王承茂, 黄润才, 顾磊欣. 基于 YOLOv5 的火灾目标检测改进算法研究[J]. 智能计算机与应用,2025,15(5):163-172. DOI: 10.20169/j.issn.2095-2163.250523

基于 YOLOv5 的火灾目标检测改进算法研究

王承茂, 黄润才, 顾磊欣
(上海工程技术大学 电子电气工程学院, 上海 201620)

摘 要: 针对目前火灾中烟雾和火焰的检测研究比较单一,且缺少烟雾火焰数据集、检测模型的抗干扰能力不强等导致检测精度低的问题,提出了一种改进 YOLOv5 的烟火目标检测算法。首先通过网络爬虫获取火灾图片,然后利用开源软件 LabelImg 在图片中标注烟雾和火焰目标,构建高质量烟雾火焰数据集,并且利用数据集增强方法扩充数据集,以此来提高模型的泛化能力,其次使用 YOLOv5 的自适应锚框技术对火焰烟雾数据集进行聚类分析,来获得最优的聚类结果,最后改进 YOLOv5 中的 CSP1 模块,对特征进行了融合处理,获取了表达能力更强的特征,提高了模型对火焰烟雾的检测效果。实验结果表明,改进的算法对火焰和烟雾的总体检测准确率和召回率为 85.9%和 88.4%,检测的平均精度(mean Average Precision, *mAP*) *mAP*@0.5 为 90%,相比原始 YOLOv5 提高了 2%,检测速度为 42.2 帧/s(Frame Per Second, FPS),能够满足火灾检测的准确性和实时性,可以对火灾进行有效检测。

关键词: 目标检测; 深度学习; 火灾检测; 特征融合; YOLOv5
中图分类号: TP391.4 **文献标志码**: A **文章编号**: 2095-2163(2025)05-0163-10

Research on improved algorithm of fire target detection based on YOLOv5

WANG Chengmao, HUANG Runcai, GU Leixin
(School of Electronic and Electrical Engineering, Shanghai University of Engineering Science, Shanghai 201620, China)

Abstract: Aiming at the problems of low detection accuracy due to the lack of smoke and flame data sets and the weak anti-interference ability of the detection model, the current research on smoke and flame detection in fires is relatively single, and an improved YOLOv5 pyrotechnic target detection algorithm is proposed. Firstly, fire pictures are obtained through Web crawlers, then the open source software LabelImg is used to mark smoke and flame targets in the pictures, a high-quality smoke and flame dataset are built, and the dataset enhancement method is used to expand the dataset to improve the generalization ability of the model. Secondly, the adaptive anchor frame technology of YOLOv5 is used to cluster and analyze the flame smoke data set to obtain the optimal clustering result. Finally, the CSP1 module in YOLOv5 is improved, and the features are fused to obtain a more expressive image, which improves the detection effect of the model on flame smoke. The experimental results show that the improved algorithm has an overall detection accuracy and recall rate of 85.9% and 88.4% for flames and smoke, and the average detection precision (mean Average Precision, *mAP*) *mAP*@0.5 is 90%, which is improved by 2% compared with the original YOLOv5, and the detection speed is 42.2 frames/s (Frame Per Second, FPS), which can meet the accuracy and real-time performance of fire detection, and can effectively detect fire.

Key words: target detection; deep learning; fire detection; feature fusion; YOLOv5

0 引 言

研究可知发生火灾时,如果不能得到及时有效控制,不仅可能会危及人们的财产及生命安全,甚至还会对生态环境造成严重的破坏,。而森林火灾的危害也不容小觑,据相关统计报告指出,全世界被火

作者简介: 王承茂(1997—), 男, 硕士研究生, 主要研究方向:图像处理,计算机视觉; 顾磊欣(1995—), 男, 硕士研究生, 主要研究方向:图像处理,人工智能。
通信作者: 黄润才(1966—), 男, 博士,副教授, 主要研究方向:计算机网络与信息安全,智能计算,服务机器人,大数据等。Email:hrc@sues.edu.cn。

收稿日期: 2023-10-27

灾损毁的森林占世界森林总面积的百分之一以上[1]。随着经济的快速发展,建筑规模的复杂性在增加,因此具有高灵敏度和准确性的火灾检测和报警对于减少火灾损失非常重要。然而,传统的传感器火灾检测技术,如烟雾和热量探测器,却并不适用于大空间、复杂建筑物或干扰多的空间。基于此,火灾检测中就经常出现漏检、误报、检测延迟等现象,火灾检测预警面临着不小难题。近年来,随着深度学习的发展,同时,考虑到图像火灾检测具有准确度高、检测速度快、能够有效检测大空间和复杂建筑结构中的火灾等优点,现已成为领域研究的重点与热点。

一般火灾发生时,会产生烟雾和火焰。因此,传统的火灾检测技术主要是基于传感器的,有烟敏火灾检测技术、温敏火灾检测技术、气体火灾检测技术、光敏火灾检测技术和复合火灾检测技术,对预防火灾起到了辅助性作用,但还是存在很多不足[2]。其中,温敏火灾传感器通过在传感器内部设置温度报警值来检测火灾现场的温度。一旦温度达到或超过该警告值,就会立即发出火灾警报[3]。但其缺点是只能适应温度相对稳定的环境,如精密仪器实验室、小型会议室等,不适用于温度变化大的环境,更不适用于一般温度低于 0℃的环境。Meili 等学者在 1950 年研制出现代离子烟雾传感器[4],在很长时间内一直占据着主导地位。烟雾感应火灾检测器检测火灾现场的烟雾。其工作原理与感温传感器类似。通过设置传感器中烟雾的最大警告值,一旦达到警告值,就会发出报警信号。烟雾传感器通常用于酒店、餐厅等。这些是生活中最常见的火灾传感器。但是,前述传感器的缺点在于对烟雾的敏感性滞后,容易出现误报和漏报。

