Towards Autonomous Early Wildfire Management: Perception, Planning, and Control of Unmanned Aerial Vehicles

Huajun Dong

A Thesis

in

The Department

of

Mechanical, Industrial & Aerospace Engineering

Presented in Partial Fulfillment of the Requirements
for the Degree of
Master of Applied Science (Mechanical Engineering) at
Concordia University
Montréal, Québec, Canada

May 2026

CONCORDIA UNIVERSITY

School of Graduate Studies

This is to certify that the thesis prepared

By:	**Huajun Dong**
Entitled:	Towards Autonomous Early Wildfire Management: Perception, Planning, and Control of Unmanned Aerial Vehicles

and submitted in partial fulfillment of the requirements for the degree of

Master of Applied Science (Mechanical Engineering)

Complies with the regulations of this University and meets the accepted standards with respect to originality and quality.

Signed by the Final Examining Committee:

	Chair
Dr. Hamid Taghavifar	
	Examiner
Dr. Hamid Taghavifar	
	Examiner
Dr. Jun Yan	
	Supervisor
Dr. Youmin Zhang	

Approved by

Dr. Lyes Kadem,
Chair of Department or Graduate Program Director

2026

Mourad Debbabi, Dean
Gina Cody School of Engineering and Computer Science

ABSTRACT

Towards Autonomous Early Wildfire Management: Perception, Planning, and Control of Unmanned Aerial Vehicles

Huajun Dong

This thesis proposes a comprehensive perception-planning-control framework for autonomous early-stage wildfire detection and suppression using Unmanned Aerial Vehicles (UAVs).

For perception, a novel dual-stream object detection model based on the YOLOv8n architecture leverages both visible and infrared imagery after image registration. By integrating a Channel Prior Convolutional Attention (CPCA) module and a Dual Modality Cross-attention Transformer Fusion (DMCTF) module, the network effectively fuses cross-modal features while actively suppressing thermal noise and visual artifacts. Its feature extraction transparency is visually validated using Grad-CAM.

For planning, to optimize flight distance, Dynamic Programming (DP) and Simulated Annealing (SA) algorithms are implemented for single-UAV, multiple-fire-spot path planning, while a Genetic Algorithm (GA) handles multi-UAV, multiple-fire-spot cooperative path planning. For control, a Linear Quadratic Tracker (LQT) ensures precise trajectory tracking.

The framework is rigorously validated through MATLAB/Simulink and DJI Assistant 2 simulations and outdoor tests. Utilizing a DJI Matrice 300 RTK UAV equipped with an H20T camera and a customized multi-drop solenoid-based mechanism, the system successfully demonstrates autonomous wildfire detection, optimal path planning and tracking, and targeted fire retardant deployment in a single mission.

Acknowledgments

Firstly, I am truly thankful to my supervisor, Dr. Youmin Zhang, for offering me the incredible opportunity to pursue my Master's studies under his guidance at Concordia University. I am deeply grateful for his visionary direction in my research and his valuable funding support. Furthermore, I greatly appreciate his generosity in providing access to state-of-the-art hardware, including the DJI M300 RTK and various other UAV platforms, as well as his selfless assistance in conducting our complex outdoor flight experiments.

Secondly, I would like to express my heartfelt gratitude to the senior members and my colleagues in the lab: Qiaomeng Qin, Jin Li, Erfan Dilfanian, Linhan Qiao, Yufei Fu, Xiaobo Wu, and Amin Taherzadeh. Their invaluable help in my research, particularly in patiently familiarizing me with the intricate software and hardware operations, has been instrumental to my progress. Thank you all for fostering such a warm, collaborative environment and making the laboratory truly feel like a home away from home. I would also like to extend a special thanks to Wanda Guo for sharing his profound insights on robotics, which greatly inspired my perspective and work.

I would like to acknowledge our industrial partners, Hamza Benzerrouk and Hakim Guiddir from RW Aerogroup, and the support from the Consortium for Research and Innovation in Aerospace in Québec (CRIAQ) through the Airtanker Visual Intelligent Tracking & Airborne Guidance System (AVITAGS) project.

Lastly, my deepest and most profound gratitude goes to my family. Their unconditional love, endless encouragement, and unwavering support throughout my entire academic journey have been my greatest source of strength. This milestone is as much theirs as it is mine.

