基于改进 YOLOv3 和卡尔曼滤波器的飞机检测追踪方法*

薛建伟[1]，史庆杰[2]，周泽强[2]，钱久超[1]，朱肖光[1]，刘佩林[1]

（1. 上海交通大学 北斗导航与位置服务上海市重点实验室·上海·201109；
2. 上海航天控制技术研究所·上海·201109）

摘　要：精确的飞机检测与追踪方法有助于提升我国军事实力，但是目前针对小目标飞机进行有效追踪的方法较少。基于深度学习的目标追踪方法较传统的方法性能更佳优越，因此针对传统方法对于小目标追踪性能不佳的问题，提出了一种基于 YOLOv3 以及卡尔曼滤波器的飞机追踪方法，以获得更好的追踪性能。该算法首先通过改进的 YOLOv3 算法对视频中的图像进行检测，在识别到视频中的飞机之后，通过卡尔曼滤波器对飞机的运动轨迹进行预测，并通过匈牙利算法进行数据关联。实验结果显示，该算法对小尺度飞机的检测性能较传统的 YOLOv3 有接近 5％的提升，且对飞机的追踪效果精度高且实时性好，具有较高的军事应用价值。

关键词：目标追踪；目标检测；飞机追踪；YOLOv3；卡尔曼滤波器

中图分类号：TP18　　**文献标志码**：A　　**文章编号**：2096-5974(2021)06-0070-08

Aircraft Detection and Tracking Method Based on Improved YOLOv3 and Kalman Filter

XUE Jianwei[1], SHI Qingjie[2], ZHOU Zeqiang[2], QIAN Jiuchao[1], ZHU Xiaoguang[1], LIU Peilin[1]

(1. Shanghai Key Laboratory of Navigation and Location Based Services, Shanghai Jiaotong University, Shanghai 201109;
2. Shanghai Aerospace Control Technology Institute, Shanghai 201109)

Abstract: Accurate aircraft detection and tracking methods can help improve a country's military strength, but there are few effective tracking methods for small target aircraft. The object tracking methods based on deep learning have better performance than traditional methods. Therefore, in view of the poor performance of traditional methods for small target tracking, this paper proposes an aircraft tracking method based on YOLOv3 and Kalman filter to obtain better performance. The improved algorithm first detects the target through the improved YOLOv3 algorithm, after that, the Kalman filter is used to predict the aircraft's trajectory, and the Hungarian algorithm is used for data association. Experimental results show that the detection performance of the algorithm for small-scale aircraft is improved by nearly 5% compared with the traditional YOLOv3, and the tracking effect of the aircraft has high accuracy and real-time performance, which has high military application value.

Keywords: object tracking; object detection; aircraft tracking; YOLOv3; Kalman filter

0　引　　言

目标追踪在计算机视觉中是一个非常具有挑战性的问题。传统的追踪方法主要是对视频中每一帧的数据进行边框标记，则所有边框的运动轨迹可以近似看作目标的运动轨迹。近些年来，目标追踪研究发展得非常迅速，已经从单目标追踪扩展到多目标追踪。但是在目标追踪领域中仍然

* 收稿日期：2020-09-19；修回日期：2021-08-12
基金项目：上海航天技术研究院-上海交大航天先进技术联合研究基金（USCAST2019-26）
作者简介：薛建伟（1994—），男，博士生。E-mail：xuejw2019@sjtu.edu.cn

存在许多挑战性问题，例如目标物的姿态变化、遮挡、背景模糊、光照变化等问题。

基于传统的追踪方法，经过多年的发展已经有了较大的进步，在一定程度上可以获得较好的追踪效果，例如，文献［1］通过特征空间以及压缩采样进行目标追踪；文献［2］提出了一种滤波器用于缓解在追踪过程中遮挡带来的影响，并通过蜂群优化算法进行多目标追踪；文献［3］采用一种核变化的稀疏学习方法设计追踪器，可以通过学习获得目标物的多个特征，并以此进行追踪。虽然上述方法可以在一定程度上实现追踪的效果，但是设计的追踪器不能充分学习到图像中目标物体的特征，因此对目标的追踪效果并不是特别稳定。

为了更好地理解图像中的信息，即提取图像特征点信息，学术界与工业界开始使用基于深度学习的方法对图像的特征点进行学习，这些方法可以自动对图像中的有效特征信息进行提取。目前，基于深度学习的方法已经广泛应用于各种领域，如计算机视觉、语音信号处理、自然语言处理等。很多学者提出了多种不同的深度学习方法对图像中的目标物体进行检测，之后再对目标添加运动估计算法进行追踪。H. Nam 等[4]提出了一种创新的多领域学习框架网络，能够在预训练阶段得出与领域无关的表示，并通过在线学习获得与领域相关的消息。文献［5］将全卷积网络嵌入到简单的追踪算法中，极大地提高了检测性能并且降低了计算时间。文献［6］提出了一种无需在线学习的前馈神经网络的追踪器。上述这些基于深度学习的追踪器，在追踪效果上得到了极大的提升，但同时其运算时间也明显增加。