图像火灾检测算法的过程主要分为图像处理、特征提取和火灾检测三部分,其中特征提取是核心部分。传统算法依赖于火灾特征的手动选择和机器学习分类,由于火灾类型和场景复杂,且实际应用中干扰因素多,导致提取复杂图像特征的算法难以区分火灾和类火,所以精度较低、泛化能力弱。基于卷积神经网络(CNN)的图像识别算法可以有效地自动学习和提取复杂的图像特征,该算法引起了广大的关注,并在视觉搜索、自动驾驶、医疗诊断等方面取得了优异的表现。因此,一些学者将 CNN 引入图像火灾检测领域,并开发了烟雾和火焰检测算法[5-6]。Maksymiv 等学者[7]提出了一种火灾检测方法,结合 AdaBoost 和局部二值模式来获取感兴趣的区域,并使用 CNN 来检测火灾以减轻误报。段锁林等学者[8]对支持向量机(SVM)进行改进来实现对疑似火焰的检测,但是对火焰的检测效果存在泛化能力差的缺点。Wang 等学者[9]将 CNN 与支持向量机相结合进行火灾检测。该方法利用 CNN 提取火灾图像的深层特征,然后利用 SVM 算法构建分类模型来实现火灾检测的功能。这些火灾探测算法的性能远优于传统的基于图像的火灾检测算法,为火灾检测算法的发展提供了新的可能性。

目前在火灾检测的研究中,同时对火焰和烟雾进行检测的算法非常少,而且缺少火焰烟雾数据集。如果能对火灾中的火焰和烟雾同时进行高准确度的实时检测,则能够辅助人群疏散工作和消防员灭火计划的制定,以最大程度减少火灾带来的各方面损失。针对这些问题,本文提出了一种改进 YOLOv5 的烟雾火焰检测算法,并且自建了高质量的烟雾火焰数据集,利用数据集增强方法将数据集扩大到了 10 320 张,以提高模型的泛化能力;利用卷积注意力机制(Convolutional Block Attention Module,CBAM)来改进 YOLOv5 中的 CSP1 结构,使得网络能够更加关注火焰烟雾的重要特征,从而提高模型对火焰烟雾的检测效果。实验结果表明,该算法能够满足火灾实时检测的准确性。

1 数据与算法

1.1 数据获取与标注方法

本文自建了 2 064 张高质量烟雾火焰数据集,所用构建数据集的烟雾和火焰图片是通过编写爬虫程序从网上获得的。该数据集的火灾情景包含了白天火灾、黑夜火灾、森林火灾、车辆火灾、建筑物火灾以及室外火灾,为了更好地对火灾中的火焰和烟雾进行检测,数据集中还包含了蜡烛火焰、火炬火焰、打火装置产生的火焰以及物体燃烧形成的烟雾。本文标注数据集所使用的软件是 LabelImg,标注图片示例如图 1 所示,该软件会将图片标注中的类别信息和目标位置信息保存在 XML 文件中,由于 YOLOv5 训练需要的数据集文件是 TXT 格式的,所以本文使用 Python 程序将 XML 文件转换为 YOLOv5 所需的 TXT 文件。因为火焰和烟雾都是没有固定形状的,所以在标注数据集时,应该尽量使得标注框中的背景信息最小化,否则容易影响训练的效果,所以本文在标注火焰和烟雾时参见图 1,尽可能地减少标注框中的背景信息。

（a）未标注图　　（b）标注图

图 1　标注图片示例

Fig. 1　Label picture example

1.2　数据集增强

神经网络的有效训练需要大量的数据。因此在数据量不足的情况下，网络不能得到有效的学习，使得训练得到的目标检测模型泛化能力差。Krizhevsky 等学者[10]通过数据集增强有效地缓解了这种情况，同时使用随机裁剪和水平翻转来评估 CIFAR 数据集上的深度卷积神经网络，证明了数据集增强的有效性。故本文在自建的 2 064 张数据集的情况下，进行数据集的增强操作，将数据集扩充到了 10 320 张，丰富了类别的坐标信息。数据集增强的目的是增加输入图像的可变性，使设计的目标检测模型对不同环境下获得的图像具有更高的鲁棒性，增强模型的泛化能力。本文数据集增强的方法包括：平移、裁剪、水平镜像和图片拼接。由于数据集在经过增强操作后，数据集图片中的类别标注位置信息会发生改变，而图片标注的类别和位置信息存储在 XML 文件中，所以数据集增强不仅会改变原来的图片，而且还会改变标注信息，所以也要对相应的 XML 文件进行修改。本文是通过编写 Python 程序来实现数据集增强的，接下来对本文的数据集增强方法进行讲解。

对于数据集的平移和裁剪操作，都是先获取 XML 标注文件中的所有 object 属性，object 中又包含了 name 和 bndbox 属性，其中 name 包含标注框的类别名称，bndbox 包含了标注框的位置信息：$(x_{\min},y_{\min},x_{\max},y_{\max})$，这里 $x_{\min}$ 和 $y_{\min}$ 表示标注框左上角的坐标值，$x_{\max}$ 和 $y_{\max}$ 表示标注框右下角的坐标值。本文对数据集进行增强后不会改变标注框的数目以及每个标注框的大小，因此会先根据所有 object 中的 bndbox 标注框位置数值来得出原图片中能包含所有标注物体的最小外框 $(\tilde{x}_{\min},\tilde{y}_{\min},\tilde{x}_{\max},\tilde{y}_{\max})$，原图的外框为 $(0,0,w,h)$，其中 w 表示原图片的宽，h 表示原图片的高，由此就能得出图片上下左右所能裁剪和平移的最大值：(x_L,x_R,y_T,y_B)，对于左右和上下平移的数值，这里取随机值，可以表示为：

$$\begin{cases} x = random.\ uniform(-(x_L-1),\ x_R) \\ y = random.\ uniform(-(y_T-1),\ y_B) \end{cases} \tag{1}$$

其中，*random. uniform*( ) 表示 Python 中用于生成一个指定区间范围内随机数的函数；x 表示原图向左或者向右移动的像素值，若 x 为正数、原图向左移动，为负数、原图向右移动；y 表示原图向上或者向下移动的像素值，若 y 为正数、原图向下移动，为负数、原图向下移动。原图经过平移后，XML 文件中的 bndbox 的坐标数值也要随之改变，变为：$(x_{\min}+x,y_{\min}+y,x_{\max}+x,y_{\max}+y)$。

对于对原图的裁剪操作，这里也是用到了 *random. uniform*( ) 取随机值的函数，具体公式为：

$$\begin{cases} c_L = x_{\min} - random.\ uniform(0,x_L) \\ c_T = y_{\min} - random.\ uniform(0,y_T) \\ c_R = x_{\max} + random.\ uniform(0,x_R) \\ c_B = y_{\max} + random.\ uniform(0,y_B) \end{cases} \tag{2}$$