Table of Contents

List of Figures

List of Tables

List of Abbreviations

2D	Two-Dimensional
3D	Three-Dimensional
AI	Artificial Intelligence
AP	Average Precision
API	Application Programming Interface
AVITAGS	Airtanker Visual Intelligent Tracking & Airborne Guidance System
BCE	Binary Cross-Entropy
BN	Batch Normalization
C2f	Cross Stage Partial with Bottleneck
CAD	Computer-Aided Design
cm	centimeter
CNN	Convolutional Neural Network
CO	Carbon Monoxide
CO_2	Carbon Dioxide
CPCA	Channel Prior Convolutional Attention
CPP	Coverage Path Planning
CRIAQ	Consortium de recherche et d'innovation en aérospatiale au Québec
CSP	Cross Stage Partial
CUDA	Compute Unified Device Architecture
DETR	Detection Transformer (End-to-End Object Detection with Transformers)
DFL	Distribution Focal Loss

DJI	Dajiang Innovation
DMCTF	Dual Modality Cross-attention Transformer Fusion
DP	Dynamic Programming
FLOP	Floating Point Operations
FN	False Negatives
FOV	Field of View
FP	False Positives
FPN	Feature Pyramid Networks
fps	frames per second
GA	Genetic Algorithm
GPS	Global Positioning System
GPU	Graphics Processing Unit
Grad-CAM	Gradient-weighted Class Activation Mapping
IoU	Intersection over Union
IR	Infrared
km	kilometer
LiDAR	Light Detection and Ranging
LQT	Linear Quadratic Tracker
M	million
M300	Matrice 300
mAP	mean Average Precision
MACs	Multiply-Accumulate Operations
MAS	Multi-Agent Systems
MLP	Multiayer Perceptron
ms	milliseconds
MTSP	Multiple Traveling Salesman Problem

NAVLab	Networked Autonomous Vehicles Laboratory
NMS	Non-Maximum Suppression
NN	Neural Network
ORB	Oriented FAST and Rotated BRIEF
OS	Operating System
OSDK	Onboard Software Development Kit
PANet	Path Aggregation Network
PC	Personal computer
PID	Proportional–Integral–Derivative
PGI	Programmable Gradient Information
QoS	Quality-of-Service
ResNet	Residual Network
RGB	Red-Green-Blue
RL	Reinforcement Learning
RoI	Region of Interest
ROS	Robot Operating System
RRT	Rapidly-exploring Random Trees
RTK	Real-Time Kinematic Positioning
SA	Simulated Annealing
SDK	Software Development Kit
SE	Squeeze-and-Excitation
SIFT	Scale-Invariant Feature Transform
SLAM	Simultaneous Localization and Mapping
SPPF	Spatial Pyramid Pooling - Fast
SSH	Secure SHell
TIR	Thermal Infrared

TP	True Positives
TSP	Traveling Salesman Problem
UAV	Unmanned Aerial Vehicle
YOLO	You Only Look Once

Chapter 1

1. Introduction

1.1 Motivation

Wildfires, also known as forest fires, are uncontrolled fires that occur in forested areas, often spreading rapidly and causing severe ecological and socioeconomic damage. They are typically ignited by natural factors, such as lightning or spontaneous combustion under extreme heat and drought conditions, or anthropogenic activities, including agricultural burning, negligence, or infrastructure failure [1].

The impacts of forest fires are multifaceted. Ecologically, they lead to biodiversity loss, habitat degradation, and soil erosion. In terms of public health, smoke from biomass burning contributes to air pollution and respiratory illnesses. Economically, wildfires damage property, disrupt infrastructure, and incur enormous firefighting costs [2].

The frequency and intensity of forest fires have increased significantly in recent decades, largely due to climate change, forest mismanagement, and urban encroachment into wildland areas. According to Natural Resources Canada, wildfires in Canada have burned an average of 7 million hectares annually over the past decade, with a record 17.6 million hectares in 2023 [3].

