Modeling and Simulation 建模与仿真, 2022, 11(2), 432-441
Published Online March 2022 in Hans. http://www.hanspub.org/journal/mos
https://doi.org/10.12677/mos.2022.112040

Hans汉斯

基于YOLOv5的机场飞行区目标检测方法研究

张斗斗，晏子奕，周良杰，蔡先乔，汪海波，惠康华

中国民航大学计算机学院，天津

收稿日期：2022年2月23日；录用日期：2022年3月14日；发布日期：2022年3月23日

摘　要

机场飞行区的飞机和车辆很多，人员活动复杂，需要检测的目标很多，而且需要实时检测目标等等。本文通过对四种主流目标检测算法的比较，找出了最适合机场飞行区的目标检测算法YOLOv5，以解决上述问题。利用YOLOv5x模型，对机场飞行区视频数据进行初步的目标检测和标注，然后手工修改和优化标注结果，组织数据集结构，合理分配训练集和测试集比例，构建适合机场飞行区的数据集。基于该数据集，本文采用深度学习算法YOLOv5训练出适用于机场飞行区的目标检测模型，实现了对机场飞行区主动目标的实时检测。通过在实验中动态调整算法策略，本文所训练的目标检测模型的效率达到了较高的水平。实验结果表明，改进后的算法具有95.5%的识别精度，满足机场飞行区的目标检测要求。

关键词

机场飞行区，深度学习，YOLOv5，目标检测

Research on Target Detection Method of Airport Flight Area Based on YOLOv5

Doudou Zhang, Ziyi Yan, Liangjie Zhou, Xianqiao Cai, Haibo Wang, Kanghua Hui

School of Computer Science, Civil Aviation University of China, Tianjin

Received: Feb. 23rd, 2022; accepted: Mar. 14th, 2022; published: Mar. 23rd, 2022

Abstract

There are many aircraft and vehicles in the airport flight area, and the personnel activities are complicated, so there are a lot of targets to be detected, and they need to be detected in real time, etc. In this paper, we find out the most suitable target detection algorithm YOLOv5 for airport flight area to solve the above problems by comparing four mainstream target detection algorithms. Using the YOLOv5x model, we perform preliminary target detection and annotation on

文章引用: 张斗斗, 晏子奕, 周良杰, 蔡先乔, 汪海波, 惠康华. 基于 YOLOv5 的机场飞行区目标检测方法研究[J]. 建模与仿真, 2022, 11(2): 432-441. DOI: 10.12677/mos.2022.112040

the airport flight area video data, and then manually modify and optimize the annotation results, organize the data set structure, reasonably allocate the training set and test set ratio, and construct a data set suitable for the airport flight area. Based on this dataset, this paper uses the deep learning algorithm YOLOv5 to train a target detection model applicable to the airport flight area and realize the real-time detection of active targets in the airport flight area. By dynamically adjusting the algorithm strategy in the experiments, the efficiency of the target detection model trained in this paper reaches a high level. The experimental results show that the improved algorithm has a recognition accuracy of 95.5%, which meets the target detection requirements of the airport flight area.

Keywords

Airport Flight Area, Deep Learning, YOLOv5, Target Detection

Copyright © 2022 by author(s) and Hans Publishers Inc.
This work is licensed under the Creative Commons Attribution International License (CC BY 4.0).
http://creativecommons.org/licenses/by/4.0/
Open Access

1. 引言

航空安全是民航人永恒的主题，不止飞机的航行安全，机场的安全管理更是需要重点关注的部分，机场当中的机场飞行区安全则是直接关系到航空系统的运作是否正常。随着航空业的发展，越来越复杂的机场飞行区环境极大程度上增加了管理人员的管理难度。现有的机场飞行区目标检测方法主要是基于遥感影像的机场区域和飞行器的检测，并没有专门针对机场飞行区内部目标检测的方法。为了协助管理人员高效管理整个机场飞行区，需要一个专用于机场飞行区环境的智能视频监控系统，对飞行区的状态进行实时监控，其中最核心的就是对机场飞行区的目标进行检测识别，由于机场飞行区属于安全性要求较高的特殊环境，其中的客机、特种车辆、工作人员较多，现有的公共数据集中缺乏许多机场特有的标注样本，因此需要创建机场飞行区目标检测专用数据集，用来训练适用于该环境的机器学习模型对飞行区目标的检测。

