

基于图像识别的火焰定位系统的研究与实现

陆晓东

徽商职业学院 信息工程系, 安徽 合肥 230022

DOI: 10.61369/TACS.2025020002

摘 要： 本文针对火焰着火点的精准定位问题展开研究，提出了一种基于图像识别技术的解决方案，并开发了相应的 Web 应用程序以实现该模型的实际应用。研究的核心在于通过 HSV 色彩空间分析和形态学操作对图像进行优化处理，从而提高火焰着火点的识别精度。此外，为增强系统的鲁棒性，引入了防御性编程策略。在应用开发方面，采用分层架构设计，结合 Flask 框架实现前端交互功能，利用 OpenCV 库实现火焰检测算法，并通过本地文件系统进行数据存储。测试分析结果表明，该系统能够准确识别火焰并有效标注识别结果。性能测试显示，系统的平均处理时间为 120 毫秒，能够支持 10 个并发请求，且响应时间均小于 500 毫秒。这些性能指标表明，该系统具备较高的实时性和稳定性，能够满足实际应用中对火焰着火点快速定位的需求。本文的研究成果为火焰着火点的精准识别与定位提供了一种有效的技术方案，并为后续自动灭火系统的开发提供了重要的技术参考和方向。

关 键 词： 图像识别；着火点；定位；OpenCV

Research and Implementation of a Flame Localization System Based on Image Recognition

Lu Xiaodong

Department of Information Engineering, Huishang Vocational College, Hefei, Anhui 230022

Abstract： This paper focuses on the precise localization of flame ignition points and proposes a solution based on image recognition technology. Additionally, a corresponding web application has been developed to realize the practical application of this model. The core of the research lies in optimizing image processing through HSV color space analysis and morphological operations, thereby improving the recognition accuracy of flame ignition points. Furthermore, defensive programming strategies have been introduced to enhance the robustness of the system. In terms of application development, a layered architecture design is employed, incorporating the Flask framework to achieve front-end interaction. The flame detection algorithm is implemented using the OpenCV library, and data storage is facilitated through a local file system. Testing and analysis results indicate that the system can accurately identify flames and effectively annotate the recognition results. Performance tests show that the system has an average processing time of 120 milliseconds and can support 10 concurrent requests with response times less than 500 milliseconds. These performance metrics demonstrate that the system exhibits high real-time capability and stability, meeting the demand for rapid localization of flame ignition points in practical applications. The research findings presented in this paper provide an effective technical solution for precise recognition and localization of flame ignition points, offering important technical references and directions for the development of subsequent automatic fire extinguishing systems.

Keywords： image recognition; ignition point; localization; OpenCV

引言

火灾是一种极具破坏性的灾害，对人员、财产、环境和社会等多方面造成严重影响。人员可能遭受烧伤、窒息、中毒等直接伤害，甚至死亡，而幸存者还可能面临心理创伤。火灾还会烧毁房屋、商业建筑和设备，导致巨大的经济损失和商业中断^[1]。此外，火灾会破坏生态系统，造成植被损毁、土壤侵蚀以及空气和水资源污染。在社会层面，火灾会增加医疗负担，破坏基础设施，导致社会秩序混乱，甚至影响文化遗产的传承。火灾的长期影响包括生态恢复缓慢、经济复苏困难以及社会心理的持续影响^[2]。因此，火灾的破坏性广

泛且深远，亟需通过加强预防和应急响应能力来减少其带来的损失。

然而，传统的人工灭火方式存在诸多弊端。人工灭火依赖消防人员到达现场操作，响应速度受限，难以在火灾初期迅速控制火势，导致火势可能蔓延扩大。同时，人工灭火劳动强度大，消防人员长时间高强度作业容易疲劳，影响灭火效率和救援效果，且近距离接触火源还面临高温、浓烟、有毒气体等危险，人身安全受到严重威胁^[9]。此外，人工灭火的效率较低，灭火剂喷射距离和覆盖范围有限，难以快速扑灭大面积火灾，且对环境和建筑物的破坏较大，大量用水可能导致二次损坏。人工灭火还受地形、天气等环境条件限制，在复杂场景下行动不便，资源消耗也较大，难以精准控制灭火剂的喷射位置和剂量，可能导致灭火剂浪费。这些弊端使得人工灭火在面对复杂火灾时存在明显不足，因此现代灭火中越来越多地引入自动化和智能化技术来弥补其不足。