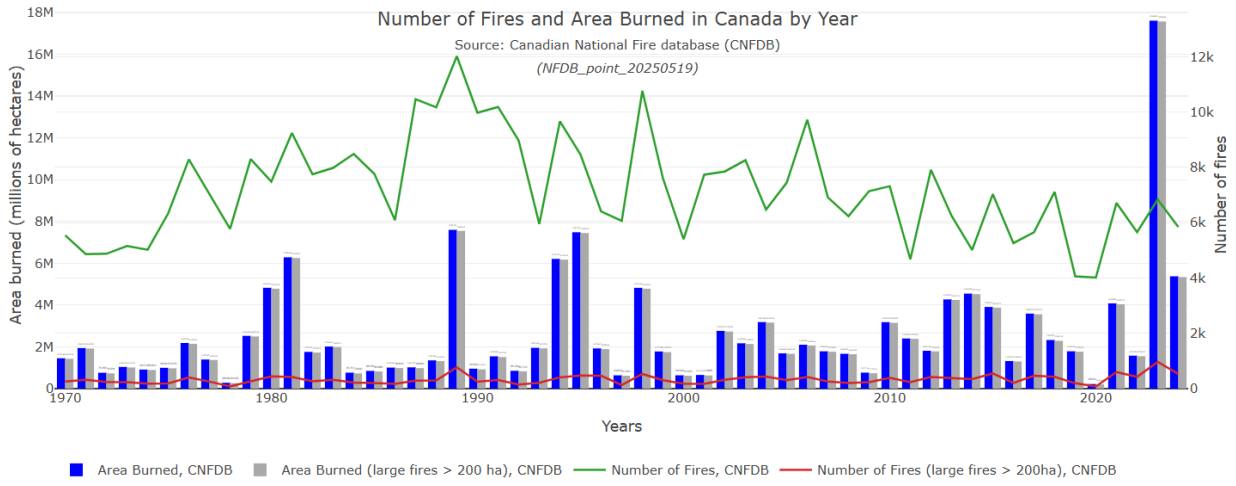

Figure 1-1 Number of fires and area burned in Canada by year [3]

The development of wildfires can generally be classified into four distinct stages: the incipient or early stage, the growth or flaming stage, the fully developed stage, and the decay or smoldering stage [4]. The early stage of a wildfire is characterized by the initial ignition and limited spread of fire. At this stage, combustion is typically localized, with relatively low flame intensity, smaller burn area, and reduced smoke emission compared to later phases. The primary indicators include subtle increases in temperature, the release of trace gases such as Carbon Monoxide (CO) and Carbon Dioxide (CO_2), and the presence of fine particulate matter before visible flames become dominant. Suppressing a wildfire in its early stage is considered the most effective strategy because the fire is still small, easier to control, and requires fewer resources for containment. Intervention at this point can prevent escalation into large-scale, destructive wildfires, significantly reducing ecological damage, health risks, and economic costs.

To effectively mitigate the risks from wildfires, particularly targeting the critical early stage, a systematic operational framework is essential. In the context of active wildfire management, the response process is generally delineated into three sequential objectives: detection, localization, and suppression (or fighting). Detection involves the timely identification of thermal anomalies or smoke plumes within vast forested regions. Once a potential threat is identified, localization follows, requiring precise coordinate mapping to guide response teams or autonomous systems. Finally, suppression entails the direct application of extinguishants to contain or eliminate the combustion before it spreads.

Wildfire detection methods are generally classified into three major categories: ground-based, aerial, and satellite-based systems [5]. Ground-based detection includes traditional observation

towers, human patrols, and sensor networks (e.g., thermal sensors, smoke detectors, or wireless sensor networks). These systems are relatively cost-effective and accurate at the local scale, but their effectiveness is limited by line-of-sight restrictions and narrow spatial coverage, making them less suitable for large-scale fire monitoring. Satellite-based detection provides extensive spatial coverage and is particularly useful for monitoring remote or inaccessible forest regions. However, the limitations of satellite-based systems include low temporal resolution, delays in data acquisition, and high operational costs, which restrict their ability to detect fires in their early stages. Aerial detection, especially using Unmanned Aerial Vehicles (UAVs), has emerged as a promising solution to bridge this gap. UAVs are capable of covering large areas quickly, providing real-time, high-resolution data, and operating under challenging conditions such as dense smoke, low visibility, or nighttime scenarios. Equipped with advanced payloads-including visible, thermal infrared, and multispectral camera sensors—UAV platforms significantly enhance the accuracy of fire localization, smoke plume monitoring, and flame intensity assessment. Therefore, UAV-based aerial detection is considered one of the most effective approaches for early-stage wildfire detection.

In addition to early wildfire detection, UAVs can also play a crucial role in wildfire suppression. Small- to medium-scale UAVs equipped with water or fire-retardant payloads can be rapidly deployed to suppress emerging fire spots before they escalate into large-scale wildfires. By integrating real-time detection with autonomous navigation and targeted suppression mechanisms, UAVs can localize their interventions to specific ignition points with high precision. Furthermore, UAV swarms offer the possibility of coordinated firefighting missions, where multiple drones operate collaboratively to cover larger areas, optimize suppression efficiency, and minimize response time. Although UAV-based firefighting cannot yet fully replace manned aerial suppression for large wildfires, it provides a valuable first-response capability that can significantly reduce fire spread during the critical early stages.

In conclusion, the use of UAVs for early wildfire detection and suppression holds significant potential. This thesis aims to design a comprehensive framework for an autonomous UAV-based system capable of detecting and fighting wildfires at their early stages.

1.2 Overview

The entire UAV-based early wildfire detecting and fighting mission can be divided into four main stages: perception, planning, control, and action.