2. 目标检测算法

RCNN 算法[1]，SPP-net 算法[2]，Faster-RCNN 算法[3]，YOLO 系列算法[4]和 SSD (Single Shot MultiBox Detector)算法[5]等是目前主流的目标检测算法。以下将介绍几个具有代表性的目标检测算法。先了解几个相关概念。

- regions 是指从原图像上获取的有可能包含待检测目标的某些区域集合[1]。
- bounding box 是指包含待检测目标的矩形框[1]。
- 交并比(Intersection over Union, IoU)是测量待检测物体准确度的一个指标，用来衡量两个 regions 之间的重叠度[6]。
- 非极大值抑制(Non-Maximum Suppression, NMS)是用来抑制不是极大值的部分，找出局部概率的最值，对于无限接近极大值的元素也要抛弃掉，最后只保留一个元素。对于之后的目标识别，使用 bounding box 进行最后的筛选。通过使用支持向量机(Support Vector Machine, SVM)判别之后得出的 bounding boxes，包含了所有类别的 bounding boxes，通过非极大值抑制 NMS 的实现，最后留下不同类的最优 bounding box [7]。

2.1. RCNN (Region Convolutional Neural Networks)

RCNN 被称为通过深度学习方法进行目标检测的开创性算法。RCNN 算法的原理大致分为四个步骤。即首先将每一张图片分割很多个候选区域，再对生成的每个候选部分，进行特征提取，将提取到的特征放到相应类别的 SVM 分类器中，判断是否属于该类型，最后修正候选框的位置[1]。

RCNN 训练特征主要经过 Pre-train 和 Fine-tune 两个阶段[1]。Pre-train 阶段使用 ILSVRC 2012 数据集[8]及简化版的 Hinton2012 [9]在 ImageNet 上的分类网络来进行预训练。Fine-tune 阶段的作用是更换掉 Pre-train 的最后一个输出层，将其转换成 21 个分类的输出层，然后通过使用 Pascal VOC 2007 数据集来训练网络。训练时要选择 IoU (交并比) > 0.5 时为正样本。

RCNN 网络主要有性能优越、通用性强、可优化点多等优点。而其缺点同样明显，首先 RCNN 训练模型的时候会通过多个阶段，其次选择性搜索(Selective Search, SS) [1]得到的 regions 是不同尺寸的，而不同尺寸的 regions 必须通过一系列操作使其达到相同尺寸，这些操作会造成某些原本存在的且需要目标丢失。而且每个 region 的提取都需要通过模型，之后再存放到磁盘，这样做增大了算力的需求，对硬件的损耗相对来说是很大的。

2.2. Fast-RCNN

提出感兴趣区域(regions of interesting, ROI) pooling layer，采用了单层金字塔结构，即单个金字塔 max-pooling 作用于特征层，简化了 regions 和特征层的映射关系[3]。网络结构通过 Fast-RCNN 算法提出的梯度传递方法进行整体训练。在两层全连接中加入奇异值分解(Singular Value Decomposition, SVD) [3]降维，加快训练速度。网络的输出使用了两个 SoftMax，分别用于 class 分类和 bounding box 回归。之后又将其优化成 Faster-RCNN。

提出区域生成网络(Region Proposal Networks, RPN)并使用神经网络生成 regions 是 Faster-RCNN 的主要优势。该方法优化了 RCNN 采用的选择查询(Selective Search)策略，减少了 regions 的提议(proposal)耗时，能够完成端对端的训练[3]。