随着人工智能技术的不断发展，目前的研究主要集中在火源识别方面，李萌娜^[14]针对无人机森林火灾图像识别与分割的挑战，提出了改进的 Deeplabv3+ 算法通过轻量网络 MobileNetv3 重构和增加浅层特征，既提升了分割精度又实现了较高的处理速度，以进一步提升无人机森林火灾监测的效率和准确性。张权力^[15]对森林火灾早期预警中烟雾检测的准确性和实时性问题，提出了一种基于改进 YOLOv5s 算法的森林防火预警系统。结果表明，改进后的 YOLOv5s+C 模型能够在 30 秒内发出预警，有效支持森林火灾的早期发现和扑救。

针对着火点定位的研究相对较少。本文将在识别到火灾的基础上，进一步定位着火点方位，以便自动灭火设备能够更精准地进行灭火。未来，火源定位技术的研究将逐步深入开展。本文主要使用图像识别技术对火灾着火点位置进行定位程序的研究和设计，该程序采用轻量化设计，旨在提高灭火效率并降低资源消耗。

一、设计理念

本系统深度融合了轻量级架构与计算机视觉技术，旨在构建一个高效、灵活且易于扩展的火焰检测系统。在系统架构方面，采用了分层处理架构，前端交互层通过 Flask 路由实现 HTTP 请求的分流，严格区分 GET 和 POST 方法的处理逻辑，以确保交互的高效性和逻辑的清晰性。业务逻辑层则通过封装函数，将火焰检测的全流程进行高内聚封装，便于维护和扩展。数据存储层通过设置 UPLOAD_FOLDER 和 RESULT_FOLDER 实现了上传文件与结果文件的物理隔离，进一步优化了数据管理的清晰性和安全性。同时，为了增强系统的鲁棒性，设计中融入了防御性编程策略，例如在文件操作前强制检查以避免空指针异常，并自动创建缺失目录。

在视觉检测策略上，系统采用了 HSV 色彩空间分析，通过双阈值检测（ $0^{\circ} - 10^{\circ}$ 和 $160^{\circ} - 180^{\circ}$ ）覆盖火焰红色光谱的全范围，有效解决了单一阈值检测容易漏检的问题。同时，通过限定饱和度（S）和明度（V）的阈值范围为 [0, 255]，有效过滤了灰暗背景的干扰。为了进一步优化图像处理效果，模型采用了先腐蚀（erode）后膨胀（dilate）的形态学操作策略，迭代次数经过实验调优，分别为 2 次和 4 次，以消除噪声并增强火焰区域的显示效果。在轮廓检测中，仅保留最外层轮廓（`RETR\_EXTERNAL`），避免了火焰内部空洞的误判。此外，模型还通过红色矩形框标注火源区域，并配合文字标签提升结果的可解释性，同时将输出图像尺寸固定为 600×400 ，确保与响应式布局的兼容性。

在工程化考量方面，系统动态构建跨平台兼容路径，避免了硬编码分隔符问题，并通过标准化结果文件命名简化了前端调用逻辑。为了平衡性能，系统在图像预处理阶段通过调整图像尺寸控制计算开销，以适配边缘设备的部署需求。同时，模型还提供了 Debug 模式，便于快速迭代开发，而在生产环境中可关闭该模式以优化性能。此外，系统的设计注重扩展性，检测算法与 Web

框架采用松耦合设计，可方便地替换为 YOLO 等深度学习模型，同时文件接口设计支持后续扩展视频流处理功能，为系统的功能扩展提供了灵活性。

模型通过传统计算机视觉方法实现了轻量化火焰检测，在保证实时性的同时，为后续的算法迭代和功能扩展预留了空间，典型适用于中小规模安防场景的快速部署需求。

二、技术方案

本系统整体架构基于三层模型设计：表现层通过 Flask 框架构建 RESTful 接口，支持用户上传文件以及展示检测结果；业务层则依托 OpenCV 实现火焰检测的核心算法；数据层则利用本地文件系统存储用户上传的图片以及检测后的结果。在通信流程方面，用户通过 POST 请求向 Flask 上传图片，Flask 接收后调用 OpenCV 中的函数进行处理，OpenCV 完成检测后将结果返回给 Flask，最后由 Flask 渲染结果页面并展示给用户。

在核心算法实现方面，系统采用双 HSV 阈值空间检测策略，通过低区间（ $0^{\circ} - 10^{\circ}$ ）捕捉明火的黄色部分，以及高区间（ $160^{\circ} - 180^{\circ}$ ）识别火焰的红色核心，同时结合饱和度（S）和明度（V）的阈值过滤，有效去除灰暗背景的干扰。

```
1 get_fire_mask(hsv_img):
2     lower1 = np.array([0, 100, 100])
3     upper1 = np.array([10, 255, 255])
4     lower2 = np.array([160, 100, 100])
5     upper2 = np.array([180, 255, 255])
6     return cv2.bitwise_or(
7         cv2.inRange(hsv_img, lower1, upper1),
8         cv2.inRange(hsv_img, lower2, upper2))
```