In the perception stage, wildfires are initially detected from imagery acquired by visible and infrared cameras. At this point, the detected fire locations in 2D image coordinates must be converted into 3D spatial coordinates. Simultaneously, the UAV must localize itself to determine its own 3D position. Once all wildfire locations and UAV positional information are obtained, the system proceeds to the planning stage, where trajectories are generated to ensure that the UAV can efficiently visit all identified fire spots. In the control stage, the UAV controller executes these trajectories, guiding the platform to accurately track the planned paths. Finally, in the action stage, the suppression mechanism is activated under appropriate conditions to extinguish the fire during the entire firefighting operation.

The hardware platform employed in this work is the DJI Matrice 300 (M300) UAV equipped with an H20 multi-sensor camera. An iCrest 2.0 onboard computer is integrated to provide computational support. On the software side, the system utilizes the DJI Onboard Software Development Kit (OSDK) for calling interfaces such as video streaming, flight control, and GPS acquisition. The overall framework is implemented within the Robot Operating System (ROS), which enables communication and coordination among different system components. In addition, a ground control computer and the DJI remote controller are used to monitor the mission in real time.

(a) DJI M300 UAV (b) DJI H20T camera (c) iCrest 2.0 onboard computer

Figure 1-2 Early wildfire detection and suppression hardware system

To validate the feasibility of the proposed framework, both simulation and outdoor flight experiments are conducted. This combination of simulation and field testing provides a

comprehensive assessment, demonstrating not only the theoretical soundness of the approach but also its practical applicability for early wildfire detection and fighting.

1.3 Thesis Organization

The rest of this thesis is organized as follows:

- Chapter 2. Literature Review: This chapter provides a review of current research on UAV-based forest fire detection and suppression.
- Chapter 3. Perception – Wildfire Detection: This chapter discusses methods for achieving efficient and accurate wildfire object detection in the perception stage using visible–infrared imagery. The topics covered include single-stream detection approaches as well as dual-stream detection frameworks, with emphasis on visible–infrared image registration, dataset construction, and the design of dual-stream neural network architectures. Gradient-weighted Class Activation Mapping (Grad-CAM) is applied to offer the explainability of the neural network.
- Chapter 4. Planning: This chapter addresses the optimal path planning problem for UAV-based wildfire suppression missions. It formulates trajectory optimization strategies for both single-UAV and multi-UAV scenarios, detailing the implementation of Dynamic Programming (DP), Simulated Annealing (SA), and Genetic Algorithm (GA) planners designed to minimize the total flight distance.
- Chapter 5. Control: This chapter focuses on the design of a control system to ensure the UAV accurately tracks the planned flight trajectories. It introduces the mathematical modeling of the quadrotor's kinematics and dynamics, followed by the development and formulation of a Linear Quadratic Tracker (LQT) for precise positional control.
- Chapter 6. Experimental Design and Analysis: This chapter presents the comprehensive validation of the proposed autonomous framework. It first details the hardware and software architectures of the experimental platform. Subsequently, it provides in-depth quantitative analyses of the perception models, alongside simulation and outdoor flight test results evaluating the performance of the trajectory planners and LQT controller.

- Chapter 7. Conclusion and Future Work: This chapter summarizes the primary contributions and key findings of the entire research. Furthermore, it discusses the practical limitations encountered during the real-world outdoor experiments and outlines potential directions for future improvements and research.

Chapter 2

2. Literature Review

Wildfires have become one of the most severe ecological disasters globally, intensified by rising temperatures, urban encroachment into wildlands, and unpredictable climate patterns. Traditional fire management methods—ground-based observation, satellite imaging, and manned aircraft—are often constrained by limited temporal resolution, high risk to human operators, and slow response times. In this context, Unmanned Aerial Vehicles (UAVs), also known as drones, have revolutionized wildfire management through their high mobility, cost efficiency, and ability to operate in hazardous environments [5], [6]. UAVs are increasingly integrated into the wildfire management pipeline—spanning perception, planning, control, and active firefighting—to achieve real-time monitoring, predictive modeling, and autonomous intervention [6]. The growing body of literature emphasizes UAVs' potential to function as intelligent agents that perceive, reason, and act collaboratively. This section reviews research progress on wildfire detection and fighting across three key domains—perception and sensing, planning and coordination, control and cooperative firefighting, and finally concludes with challenges and future directions.

2.1 Perception and Sensing

Perception constitutes the cornerstone of UAV-based wildfire operations, enabling precise fire detection, environmental understanding, and situational awareness. Modern UAVs equipped with multi-spectral, Thermal Infrared (TIR), and visible-light imaging systems provide granular, near-real-time data essential for identifying fire perimeters, flame intensity, smoke dispersion, and vegetation moisture content—parameters crucial for accurate fire behavior modeling [6].