RPN 技术大概的思路是通过滑动窗口的方法将从图集中提取到的特征进行分析预测。具体操作是用 3 × 3 的卷积核对提取的 256 维特征图进行滑动卷积，然后用 1 × 1 的卷积将其分为两个独立的路径，一条路径分别输出所有的 anchors 检测目标和未检测背景的概率，第二条路径输出 anchors box 相应的四条参数，即中心坐标点(x, y)、长(height)、宽(weight)。卷积进行一次滑动，就会输出 k 个信息，来反馈出 anchors 中是否存在待测物体及其具体位置。所以经过 RPN 技术得到的输出一路有 $2k$ 个信息，具体表示 anchors 是否包含物体，而第二路有 $4k$ 个信息，表示具体 anchors 的位置。

训练方法如下所示[3]：

1) 首先对 RPN 网络进行训练，RPN 需要的参数由预设的模型输入。

2) 接着只训练 Fast-RCNN 网络，将从第一次 RPN 网络训练得到的结果域作为 RCNN 网络的输入。

3) 再次对 RPN 网络进行训练，只需改变 RPN 唯一部分的数据，保留网络共同域的数据。

4) 用第二次训练 RPN 得到的结果再次对 Fast-RCNN 网络进行调整，也只改变 Fast-RCNN 唯一部分的数据，保留网络共同域的数据。

2.3. YOLO 算法

YOLO 算法的核心思想是不对图像进行分割，而是将单张图像作为一个整体输入网络，并在输出层直接返回 bounding box 的位置和类别。相比较 YOLO 算法，Fast-RCNN 虽然也是将整个图像作为网络的

输入，但实际上还是采用了 RCNN 的输入思想，不过是改变了网络中提取 proposal 的位置。

YOLO 算法的大致思路是将一张图片分割成 $K \times K$ 个网格，每个网格都要负责检测落于本网格的待检测目标物。每个网格要预测本网格中的待检测目标类别以及 n 个 bounding box 的预测信息，这 n 个预测信息分别表示各自网洛络中目标的位置信息：中心坐标(x, y)、长(h)、宽(w)和 confidence 值。而 confidence 值用来表示在网络的 box 中目标的置信度和预测的准确性(通过 ground region 的 IoU 体现)这两个数据。

$$\text{confidence} = \Pr(\text{object}) \times \text{IOU}(\text{pred}, \text{truth}) \tag{1}$$

2.4. 小结

RCNN 算法、Fast-RCNN 算法、SPP-net 算法、Faster-RCNN 算法都可以分为两个主要部分：region 划分和提取 regions。通过分割及 anchor 方法来代替大批量的 regions 是 YOLO 算法和 SSD 算法与上述算法的不同之处，相对而言 YOLO 算法及 SSD 算法的计算量小，使得模型训练时更快，满足机场飞行区实时目标检测的需求，因此本文将采用 YOLO 算法来进行实验。

3. YOLOv5 方法流程

3.1. YOLOv5 网络模型介绍

YOLOv5 在网络结构上和 YOLOv4 很相似，如图 1 所示，YOLOv5 网络结构分为四部分：输入端、Backbone、Neck、Prediction。其目标检测网络到目前为止主要有四个版本 YOLOv5s/m/l/x，其中 YOLOv5s 网络体积最小，速度最快，但是相应的精度(Average Precision, AP)也最低，深度、特征图的宽度也最小，其他的三种网络版本在 YOLOv5s 网络的基础上进行扩展，网络深度逐渐增加的同时对待检测目标的特征融合和提取能力也在逐渐增强。同时 YOLOv5 的四种不同网络在不断阶段的卷积核数量也不相同，卷积核的数量越多，网络的宽度越大。由于卷积操作需要大量运算，而卷积核越多网络中的卷积操作也就越多，因此卷积核的数量决定了网络所需要的计算量。这也就意味着一个网络的整体计算量越大也就是卷积核的数量越多其特征提取能力也就越强。更快的目标检测速度和更高的准确性是对机场飞行区目标检测模型的要求，虽然 YOLOv5 网络越复杂精度越高计算量也就越大，但随着近几年 GPU 算力的飞速发展，现在的计算机算力已经符合了这两个要求，因此本文也将基于 YOLOv5x 号模型进行研究。