其中，c_L 表示对原图左侧裁剪的像素值大小；c_T 表示对原图上侧裁剪的像素值大小；c_R 表示对原图右侧裁剪的像素值大小；c_B 表示对原图下侧裁剪的像素值大小。对原图进行裁剪后，对应的 XML 文件中的 bndbox 坐标数值变为：$(x_{\min}-c_L,y_{\min}-c_T,x_{\max}-c_L,y_{\max}-c_T)$。

对数据集的水平镜像操作是调用 Opencv 库中的 flip( ) 函数进行处理的，水平镜像后，对应的 XML 文件中的 bndbox 坐标数值变为：$(w-x_{\min},y_{\min},w-x_{\max},y_{\max})$，其中 w 为原图的宽。

为了更好地说明本文的数据集增强方法，这里先用 Python 读取 XML 文件中图片的标注框的位置信息和类别信息，然后利用 Opencv 将标注框显示在图片中，形成标注显示图，如图 2 所示。图 2(a) 为原图标注显示图，其中红色框标注的是烟雾，绿色框标注的是火焰，整体上表示的是数据集增强前的标

注显示图。图 2(b)、图 2(c)和图 2(d)分别表示原图经过平移、裁剪和水平镜像后的效果标注显示图，对数据集图片的平移和水平镜像不会改变图片的大小，裁剪则会改变图片的大小。

(a) 原图标注显示图

(b) 平移后标注显示图

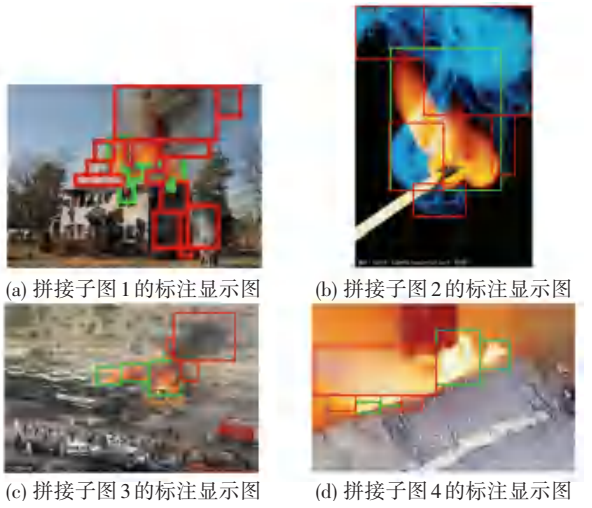
(c) 裁剪后标注显示图

(d) 水平镜像后标注显示图

图 2 标注显示图

Fig. 2 Label display picture

对于数据集的拼接操作，本文是从数据集图片文件夹中随机选出 4 张图片用来以对角的方式进行拼接，且 4 张图的大小不一定相同，拼接后同样会生成对应的图片和 XML 文件。同上，为了更好地说明数据集增强方法，这里同样将数据集中的标注文件 XML 中的标注框直接显示在原图上，红色框标注的是烟雾，绿色框标注的是火焰，4 张拼接子图的标注显示如图 3 所示。



(a) 拼接子图 1 的标注显示图　(b) 拼接子图 2 的标注显示图

(c) 拼接子图 3 的标注显示图　(d) 拼接子图 4 的标注显示图

图 3 4 张拼接子图的标注显示图

Fig. 3 Label display picture of four splicing subgraphs

本文首先会根据随机选出的 4 张图作为拼接子图，由 4 张拼接子图的宽(w_1、w_2、w_3、w_4) 和高(h_1、h_2、h_3、h_4) 求出能容纳 4 张拼接子图(2×2 的排列方式) 的大图的宽(W) 和高(H)，大图的 W 和 H 计算过程可以定义为：

$$\begin{cases} W = \max(w_1, w_3) + \max(w_2, w_4) \\ H = \max(h_1, h_2) + \max(h_3, h_4) \end{cases} \tag{3}$$

其中，max( ) 表示 Python 中求取最大值的函数。

接下来，找到 4 张图片的拼接交点 $P(x,y)$，其求解过程如下：

$$\begin{cases} P.x = \max(w_1, w_3) \\ P.y = \max(h_1, h_2) \end{cases} \tag{4}$$

由求得的 4 张图片的拼接交点 $P(x,y)$ 得到拼接大图的标注框信息，将其存储在新的 XML 文件中，该 XML 文件自然要包含用于拼接的 4 张图的所有标注框信息，由于大图是 4 张图拼接而成，所以大图完全可以分为 4 张子图，4 张子图的所有标注框的位置对应在大图中的位置其实是相当于平移处理的。第 1 张子图在大图中的所有标注框位置相当于将拼接子图 1 的所有标注框平移了 ($P.x - w_1$, $P.y - h_1$) 个像素单位，第 2 张子图在大图中的所有标注框位置相当于将拼接子图 2 的所有标注框平移了 ($P.x$, $P.y - h_2$) 个像素单位，第 3 张子图在大图中的所有标注框位置相当于将拼接子图 3 的所有标注框平移了 ($P.x - w_3$, $P.y$)，第 4 张子图在大图中的所有标注框位置相当于将拼接子图 4 的所有标注框平移了 ($P.x$, $P.y$)，最终拼接的大图的标注显示如图 4 所示。

图 4　拼接后的大图的标注显示图

Fig. 4　The label display picture of the spliced large picture

1.3　YOLOv5 算法

1.3.1　YOLOv5 结构

YOLOv5 是一种单阶段的目标检测算法，是在 YOLOv4[11] 的基础上加以改进的。YOLOv5s 的权重数据为 27 M，相当于 YOLOv4 的 1/9，这使其具有更快的检测速度。据 YOLOv5 的官方信息中显示，该算法最快 0.007 s 处理一张图片，可以满足实时检测。同时，也适合部署在计算、内存和能耗要求有限的机载设备中。YOLOv5 的网络结构如图 5 所示，主要由 Input、Backbone、Neck、Prediction 四个模块组成。