近年来，关于目标的检测与追踪已有一些相关的研究[7-11]，但是其主要集中在行人的检测和追踪应用中，对于飞机的追踪研究较少。飞机作为一种重要的运输工具以及军事装备，有效地对其进行追踪具有十分重要的现实意义。YOLO 算法作为一种高效的识别算法，对于目标的检测识别应用有着十分优秀的性能[12-15]，可以作为目标追踪中检测阶段的目标检测器，用于提供追踪目标的初始位置。为了实现全天候的飞机追踪，本文使用红外飞机数据集，并针对在飞机追踪场景中飞机存在多角度、多尺度等情况，且需要满足实时性和准确性的需求，采用改进的 YOLOv3 目标检测算法，实现对多种情况下的高精度飞机检测。在此基础上，提出了一种结合卡尔曼滤波器的飞机追踪方法，实现了对红外飞机的准确跟踪。

1　算法设计

1.1　基于改进 YOLOv3 和卡尔曼滤波器的目标追踪算法流程

本文提出了一种基于检测的跟踪算法，其框架图如图 1 所示。当将视频图像输入算法时，首先通过改进的 YOLOv3 算法对图像进行目标检测，得到视频中所有物体的位置及边框大小。之后通过卡尔曼滤波算法对目标的运动状态进行预测，即通过视频中上一帧的目标物体的位置及速度等参数，预测出当前帧目标框的位置和速度等参数，并将预测的目标物边框与前一帧中目标边框通过匈牙利算法进行匹配。此时如果可以正确匹配，则将状态进行更新；如果发现不能进行匹配，则确定图像中是否存在新目标，如果存在则新增一条运动轨迹，否则删除该预测信息，显示当前的检测结果。

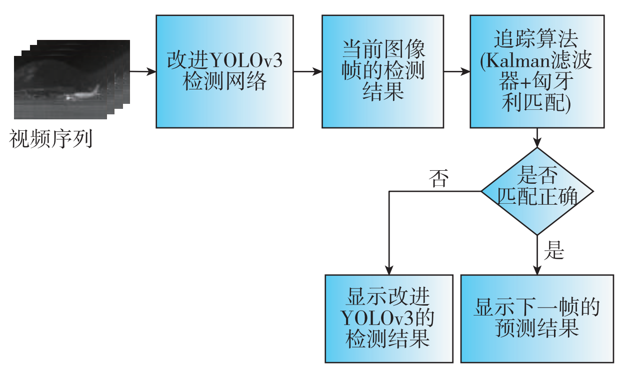


图 1　算法流程图

Fig. 1　Framework of the algorithm

1.2　YOLOv3 算法及改进

根据算法框架图可知，为了更好地实现系统的追踪，首先需要对视频中的目标进行有效的检测，因此本文选用了目前比较流行的 YOLOv3 目标检测算法对视频中的飞机进行检测。YOLOv3 是 J. Redmon 等于 2018 年提出的一种单阶段的目标检测算法[9]，其结合了残差网络、特征金字塔以及多特征融合等方法，具有非常好的识别速度和检测精确度。其原型网络以 Darknet53 为骨干网络架构，通过进行 32 倍、16 倍、8 倍降采样获取三种尺度下

的特征图，以适应不同尺度目标的检测输出；同时为了减少底层目标特征的损失，通过对顶层特征图的 2 次上采样并与底层特征图进行融合，增强了对不同尺度目标的适应性，其结构如图 2 所示。