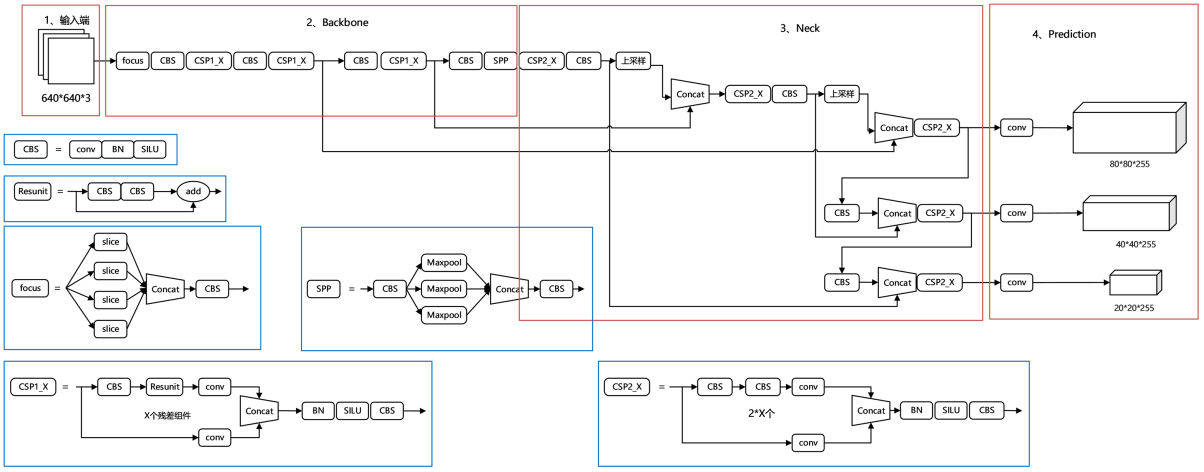

Figure 1. Network structure diagram
图 1. YOLOv5 网络结构图

3.2. 输入端数据处理

YOLOv5 在输入端改进了 CutMix 数据增强方式，采用了 Mosaic 数据增强方式。相较于 CutMix 数据增强方式，Mosaic 数据增强将图片数量从两张增加到四张，通过对四张图片进行拼接形成一张新的图片这种方式，丰富了待检测目标的背景并且极大地丰富了数据集。

YOLO 系列算法为不同的数据集设定了不同了初始锚框。YOLOv5 在 Coco 数据集[10]上设定的初始锚框为(116, 90, 156, 198, 373, 326)、(30, 61, 62, 45, 59, 119)、(10, 13, 16, 30, 33, 23)，网络训练的时候，会在初始锚框的基础上生成预测框，再将预测框同真实框对比，并根据对比的差距来修改参数，迭代网络模型参数。

