第 59 卷 第 4 期/2022 年 2 月/激光与光电子学进展 研究论文

激光与光电子学进展 先进成像

基于自适应空间特征融合的轻量化目标检测算法

罗禹杰，张剑*，陈亮，张侣，欧阳婉卿，黄代琴，杨羽翼

湖南科技大学信息与电气工程学院，湖南 湘潭 411100

摘要 针对目前深度学习中单阶段目标检测网络结构复杂、训练困难与在移动与嵌入式设备难以部署的问题，提出了一种基于自适应空间特征融合的轻量化目标检测算法。所提算法以YOLOv4为网络基础框架，采用轻量级MobileNet作为特征提取网络，降低网络深度与训练难度，提高检测速度；采用一种自适应空间特征融合(ASFF)方式改进PANet对多尺度特征融合效果差的不足；通过增加网络的输出维度，利用Gaussian算法对新增维度建模并输出预测框位置的不确定性；最后对位置损失函数进行重新定义，提高位置回归的准确性。所提算法以疫情期间口罩佩戴检测机器人为部署载体，对人脸口罩佩戴情况进行了测试，实验结果表明，所提算法的检测精度达到了95.92%，检测速度达到了19 frame/s，相比于原始算法和其他主流检测算法，更适合部署于移动与嵌入设备实现实时检测。

关键词 机器视觉；模式识别；特征提取网络；特征融合；损失函数

中图分类号 TP391.4 **文献标志码** A **doi**：10.3788/LOP202259.0415004

Lightweight Target Detection Algorithm Based on Adaptive Spatial Feature Fusion

Luo Yujie, Zhang Jian*, Chen Liang, Zhang Lü, Ouyang Wanqing, Huang Daiqin, Yang Yuyi

School of Information and Electrical Engineering, Hunan University of Science and Technology, Xiangtan, Hunan 411100, *China*

Abstract Aiming at the problems of complex network structure, difficult training and difficult deployment in mobile , and embedded devices of single-stage target detection in deep learning, a lightweight target detection algorithm based on adaptive spatial feature fusion is proposed. The proposed algorithm takes YOLOv4 as the basic framework of the network and uses lightweight MobileNet as the feature extraction network to reduce the network depth and training difficulty and improve the detection speed; an adaptive spatial feature fusion (ASFF) method is used to improve the poor effect of PANet on multi-scale feature fusion; by adding the output dimension of the network, the Gaussian algorithm is used to model the new dimension and output the uncertainty of the position of the prediction box; finally, the position loss function is redefined to improve the accuracy of position regression. The proposed algorithm takes the mask wearing detection robot during the epidemic as the deployment carrier to test the face mask wearing. The experimental results show that the detection accuracy of the proposed algorithm reaches 95.92% and the detection speed reaches 19 frame/s. Compared with the original algorithm and other mainstream detection algorithms, the proposed algorithm is more suitable for deployment in mobile and embedded devices to realize real-time detection.

Key words machine vision; pattern recognition; feature extraction network; feature fusion; loss function

收稿日期：2021-02-18；**修回日期**：2021-03-19；**录用日期**：2021-04-06

基金项目：国家自然科学基金项目(61972443)、湖南省自然科学基金(2020JJ5170)、湖南省教育厅一般项目(180C0299)

通信作者：*jzhang@hnust.edu.cn

1 引 言

目标检测的主要任务是从输入图像中定位感兴趣的目标，然后准确地确定每个目标的类别。当前目标检测技术已经广泛地应用于日常生活安全、机器人导航、智能视频监控、交通场景检测及航天航空等领域[1]。当前主流目标检测算法主要分为两类：一类是以RCNN[2-4]为代表的双阶段(two-stage)目标检测算法，是一种先生成候选框，再检测的目标检测算法；另一类是以SSD[5]、YOLO为代表的单阶段(one-stage)目标检测算法，是一种基于回归的目标检测算法。Two-stage比one-stage具有更高的精确度，但one-stage的检测速度要比two-stage更快。One-stage的目标检测算法更适用于某些需要更高实时性的目标检测任务。

2016年，Redmon等[6]提出了YOLO目标检测算法，是第一个基于回归的one-stage目标检测算法。2017年，Redmom等[7]又提出了YOLOv2目标检测算法，YOLOv2是在YOLOv1的基础上，删除了全接连层与最后一个的池化层得到的，YOLOv2借鉴Faster-RCNN与SSD的思路，引入一种先验框(anchor boxes)来预测边界框(bounding boxes)，并命名为DarkNet-19。2018年，Redmon等[8]提出了YOLOv3，YOLOv3引入了一种基于金字塔结构的特征提取网络并命名为DarkNet-53，将损失函数由Softmax替换成Logistic损失。2020年，在YOLOv3的作者宣布放弃对YOLOv3进行更新后，Bochkovskiy等[9]提出了YOLOv4目标检测算法，该算法已被YOLOv3的作者认可。YOLOv4使用CSPDarkNet-53[10]、空间金字塔池化(SPP)、PANet[11]及YOLOv3的检测头(Head)作为YOLOv4的网络结构，并引入了一种Mosaic的数据扩充方式与自我对抗训练方式。

YOLO系列的目标检测算法，具有复杂的网络结构与大量的网络参数，需要强大的GPU来实现实时目标检测。然而在现实应用中需要利用某些计算能力与内存占用较低的移动设备和嵌入式设备进行实时检测，如智能手机上的人脸识别和嵌入式的实时监控[12]，这对目标检测算法是一个巨大的挑战。因此就有许多研究员提出了轻量级的目标检测算法，YOLOv2-tiny是轻量级YOLO算法[13-29]之一，该算法将DarkNet19网络中的卷积层删除到了9层，以降低网络的复杂度。YOLOv3-tiny[20]是通过压缩YOLOv3的网络模型得到的，压缩卷积层与最大池化层，舍弃了残差(ResBlock)结构，且输出分支从三层变为两层。

同样YOLOv4也存在着网络结构复杂、无法在移动与嵌入式设备实现实时检测的问题，故本文提出了一种基于自适应空间特征融合的轻量化目标检测算法。所提算法以YOLOv4为网络框架，利用轻量级的MobileNet[21]代替YOLOv4中的CSPDarkNet-53作为特征提取网络，降低网络深度、提高网络的检测速度、加快训练收敛速度，使其更适用于移动设备；引入自适应空间特征融合(ASFF)方式[22]对PANet结构进行改进，提高检测精度；通过增加网络输出并修改损失函数，提高预测框的可靠性和检测精度。

