "bbox": [555, 122, 26, 42], "category_id": 0, "ignore": 0, "image_id": 17, "bbox": [58, 163, 163], "area": 6399, "iscrowd": 0, "ignore": 0, "image_id": 18, "bbox": [414, 73, 414, 34], "area": 1599, "iscrowd": 0, "ignore": 0, "image_id": 19, "bbox": [128, 166, 223, 223, 128, 128], "area": 6633, "iscrowd": 1}, {"segmentation": [[112, 139, 112, 174, 174, 174, 174, 139, 112]], "id": 21}, {"segmentation": [[304, 243, 304, 258, 348, 258, 348, 243, 304]], "category_id": 0, "id": 22}, {"segmentation": [[337, 208, 337, 208, 30, 32], "category_id": 1, "id": 23}, {"segmentation": 24, "bbox": [169, 267, 23, 25], "category_id": 1, "id": 24}, {"image_id": 25, "bbox": [454, 237, 126, 131], "category_id": 0, "ignore": 0, "image_id": 26, "bbox": [515, 195, 54, 64], "category_id": 0, "ignore": 0}],
```

(c) 训练索引

图 9 样本集内容

3.2 实验结果分析

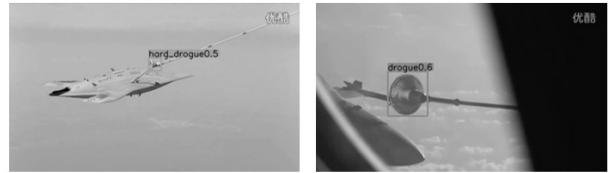
利用自行制作的样本数据集对本文中提出的算法进行实验验证,效果如图 10 所示,分别包含了多背景场景、小目标场景、多视角场景以及遮挡场景。



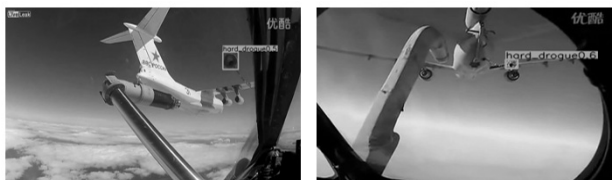
(a) 多场景检测



(b) 多视角检测



(c) 遮挡检测



(d) 小目标检测

图 10 算法测试效果

为验证算法具体性能,将传统 CenterNet 算法及目前在目标检测领域领先的 YOLOv3 算法^[13]与 Snappy-CenterNet 算法进行对比,分别从准确性、可靠性与实时性 3 方面对算法性能展开测试与分析。

3.2.1 准确性指标

本文中定义算法的准确性由准确率与成功率进行定量评估。

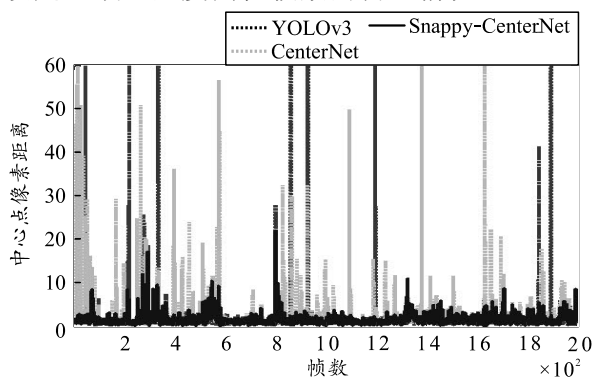
1) 准确率。

准确率与结果框位置误差有关,位置误差由结果目标框与真实目标框两者中心点的欧式距离表示。笔者取误差阈值范围为[0, 30],图 11(a)展示的是每帧图像的位置误差,图 11(b)展示的是在 2 000 帧图像中,中心点距离小于设定的位置误差阈值帧数的占比情况。

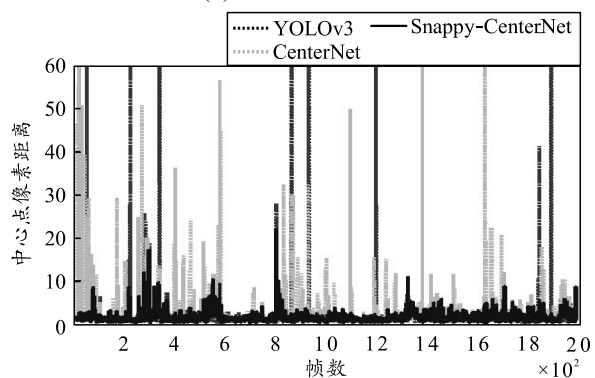
2) 成功率。