另外，由于输入的图片大小都不尽相同，因此需要将所有的图片大小做统一化处理再送入检测网络，YOLOv5 的图片缩放方式采取自适应方式，具体缩放过程如图 2 所示。

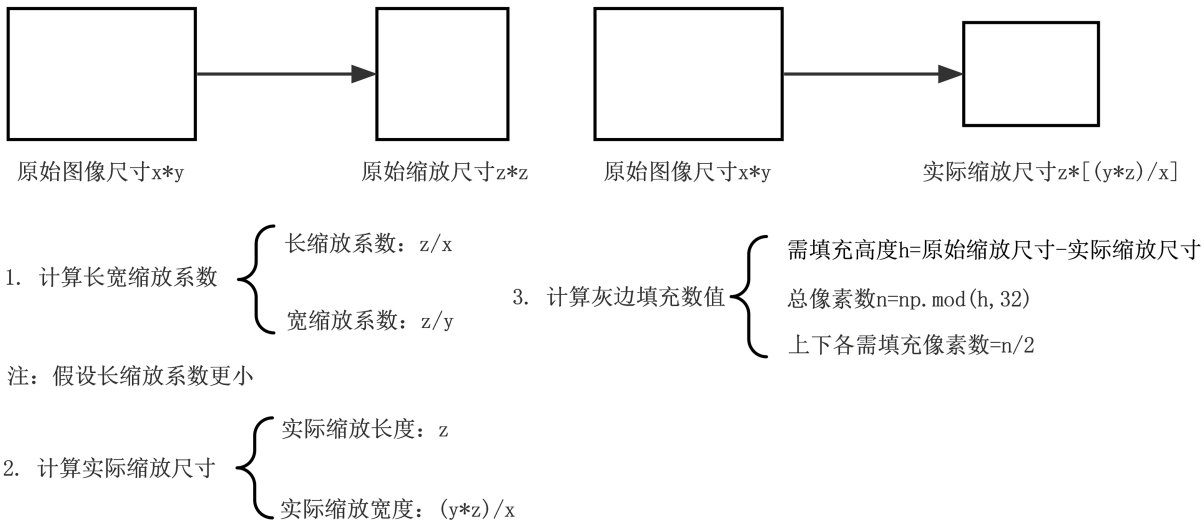

Figure 2. Adaptive scaling process
图 2. 自适应缩放过程

3.3. Backbone 部分

在主干部分 Backbone 中采用 Focus 结构，通过将原始图像进行切片操作，再经过卷积操作，得到需要的特征图，以 YOLOv5x 为例，Focus 结构最后使用了 80 个卷积核。同时在 YOLOv5 中设计了两种 CSP 结构。不同于 YOLOv4 中 CSP 结构只用在主干网络中，YOLOv5 的 CSP1_X 用于主干网络，CSP2_X 则用于 Neck 中。

3.4. Neck 部分

在 Neck 部分采用 FPN (feature pyramid networks)和 PAN (Path Aggregation Network)组合的结构，FPN 通过上采样的方式将网络中传递的上层特征信息进行融合并传递给网络中的下一层，是一种从上到下的结构，同时通过利用两个 PAN 结构，形成了一种从下向上传递强定位特征的结构，加强了网络的特征提取能力。

3.5. 输出端损失函数

在输出端采用 CIoU_Loss 做 Bounding box 的损失函数，在 DIoU_Loss 的基础上，增加了边界框长宽

比的尺度信息[6]。CIoU_Loss 损失函数见公式(2)、(3)、(4)所示，其中 b 代表预测框，b^{gt} 代表 GT 框的中心点，$R\left(B,B^{GT}\right)$ 代表预测框与 GT 框的惩罚项。c 代表预测框和 GT 框的最小外接矩形的对角线距离，ρ 为欧式距离。最后通过 NMS 对目标框进行处理，获得更好的检测效果。

下列公式为 CIoU_Loss 损失函数：

$$\text{CIOU\_Loss}=1+\text{IOU}+R\left(B,B^{GT}\right)+\frac{v^2}{\left(1-\text{IOU}\right)+v} \tag{2}$$

$$v=\frac{4}{\pi^2}\left(\arctan\frac{w^{gt}}{h^{gt}}-\arctan\frac{w^{p}}{h^{p}}\right)^2 \tag{3}$$

$$R\left(B,B^{GT}\right)=\frac{\rho^2\left(b,b^{gt}\right)}{c^2} \tag{4}$$

4. 实验

4.1. 数据集

实验数据集来自国内某机场真实视频监控数据，共计 14,930 张图片数据集示例如图 3 所示。

Figure 3. Dataset example
图 3. 数据集示例

对初始数据使用 YOLOv5x 模型进行初步识别，再进行人工筛选与标定后得到可用数据共 4457 张图片，共六类包括：特种车辆 truck、无动力设备 unpowered、人 person、飞机 airplane、摆渡车 bus、汽车 car。对数据集进行分析后得到的可视化结果如图 4 所示。