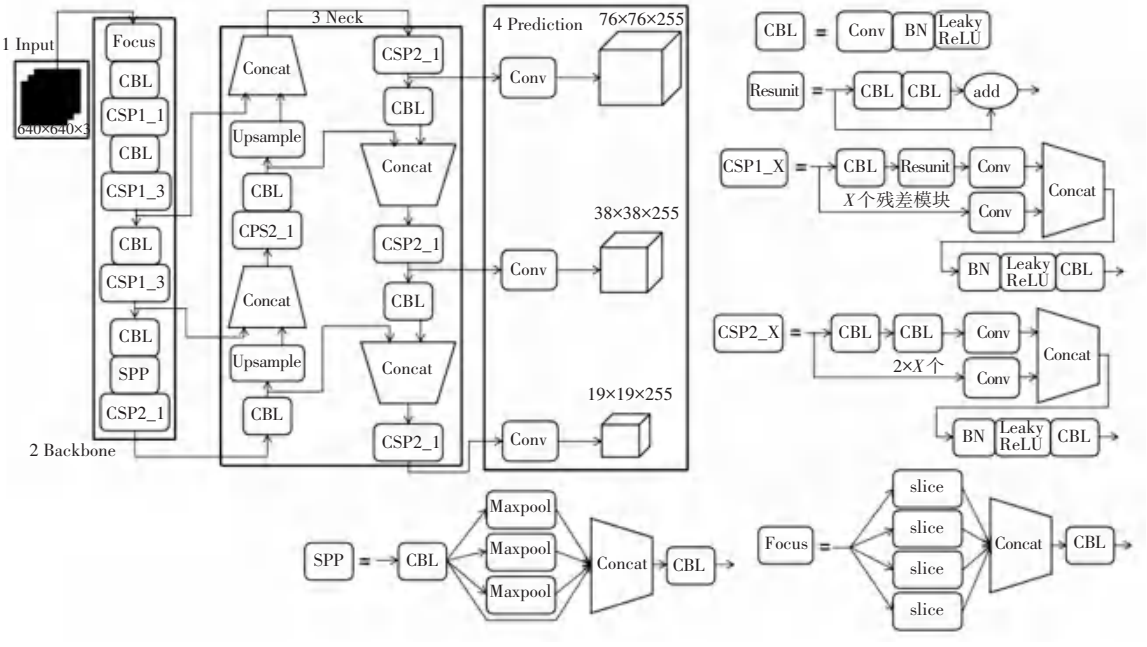


图 5　YOLOv5 网络结构图

Fig. 5　YOLOv5 network structure diagram

Input 模块可以处理 Mosaic 数据增强、自适应锚框的计算和自适应图像的缩放。Mosaic 数据增强是在 YOLOv4 的研究中提出的，方法是随机裁剪 4 张图片，拼接成一张图片，然后将图像作为训练数据来增强背景信息并提高 *batch_size*。批量归一化可以同时计算 4 张图像，让单个 GPU 可以很好地训练模型。自适应锚框的计算就是预先定义一组默认的边框，训练时的训练样本由真实边框与默认边框的偏移量构建。自适应图像的缩放可以在进入训练时实现各种分辨率图像的不匹配。常用的目标检测算法中不同图像的长宽不一样，因此常用的方法是将原始图像缩放成标准尺寸，然后加载到检测网络中。

Backbone 模块由 Focus 结构和 CSP(Cross Stage Partial)结构组成。Focus 结构是隔行采样拼接结构。YOLOv5 的默认输入是 640×640×3。研究中，首先复制 4 张图片，然后将 4 张图片通过切片切割成 320×320×3 的 4 个部分；接下来利用 Concat 从深度方向将这 4 个部分连接起来，将其输出到 320×320×12；之后再采用卷积核数为 32 的卷积层生成 320×320×32 的输入；最后通过归一化和激活函数输入到下一个卷积层。CSP 结构利用了 CSPNet(Cross Stage Partial Network)的设计理念。YOLOv5 中设计了 2 种 CSP 结构，以 YOLOv5s 的网络为例，CSP1_X 结构应用于 Backbone 网络，CSP2_X 结构应用于 Neck 模块中。CSP1_X 和 CSP2_X 都展示在图 5 中，每个 CSP 模块前面的卷积核大小都是 3×3、*stride* = 2，起到了下采样的作用。采用 CSP 模块，能在保证精度的同时减少计算量。

YOLOv5 的 Neck 模块采用 FPN+PAN 结构。FPN 可以改变不同层次特征的尺度，然后整合信

息,从而提取出较低层次的信息。通过不同层次特征的融合,FPN 可以比较完整地反映小物体的信息。YOLOv5 中添加了具有自下而上特征的金字塔结构,包括 2 个 PAN 结构。强语义特征可以通过 FPN 层从上到下传递。特征金字塔可以自下而上传达强定位特征,因此不同的主干层可以对不同的检测层进行参数聚合,从而进一步提高特征提取的能力。

Prediction 模块用来完成对目标的预测。YOLOv5 可以将输入的图像划分为 $N \times N$ 的识别网格。每个网格都有几个预测的边界框和相应的置信度,以获得类概率图[12],除此之外,还可以获得最终的检测效果。每个网格可以预测 α 个边界框以及相应的置信度分数。置信度分数反映了模型对网格的预测,其计算公式为:

$$confidence = P_r(Object) \times IoU_{\text{pred}}^{\text{truth}} \tag{5}$$

若当前 box 有对象的概率 $P_r(Object)$ 不存在,则置信度为 0;否则为 1,置信度为预测框和真实框的交集与并集的比值。YOLOv5 对每个预测的边界框有 5 个预测值:x、y、w、h、$confidence$。坐标 (x,y) 表示预测边界框中心和网格边界的相对值,w 和 h 为预测的边界框的宽和高,$confidence$ 是预测边界框和真实框的 IoU 值,其计算公式为:

$$IoU = \frac{|A \cap B|}{|A \cup B|} \tag{6}$$

YOLOv5 的预测部分计算 $GIoU\_Loss$ 损失函数的损失值,可由下式进行计算:

$$GIoU = IoU - \frac{|C \backslash (A \cup B)|}{|C|} \tag{7}$$

其中,C 表示覆盖框 A 和框 B 的最小外框。

1.3.2 YOLOv5 的自适应锚框技术

YOLOv5 优化了对数据集的预处理,自动学习边界框锚点以此来获得适合预测自定义数据集中物体边界框的预设锚框。自适应锚框技术是通过更新每次迭代的预测框区域来更新目标框,因为目标检测的准确率与预测框的设置密切相关。预测框越准确,其检测的准确率就越高。这里使用 K-means 聚类算法自动对自定义的数据集中标记的目标边界框进行聚类,产生不同数量和大小的先验框。这种方法可以提高先验框与实际目标框的匹配度,从而进一步提高检测精度。