2 YOLOv4算法的改进

2.1 主干网络轻量化

为提高YOLOv4检测速度、降低训练难度、方便嵌入移动端，所提算法引入MobileNet对YOLOv4骨干网络(backbone)进行改进。

MobileNet是一个含有深度可分离卷积的流线型轻量级神经网络，它将神经网络中的标准卷积分解成一个深度卷积和一个1×1的点卷积，其中深度卷积作用于通道，通过一一对应保证每一个通道只被一个卷积核提取特征，点卷积来组合深度卷积后得到的特征图，维持特征的完整性，如图1所示，这种结构可以在减少输出通道的同时实现跨通道的信息整合，在牺牲少量精度的情况下计算速度提升了8～9倍。

标准卷积与深度可分离卷积计算量之比为

$$\frac{D_s\times D_s\times M\times 1+1\times M\times N}{D_s\times D_s\times M\times N}=\frac{1}{N}+\frac{1}{D_s^2},\quad(1)$$

式中：D_s为卷积核的尺寸；M和N为输入通道数和输出通道数。从(1)式可以看出，在相同维度的特征图下，深度可分离卷积相较于标准卷积可以大幅度降低计算量。MobileNet结构如表1所示，其中“Conv”表示标准卷积，“Conv dw”表示深度卷积，一共含有27层卷积和220万参数。CSPDarkNet-53结构如表2所示，含有53层卷积和2760万参数。两者对比，MobileNet网络层数少、结构简单。综上所述，利用MobileNet替换CSPDarkNet-53骨干网络，简化结构、降低计算消耗内存，使算法模型的训练效率与检测速度有了大幅度的提升。

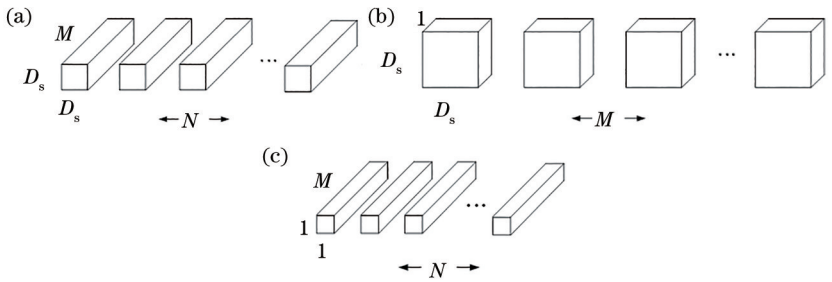

图 1　标准卷积和深度可分离卷积。(a)标准卷积;(b)深度卷积;(c)点卷积

Fig. 1　Standard convolution and depth separable convolution. (a) Standard convolution; (b) deep convolution; (c) point convolution

表 1　MobileNet 结构

Table 1　MobileNet structure

	Type	Number of filters	Size	Output
	Conv	32	3×3	416×416
	Conv dw	32	3×3/2	208×208
	Conv	64	1×1	208×208
	Conv dw	64	3×3/2	104×104
	Conv	128	1×1	104×104
	Conv dw	128	3×3	104×104
	Conv	128	1×1	104×104
	Conv dw	128	3×3/2	52×52
	Conv	256	1×1	52×52
	Conv dw	256	3×3	52×52
	Conv	256	1×1	52×52
	Conv dw	256	3×3/2	26×26
	Conv	512	1×1	26×26
5×	Conv dw	512	3×3	26×26
	Conv	512	1×1	26×26
	Conv dw	512	3×3/2	13×13
	Conv	1024	1×1	13×13
	Conv dw	1024	3×3	13×13
	Conv	1024	1×1	13×13

表 2　CSPDarkNet-53 结构

Table 2　CSPDarkNet-53　structure

	Type	Number of filters	Size	Output
	Convolutional	32	3×3	416×416
	Convolutional	64	3×3/2	208×208
	Convolutional	32	1×1	
1×	Convolutional	64	3×3	
	Residual			208×208
	Convolutional	128	3×3/2	104×104
	Convolutional	64	1×1	
2×	Convolutional	128	3×3	
	Residual			104×104
	Convolutional	256	3×3/2	52×52
	Convolutional	128	1×1	
8×	Convolutional	256	3×3	
	Residual			52×52
	Convolutional	512	3×3/2	26×26
	Convolutional	256	1×1	
8×	Convolutional	512	3×3	
	Residual			26×26
	Convolutional	1024	3×3/2	13×13
	Convolutional	512	1×1	
4×	Convolutional	1024	3×3	
	Residual			13×13

2.2　增加特征融合效果

YOLOv4 采用 PANet 结构对多尺度特征图进行融合并输出，PANet 在 FPN 的基础上加入了自下而上的增强结构，从原来的单项融合转为双向融合，如图 2 所示。PANet 是由自上而下融合路径(a)和自下而上融合路径(b)两部分构成的，其中自上而下的融合方式如图 3 所示，将特征图 $\boldsymbol{X}$ 通过最近邻的方式上采样(upsampling)2 倍，再与通过 1×1 的卷积调整通道后的前一层特征图 $\boldsymbol{Y}$ 相加；自下而上的融合是自上而下融合的逆过程，将上采样改为下采样即可。

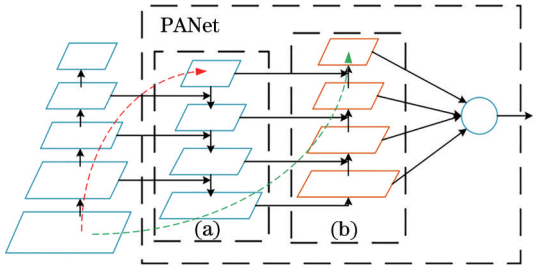

图 2　PANet 结构

Fig. 2　PANet structure

PANet 的融合方式只是简单地将特征图变换成相同尺寸再相加，无法充分利用不同尺度的特征。所提算法引入一种新的空间融合方式——自适应空间特征融合(ASFF)，对 PANet 结构进行改

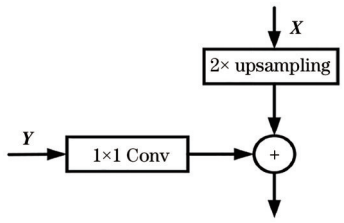

图 3　自上而下融合方式

Fig. 3　Top-down integration

进，并通过学习得到权重参数，对不同阶层的特征图进行融合，具体设计如图 4 所示，$\boldsymbol{X}_1$、$\boldsymbol{X}_2$、$\boldsymbol{X}_3$ 分别代表通过 MobileNet 主干网络提取到的特征图，以 ASFF3 为例，经过 PANet 结构后得到的特征图 level 1 与 level 2，先通过 1×1 卷积压缩成与 level 3 相同的通道数，然后分别进行 4 倍上采样和 2 倍上采样形成与 level 3 相同维度的特征图，记为 resize_level 1 与 resize_level 2，将 resize_level 1、resize_level 2 与 level 3 通过 1×1 卷积得到权重参数 α、β、γ，最后把得到 resize_level 1、resize_level 2 与 level 3 分别乘以 α、β、γ 再相加得到新的融合特征。