由可视化结果经分析可以看出 person 类的数量较少，反应了机场数据集的特征之一是人员活动不密集。从 label 大小分布可以看出 label 的大小主要分布在(0, 0)~(0.3, 0.4)之间，主要集中在(0.1, 0.07)之间，也反映了机场数据的另一个特点就是相对机场环境而言小目标较多。这两个特征将直接影响模型对目标检测的准确性，也是机场飞行区目标检测的难点。

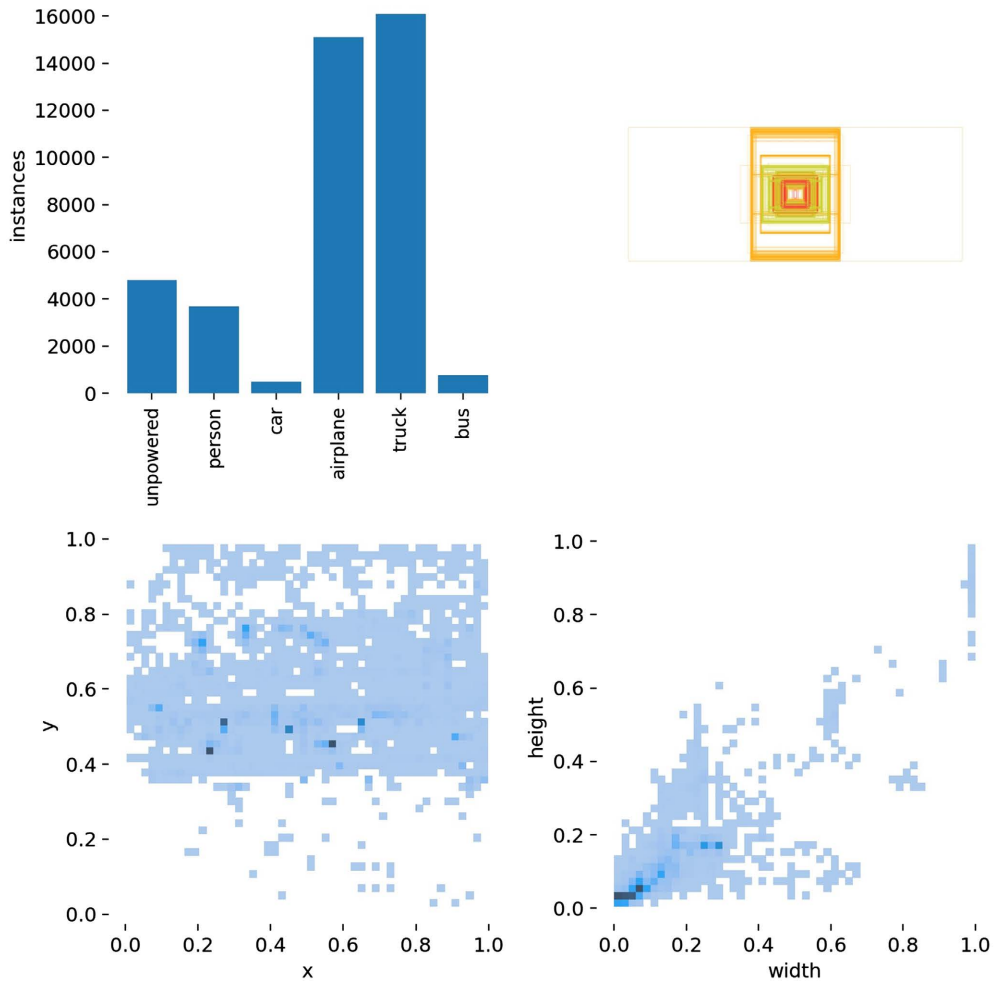

Figure 4. Number, size and center point distribution of labels
图 4. 标签数量、大小及中心点分布