K-means 是数据挖掘中典型的聚类算法,广泛用于对数据集进行聚类的操作中。K-means 算法由 MacQueen 等学者在 1967 年首先提出,是最简单的非监督学习算法之一,可用于解决众所周知的聚类问题[13]。K-means 是一种分区聚类算法,这种方法是通过迭代将给定的数据对象分类到 k 个不同的簇中,收敛到局部最小值。因此生成的聚类结果紧凑且独立。该算法由 2 个独立的阶段组成。

(1)第一个阶段:随机选择 k 个中心,其中 k 值是预先固定的。

(2)第二个阶段:是将每个数据对象带到最近的中心。一般都用欧氏距离来确定每个数据对象与聚类中心之间的距离。当所有的数据对象都包含在一些簇中时,第一步就完成了,并进行了早期分组,重新计算早期形成的簇的平均值。这个迭代过程不断重复,直到准则函数变为最小值。

1.4 改进 YOLOv5 中的 CSP1 模块

网络模块中融入特征增强技术能提高网络抗干扰能力,也能提高网络模型对物体目标检测的识别效果。注意力在人类感知中起着重要作用[14],人类视觉系统的一个重要特征是不会尝试一次性处理整个场景。人类是利用一系列局部瞥见并选择性地关注显著部分,以便更好捕捉视觉结构[15]。近年来,注意力网络机制在增强特征提取的操作方面效果显著,能够关注网络中的重要特征,抑制网络中的不重要的特征,有效提升网络的性能。受 Woo 等学者[16]提出的通道和空间注意力机制(CBAM)的启发,本文在 YOLOv5 中的 CSP1 模块中融入了 CBAM 注意力机制,以此来加强网络对火焰和烟雾的检测效果。

1.4.1 通道和空间注意力模块

CBAM 包括通道注意力模块和空间注意力模块,其整个注意力过程可以用下式来表示:

$$F' = M_C(F) \otimes F \tag{8}$$

$$F'' = M_S(F') \otimes F' \tag{9}$$

其中,"⊗"表示的是逐元素乘法;F' 表示通道注意力的输出特征结果;F'' 表示最终的输出特征结果。下面分别对通道注意力模块和空间注意力模块的细节部分展开研究论述。

通道注意力模块使模型更加关注特征图的通道信息。首先通过使用平均池化和最大池化操作来聚合特征图的空间信息,得到平均池化特征 F_{avg}^C 和最大池化特征 $F_{\max}^C$;接着将得到的这 2 个特征输入到共享网络以生成通道注意力图 $M_C \in R^{C\times1\times1}$,共享网络由具有一个隐藏层的多层感知器(MLP)组成;为了减少参数的开销,隐藏激活函数大小设置为 $R^{C/r\times1\times1}$,其中 r 是缩减率;最后使用逐元素求和合并输出特征向量。通道注意力的计算公式具体如下:

$$M_C(F) = \sigma(MLP(AvgPool(F)) + MLP(MaxPool(F))) = \sigma(W_1(W_0(F^C_{avg})) + W_1(W_0(F^C_{max}))) \quad (10)$$

其中，σ 表示 Sigmoid 函数；$W_0 \in R^{C/r\times 1}$；$W_1 \in R^{C\times C/r}$。通道注意力模块示意如图 6 所示。

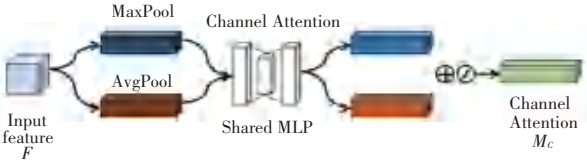


图 6　通道注意力模块示意图

Fig. 6　Schematic diagram of channel attention module

空间注意力模块使模型更加关注特征图的空间信息。首先，使用 2 个池化操作来聚合特征图的通道信息，生成跨通道的平均池化特征图 $F^S_{avg} \in R^{1\times H\times W}$ 和跨通道的最大池化特征图 $F^S_{max} \in R^{1\times H\times W}$，然后通过标准卷积层对其进行连接和卷积操作，生成二维的空间注意力图 $M_S(F) \in R^{H\times W}$。空间注意力的计算公式具体如下：

$$M_S(F) = \sigma(f^{7\times 7}([AvgPool(F); MaxPool(F)])) = \sigma(f^{7\times 7}(F^S_{avg}; F^S_{max})) \quad (11)$$

其中，σ 表示 Sigmoid 函数，$f^{7\times 7}$ 表示滤波器大小为 7×7 的卷积运算。空间注意力模块示意见图 7。

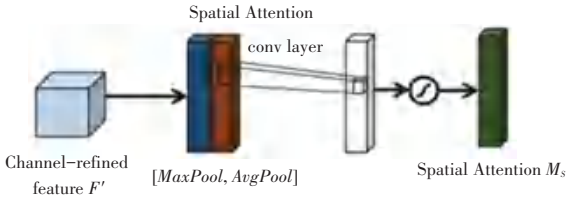


图 7　空间注意力模块示意图

Fig. 7　Schematic diagram of spatial attention module

1.4.2　融入通道和空间注意力模块

本文结合通道注意力模块和空间注意力模块的特点，在 YOLOv5 中的 CSP1 结构中融入通道和空间注意力模块，能够让网络模型在关注通道和空间信息的同时也关注特征图，进而提高网络模型对火焰和烟雾的特征提取能力，提高模型的识别效果。其网络示意如图 8 所示。