Early research on single-UAV systems laid the groundwork for many of these perception capabilities. Low-cost fixed-wing and rotary UAVs equipped with TIR and RGB sensors proved highly effective for detecting thermal anomalies and smoke signatures in small or early-stage wildfires, where rapid deployment was more critical than spatial coverage. Notably, [7] developed a solar-powered UAV integrating AI-based pattern recognition to autonomously detect ignition points and predict likely spread directions before firefighting teams reached the scene. Similarly, [6] demonstrated that a single UAV using onboard computer vision algorithms could perform real-time segmentation of fire fronts, employing adaptive thresholding and neural inference to identify fire pixels under varying illumination and smoke conditions.

More recent studies have extended UAV perception from passive observation to inferential modeling of wildfire dynamics. For example, [8] introduced an active sensing framework in which UAVs autonomously adjust their flight paths to maximize information gain regarding fire propagation. Their human-centered design exemplifies a transition toward adaptive autonomy, where UAVs collaborate with ground firefighters by forecasting fire evolution and updating situational models in real time.

Finally, the integration of edge computing and onboard AI has made real-time decision-making possible without dependence on ground control infrastructure. This independence is particularly advantageous in remote or communication-degraded regions, where centralized data processing is limited [9]. When combined with cloud-based fusion networks, these perception systems deliver continuous, high-resolution situational awareness to incident command centers, greatly improving the timeliness and precision of wildfire response operations.

2.2 Planning and Coordination

The planning layer focuses on single or multiple UAV path optimization, coverage strategies, and cooperative mission design. Single UAVs must achieve optimal information collection and situational coverage despite limited endurance and field of view. Therefore, path planning and trajectory optimization are central to their effectiveness. Single-UAV planning problems are often formulated as Coverage Path Planning (CPP) or active exploration problems. Researchers have employed algorithms such as Rapidly-exploring Random Trees (RRT) and A* to enable UAVs to dynamically re-plan trajectories based on evolving fire fronts. For instance, [10] proposed a real-time adaptive flight planning framework where a single UAV adjusts its observation trajectory using predictive fire modeling, optimizing between image resolution and flight time. In addition, AI-based planners have emerged that integrate Reinforcement Learning (RL) to train UAVs for autonomous decision-making in dynamic fire environments. These models allow the UAV to learn effective scanning behaviors under uncertainty, adjusting altitude, heading, and revisit intervals to maintain complete fire coverage without external supervision.

Regarding multiple UAVs, early studies such as [11] formulated the problem as utility-based cooperative control, enabling UAV swarms to monitor large wildfire zones efficiently while minimizing redundant coverage and communication costs.

Subsequent developments have introduced multi-agent optimization frameworks that account for fire propagation models, UAV endurance, and line-of-sight communication. For instance, [12] proposed a multi-UAV coverage and tracking algorithm with Quality-of-Service (QoS) guarantees, ensuring both consistent observation and timely updates for human teams. Their predictive planning model integrates wildfire physics and environmental uncertainty, allowing UAVs to anticipate future fire states rather than react to past data.

Meanwhile, [10] introduced a real-time coordination strategy for fleets of fixed-wing UAVs, demonstrating efficient trajectory planning that balances observation precision and flight endurance. Their system employs a Model Predictive Control (MPC) approach for dynamic re-tasking based on live fire data.

Newer studies envision UAVs as part of decentralized Multi-Agent Systems (MAS), where each UAV acts as a semi-autonomous node sharing data through consensus algorithms. [13]

describes a conceptual MAS for wildfire management, highlighting decentralized control and self-organizing behaviors as key enablers for resilience and scalability in large-scale operations.

2.3 Control and Cooperative Firefighting

Beyond observation, UAVs are increasingly utilized for direct firefighting interventions. This includes retardant deployment, ignition control for backburn operations, and communication relay for ground teams. Early implementations focused on simple trajectory control, but newer works integrate cooperative control algorithms and multi-agent coordination for dynamic firefighting.

Single-UAV control focuses on stability, autonomy, and adaptive response in extreme thermal and wind conditions. UAV control systems for wildfire management must handle strong convection currents, low visibility, and GPS degradation due to smoke or terrain occlusion.

Traditional PID and LQR controllers are now often augmented with fuzzy logic and adaptive neural control to maintain flight stability. For example, [14] explored autonomous flight stabilization that adapts to temperature-induced turbulence by integrating onboard IMU data and atmospheric sensors. More advanced systems also employ vision-based control, where the UAV's flight behavior directly reacts to visual feedback from the fire front, effectively linking perception and control layers in real time.