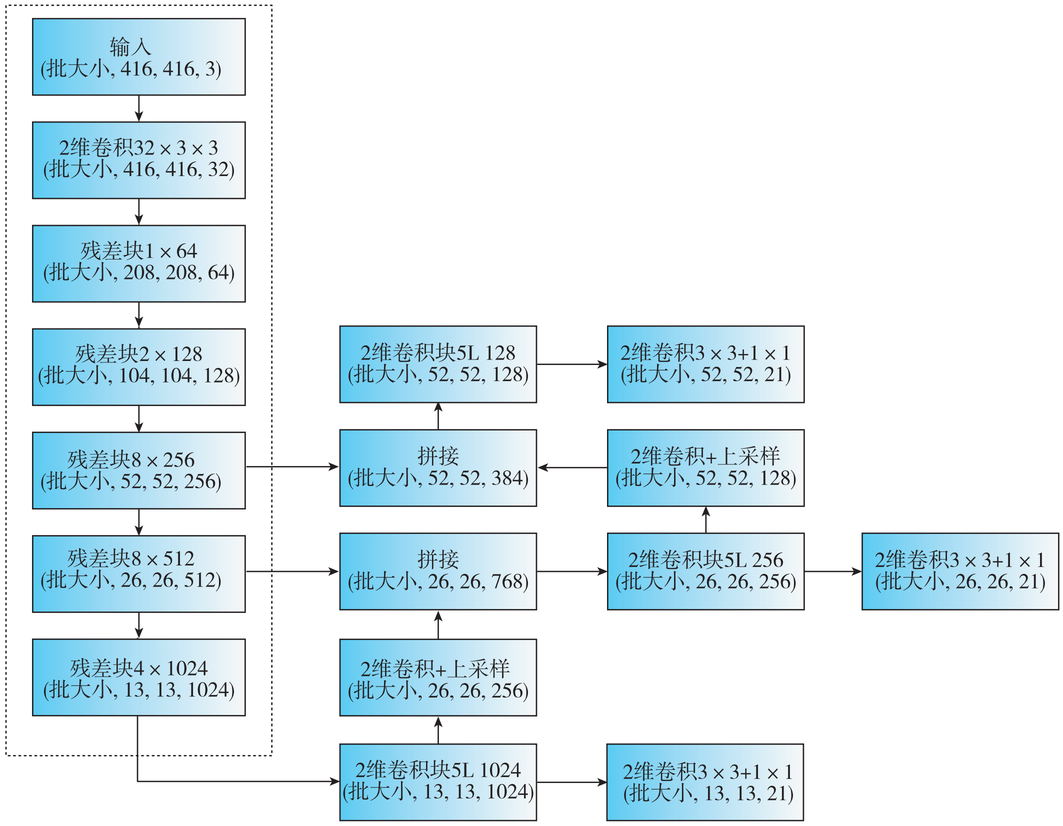


图 2 YOLOv3 的网络结构

Fig. 2 YOLOv3 network structure

从 YOLOv3 的网络结构可以看出，其使用了大量的卷积结构，在卷积结束后使用 L2 正则化处理，并使用 Leaky Relu 函数进行激活处理。此外，YOLOv3 使用了多尺度特征进行预测，其尺寸分别为（52，52，256）、（26，26，512）和（13，13，1024），并对特征层进行卷积处理用于预测。结果显示，YOLOv3 算法具有很好的检测效果，并且可以对不同尺度的目标进行检测输出。但是本文中使用的飞机数据集，包含各种尺度大小的飞机，并且为了保证夜晚的检测能力，还使用了红外数据集，其分辨率相对较低，因此如果继续使用上述算法，将会导致目标物体的信息量随着层数的增加而急剧下降或者消失，造成目标漏检。

为了实现对飞机的全天候追踪，首先需要 YOLOv3 实现对飞机的全天候检测，因此需要提高网络对于红外图像下不同尺度目标的识别能力。本文通过将图像中底层和深层特征图相结合，以提高算法对红外图像中目标检测的效果，其具体方法如下：

1）将网络结构中第 103 层的特征图通过 1×1 卷积层进行特征信息的融合，通道数从 128 降到 64，并输出到 108 层，以精简结构。

2）对网络结构中 108 层的特征图进行上采样，从 52×52 扩展到 104×104，并在网络结构的 109 层降低目标的感受野尺度，以提高对小目标检测的能力。

3）将网络结构中 11 层的特征图与 109 层的特征图进行融合，由于 11 层的特征信息属于底层的特征，经过多次的卷积处理，其保存了小目标物体更多的细节，将两层融合可以带来更多的信息。

4）在网络结构第 81 层、93 层、105 层、117 层中，其输出的尺度大小分别为 13×13、26×26、53×53、104×104，并通过逻辑回归对不同尺度情况下的目标分别进行检测，减少漏检、虚警等情况。

1.3　卡尔曼滤波追踪算法

本文使用卡尔曼滤波器[9]对运动过程中的状态变量进行最优估计。卡尔曼滤波器算法是用一组数学方程式来表示的，其可以对运动物体的状态进行估计。在估计的过程中，将观察到的目标物体的状态值提供给卡尔曼滤波器以对其进行预测，被预测的物体的状态可以由式（1）进行确定

$$\boldsymbol{x}=[u,v,s,r] \tag{1}$$

其中，u 和 v 分别表示目标物当前的中心点的横坐标和纵坐标；s 和 r 分别表示目标边框的长宽比。卡尔曼滤波器模型假设第 k 时刻的状态是由 $k-1$ 时刻的状态演化而来，其可以表示为

$$\boldsymbol{x}_k=\boldsymbol{A}_k\boldsymbol{x}_{k-1}+\boldsymbol{B}_k\boldsymbol{u}_k+\boldsymbol{w}_k \tag{2}$$