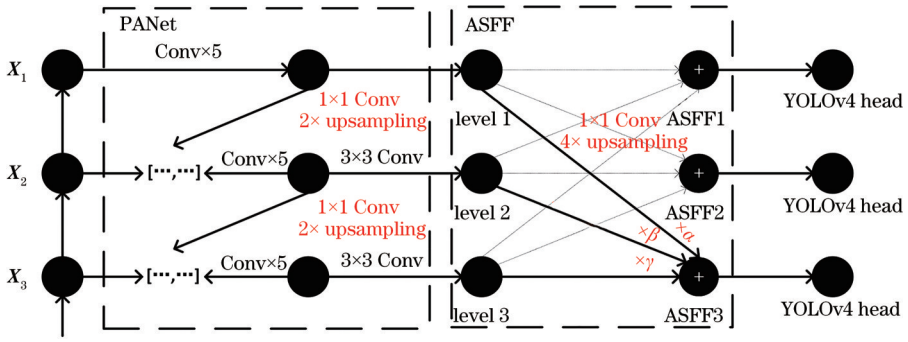

图 4　改进的 PANet 结构

Fig. 4　Improved PANet structure

ASFF，这种通过学习参数的融合方式，能过滤其他层次的特征，只保留该阶层有用的信息，不仅使得信息层次化，而且模型的训练效率也会更佳，可以描述为

$$\boldsymbol{y}_{ab}^{l}=\alpha_{ab}^{l}\cdot\boldsymbol{x}_{ab}^{1\to l}+\beta_{ab}^{l}\cdot\boldsymbol{x}_{ab}^{2\to l}+\gamma_{ab}^{l}\cdot\boldsymbol{x}_{ab}^{3\to l},\tag{2}$$

式中：$\boldsymbol{y}_{ab}^{l}$ 表示通过 ASFF 得到的新特征图；$\boldsymbol{x}_{ab}^{n\to l}$ 表示 n 层到 l 层的特征图上的特征向量；α_{ab}^{l}、β_{ab}^{l}、γ_{ab}^{l} 表示 3 个不同层次特征图的权重值，通过 Softmax 函数使其满足 $\alpha_{ab}^{l}+\beta_{ab}^{l}+\gamma_{ab}^{l}=1$，$\alpha_{ab}^{l}$、$\beta_{ab}^{l}$、$\gamma_{ab}^{l}\in[0,1]$。

2.3　增加预测框位置置信度

YOLOv4 的输出如图 5 所示，输入图片被划分为 16 个网格，每个网格的输出为 $3\times\left[\left(t_x,t_y,t_w,t_h\right)+p_{\mathrm{obj}}+p_n\right]$，其中 t_x，t_y，t_w，t_h 代表中心点的坐标和边界框的宽与高，p_{obj} 指的是目标分类的置信度，p_n 是目标分类的类别。从 YOLOv4 的输出可以看出，原始 YOLOv4 的输出只有目标分类有置信度，而边界框只有位置信息，没有对应的置信度，边界框的不确定性是未知的，这样容易造成一些误检情况。针对这个问题，在基本不改变 YOLOv4 结构和计算量的情况下，利用 Gaussian 算法对网络输出进行建模，通过增加网络输出来增加每个边界框的可靠性，如图 6 所示。在进行坐标预测时，输出从

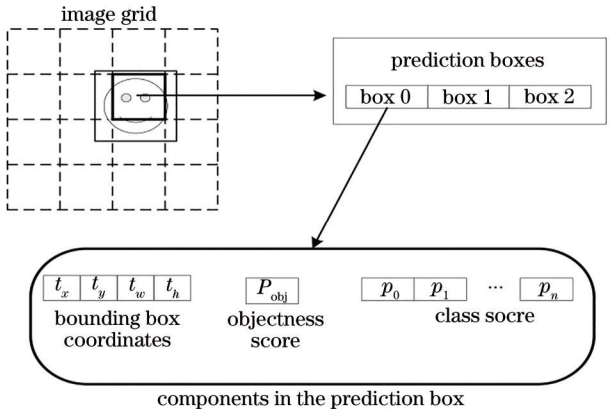

图 5　YOLOv4 的网络输出

Fig. 5　Network output of YOLOv4

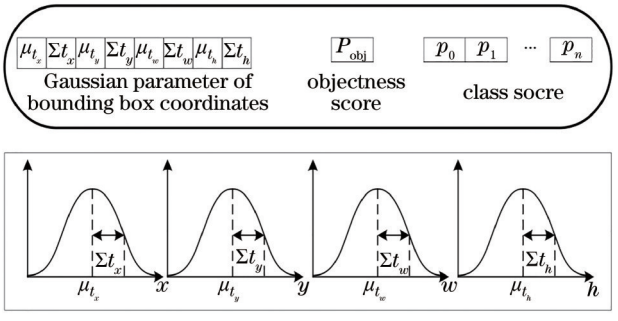

图 6　新增网络输出

Fig. 6　New network output

4个维度提升到8个维度，这8个维度分别对应预测边界框的中心坐标、高宽及对应预测边界框的置信度，以预测边界框位置作为均值，置信度作为方差，进行高斯算法建模。

$$p(y|x)=N[y;\mu(x),\Sigma(x)], \tag{3}$$

式中：$\mu(x)$为均值；$\Sigma(x)$为方差。高斯密度函数的表达式为

$$X=N(\mu,\sigma^2)=\frac{1}{\sqrt{2\pi}\,\sigma}\exp\left[-\frac{(x-\mu)^2}{2\sigma^2}\right]。 \tag{4}$$

3　自适应空间特征融合的轻量化目标检测算法模型

3.1　网络结构设计

所提算法基于YOLOv4的基础框架，由改进后的backbone、neck、head与损失函数4部分组成，如图7所示。将416×416大小的图片输入MobileNet特征提取网络，输出尺寸分别为52×52、26×26、13×13的特征图，将其输入到带有ASFF结构的neck结构中进行特征融合，并将融合特征输入到提升输出维度后的head检测器中进行检测。