4.2. 实验环境与模型训练

训练参数：预训练权重为 yolov5x.pt，输入图像尺寸为 640 × 640，最大迭代次数 300 轮，batch_size 按 GPU 规格设置为 32，训练线程设置为 8，其余参数默认。

本实验环境配置如表 1 所示。

Table 1. Experimental environment configuration
表 1. 实验环境配置

参数	配置
系统环境	Ubuntu 18.04.5 LTS
CPU	Intel®CoreTMi5-8400CPU@2.80GH
GPU	Quadro GV100
GPU 加速	CUDA10.2
训练框架	PyTorch
语言	Python3.6

4.3. 性能评估

本文算法性能指标将通过准确率(Accuracy)、精确率(Precision, P)、召回率(Recall, R)、平均精度均值(mean Average Precision, mAP)、PR 曲线等对模型进行评估。其中：

$$\text{Accuracy} = \frac{\text{TP}+\text{TN}}{\text{TP}+\text{TN}+\text{FP}+\text{FN}} \times 100\% \tag{5}$$

$$\text{Precision} = \frac{\text{TP}}{\text{TP}+\text{FP}} \times 100\% \tag{6}$$

$$\text{Recall} = \frac{\text{TP}}{\text{TP}+\text{FN}} \times 100\% \tag{7}$$

4.4. 实验结果

混淆矩阵(confusion matrix)是一种可以直观反映算法性能的可视化效果图，通过对混淆矩阵的分析可以直观的看到模型对于每一类目标的检测准确率，以及识别错误的主要原因。以下将通过分析混淆矩阵得到本文训练的模型的准确率以及影响准确率的主要因素。

由图 4 可知本实验训练的模型对于 unpowered、airplane、truck 的识别准确率较高，达到了 0.92 及以上的准确率。结合每一类目标在数据集中对应的数量可知，首先这三类目标的数量比较多，而且特征比较明显，因此模型在检测的过程中能够较好的区分和识别这三类物体。

person 类与背景的混合率较高是导致模型识别准确率较低的主要原因，是机场数据集的两个特征导致的，因此模型对于 person 类的识别准确率一般。

car 类与 truck 类以及背景的混合率较高，其识别率也比较低，原因在于 car 类在图片中的比例较小，且与 truck 的特征较为相似。

由图 5 数据可知，本实验训练的目标检测框架对于机场目标的检测有 0.807 mAP，精确率为 0.955。除 person 类外的其他类 mAP 值和精确率都相对较高，造成这一现象的主要原因在于机场环境的特殊性，人在该环境中相对较小其特征相对不明显，而且 YOLOv5 网络在训练时会进一步压缩图片，导致小目标的特征提取较困难。因此该目标检测框架基本能满足机场环境的目标检测需求，但仍存在改进空间。