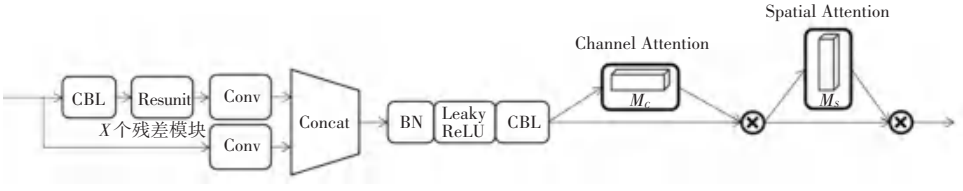


图 8　CSP1 中融入注意力模块示意图

Fig. 8　Schematic diagram of integrating attention module in CSP1

1.4.3　损失函数

YOLOv5 的损失函数由 3 部分组成，分别是坐标损失函数（$Loss_{iou}$）、置信度损失函数（$Loss_{coord}$）和分类损失函数（$Loss_{class}$）。YOLOv5 的损失函数（$Loss$）计算公式具体如下：

$$Loss = Loss_{iou} + Loss_{coord} + Loss_{class} \quad (12)$$

$$Loss_{iou} = 1 - GIoU \quad (13)$$

$$Loss_{coord} = \sum_{i=0}^{s^2}\sum_{j=0}^{B} I^{obj}_{ij}[\hat{C}^j_i \log(C^j_i) + (1-\hat{C}^j_i)\log(1-\hat{C}^j_i)] - \lambda_{nobj}\sum_{i=0}^{s^2}\sum_{j=0}^{B} I^{nobj}_{ij}[\hat{C}^j_i \log(C^j_i) + (1-\hat{C}^j_i)\log(1-\hat{C}^j_i)] \quad (14)$$

$$Loss_{class} = \sum_{i=0}^{s^2} I^{obj}_{ij} \sum_{C\in classes}[\hat{P}^j_i \log(P^j_i) + (1-\hat{P}^j_i)\log(1-\hat{P}^j_i)] \quad (15)$$

式(13)中的 $GIoU$ 计算公式在 1.3.1 节中的式(7)已经作了说明，式(14)中的 I^{obj}_{ij} 和 I^{nobj}_{ij} 分别表示第 i 个网格中第 j 个检测框内是否有目标，有目标、为 1，没有目标则为 0；λ_{nobj} 为坐标误差的损失权重；C^j_i 和式(15)中的 P^j_i 为真实值，$\hat{C}^j_i$ 和式(15)中的 $\hat{P}^j_i$ 为网络预测值。

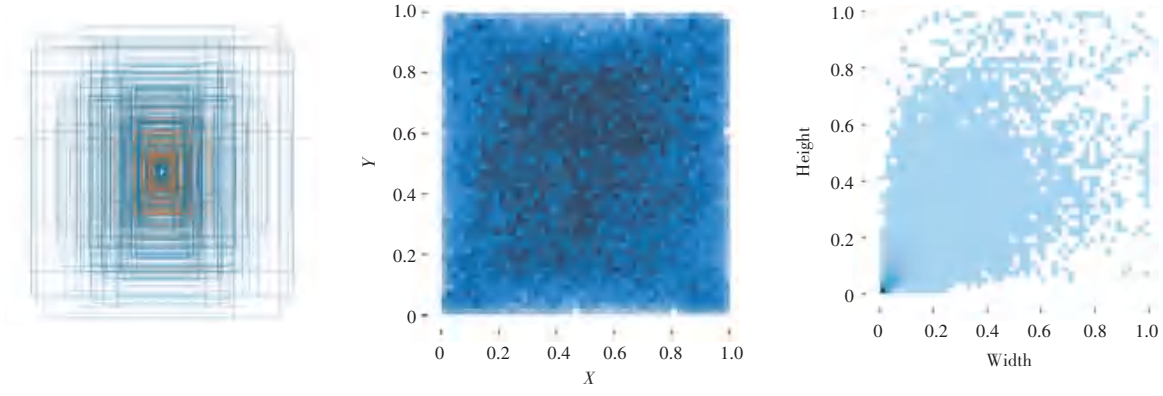
2　实验结果及分析

2.1　实验平台与训练过程

本文模型的训练是基于 64 位 ubuntu18.04 操作系统和深度学习框架 Pytorch1.80 下完成的。测试设备的 CPU 型号为 Intel Core i5-10200H，内存 16 G，GPU 型号为 NVIDIA GTX1650Ti 4G 独立显卡，仿真软件环境为 CUDA 10.2、CUDNN 7.6、Python 3.8，对于代码的调试以及训练使用的软件是 Pycharm。

本文火灾数据集的图片来源于网络爬虫的获取，构建了 2 064 张高质量火焰烟雾数据集，并且通过数据增强技术，将数据集扩充到了 10 320 张。数

据集包括 2 类:火焰和烟雾。数据集分为 7 224 张图片作为训练集、1 548 张图片作为验证集和 1 548 张图片作为测试集。通过原始 YOLOv5 和改进后的 YOLOv5 分别进行训练。训练过程中的参数做了如下设置:输入图像的尺寸为 640×640,批处理大小设置为 8,迭代次数为 1 000,动量为 0.937,初始学习率为 0.01,权重衰减系数为 0.000 5。在训练前会通过 YOLOv5 的自适应锚框技术对火焰烟雾数据集进行聚类分析,以便获得更好的检测效果,如图 9(a)所示,直观展示了火焰烟雾数据标签的锚框分布情况,图 9(b)为数据标签上的标准化目标位置图,图 9(c)为数据标签的归一化目标尺寸图。



(a) 锚框分布图 (b) 标准化目标位置图 (c) 归一化目标尺寸图

图 9 烟雾火焰数据集统计结果图

Fig. 9 Statistical results of smoke and flame dataset

2.2 实验结果与分析

2.2.1 实验评价指标

本文引入准确率(P)、召回率(R)和均值平均准确率(mAP)来评估火焰烟雾模型的检测性能,其中 P 和 R 的表达式如下:

$$P = \frac{TP}{TP + FP} \tag{16}$$

$$R = \frac{TP}{TP + FN} \tag{17}$$

其中,TP(True Positive)表示真实情况是正样本,预测结果也是正样本;TN(True Negative)表示真实情况是负样本,预测结果也是负样本;FP(False Positive)表示真情况是负样本,预测结果为正样本;FN(False Negative)表示真实情况为正样本,预测结果为负样本。

AP 为平均准确率,是 P 指标对 R 指标的积分,即 P-R 曲线下的面积;mAP 是均值的平均准确率,是把每个类别的 AP 值相加,再除以类别的数量,即求取平均值。AP 和 mAP 的表达式如下:

$$AP = \int_0^1 P(R)\,\mathrm{d}R \tag{18}$$

$$mAP = \frac{1}{|Q_R|}\sum_{q=Q_R} AP(q) \tag{19}$$

其中,Q_R 表示类别的数量。

2.2.2 消融实验

为了验证本文改进算法的优越性,将原始 YOLOv5 算法训练得到的模型(原始模型)和本文改进的算法训练得到的模型(改进模型)对火灾数据集中的测试集进行检测,对比检测效果。