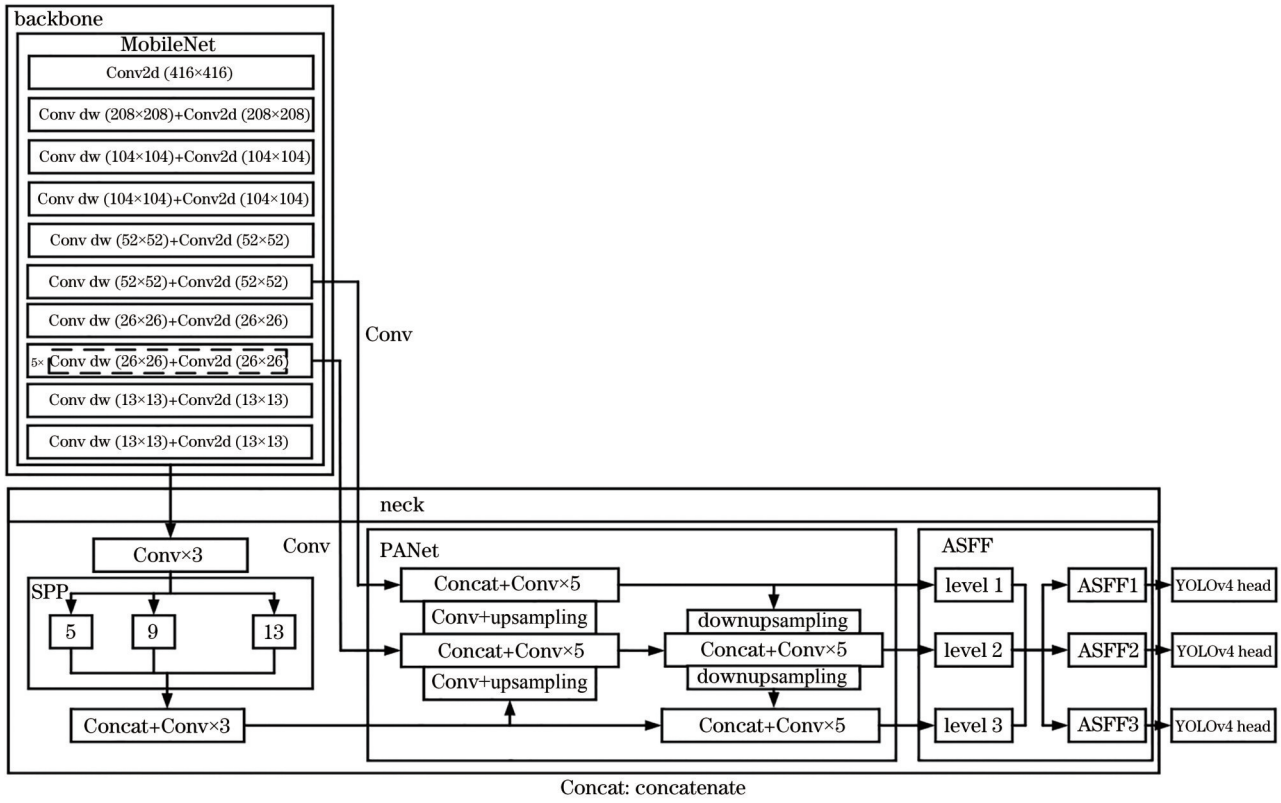

图7　所提网络结构

Fig. 7　Structure of propoesd network

3.2　损失函数设计

所提算法的损失函数包括边界框的坐标误差、边界框置信度误差与类别误差3部分。所提算法增加了网络输出来提高预测边界框的可靠性，故对预测边界框坐标误差的损失函数进行对应修改。秉承YOLOv4的思想，在预测值前输出前使用Sigmoid函数进行预处理，对网络的输出$\hat{\mu}_{t_x}$、$\hat{\mu}_{t_y}$、$\hat{\mu}_{t_w}$、$\hat{\mu}_{t_h}$、$\hat{\Sigma}_{t_x}$、$\hat{\Sigma}_{t_y}$、$\hat{\Sigma}_{t_w}$、$\hat{\Sigma}_{t_h}$进行以下变换，使其值在[0,1]。

$$\begin{cases}\mu_{t_x}=\sigma\left(\hat{\mu}_{t_x}\right)\\ \mu_{t_y}=\sigma\left(\hat{\mu}_{t_y}\right)\\ \mu_{t_w}=\hat{\mu}_{t_w}\\ \mu_{t_h}=\hat{\mu}_{t_h}\end{cases}, \tag{5}$$

$$\begin{cases}\Sigma_{t_x}=\sigma\left(\hat{\Sigma}_{t_x}\right)\\ \Sigma_{t_y}=\sigma\left(\hat{\Sigma}_{t_y}\right)\\ \Sigma_{t_w}=\sigma\left(\hat{\Sigma}_{t_w}\right)\\ \Sigma_{t_h}=\sigma\left(\hat{\Sigma}_{t_h}\right)\end{cases},\tag{6}$$

$$\sigma(x)=\frac{1}{1+\exp(-x)},\tag{7}$$

式中：μ_{t_x}、μ_{t_y}为预测边界框的中心坐标在划分网格(grid)中的偏移值；μ_{t_w}、μ_{t_h}为预测边界框的宽与长。所提算法预测框的坐标都满足均值为μ、方差为σ的高斯分布，故采用 negative log likelihood loss 代替坐标回归交叉熵函数，有

$$L_m=-\sum_{i=1}^{W}\sum_{j=1}^{H}\sum_{k=1}^{K}\gamma_{ijk}\log\left\{N\left[m_{ijk}^{\mathrm{G}}\middle|\mu_{t_m}(m_{ijk}),\sum t_m(m_{ijk})\right]+\varepsilon\right\},m\in(x,y,w,h),\tag{8}$$

$$\begin{cases}\gamma_{ijk}=\dfrac{\omega_{\mathrm{scale}}\times\delta_{ijk}^{\mathrm{obj}}}{2}\\ \omega_{\mathrm{scale}}=2-w^{\mathrm{G}}\times h^{\mathrm{G}}\end{cases},\tag{9}$$