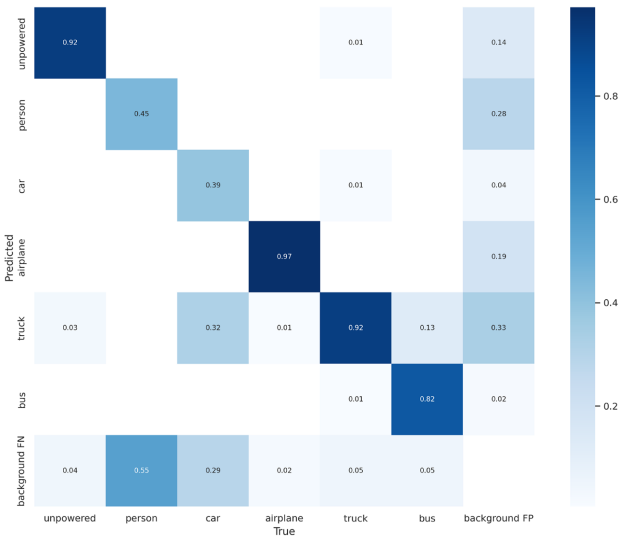

Figure 5. Confusion matrix
图 5. 混淆矩阵

部分目标检测效果图如图 6 所示。

Figure 6. Test result example
图 6. 检测结果示例

5. 总结

本文通过对现有主流目标检测算法的分析，提出了基于 YOLOv5 的深度学习模型，结合机场真实数据集，本文所训练的模型一定程度上改善了原有模型的目标检测准确率并且增加了新类别，进一步完善了 YOLOv5 机器学习模型对于机场环境下的目标检测能力。该模型在准确性和实时性方面达到了机场飞行区目标检测要求，但不足的是由于机场飞行区数据集的两个特征和算法本身的因素导致模型对于某些类的识别准确率较低，因此在后续的研究中将在这方面做进一步的研究和改进。

基金项目

中国民航大学 2021 年度大学生创新创业训练国家级项目(202110059020)资助。

参考文献

[1] Girshick, R., Donahue, J., Darrell, T. and Malik, J. (2014) Rich Feature Hierarchies for Accurate Object Detection and Semantic Segmentation. *Proceedings of the* 2014 *IEEE Conference on Computer Vision and Pattern Recognition*, Columbus, 23-28 June 2014, 580-587. https://doi.org/10.1109/CVPR.2014.81

[2] He, K., Zhang, X., Ren, S. and Sun, J. (2015) Spatial Pyramid Pooling in Deep Convolutional Networks for Visual Recognition. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, **37**, 1904-1916. https://doi.org/10.1109/TPAMI.2015.2389824

[3] Ren, S., He, K., Girshick, R. and Sun, J. (2015) Faster R-CNN: To-Wards Real-Time Object Detection with Region Proposal Networks. 28*th Conference on Neural Information Processing Systems*, Montreal, 8-13 December 2014, 91-99.

[4] Redmon, J., Divvala, S., Girshick, R. and Farhadi, A. (2016) You Only Look Once: Unified, Real-Time Object Detection. *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, Las Vegas, 27-30 June 2016, 779-788. https://doi.org/10.1109/CVPR.2016.91

[5] Liu, W., Anguelov, D., Erhan, D., Szegedy, C., Reed, S., Fu, C.-Y., *et al.* (2016) SSD: Single Shot Multibox *Detector*. *European Conference on Computer Vision*, Amsterdam, 8-16 October 2016, 21-37. https://doi.org/10.1007/978-3-319-46448-0_2

[6] Zheng, Z., Wang, P., Liu, W., Li, J., Ye, R. and Ren, D. (2020) Distance-IoU Loss: Faster and Better Learning for Bounding Box Regression. *Proceedings of the AAAI Conference on Artificial Intelligence*, **34**, 12993-13000. https://doi.org/10.1609/aaai.v34i07.6999

[7] Neubeck, A. and Van Gool, L. (2006) Efficient Non-Maximum Suppression. 18*th International Conference on Pattern Recognition* (*ICPR*'06), **3**, 850-855. https://doi.org/10.1109/ICPR.2006.479

[8] Erhan, D., Szegedy, C., Toshev, A. and Anguelov, D. (2014) Scalable Object Detection Using Deep Neural Networks. *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, Columbus, 23-28 June 2014, 2147-2154. https://doi.org/10.1109/CVPR.2014.276

[9] Krizhevsky, A., Sutskever, I. and Hinton, G.E. (2012) ImageNet Classification with Deep Convolutional Neural Networks. *Proceedings of the* 25*th International Conference on Neural Information Processing Systems*, Lake Tahoe, December 2012, 1097-1105.

[10] Lin, T.Y., Maire, M., Belongie, S., Hays, J., Perona, P., Ramanan, D., *et al.* (2014) Microsoft Coco: Common Objects in Context. *European Conference on Computer Vision*, Zurich, 6-12 September 2014, 740-755. https://doi.org/10.1007/978-3-319-10602-1_48