原始模型和改进模型的测试结果见表 1。对比原始模型的检测效果,改进后的模型在准确率和召回率上都有提升。改进的模型在对火焰检测的准确率上提升了 1.5%,在对火焰检测的召回率上提升了 0.4%,在对烟雾检测的准确率上提升了 2.4%,在对烟雾检测的召回率上提升了 0.6%。在 IoU 阈值为 50%的情况下,原始模型检测的 $mAP@0.5$ 为 88%,改进模型检测的 $mAP@0.5$ 为 90%,相比原始模型提高了 2%。通过测试发现,尽管模型的复杂度增加了,但改进后的网络模型大小和原始模型大小相同,在检测速度方面,原始模型为 23.6 ms/帧,改进模型为 23.7 ms/帧,完全满足对火焰和烟雾的实时检测要求。

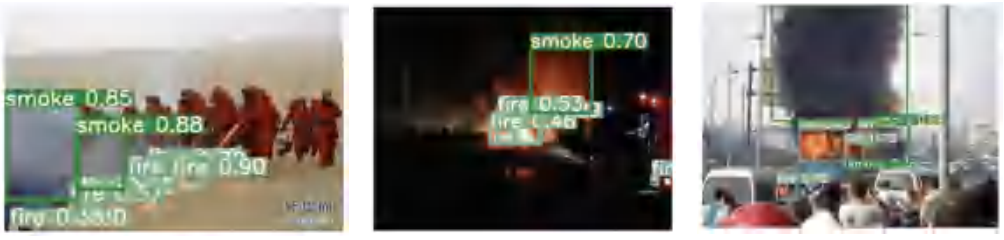
表 1 原始模型和改进模型之间的准确率、召回率以及 *mAP* 对比

Table 1 Comparison of accuracy, recall and *mAP* between the original model and the improved model %

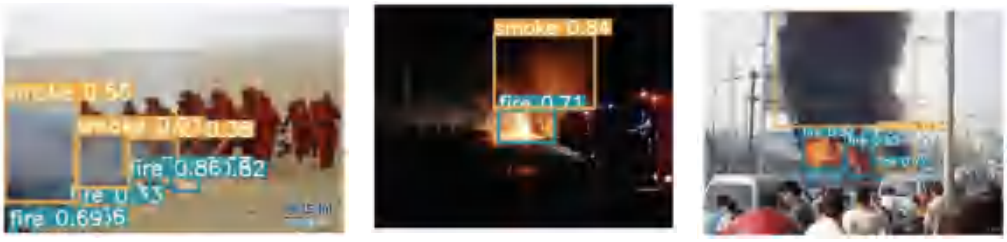
Algorithm	Category	*Precision*	*Recall*	*mAP*
Original YOLOv5	fire	86.3	88.0	88
	smoke	81.6	87.9	88
Improved YOLOv5	fire	87.8	**88.4**	90
	smoke	**84.0**	**88.5**	90

本文算法的实际检测效果如图 10 所示。从图 10(a1)和图 10(b1)的对比可以看出,原始 YOLOv5

漏检了一部分烟雾区域，由于本文在原始算法中融入了注意力机制，使得算法能够更好地关注目标区域，因此也能得到更好的检测效果。从图 10(a2)和图 10(b2)的对比可以看出原算法把发光区域也检测成了火焰，造成了误检，这说明本文改进的算法抗干扰能力强，不易受灯光等物体的影响。在图 10(a3)和图 10(b3)的对比中，原算法把人脸也检测成了火焰，但改进后的 YOLOv5 算法有效地解决了上述问题，并且在复杂环境下也具有良好的鲁棒性，证明了网络的有效性，本文改进后的算法模型大小仅为 14.4 MB，可以很好地应用到实际火灾检测场景的预警防护中。



(a1) 原始 YOLOv5 训练后检测结果 1　(a2) 原始 YOLOv5 训练后检测结果 2　(a3) 原始 YOLOv5 训练后检测结果 3



(b1) 改进的 YOLOv5 训练检测结果 1　(b2) 改进的 YOLOv5 训练检测结果 2　(b3) 改进的 YOLOv5 训练检测结果 3

图 10　不同算法识别效果对比

Fig. 10　Comparison of recognition effects of different algorithms

2.3　对比实验

为了进一步验证本文改进的 YOLOv5 算法的性能，将其与目前主流的检测算法(Faster RCNN[17]、SSD[18]、YOLOv3[19])，以及最近非常受欢迎的目标检测算法 Efficient Det[20] 和 Center Net[21] 进行比较，对比各算法在本文数据集下检测结果的 *mAP* 以及 *FPS*。*FPS* 表示目标检测方法每秒可以处理的图片数量，也就是检验了检测方法的实时性能。

本文改进算法与上面提到的检测算法的比较结果见表 2。根据表 2 的检测结果可以看出，与主流检测算法相比，本文改进的 YOLOv5 网络具有更高的 *mAP*，相比于 Faster RCNN、SSD、YOLOv3，*mAP* 分别提高了 2.7%、13.3%、16.8%，优于目前主流的这几种检测方法。虽然 Faster RCNN 也有很高的 *mAP*，但是其在 *FPS* 方面仍然逊色于本文改进的算法，在对火焰和烟雾的实时检测速度方面明显弱于本文改进的算法。再来分析最近备受瞩目的 2 个目标检测算法 Efficient Det 和 Center Net，两者在检测精度上都要优于主流的目标检测算法，而且检测速度也非常快，这 2 个算法是近年来提出的检测效果非常好的算法，在检测速度上都高于本文改进的算法。但是相比于检测精度，本文改进算法的表现则更好，*mAP* 分别提高了 0.8%和 1.3%。经过对比以上优秀目标检测算法与本文改进的算法的检测结果，表明本文改进算法的优越性。

表 2　改进 YOLOv5 与主流检测算法对比

Table 2　Comparison between improved YOLOv5 and mainstream detection algorithms

算法	*mAP*/ %	*FPS*
Faster RCNN	87.3	28
SSD	76.7	47
YOLOv3	73.2	45
Efficient Det	89.2	43
Center Net	88.7	46
Improved YOLOv5	**90.0**	**42**