式中：W、H、K为特征图宽与高的 gird 数及预测边界框(anchor)数；$\mu_{t_m}(m_{ijk})$和$\sum t_m(m_{ijk})$为预测边界框位置坐标值和对应位置的置信度；m_{ijk}^{G}为实际位置坐标值；γ_{ijk}为权重的惩罚系数；ω_{scale}由实际坐标位置的长(h^{G})与宽(w^{G})计算得出；$\delta_{ijk}^{\mathrm{obj}}$为一个参数，当实际坐标位置值与预测边界框坐标值的截面交并比(IOU)最大时为 1，否则取零；$\varepsilon=10^{-9}$是一个为了稳定对数函数的数。综上所述，所提算法损失函数的表达式为

$$\begin{aligned}L_{\mathrm{tot}}=&L_x+L_y+L_w+L_h+\lambda_{\mathrm{obj}}\sum_{i=0}^{S^2}\sum_{j=0}^{B}l_{ij}^{\mathrm{obj}}\left(C_i-\hat{C}_i\right)^2+\\&\lambda_{\mathrm{noobj}}\sum_{i=0}^{S^2}\sum_{j=0}^{B}l_{ij}^{\mathrm{noobj}}\left(C_i-\hat{C}_i\right)^2+\sum_{i=0}^{S^2}l_i^{\mathrm{noobj}}\sum_{c\in N_{\mathrm{class}}}^{S^2}\left[p_i(c)-\hat{p}_i(c)\right]^2,\end{aligned}\tag{10}$$

式中：第 1 项到第 4 项为预测边界框坐标值x、y、w、h的损失函数；第 5 项和第 6 项是存在与不存在物体的预测边界框置信度损失函数；第 7 项为分类误差损失函数；S是 YOLOv4 的划分网格大小(grid size)；B为每个网格的预测边界框(anchor)的个数；l_{ij}^{obj}在有物体时为 1 否则为 0；l_{ij}^{noobj}在没有物体时为 1 有物体时为 0；C_i表示类别C在此 grid size 的置信度；$\hat{C}_i$表示类别C在此 grid size 的置信度的真实值；N_{class}表示所有类别的集合；$p_i(c)$表示类别c在此 grid size 的概率；λ_{obj}与λ_{noobj}为损失函数的权重系数。

4　实验与结果分析

4.1　实验平台与数据集

训练平台使用 python 3.6 进行编译和测试，对应的开发工具为 Pycharm 2018.1.4，主计算机视觉库为 Python-OpenCv 3.4.2，操作系统为 Windows 10，CPU 为 i7-6700K 的四核处理器，主频为 4 GHz，显卡为 GTX 1070，内存为 8 GB，硬盘容量 2 TB。

为了更好验证所提检测算法的有效性，以树莓派 4B 的仿生机器人为算法搭载平台代表移动与嵌入式设备进行检测，该平台 SOC 为 Broadcom BCM2711，CPU 为 64 位 1.5 GHz 四核处理器，GPU 采用的是 Broadcom VideoCore Vl@500 MHz，HDMI 是支持 4K 60 Hz 的 micro HDMI。

以未佩戴口罩(no_mask)、正确佩戴口罩(mask)与不正确佩戴口罩(no_mask_well)3 种情况为识别目标进行实验验证。由于目前没有公开的不正确佩戴口罩的数据集，故采用移动摄像头对 100 人在不同环境和不正确佩戴口罩的情况下进行拍照取样，得到 3000 张人脸不正确佩戴口罩照片，再从 WIDER FACE、masked faces (MAFA)和 real-world masked face dataset (RMFD)3 个数据集中筛选出 3000 张人脸未佩戴口罩照片和 3000 张人脸正确佩戴口罩照片。最终取得 no_mask、mask、no_mask_well 各 3000 张图像并对其进行手工标注。

4.2　评价指标

所提算法以平均准确率(AP)、平均准确率均值(mAP)及检测速度作为评价指标。AP 和 mAP 的表达式为

$$\begin{cases} P=\dfrac{N_{TP}}{N_{TP}+N_{FP}} \\ R_{AP}=\displaystyle\int_0^1 P(R)\mathrm{d}R \end{cases}, \tag{11}$$

$$R_{mAP}=\frac{\sum_{n=1}^{N} R_{AP_n}}{N}, \tag{12}$$

式中：N_{TP}为分类正确的正样本；N_{FP}为分类错误的正样本；$P(R)$为PR曲线中P值，其中R是预测的召回率，P是预测的准确率；N指目标物体的种类数；R_{AP_n}为n类目标体的AP值。

检测速度是指目标检测网络每秒能够检测的图片数量(帧数)，单位为frame/s。

4.3　结果分析

4.3.1　所提算法与YOLOv4算法训练效果对比

实验1：对比验证修改前后的模型在少样本下的训练效果，不使用预训练权重，随机挑选未佩戴口罩图像、正确佩戴口罩图像与不正确佩戴口罩图像各300张，按9∶1的比例划分为训练集与验证集，训练轮数(epoch)为100，初始学习率设为0.01，采用余弦退火衰减学习率，batch_size设为8，使用早停法(early stopping)避免出现过拟合，验证损失位于epoch后测试并添加正则化。在相同的实验环境与参数设置下，训练YOLOv4网络进行比较。

两种模型的训练损失曲线与验证损失曲线如图8所示，其中YOLOv4_improvement代表所提算法。从图中可以看出：YOLOv4在少量数据样本下，训练效果差，损失曲线与验证损失曲线下降速度缓慢，收敛效果欠佳，损失与验证损失最低也只能下降到16.5和16.0；而所提算法相比于YOLOv4，损失曲线与验证损失曲线下降速度快，在epoch为15时，损失就达到了10以下，最后能达到1.5左右，表明所提算法更容易收敛，训练效果好，即使在少量数据样本的情况下也能训练。

实验2：对比验证修改前后的模型在大量样本下的训练效果，不加载预训练权重，随机挑选用未佩戴口罩图像、正确佩戴口罩图像与不正确佩戴口罩图像各3000张，按9∶1的比例划分为训练集与验证集，训练轮数(epoch)为200，初始学习率设为0.01，采用余弦退火衰减学习率，batch_size设为8，使用早停法(early stopping)避免出现过拟合，验证损失位于epoch后测试并添加正则化。在相同的实验环境与参数设置下，训练YOLOv4进行比较。

两种模型的训练损失曲线图如9所示。从图中

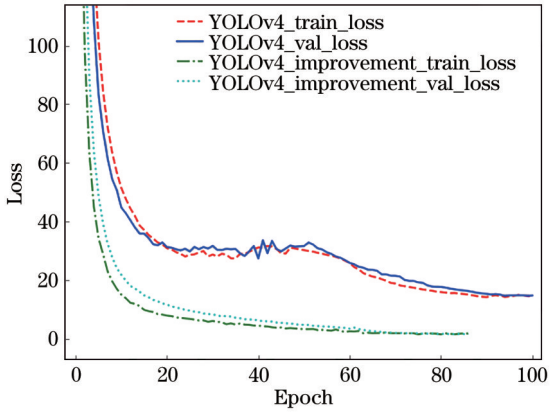

图8　少量数据样本下所提算法与YOLOv4训练效果对比

Fig. 8　Comparison of training effect between proposed algorithm and YOLOv4 under a small amount of data samples