图 1 火焰检测核心代码

在图像优化处理环节，系统通过形态学操作组合，先利用腐蚀操作消除细小噪声，再通过膨胀操作增强火焰区域，同时在轮廓分析中仅检测外部边界，避免火焰内部空洞的误判。

在技术创新方面，系统引入了动态路径管理机制，自动适配不同操作系统的路径格式，并结合的目录自动创建机制，显著提升了系统的部署便捷性。

在可视化方面，系统采用红色矩形框（RGB: 0,0,255）标注火源区域，同时将输出图像尺寸固定为 600×400，确保与响应式布局的兼容性，从而提高结果的警示效果和展示效果。

```
1 locate_fire_areas(mask):
2     contours, _ = cv2.findContours(mask,
3         cv2.RETR_EXTERNAL, cv2.CHAIN_APPROX_SIMPLE)
4     return [cv2.boundingRect(c) for c in contours
5         if cv2.contourArea(c) > 100]
```

图2 定位算法实现代码

在性能优化方面，系统通过图像预处理降低计算复杂度，并引入结果缓存机制避免重复检测，显著提升了系统的运行效率。

```
1
2 def preprocess_image(image_path):
3     img = cv2.imread(image_path)
4     if img is None:
5         return None
6     return cv2.resize(img, (600, 400))
```

图3 数据预处理代码

通过三层架构模型清晰划分了各层的功能，确保系统的高效运行和易于维护；核心算法通过双 HSV 阈值检测和形态学优化，精准识别火焰特征并优化检测结果；动态路径管理、可视化增强和性能优化策略等关键技术创新点，进一步提升了系统的实用性和用户体验。

三、程序测试分析

本系统兼容 Windows、Linux 和 macOS 操作系统，要求 Python 版本不低于 3.0，并依赖 Flask（版本≥2.0）、OpenCV（版本≥4.5）以及 numpy 等核心库。为确保依赖环境的独立性，建议在安装过程中创建虚拟环境。

该应用支持上传 PNG 和 JPG 格式的图片，单文件大小限制

为 10MB 以内。上传的图片会被保存至指定目录。火焰检测算法能够准确识别包含火焰的图片，并在检测到火焰区域时标注红色矩形框及“Fire Detected”文字。检测完成后，页面会自动刷新并显示标注后的结果图片，同时检测结果会自动保存。



图4 功能测试图

在性能测试中，应用对测试图片（统一调整为 600×400 分辨率）的平均处理时间约为 120ms，涵盖图像读取、处理及保存的全流程。此外，使用 JMeter 模拟 10 并发请求时，平均响应时间小于 500ms，且无请求失败，表现出良好的并发处理能力。

在异常处理方面，当用户上传非图片文件时，OpenCV 读取会返回 `None`，但程序能够正确处理该异常，避免系统崩溃。对于 10MB 以上的大文件，由于应用在处理前会进行 `resize` 操作，因此内存消耗保持稳定，但处理时间会略有增加。

四、结论

本文介绍了一种基于轻量级架构和计算机视觉技术的火焰检测 Web 应用，旨在提高火灾应对效率。该应用采用分层架构，结合 Flask 框架实现前端交互，利用 OpenCV 实现火焰检测算法，并通过本地文件系统存储数据。系统通过 HSV 色彩空间分析和形态学操作优化图像处理，同时引入防御性编程策略增强鲁棒性。性能测试表明，平均处理时间为 120ms，支持 10 并发请求，响应时间小于 500ms。该技术方案适用于中小规模安防场景的快速部署需求，具有较高的应用价值和推广前景。

参考文献

- [1] 刘伟鹏, 曹玉东. 基于深度学习的林火图像识别算法及实现探究 [J]. 长江信息通信, 2022, 35(01): 29–31.
- [2] 何乃磊, 张金生, 林文树. 基于深度学习多目标检测技术的林火图像识别研究 [J]. 南京林业大学学报 (自然科学版), 2024, 48(03): 207–218.
- [3] 陈欣, 施林苏, 丁洪涛, 等. 图像识别技术在林草火灾预警中的应用及优化 [J]. 广东通信技术, 2020, 40(08): 61–65.
- [4] 李萌娜. 基于深度学习的无人机森林火灾图像识别与分割研究 [D]. 西安理工大学, 2022.DOI: 10.27398/d.cnki.gxalu.2022.000591.
- [5] 张权力. 基于图像识别技术的森林防火预警系统的研究与应用 [D]. 湖北文理学院, 2024.DOI: 10.44305/d.cnki.ghbwl.2024.000040.