其中，$\boldsymbol{x}_k$ 是第k 时刻的状态量；$\boldsymbol{u}_k$ 是第k 时刻输入的外部控制信号；$\boldsymbol{w}_k$ 是在 k 时刻处理过程中的噪声信号；$\boldsymbol{A}_k$ 是第 k 时刻的状态转移矩阵；$\boldsymbol{B}_k$ 是第 k 时刻的输入控制信号模型。此外，在时刻 k，对真实状态 $\boldsymbol{x}_k$ 的一个测量值 $\boldsymbol{z}_k$ 满足以下公式

$$\boldsymbol{z}_k=\boldsymbol{H}_k\boldsymbol{x}_k+\boldsymbol{v}_k \tag{3}$$

其中，$\boldsymbol{v}_k$ 是观察噪声；$\boldsymbol{H}_k$ 是观察模型。假设随机变量 $\boldsymbol{w}_k$ 和 $\boldsymbol{v}_k$ 服从正态分布、均值为 0、协方差矩阵分别为 $\boldsymbol{Q}$ 和 $\boldsymbol{R}$ 的多元正态分布，即 $\boldsymbol{w}_k\sim N(0,\boldsymbol{Q})$，$\boldsymbol{v}_k\sim N(0,\boldsymbol{R})$。卡尔曼滤波算法是一种迭代的状态估计算法，通过之前的状态估计值以及新的观测值获得当前时刻的最佳状态估计值 $\hat{\boldsymbol{x}}$。这一过程由 2 个阶段组成：预测阶段以及状态更新阶段，其中预测阶段包括式（4）和式（5），更新阶段为式（6）～式（8）

$$\hat{\boldsymbol{x}}_{k|k-1}=\boldsymbol{A}_k\hat{\boldsymbol{x}}_{k-1}+\boldsymbol{B}_k\boldsymbol{u}_k \tag{4}$$

$$\boldsymbol{P}_{k|k-1}=\boldsymbol{A}_k\boldsymbol{P}_{k-1|k-1}\boldsymbol{A}_k^{\mathrm{T}}+\boldsymbol{Q} \tag{5}$$

$$\boldsymbol{K}_k=\boldsymbol{P}_{k|k-1}\boldsymbol{H}_k^{\mathrm{T}}(\boldsymbol{H}_k\boldsymbol{P}_{k|k-1}\boldsymbol{H}_k^{\mathrm{T}}+\boldsymbol{R})^{-1} \tag{6}$$

$$\hat{\boldsymbol{x}}_{k|k}=\hat{\boldsymbol{x}}_{k|k-1}+\boldsymbol{K}_k(\boldsymbol{z}_k-\boldsymbol{H}_k\hat{\boldsymbol{x}}_{k|k-1}) \tag{7}$$

$$\boldsymbol{P}_{k|k}=(\boldsymbol{I}-\boldsymbol{K}_k\boldsymbol{H}_k)\boldsymbol{P}_{k|k-1} \tag{8}$$

其中，$\boldsymbol{P}_{k|k-1}$ 为预测估计协方差矩阵；$\boldsymbol{P}_{k|k}$ 为更新后的协方差估计矩阵；$\boldsymbol{K}_k$ 为最优卡尔曼增益。在观测 $\boldsymbol{z}_k$ 之前计算最佳状态 $\hat{\boldsymbol{x}}_{k-1}$，更新阶段时根据观察的 $\boldsymbol{z}_k$ 计算最佳状态 $\hat{\boldsymbol{x}}_k$。

1.4　多目标数据关联

在对图像中的各个检测目标进行预测之后，数据关联是一个非常重要的部分，其目的是确定当前预测的目标与之前数据帧中目标的对应关系，或者检测新的图像中是否出现了一个新的对象。当新物体进入框架，但是系统未检测到该物体或者物体超出了图像的范围，在这种情况下，检测算法可能会产生虚警值（即检测不存在的对象）或者预测位置与实际位置差异很大。