可以看出：YOLOv4在大量数据样本下，虽然损失下降到3，但是下降速率依旧很慢，收敛缓慢；反观所提算法，虽然损失值有所上升，但是收敛速度依

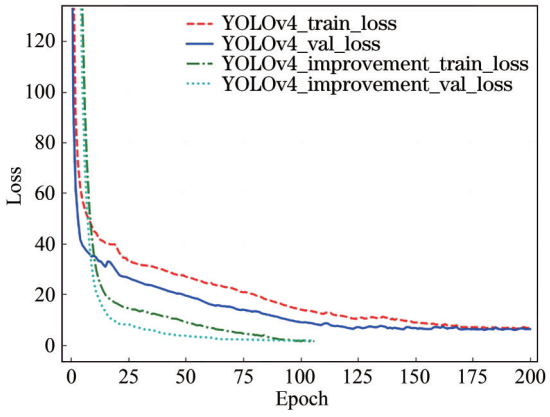

图9　大量数据样本下本文算法与YOLOv4训练效果对比

Fig. 9　Comparison of training effect between proposed algorithm and YOLOv4 under a large number of data samples

旧很快，损失最低能达到1，表明所提算法大量数据下收敛速度与收敛效果均优于原始YOLOv4。

两次对比实验表明，YOLOv4的网络太大难以训练，采用轻量级MobileNet网络代替原网络中CSPDarkNet-53特征提取网络，使得网络层数下降，简化网络，达到更容易训练的目的。

4.3.2　所提算法与YOLOv4检测效果对比

将实验2中训练好的模型搭载到检测机器人上，对未佩戴口罩、正确佩戴口罩、不正确戴口罩3种情况进行具体的检测，检测结果如图10所示。从图10(a)、(b)、(c)、(d)中可以看出，YOLOv4与所提算法都能检测未佩戴口罩的情况，但是YOLOv4算法的精确度最高只有89%，而所提算法的精确度基

本维持在 90% 以上，最高能达到 99% 以上。从图 10(e)、(f)中可以看出，在人群密集且图片像素不高的情况下，YOLOv4算法的识别效果欠佳，漏检和错检较多，而所提算法相比YOLOv4算法不仅在识别效果上有了很大的提高，而且在精度上也有很大提高。从图 10(g)、(h)中可以看出，YOLOv4漏检了两种情况，对于未佩戴好口罩情况的精确度也较低，而本所提算法不仅能检测出全部情况，对未佩戴好口罩情况在精确度上要比YOLOv4高出15个百点以上。综上所述，所提算法在未佩戴口罩、正确佩戴口罩、不正确佩戴口罩人脸检测的识别率与识别精确度均明显优于YOLOv4算法。

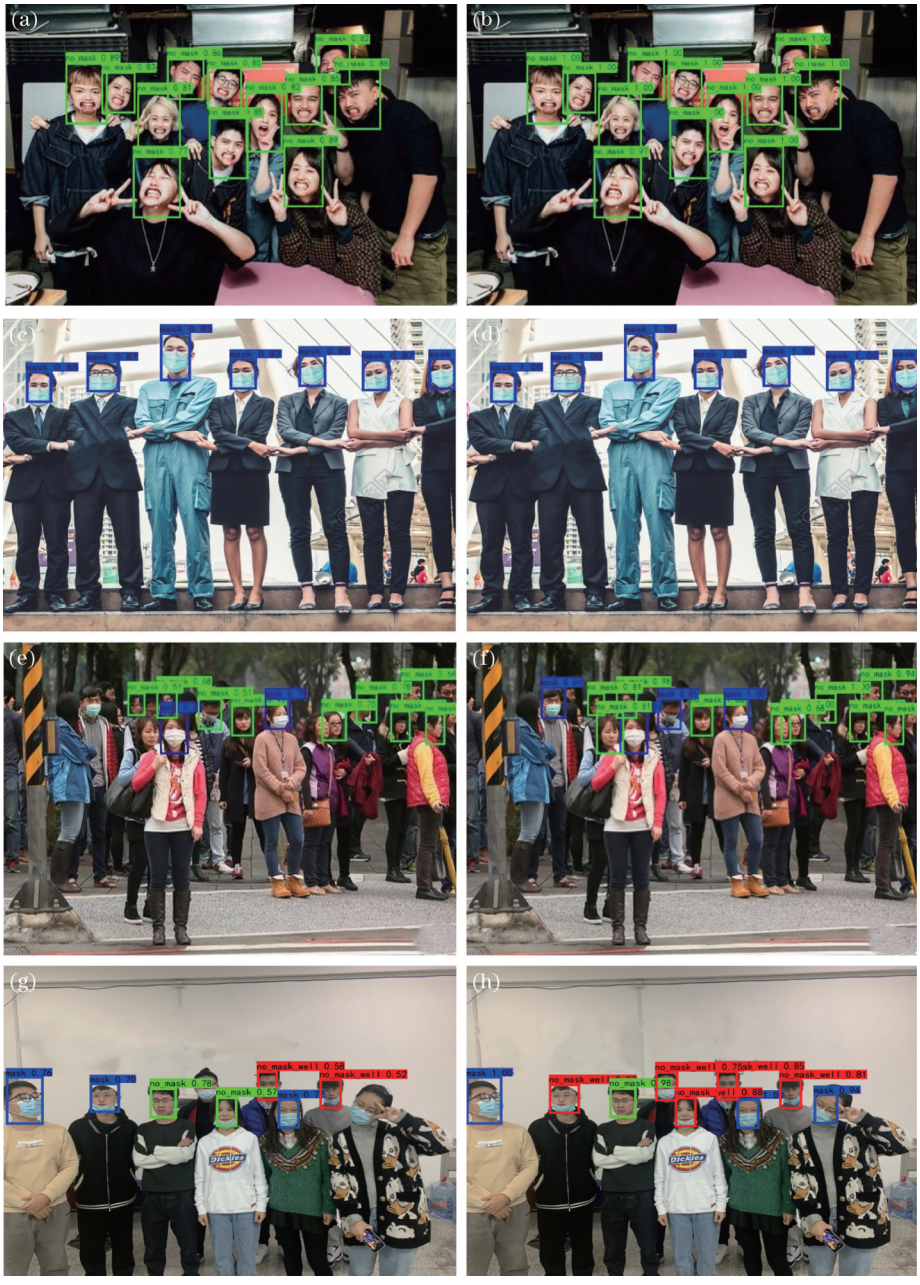

图 10　所提算法与YOLOv4检测效果对比。(a)(c)(e)(g) YOLOv4算法；(b)(d)(f)(h)所提算法

Fig. 10　Comparison of detection effect between proposed algorithm and YOLOv4.(a)(c)(e)(g) YOLOv4 algorithm; (b)(d)(f)(h) proposed algorithm

