Fire and Water Stream Video Recognition Software to Redirect a Water Stream onto a Fire

By Fintan Lyons

Introduction

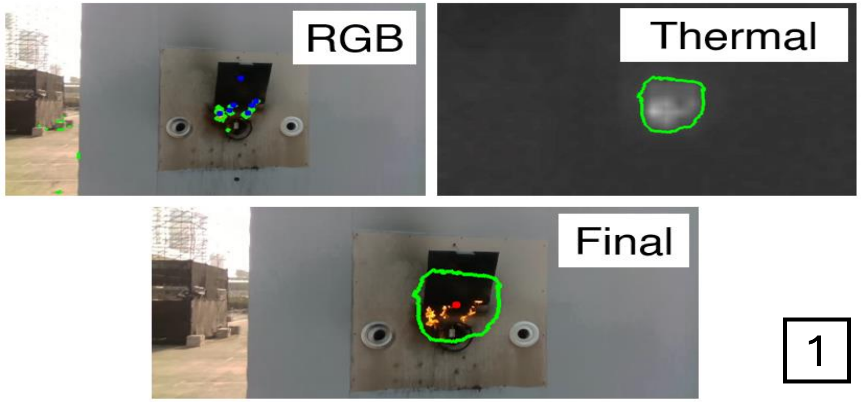
Fires remain a serious global threat, causing loss of life, property damage, and environmental harm. While firefighting techniques and legislation have reduced fire incidents, firefighters still face significant risks when tackling blazes directly. Firefighting drones are emerging as a safer and more versatile alternative, capable of accessing hazardous areas, providing aerial surveillance, and assisting in fire detection. However, their effectiveness is currently limited by the need for human operators to manually aim water nozzles.

Aim

To develop fire and water stream video recognition software to accurately aim and shoot a water stream at a fire.

Fire Detection

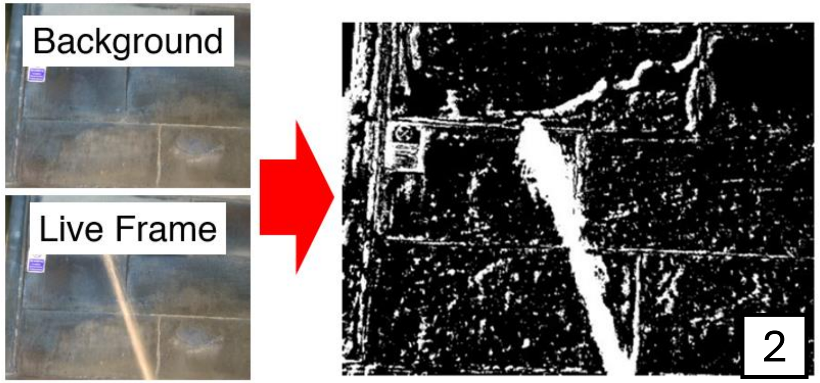
Traditional heat and smoke detectors are limited in accuracy and range, so this project developed a vision-based fire detection (VFD) algorithm using both RGB and thermal cameras. The method identifies fire by combining two key features—orange colour and heat—to locate the fire's hotspot, which is the optimal target for extinguishing. Tested on recorded UAV fire videos, the algorithm achieved an average accuracy of 75.5%. Figure 1 show the different features detected and the final detected fire.



1

Water Detection

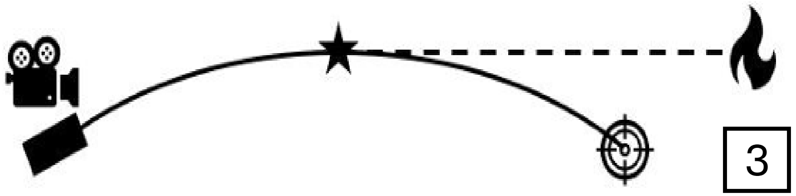
The project developed a water stream detection algorithm using an RGB camera and the OpenCV function cv2.absdiff to highlight differences between background and live frames as seen in Figure 2. Thermal and depth cameras were unsuitable due to technical limitations, so the RGB camera provided the most reliable input. Tested across eight cases, the algorithm achieved a 75% success rate, correctly identifying the water stream in six cases.



2

Aiming

The aiming system was calibrated so that the water nozzle could direct the stream onto a detected fire rather than just its visible peak in the camera frame as shown in Figure 3. Experiments mapped nozzle pitch to water landing distance and nozzle yaw to horizontal pixel position, producing equations that relate fire coordinates and depth data to the required input angles.



3

Testing

For the live fire test, the software was run on a Raspberry Pi which processed RGB, thermal, and depth camera inputs to detect fire and control the water nozzle. As Figure 4, the RGB feed highlighted the fire and other fire-coloured objects and the thermal feed successfully identifying the fire as a heat contour. Although the system was able to detect the fire and activate the water pump, there were processing delays and inconsistencies in fire location. Once the pump was triggered the nozzle did not adjust to aim the water stream, indicating that while the integrated code functioned in real time, hardware constraints prevented full operation of the aiming system.

4

Conclusion

This project successfully developed and tested fire detection, water stream detection, aiming calibration, and control code, achieving all key objectives. The system integrated these elements into a combined real-time code that could detect fire and activate the water pump, though hardware limitations prevented the nozzle from moving the stream onto the target. Despite this, the work provides a strong foundation for future development, with clear pathways for improvement including more powerful processing hardware, better camera alignment, and advanced detection methods. The outcomes demonstrate proof of concept and highlight the potential for further refinement toward a fully autonomous firefighting drone.