假设检测出的物体可以用 $\boldsymbol{x}_{nt}$ 表示，其中 n 表示视频中第 n 个目标物体，t 表示对应的图像序列，其取值范围为[1，i]，其中 i 表示最大的数据帧数，且 $\boldsymbol{X}=(\boldsymbol{x}_{11}，\boldsymbol{x}_{21}，\cdots，\boldsymbol{x}_{n1}，\boldsymbol{x}_{12}，\cdots，\boldsymbol{x}_{1i}，\boldsymbol{x}_{2i}，\cdots，\boldsymbol{x}_{ni})$。对于观测的有序序列，即 $\boldsymbol{T}_k=(\boldsymbol{x}_{k1}，\boldsymbol{x}_{k2}，\cdots，\boldsymbol{x}_{ki})$，这些序列构成单个目标的运动轨迹信息，其中任意时刻的观测状态 $\boldsymbol{x}_{ki}\in\boldsymbol{X}$。关联假设 $\boldsymbol{T}$ 定义为一组单目标轨迹的集合，即 $\boldsymbol{T}=\boldsymbol{T}_k$，数据关联的目的是在给定观测集 $\boldsymbol{X}$ 的情况下最大化 $\boldsymbol{T}$ 的后验概率

$$T^*=\boldsymbol{T}P(\boldsymbol{T}|\boldsymbol{X}) \tag{9}$$

$$T^*=\boldsymbol{T}P(\boldsymbol{X}|\boldsymbol{T})P(\boldsymbol{T}) \tag{10}$$

$$T^*=\boldsymbol{T}\prod_i P(\boldsymbol{T}|\boldsymbol{X})P(\boldsymbol{T}) \tag{11}$$

其中，T^* 为最佳的预测值；$P(\boldsymbol{T}|\boldsymbol{X})$ 为给定当前观测集 $\boldsymbol{X}$ 时当前轨迹集合 $\boldsymbol{T}$ 发生的概率；$P(\boldsymbol{T})$ 为当前轨迹集合 $\boldsymbol{T}$ 发生的概率；$P(\boldsymbol{X}|\boldsymbol{T})$ 为给定当前轨迹集合 $\boldsymbol{T}$ 时给定观测集 $\boldsymbol{X}$ 发生的概率。

为此，本文采用匈牙利算法将检测的边框与预测的轨迹边框进行最佳匹配。此时将交并比（Intersection over Union，IoU）作为损失函数进行优化。在匈牙利算法中，假设其损失矩阵为 $\boldsymbol{C}$，其中元素 C_{ij} 表示第i 个检测的边框与第j 个预测的边框的交并比。为了计算预测和检测之间的损失矩阵 $\boldsymbol{C}$，将交并比定义为 IOU，其计算公式如式（12）所示，其中 $\boldsymbol{A}$ 和 $\boldsymbol{B}$ 分别表示图像中目标的检测框和预测框

$$IOU=\frac{|\boldsymbol{A}\cap\boldsymbol{B}|}{|\boldsymbol{A}\cup\boldsymbol{B}|} \tag{12}$$

为了最大程度地降低损失矩阵，需设定一个最大的检测关联的阈值，当超出该阈值时，不会追踪新的目标；如果经过一段时间的遮挡之后其重新出现，则重新对该目标进行检测以及跟踪预测。此外，使用交并比作为损失函数，在一定程度上缓解了短期遮挡带来的跟踪误差。

2　实验结果与分析

2.1　红外训练数据集

当前针对军事环境中的飞机检测追踪场景，识别是追踪的第一阶段，因此如何进行识别是非

常重要的一环，而且如果想实现全天候的追踪，首先需要实现全天候的识别，这对夜间红外探测能力提出了较高的要求。随着深度学习的发展以及各类开源数据库的兴起，如 ImageNet、COCO 等开源图像数据库得到了广泛的应用，给相关的算法研究带来了极大的便利。但是开源数据库中一般只包含普通场景下的照片用于检测，红外图像的资源较少，而且用于飞机的红外图像数据也比较少，给识别红外飞机图像的检测带来了困难。同时，由于红外探测的成像机理不同，导致同一目标的纹理、形状信息较少，分辨率普遍较低，容易受到噪声的影响，不利于红外图像中目标特征的提取。本文采用的数据集是针对飞机滑行、起飞、降落等条件下的红外数据，并通过 LabelImag 软件进行标注，图 3 所示为本文中使用的部分数据集照片，可以看到红外图像数据的清晰度较低，且与周围环境区分不明显。本文用于检测的训练集为 500 张红外飞机图像，验证集为随机的 200 张红外飞机图像，测试集为 500 张红外飞机图像。

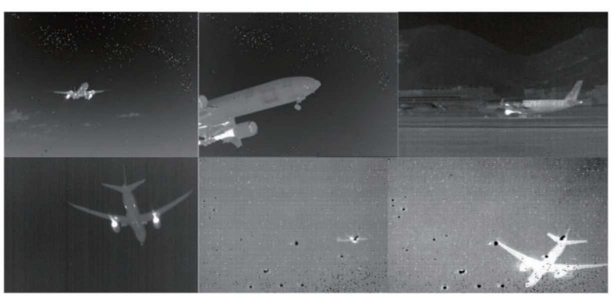


图 3 目标检测数据集

Fig. 3 Target detection data set

2.2 模型训练

在进行飞机检测模型的训练时，直接通过 YOLOv3 网络架构进行训练需要耗费大量的时间，且由于硬件资源计算能力有限以及数据集样本较少，从头进行训练不仅速度慢而且容易产生过拟合现象。为加快模型的训练速度，增强模型泛化能力，本文通过迁移学习将已有的预训练模型作为本次训练的初始化权重，并在使用的红外数据集上进行模型训练。