4.3.3　所提算法与其他算法对比

为了进一步验证所提算法的有效性，将所提算法与其他主流算法搭载到检测机器人上并进行了对比实验，结果如表3所示。

从表3可以看出，所提算法对口罩佩戴取得了较好的检测效果，对比Faster-RCNN检测算法，检

表 3 不同算法性能对比

Table 3 Performance comparison of different algorithms

Algorithm	AP /%			mAP /%	Detection speed / (frame·s^{-1})
	Mask	No_mask	No_mask_well		
Faster-RCNN	97.77	97.56	95.33	96.88	2
RetinaFace	74.51	97.35	70.45	80.77	6
Attention-RetinaFace	75.76	98.67	73.89	82.77	8
SSD	77.33	75.55	73.34	74.07	9
YOLOv3	81.62	80.96	77.65	80.07	11
YOLOv3-tiny	78.67	77.92	75.33	77.30	15
YOLOv4	88.92	87.34	85.84	87.36	13
Improved_YOLOv4	96.38	96.96	94.44	95.92	19

测速度提升了 17 frame·s^{-1}，但是对于检测精度，Faster-RCNN 算法的 AP 和 mAP 是要优于所提算法的。原因在于 Faster-RCNN 属于 two-stage 检测算法，会先对图形产生一些候选区域，然后再对候选的区域进行检测，因此 Faster-RCNN 的精度更高，但是这类 two-stage 算法的检测速度很低，不适合配置于移动设备上。与 RetinaFace、Attention-RetinaFace 算法相比，所提算法对正确佩戴口罩情况的 AP 值分别提升了 21.87 个百分点和 20.62 个百分点，在不正确佩戴口罩情况的 AP 值分别提升了 23.99 个百分点与 20.55 个百分点，检测速度提升了 13 frame·s^{-1}和 11 frame·s^{-1}，但是所提算法对于未佩戴口罩情况情况的 AP 是低于 RetinaFace 与 Attention-RetinaFace 算法的，原因在于这两种算法是基于人脸对齐、人脸密集与像素级人脸三维分析的单目标检测算法，因此在检测未佩戴口罩情况上，这两种算法的精度要更高。对比单阶段的 SSD、YOLOv3、YOLOv4 算法，所提算法在未佩戴口罩的检测上 AP 分别提升 21.41 个百分点、16.00 个百分点、9.62 个百分点；在正确佩戴口罩的检测上 AP 分别提升 19.05 个百分点、14.76 个百分点、7.46 个百分点；在不正确佩戴口罩的检测上 AP 分别提升 21.10 个百分点、16.79 个百分点、8.60 个百分点，检测速度分别提升了 10 frame·s^{-1}、8 frame·s^{-1}和 6 frame·s^{-1}。对比相同的轻量级目标检测算法 YOLOv3-tiny，所提算法在未佩戴口罩、正确佩戴口罩、不正确佩戴口罩检测上 AP 提高了 19.04 个百分点、17.71 个百分点、19.11 个百分点，检测速度提高了 4 frame·s^{-1}。总而言之，所提算法在人脸口罩佩戴目标检测下相较于其他主流目标检测算法具有一定的优势，更适配置于移动与嵌入设备实现实时监测。

4.3.4 消融实验结果及其分析

消融实验是深度学习领域中常用的实验方法，主要用来分析不同的网络分支对整个模型的影响。为了进一步分析所提改进算法对于 YOLOv4 模型的影响，将所提算法裁剪成 4 组分别进行训练，第 1 组为未改动的 YOLOv4 算法，第 2 组为特征提取网络替换成 MobileNet 的 YOLOv4 算法，第 3 组在第 2 组的基础上把 PANet 结构改进成 ASFF 结构，第 4 组在第 3 组的基础之上改变了损失函数，增加了分析预测框的可靠性。4 组消融实验结果如表 4 所示，其中“√”表示含有该结构，“×”表示未含有该结构。

从表 4 可以看出：第 1 组 YOLOv4 的实验在正确佩戴口罩、未佩戴口罩、不正确佩戴口罩的目标检测上 AP 值分别为 88.92%、87.34%、85.84%，mAP 值为 87.36%，检测速度为 13 frame·s^{-1}。第 2 组实验将 CSPDarkNet-53 特征提取网络替换成了

表 4 消融实验对比

Table 4 Comparison of ablation experiments

Grouping	MobileNet	ASFF	Modify_loss	AP /%			mAP /%	Detection speed / (frame·s^{-1})
				Mask	No_mask	No_mask_well		
G1	×	×	×	88.92	87.34	85.84	87.36	13
G2	√	×	×	86.75	86.02	83.98	85.58	21
G3	√	√	×	91.88	92.32	89.45	91.21	20
G4	√	√	√	96.38	96.96	94.44	95.92	19

轻量级的MobileNet网络后，检测精度下降，检测速度提升了8 frame·s^{-1}，原因是特征提取网络层数减少，导致特征图提取的效果下降，但是降低了内存消耗，从而提高了检测速度。第3组实验，由于在第2组的基础上把PANet结构改进成ASFF结构，相比第2组实验，虽然检测速度下降了1 frame·s^{-1}，但是各类AP分别提升了5.13个百分点、6.30个百分点、5.47个百分点，mAP值提升了5.63个百分点，表明了改进后的ASFF结构以增加一点计算量的代价，提升了对特征图的提取效果，从而提升模型的性能。第4组实验在第3组实验上改变了损失函数，增加了对预测框的可靠性分析，对比第3组，各类AP值分别提升了4.50个百分点、4.64个百分点、4.99个百分点，mAP值提升了4.71个百分点，表明损失函数的改进增强了检测的精确度。