成功率与区域交并比指数(intersection over union, IoU)有关,该指数为 2 个区域交集与并集面积的比值,表示两区域重合部分占总面积的大小,又称区域重叠度。笔者取区域重叠度阈值范围为[0, 1],图 12(a)展示的是每帧图像的区域重叠度数值,图 12(b)展示的是在 2 000 帧图像中,真实重叠度小

于设定区域重叠度阈值帧数的占比情况。

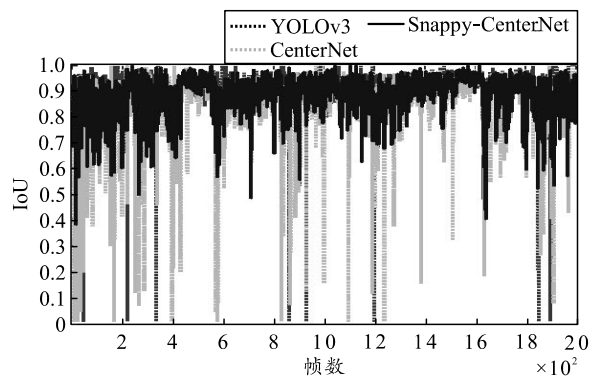


(a) 位置误差对比

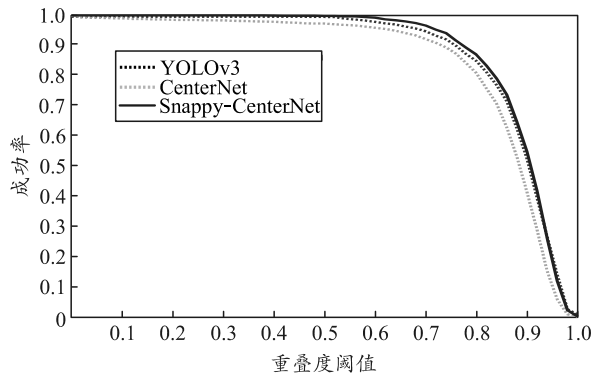


(b) 准确率对比

图 11 位置误差与准确率情况对比结果



(a) 区域重叠度对比



(b) 成功率对比

图 12 区域重叠度与成功率情况对比结果

其中出现尖峰值的帧即为小目标帧，由此可见

当小目标出现时，传统 CenterNet 网络会出现位置误差大、区域重叠度低等情况，YOLOv3 网络对小目标情况的泛化能力相对较好，但仍会出现检测失败或结果错误的情况，而 Snappy-CenterNet 算法相较而言可有效改善对小目标检测的准确程度。由准确率与成功率对比图中可知，Snappy-CenterNet 算法的准确率与成功率在三者中表现最优。

3.2.2 可靠性与实时性指标

本文中定义算法的可靠性由精确率 P 与召回率 R 2 项指标进行定量评估，计算公式如下：

$$\left. \begin{aligned} P &= TP / (TP + FP) \times 100\% \\ R &= TP / (TP + FN) \times 100\% \end{aligned} \right\} \quad (6)$$

式中： TP 为分类正确的正样本； FP 为错误分类的负样本； FN 为错误分类的正样本。

算法的实时性效果由处理时间进行定量评估，以 ms 作为计量单位。

若以位置误差阈值为 10 时的准确率结果代表算法整体的位置精度，以重叠度阈值为 0.7 时的成功率结果代表算法整体的区域精度，3 种算法的整体情况如表 3 所示。无论是在准确性、可靠性还是实时性方面，Snappy-CenterNet 算法均优于其余对比算法，满足空中加油对于视觉导航系统的基本要求。

表 3 算法整体性能情况

算法	精确率	召回率	位置精度	区域精度	耗时/ms
CenterNet	0.97	0.98	0.96	0.92	43.79
YOLOv3	0.98	0.98	0.98	0.95	32.42
Snappy-CenterNet	0.99	0.99	0.99	0.96	29.69

从以上对比结果可以看出，Snappy-CenterNet 网络降低了 32.2% 的网络运算时间，将检测结果精度提升了 3.5%，同时实现了小目标情况下的目标准确检测。

4 结束语

笔者基于深度学习的方法，优化了 Hourglass 主干网络结构，并以此为框架设计了基于级联式 Snappy-CenterNet 的目标检测网络，提升了目标检测算法的精度与速度，并有效提升了小目标情况下的检测准确度。

实验结果表明：笔者提出的基于级联式 Snappy-CenterNet 的空中加油锥套目标检测算法的可靠性更强、准确性更优、实时性更高，对于空中加油过程中的视觉导航系统的实际应用具有较为重要的参考价值。