本次实验的硬件平台主要采用的 CPU 为 i7-8700K，内存为 16G，显卡为 NVIDIA RTX2070。软件平台为 Ubuntu16.04 系统环境，Python 3.6.9 作为开发语言，采用 NVIDIA 推出的 CUDA（Compute Unified Device Architecture）并行计算架构和 CUDNN（CUDA Deep Neural Network）深度学习专用加速库，对检测算法的训练和推理进行加速，模型训练方式采用 Adam 优化器对网络进行优化。

分别使用 YOLOv3 算法以及改进的 YOLOv3 算法对红外数据集进行网络的训练，分别记录模型过程中损失值的变化，为了防止过拟合，训练过程中使用了早停法，其损失函数曲线如图 4 所示。从图 4 中可以看出，改进之后的检测算法其损失值下降更快，可以更快地收敛到稳定值，证明了改进算法在红外飞机图像检测中的有效性。

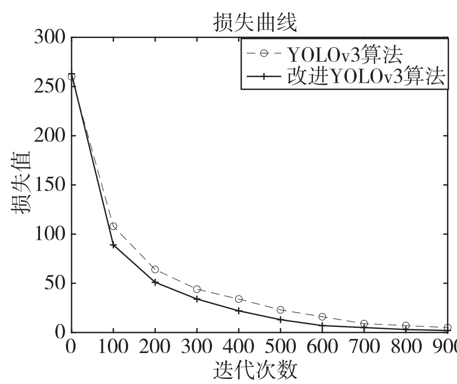


图 4 损失值变化曲线

Fig. 4 Loss change curve

2.3 检测结果分析

在本文中，使用识别的精确率（$P_{\text{precision}}$）和召回率（P_{recall}）作为检测算法性能的评价指标，其定义分别为

$$P_{\text{precision}} = \frac{TP}{TP + FP} \tag{13}$$

$$P_{\text{recall}} = \frac{TP}{TP + FN} \tag{14}$$

其中，TP 表示模型可以正确预测的飞机样本个数；FP 为模型误检的飞机样本个数（实际不存在）；FN 为检测到的飞机样本个数。此外，基于精确率和召回率，使用综合评价指标 F 进行评估，其表达式为

$$F = \frac{2 \times P_{\text{precision}} \times P_{\text{recall}}}{P_{\text{precision}} + P_{\text{recall}}} \tag{15}$$

为了验证改进 YOLOv3 算法的有效性，使用了两种算法对飞机图像进行识别，实验结果如表 1

所示。根据结果可知，改进后的 YOLOv3 模型在红外飞机图像的检测任务中较 YOLOv3 算法的准确度更高。

表 1　不同模型的检测对比结果

Tab. 1　Comparison of detection results of different model

组号	算法	$P_{precision}$/%	P_{recall}/%	F/%
1	YOLOv3	78.4	79.2	78.9
2	改进 YOLOv3（本文检测算法）	83.7	82.3	83.1

如表 1 所示，改进的 YOLOv3 检测算法对样本数据的检测准确率可以达到 83.7%，召回率可以达到 82.3%，较 YOLOv3 检测算法在本数据集上测试的检测准确率提高了 5.3%，召回率提高了 3.1%。图 5 所示为改进的 YOLOv3 算法对数据集中飞机在不同尺度、不同场景下进行检测的结果示意图。改进 YOLOv3 模型通过增加网络模型的感受野，增加了网络对于红外图像特征的检测和利用能力，提高了模型检测的平均准确率，因此也为进一步进行检测跟踪的算法模型提供了良好的条件。