5 结　论

当前的目标检测算法主要分为one-stage与two-stage两类算法，one-stage算法相比two-stage算法具有更快的检测速度，更适合用于某些需要更高实时性的目标检测任务。针对目前深度学习中one-stage目标检测网络结构复杂、训练困难与在移动与嵌入式设备难以部署的问题，提出了一种基于自适应空间特征融合的轻量化目标检测算法。所提算法以YOLOv4为网络基础框架，采用MobileNet作为特征提取网络；在FPN结构中增添了ASFF特征融合方式；增加网络输出维度，对其进行高斯建模，提高边界框的可靠性；修改网络损失函数，改变边界框坐标的回归策略。以疫情期间口罩佩戴检测机器人为算法部署载体，人脸口罩佩戴情况为检测目标，实验结果表明：1）主干网络的轻量化提高了训练的收敛效率，在相同硬件配置的基础上，检测速度达到了19 frame·s^{-1}，相比YOLOv4提高了8 frame·s^{-1}；2）通过ASFF的增添与改变边界框坐标的回归策略，提高了多尺度特征融合效率与边界框的可靠性，从而增加了算法的检测精度。相较于其他主流检测算法，所提算法在保证精度的情况下具有一定的检测速度优势，更适合部署于移动与嵌入设备实现实时检测。

参　考　文　献

[1] Luo H L, Chen H K. Survey of object detection based on deep learning[J]. Acta Electronica Sinica, 2020, 48(6): 1230-1239.
罗会兰, 陈鸿坤. 基于深度学习的目标检测研究综述[J]. 电子学报, 2020, 48(6): 1230-1239.

[2] Girshick R, Donahue J, Darrell T, et al. Rich feature hierarchies for accurate object detection and semantic segmentation[C]//2014 IEEE Conference on Computer Vision and Pattern Recognition, June 23-28, 2014, Columbus, OH, USA. New York: IEEE Press, 2014: 580-587.

[3] Girshick R. Fast R-CNN[C]//2015 IEEE International Conference on Computer Vision (ICCV), December 7-13, 2015, Santiago, Chile. New York: IEEE Press, 2015: 1440-1448.

[4] Ren S Q, He K M, 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.

[5] Liu W, Anguelov D, Erhan D, et al. SSD: single shot MultiBox detector[M]//Leibe B, Matas J, Sebe N, et al. Computer vision-ECCV 2016. Lecture notes in computer science. Cham: Springer, 2016, 9905: 21-37.

[6] Redmon J, Divvala S, Girshick R, et al. You only look once: unified, real-time object detection[C]//2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), June 27-30, 2016, Las Vegas, NV, USA. New York: IEEE Press, 2016: 779-788.

[7] Redmon J, Farhadi A. YOLO9000: better, faster, stronger[C]//2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), July 21-26, 2017, Honolulu, HI. New York: IEEE Press, 2017: 6517-6525.

[8] Redmon J, Farhadi A. YOLOV3: an incremental improvement[EB/OL]. (2018-04-08) [2021-02-10]. https://arxiv.org/abs/1804.02767.

[9] Bochkovskiy A, Wang C Y, Liao H Y Mark. YOLOv4: optimal speed and accuracy of object detection[EB/OL]. (2020-04-23)[2021-02-10]. https://arxiv.org/abs/2004.10934.

[10] Li B, Wang C, Wu J, et al. Surface defect detection of aeroengine components based on improved YOLOv4 algorithm[J]. Laser & Optoelectronics Progress, 2021, 58(14): 1415004.
李彬, 汪诚, 吴静, 等. 改进YOLOv4算法的航空发动机部件表面缺陷检测[J]. 激光与光电子学进展, 2021, 58(14): 1415004.

[11] Liu S, Qi L, Qin H F, et al. Path aggregation

network for instance segmentation[C]//2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition, June 18-23, 2018, Salt Lake City, UT, USA. New York: IEEE Press, 2018: 8759-8768.

[12] Mao Q C, Sun H M, Liu Y B, et al. Mini-YOLOv3: real-time object detector for embedded applications[J]. IEEE Access, 2019, 7: 133529-133538.

[13] Zhao H P, Zhou Y, Zhang L, et al. Mixed YOLOv3-LITE: a lightweight real-time object detection method [J]. Sensors, 2020, 20(7): 1861.

[14] Huang R, Pedoeem J, Chen C X. YOLO-LITE: a real-time object detection algorithm optimized for non-GPU computers[C]//2018 IEEE International Conference on Big Data (Big Data), December 10-13, 2018, Seattle, WA, USA. New York: IEEE Press, 2018: 2503-2510.

[15] Xiao D, Shan F, Li Z, et al. A target detection model based on improved tiny-Yolov3 under the environment of mining truck[J]. IEEE Access, 2019, 7: 123757-123764.

[16] Huang R, Gu J, Sun X, et al. A rapid recognition method for electronic components based on the improved YOLO-V3 network[J]. Electronics, 2019, 8(8): 825.

[17] Liu W J, Gao M Y, Qu H C, et al. Light-weight multi-object detection network based on inverted residual structure[J]. Laser & Optoelectronics Progress, 2019, 56(22): 221003.

刘万军, 高明月, 曲海成, 等. 基于反残差结构的轻量级多目标检测网络[J]. 激光与光电子学进展, 2019, 56(22): 221003.

[18] Ma Q, Zhu B, Zhang H W, et al. Low-altitude UAV detection and recognition method based on optimized YOLOv3[J]. Laser & Optoelectronics Progress, 2019, 56(20): 201006.

马旗, 朱斌, 张宏伟, 等. 基于优化YOLOv3的低空无人机检测识别方法[J]. 激光与光电子学进展, 2019, 56(20): 201006.

[19] Guo J X, Liu L B, Xu F, et al. Airport scene aircraft detection method based on YOLO v3[J]. Laser & Optoelectronics Progress, 2019, 56(19): 191003.

郭进祥, 刘立波, 徐峰, 等. 基于YOLO v3的机场场面飞机检测方法[J]. 激光与光电子学进展, 2019, 56(19): 191003.

[20] Adarsh P, Rathi P, Kumar M. YOLO v3-Tiny: object detection and recognition using one stage improved model[C]//2020 6th International Conference on Advanced Computing and Communication Systems (ICACCS), March 6-7, 2020, Coimbatore, India. New York: IEEE Press, 2020: 687-694.

[21] Howard A G, Zhu M L, Chen B, et al. MobileNets: efficient convolutional neural networks for mobile vision applications[EB/OL]. (2017-04-17)[2021-02-10]. https://arxiv.org/abs/1704.04861.

[22] Liu S T, Huang D, Wang Y H. Learning spatial fusion for single-shot object detection[EB/OL]. (2019-11-21)[2021-02-10]. https://arxiv.org/abs/1911.09516.