Furthermore, fail-safe mechanisms and autonomous return-to-base protocols are vital for single-UAV systems operating beyond line-of-sight. These ensure operational continuity and data preservation in case of environmental interference or energy depletion.

Regarding multiple UAVs, [11] demonstrated one of the first control-theoretic models for multi-UAV cooperation in wildfire suppression, assigning UAVs to sensing, monitoring, and suppression roles dynamically. Their study laid the groundwork for later AI-driven control architectures that allow real-time role reassignment and load balancing. [15] provides a comprehensive review of swarm coordination, distributed control, and real-time adaptation applicable to wildfire UAV teams.

Control systems are also evolving toward human-centered hybrid architectures, where UAVs operate autonomously but within a supervised autonomy loop managed by human incident commanders. This approach maintains accountability while leveraging UAV speed and precision, aligning with ethical and safety requirements for autonomous systems.

2.4 Challenges and Future Directions

While UAV-based wildfire detection and fighting technologies have proven transformative, several challenges remain:

1. **Energy and Endurance** — UAV missions are constrained by battery life and payload weight, limiting flight duration in large-scale fires.
2. **Communication in Adverse Conditions** — Smoke and terrain can disrupt GPS and radio links. Decentralized communication and ad hoc mesh networks are emerging as resilient alternatives.
3. **Safety and Regulation** — Integration into manned airspace requires stringent compliance with aviation laws and robust fail-safe mechanisms.
4. **AI Reliability and Ethics** — Decision-making transparency and accountability remain concerns, particularly for autonomous firefighting interventions.

Future trends point toward swarm intelligence, bio-inspired coordination, and cloud-edge integration that combines UAV data streams with predictive fire models for proactive decision-making [13].

Chapter 3

3. Perception – Wildfire Detection

3.1 Overview

This section explains the requirements of the perception stage in UAV-based wildfire detection and suppression missions, compares various perception methods, and concludes by highlighting the advantages of the selected approach.

In the UAV-based wildfire detection and suppression mission, DJI Matrice 300 (M300) UAV is equipped with an H20T camera that provides real-time video streams. These data are processed onboard using the iCrest 2.0 computer, an edge-computing device integrated with an NVIDIA Jetson Xavier NX GPU. Due to the limited computational resources of this platform, the deployed models must be lightweight with low computational complexity to ensure real-time performance.

To detect wildfires within an image, computer vision techniques are required. Traditional computer vision methods refer to approaches that rely on manually engineered features and rule-based algorithms to interpret and analyze visual data. These methods typically involve techniques like edge detection, color histograms, and handcrafted feature descriptors such as SIFT [16], HOG [17], and SURF [18], followed by classical machine learning models like SVMs or decision trees for classification and detection tasks. These approaches have some advantages, such as low

computational requirements, interpretability, and robustness in low-data environments. However, they suffer from limited generalization ability, sensitivity to variations in scale, lighting, and occlusion, and the need for extensive domain expertise in designing effective features.

In recent years, traditional methods have been largely replaced by deep learning approaches. “Deep learning methods have surpassed traditional techniques in almost every computer vision benchmark, making handcrafted features largely obsolete in modern research and applications” [19]. The shift is due to the superior accuracy, scalability, and adaptability of deep learning models in complex, real-world visual tasks. Deep learning-based computer vision can be broadly categorized into several core tasks, including image classification, object detection, and semantic segmentation, among others.

For our task, image classification can only provide binary results (e.g., fire or no fire) without spatial context, making it insufficient for downstream operations like planning. In contrast, semantic segmentation offers pixel-level precision, allowing detailed localization of fire regions. However, this level of granularity comes at a high computational cost. For instance, state-of-the-art segmentation models like DeepLabV3+ require 4 to 5 times more MACs (Multiply-Accumulate Operations) and inference times up to 3 times longer than comparable object detection models when deployed on edge devices [20]. These requirements make semantic segmentation impractical for real-time inference on low-power edge hardware such as NVIDIA Jetson Xavier. Conversely, object detection models (e.g., YOLO or SSD variants) strike a balance between spatial localization and computational efficiency. Lightweight object detectors can operate at 30+ FPS with <1 GFLOP on ARM-based edge devices, making them highly suitable for our mission-critical wildfire detection scenarios [21]. Therefore, this study focuses on the investigation of object detection models for wildfire detection.