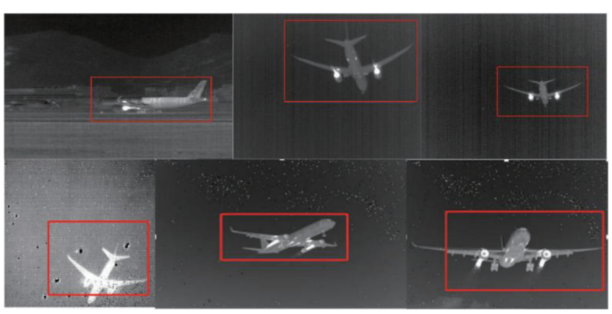


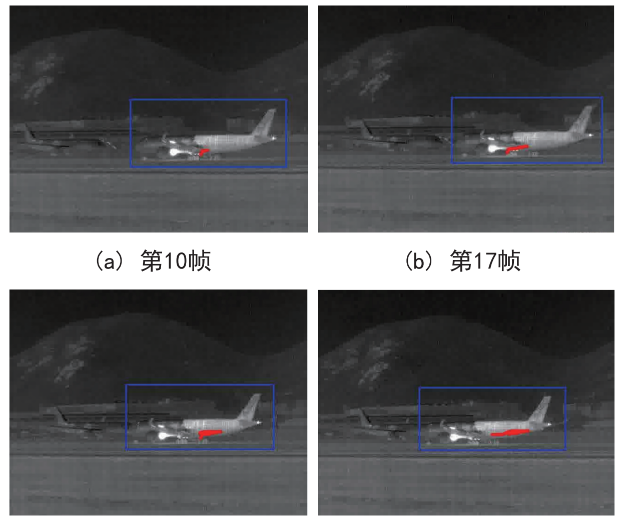
图 5　检测结果

Fig. 5　Detection results

2.4　跟踪效果分析

本文采用了改进的 YOLOv3 与卡尔曼滤波器算法进行飞机的跟踪实验。在视频测试中，运行了本文提出的基于改进的 YOLOv3 和卡尔曼滤波器的追踪算法与 YOLOv3 和卡尔曼滤波器的追踪算法的对比实验，对测试集中的 20 段红外飞机视频进行跟踪测试。图 6 和图 7 所示分别为针对飞机滑行状态以及飞行状态的追踪效果，并分别选取视频中的第 10 帧、第 17 帧、第 34 帧及第 70 帧的检测效果进行显示对比。

图 6 所示为本文追踪算法对于飞机滑行状态时的跟踪效果图，其中蓝色框为通过改进 YOLOv3 算法检测所得到的飞机的目标框，红色的线为算法追踪到的轨迹示意图。分别选取飞机滑行时视频数据中的第 10 帧、17 帧、34 帧以及 70 帧的追踪效果图，由于飞机在滑行时运动轨迹平稳，本文的追踪算法对滑行状态时的飞机具有可靠的追踪效果。图 7 所示为本文追踪算法对于飞机飞行状态时的跟踪效果图，其中蓝色框和红色线的意义同图 6，由于飞机在飞行状态时飞行较

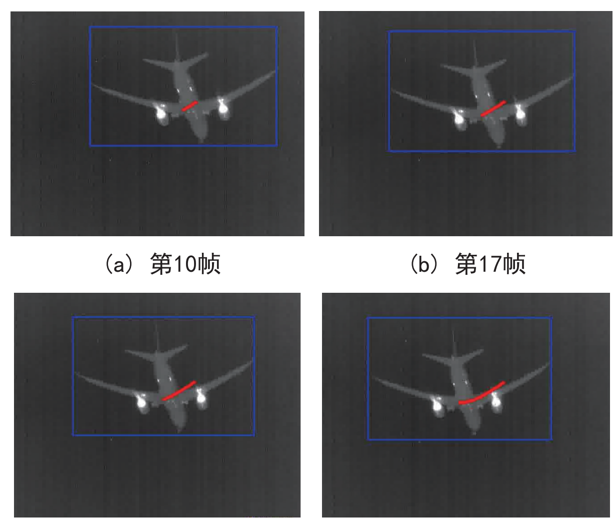


(a) 第10帧　(b) 第17帧

(c) 第34帧　(d) 第70帧

图 6　滑行状态时的追踪结果

Fig. 6　Tracking results when taxiing



(a) 第10帧　(b) 第17帧

(c) 第34帧　(d) 第70帧

图 7　飞行状态时的追踪结果

Fig. 7　Tracking results during flight status

平稳，因此本文的追踪算法具有可靠的追踪效果。

针对 20 段 100 帧的视频数据，其中包含不同角度、不同尺度的飞机运动轨迹。本文以飞机追踪的准确性评价追踪方法的性能，通过 YOLOv3 和卡尔曼滤波器的检测追踪算法与本文提出的改进 YOLOv3 和卡尔曼滤波器的检测追踪算法在数据集中进行测试，算法的对比结果如表 2 所示。从表 2 可以看出，本文提出的方法在检测效果上较未改进的 YOLOv3 结合卡尔曼滤波器的追踪算法而言，追踪的准确率提升了 10%，但在追踪的帧率上提升并不明显，平均帧率只提升了 2 帧。这是由于这些方法都是在基于检测的效果下进行追踪的，检测的时间占据主要的因素，因此检测的性能决定了追踪的准确率及追踪帧率。本文提出的改进结构通过提高感受野，提升了对小角度、小目标的飞机的检测效果，因此带来了一定的追踪性能的提升。整体而言，本文提出的方法可以满足国防军事场景下对红外飞机图像的检测和跟踪的应用需求。