3　结束语

本研究将深度学习技术应用于火灾的火焰和烟雾检测，由于目前对于火灾检测的研究，要么只检测火焰，要么只检测烟雾，很少有同时检测火焰和烟雾的研究，并且也缺少火焰烟雾数据集。在火灾发生后，对火焰和烟雾的检测都很重要，如果能够同时检测出火焰和烟雾，就可以从中得到安全区域和火焰烟雾区域，对于火灾发生后的人群疏散工作以及辅

助制定消防灭火计划都可提供强大助益,对保护人们的生命安全以及减少经济损失都起着至关重要作用。因此,本文自建了高质量的火焰烟雾数据集,包含了多种复杂场景,并且对数据集进行了数据增强,使得到的火焰烟雾检测模型对不同环境下获得的图像检测具有更高的鲁棒性,增强了模型的泛化能力;针对原始 YOLOv5 算法对火焰和烟雾的检测效果不好、容易造成漏检和误检的问题,本文结合通道注意力模块和空间注意力模块这些特征增强技术的特点,在 YOLOv5 中的 CSP1 结构中融入通道和空间注意力模块,提高了网络模型对火焰和烟雾的特征提取能力,解决了原始 YOLOv5 算法造成的漏检和误检的问题,提高了模型的检测效果。实验结果表明,改进后的算法在对火焰和烟雾的检测中,相比原始模型在检测准确率和召回率上都有提升,改进后的算法对火焰和烟雾的总体检测准确率和召回率分别为 85.9% 和 88.4%,检测速度能达到 42.2 FPS,能够用于火灾中火焰和烟雾的实时检测,对辅助火灾发生后的人群疏散以及制定灭火计划都具有非常重要的意义。

本文主要针对实时检测要求下的火焰和烟雾进行研究和开发。但是,快速检测仍然需要好的硬件配置。未来会继续优化本文改进后的模型,进一步提升模型对火焰和烟雾的检测精度以及实时检测速度,同时也会继续丰富火焰烟雾数据集,提高数据集场景的复杂度,让模型能够有更好的检测效果。

参考文献

[1] 高文. 森林防火存在的问题及建议[J]. 现代农业科技, 2018(21): 142-143.

[2] SUN Xuetao. Independent photoelectric smoke fire detection alarm [J]. Fire Protection Technology and Products Information, 2017(1):00297-00298.

[3] LIU Aiqun. Design of photoelectric smoke and temperature sensitive composite fire detector[J]. J. Prospects for Science and Technology, 2017, 27(22): 158-160.

[4] ZHONG Chen, LI Xiaobai, DING Hongjun, Sensitivity detection of image smoke fire detector based on pixel ratio[J]. Fire Science and Technology, 2017, 36(2): 220-223.

[5] MAO Wentao, WANG Wenpeng, DOU Zhi, et al. Fire recognition based on multi-channel convolutional neural network [J]. Fire Technology, 2018, 54: 531-554.

[6] HU Yaocong, LU Xiaobo. Real-time video fire smoke detection by utilizing spatial-temporal ConvNet features [J]. Multimedia Tools and Applications, 2018, 77: 29283-29301.

[7] MAKSYMIV O, RAK T, PELESHKO D. Real-time fire detection method combining AdaBoost, LBP and convolutional neural network in video sequence [C]//2017 14th International Conference the Experience of Designing and Application of CAD Systems in Microelectronics (CADSM). Piscataway, NJ: IEEE, 2017: 351-353.

[8] 段锁林, 任珏朋, 毛丹,等. 基于改进的 PSO 优化 SVM 火灾火焰识别算法研究[J]. 计算机测量与控制, 2016, 24(4): 202-205.

[9] WANG Zhicheng, WANG Zhiheng, ZHANG Hongwei, et al. A novel fire detection approach based on CNN-SVM using tensorflow[C]//Proceedings of the 13th International Conference on Intelligent Computing Methodologies (ICIC 2017). Cham: Springer, 2017: 682-693.

[10] KRIZHEVSKY A, SUTSKEVER I, HINTON G E. Imagenet classification with deep convolutional neural networks [J]. Communications of the ACM, 2017, 60(6): 84-90.

[11] BOCHKOVSKIY A, WANG C Y, LIAO H Y M. YOLOv4: Optimal speed and accuracy of object detection[J]. arXiv preprint arXiv, 2004.10934, 2020.

[12] HE Qiwei, ZHAO Yukun, ZONG Zhaoxiang. The design and realization of visual detection system based on Yolov3 algorithm [J]. Digital Technology and Application, 2020, 38(8): 128-131.

[13] SUN Jigui, LIU Jie, ZHAO Lianyu. Clustering algorithms Research[J]. Journal of Software, 2008, 19(1): 48-61.

[14] RENSINK R A. The dynamic representation of scenes[J]. Visual Cognition, 2000, 7(1-3): 17-42.

[15] LAROCHELLE H, HINTON G E. Learning to combine foveal glimpses with a third-order Boltzmann machine[C]//Proceedings of the 24th Annual Conference on Neural Information Processing Systems. Vancouver, Canada: dblp, 2010: 1-9.

[16] WOO S, PARK J, LEE J Y, et al. CBAM: Convolutional block attention module[C]//Proceedings of the European Conference on Computer Vision (ECCV). Cham: Springer, 2018: 3-19.

[17] REN Shaoqing, HE Kaiming, GIRSHICK R, et al. Faster R-CNN: Towards real-time object detection with region proposal networks[J]. arXiv preprint arXiv, 1506.01497, 2016.

[18] LIU Wei, ANGUELOV D, ERHAN D, et al. SSD: Single shot multibox detector [C]//Proceedings of the 14th European Conference on Computer Vision (ECCV 2016). Cham: Springer, 2016: 21-37.

[19] REDMON J, FARHADI A. YOLOv3: An incremental improvement[J]. arXiv preprint arXiv, 1804.02767, 2018.

[20] TAN Mingxing, PANG Ruoming, LE Q V. EfficientDet: Scalable and efficient object detection [C]//Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. Piscataway, NJ: IEEE, 2020: 10781-10790.

[21] DUAN Kaiwen, BAI Song, XIE Lingxi, et al. CenterNet: Keypoint triplets for object detection [C]//Proceedings of the IEEE/CVF International Conference on Computer Vision. Piscataway, NJ: IEEE, 2019: 6569-6578.