3.2 Visible-infrared Wildfire Image Dataset

To build a dataset for training and evaluating the proposed wildfire object detection model, visible–infrared image pairs are first collected using the DJI M300 drone equipped with the DJI H20T camera. The collected data include high-quality imagery of simulated wildfire scenarios, where fire was generated in a metal pot filled with burning wood to replicate realistic combustion conditions. To ensure the dataset captures diverse challenges encountered in real-world

environments, images were acquired under varying conditions, such as occlusion by vegetation, nighttime with limited illumination, and the presence of additional heat sources that may act as potential disturbances. Representative examples of these captured images are shown in Figure 3-1.

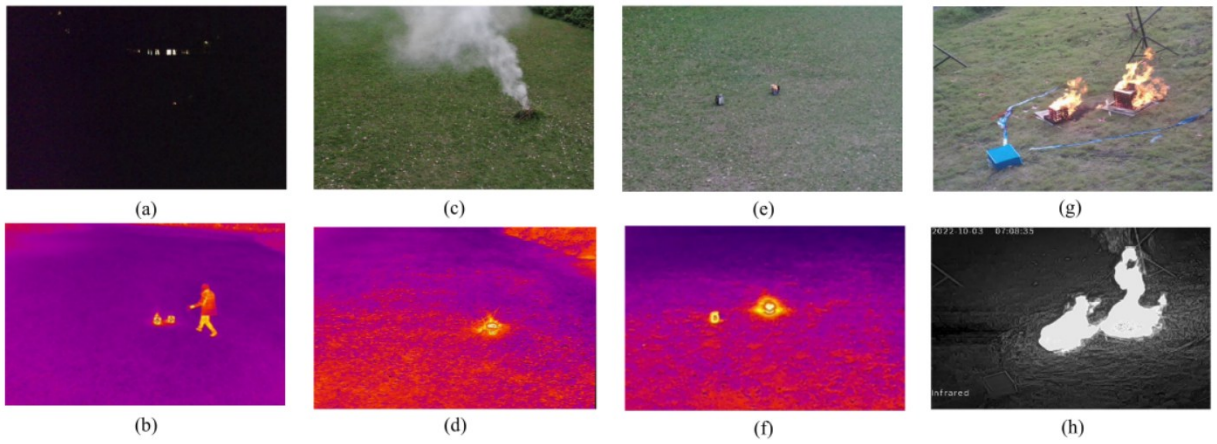

Figure 3-1Visible–infrared paired image examples from our dataset and the dataset in [18]. (a) & (b): At nighttime, visible images are hard to acquire features, while infrared images clearly depict the fire and other heat sources. (c) & (d): Due to occlusion by vegetation, fires are invisible in visible images, whereas in infrared images, the fire stands out due to its heat. (e) & (f): In infrared images, the features are less salient, making it hard to distinguish fire from other heat sources (such as kettles here). (g) & (h): Due to the thermal radiation effect, the fire area in infrared images appears larger than in visible images. This discrepancy can reduce the accuracy of retardant deployment for wildfire fighting.

In addition to this self-collected dataset, two open-source datasets of visible–infrared wildfire image pairs [22] are included to further broaden the range of scenarios and improve the generalization capability of the model. In total, the combined dataset contains 2596 images. A detailed summary of the dataset is provided in Table 3-1.

Table 3-1 Visible–infrared wildfire image dataset analysis

	Number of visible–infrared image pairs
Self-collected dataset	668
Dataset from [22]	1291
Dataset from [23]	637

All images in the dataset were manually labeled using the open-source tool LabelImg and saved in the YOLO format. Finally, the dataset was randomly partitioned into training and testing sets at an 8:2 ratio.

3.3 Baseline – Single-stream Detection Model (YOLOv8n)

3.3.1 Development of Object Detection Models

Object detection models can generally be divided into two main categories: two-stage and one-stage detectors. Two-stage detectors, represented by the R-CNN family (e.g., R-CNN [24], Fast R-CNN [25], Faster R-CNN [26]), first generate region proposals and then perform classification and bounding-box regression on these candidate regions. This design typically achieves higher accuracy, especially for complex scenes and small objects, but at the expense of computational efficiency. In contrast, one-stage detectors such as YOLO [27] and SSD [28] eliminate the proposal generation step and directly predict class probabilities and bounding boxes from dense sampling of feature maps. This leads to real-time inference capability, making one-stage models more suitable for resource-constrained or real-time applications, though sometimes with a trade-off in accuracy compared to two-stage methods. More recently, the boundaries between these two categories have been blurred by transformer-based detectors such as DETR [29], which adopt end-to-end learning and attention mechanisms, offering a new paradigm that combines accuracy with simpler architectures.