表 2 不同算法的追踪结果对比

Tab. 2 Comparison of tracking results of different algorithms

方法	视频数量	算法追踪数量	准确率/%	帧率/（帧/s）
YOLOv3+卡尔曼滤波器	20	15	75	22
改进 YOLOv3+卡尔曼滤波器	20	17	85	24

3 结 论

本文研究了基于改进 YOLOv3 和卡尔曼滤波器的飞机追踪算法，首先通过改进的 YOLOv3 算法提高了网络结构的感受野，使其可以更好地检测不同尺度大小的飞机，以提供飞机的位置信息。之后结合卡尔曼滤波器对飞机的位置信息进行预测，并采用匈牙利算法对其进行匹配，进而实现追踪。仿真实验证明了本文提出的方法对于飞机的识别性能有了较大的提升，达到了预期追踪效果，且本文算法的帧率可以达到实时性要求，满足军事领域中对于小角度、小目标的飞机的识别和追踪的需求。此外，由于红外视频的特性，该方法仍然存在多种未检测到目标而导致追踪失败的情况，下一步将继续增加训练样本的数量，以提高系统的性能。

参考文献（References）

[1] JIANG B, YANG J, LYU Z, et al. Wearable vision assistance system based on binocular sensors for visually impaired users [J]. IEEE Internet of Things Journal, 2019, 4: 1375-1383.

[2] LI J, WANG J. Adaptive object tracking algorithm based on eigen basis space and compressive sampling [J]. IET Image Process, 2012, 6 (8): 1170-1180.

[3] LIN S D, LIN J J, CHUANG C Y. Particle filter with occlusion handling for visual tracking [J]. IET Image Process, 2015, 9 (11): 959-968.

[4] KANG B, ZHU W P, LIANG D. Robust multi-feature visual tracking via multi-task kernel-based sparse learning [J]. IET Image Process, 2017, 11 (12): 1172-1178.

[5] NAM H, HAN B. Learning multi-domain convolutional neural networks for visual tracking [C] //Proceedings of IEEE Conference on Computer Vision and Pattern Recognition. Las Vegas, USA: IEEE, 2016: 4293-4302.

[6] MA C, HUANG J B, YANG X, et al. Hierarchical convolutional features for visual tracking [C] //Proceedings of IEEE Conference on Computer Vision. Santiago, Chile: IEEE, 2015: 3074-3082.

[7] BERTINETTO L, VALMADRE J, HENRIQUES J F, et al. Fully-convolutional Siamese networks for object tracking [C] //Proceedings of European Conference on Computer Vision. Amsterdam, Netherlands: IEEE, 2016: 850-865.

[8] HELD D, THRUN S, SAVARESE S. Learning to track at 100 fps with deep regression networks [C] //Proceedings of European Conference on Computer Vision. Amsterdam, Netherlands: IEEE, 2016: 749-765.

[9] REDMON J, FARHADI A. YOLOv3: an incremental improvement [C] // Proceedings of 2018 IEEE Conference on Computer Vision and Pattern Recognition. Washington, USA: IEEE, 2018: 1-6.

[10] 宋焕生，李莹，杨瑾，等. 基于高速公路场景的车辆目标跟踪 [J]. 计算机系统应用，2019，28 (6)：82-88.
SONG H S, LI Y, YANG J, et al. Vehicle target tracking based on highway scene [J]. Computer System Application, 2019, 28 (6): 82-88 (in Chinese).

[11] CHENG Y, JIN Z G, GAO T, et al. An improved collaborative representation based classification with regularized least square (CRC - RLS) method for robust face recognition [J]. Neurocomputing, 2016, 215: 250-259.

[12] 杨恺，徐友春，安相璧，等. 基于深度学习的车辆检测方法 [J]. 计算机与网络，2018，44 (19)：58-61.
YANG K, XU Y C, AN X B, et al. Vehicle detection method based on deep learning [J]. Computer and Network, 2018, 44 (19): 58-61 (in Chinese).

［13］ SANDLER M，HOWARD A，ZHU M L，et al. MobileNetV2：inverted residuals and linear bottlenecks［C］// Proceedings of IEEE Conference on Computer Vision and Pattern Recognition（CVPR）. Salt Lake City，USA：IEEE，2018：4510-4520.

［14］ REDMON J，FARHADI A. YOLO9000：better，faster，stronger［C］// Proceedings of IEEE Conference on Computer Vision and Pattern Recognition（CVPR）. Honolulu，USA：IEEE，2017：7263-7271.

［15］ TIAN Y N，YANG G D，WANG Z，et al. Apple detection during different growth stages in orchards using the improved YOLO-V3 model［J］. Computers and Electronics in Agriculture，2018，157：417-426.