3.3.2 Development of YOLO

The YOLO (You Only Look Once) family of object detection models has experienced continuous evolution since its introduction in 2015, progressively improving accuracy, speed, and adaptability across diverse applications. The first version, YOLOv1, pioneered the concept of unifying classification and localization into a single convolutional network, enabling real-time detection. YOLOv2 introduced batch normalization and multi-scale training, boosting robustness and accuracy. YOLOv3 employed the deeper Darknet-53 backbone and incorporated multi-scale detection strategies. Later, YOLOv4 enhanced feature representation with SPP, CSP modules, and Mosaic data augmentation. The release of YOLOv5 emphasized lightweight architecture, data augmentation, and efficient training, making it widely adopted in practical deployments despite being unofficial. Subsequent versions, such as YOLOv6 and YOLOv7, further advanced industrial optimization, parameter reallocation, and ELAN network structures. More recently, YOLOv8 adopted anchor-free detection heads and novel loss functions, while YOLOv9 integrated

Programmable Gradient Information (PGI) and GELAN networks. YOLOv10 proposed consistent dual-branch strategies to overcome redundancy in prediction, reflecting the trend toward balancing accuracy, efficiency, and edge deployment readiness. The latest YOLOv11 marks another leap, introducing lightweight yet powerful backbone designs, enhanced multi-scale detection layers, and specialized variants such as YOLOv11-Seg and YOLOv11n, targeting segmentation tasks and edge-device deployment. This evolutionary trajectory illustrates how YOLO has shifted from a groundbreaking real-time detector to a versatile, high-performance family of models adaptable to both academic research and real-world edge computing scenarios.

3.3.3 YOLOv8n

Although the YOLO series has been updated at a rapid pace in recent years, YOLOv8 remains one of the most widely used models in real-world applications. This is primarily due to its stable support from Ultralytics, comprehensive documentation, and broad compatibility with industrial deployment frameworks such as ONNX, TensorRT, and OpenVINO. Consequently, YOLOv8 continues to dominate industrial applications, striking an effective balance between accuracy, inference speed, and deployment flexibility. Therefore, YOLOv8 is employed as the baseline model in this work.

Figure 3-2 demonstrates the architecture of YOLOv8. Its architecture consists of three main parts: Backbone, Neck, and Head. Backbone serves as the primary feature extractor, processing the input image (commonly resized to 640×640 pixels) through a series of layers that progressively reduce spatial resolution while increasing feature channels. It incorporates convolutional layers, Cross Stage Partial with Bottleneck (C2f) blocks, and a Spatial Pyramid Pooling - Fast (SPPF) block. C2f block improves upon the original CSPNet by splitting the input feature maps into multiple paths, enabling a portion of the features to undergo deeper convolutional processing while another portion bypasses the transformation and is later concatenated. This design reduces computational redundancy, improves gradient flow, and enables a lighter yet more effective network. SPPF block, positioned near the end of the Backbone, captures multi-scale contextual information by applying pooling operations with different receptive fields in parallel. This enables the model to recognize objects of varying sizes and aspect ratios more reliably, even under challenging visual conditions.

Neck plays a crucial role in aggregating and refining features across different scales. It is based on the PANet (Path Aggregation Network) structure, which extends the functionality of traditional Feature Pyramid Networks (FPN). While FPNs propagate semantic information from deeper layers upward to facilitate multi-scale detection, PANet introduces a bidirectional flow, enhancing the bottom-up path to strengthen localization signals from shallower layers. In YOLOv8, this is achieved by a combination of upsampling operations, concatenations, and further C2f modules. The process begins by upsampling deeper, semantically rich but spatially coarse feature maps and merging them with higher-resolution feature maps from earlier stages of the Backbone. This design ensures that small objects benefit from strong semantic context, while large objects retain sufficient spatial detail. The result is a set of feature representations that are both semantically enriched and spatially precise, significantly improving detection robustness across diverse object scales.

Finally, Head employs an anchor-free detection mechanism, which directly regresses bounding box coordinates, object scores, and class probabilities without reliance on predefined anchor boxes. Traditional anchor-based detectors, such as YOLOv5, relied on predefined anchor boxes of fixed aspect ratios and scales, which increased both the complexity of training and the sensitivity of performance to anchor design choices. In contrast, this anchor-free approach simplifies the model design, accelerates training convergence, and improves generalization across datasets with varying object size distributions. The detection head operates at three scales—P3, P4, and P5—corresponding to feature maps of different resolutions, which are specialized for detecting small, medium, and large objects, respectively. This design simplifies training, accelerates inference, and reduces the sensitivity to anchor selection compared to earlier YOLO versions. The learning objective of YOLOv8 integrates multiple loss functions to optimize both classification and localization. Bounding box regression is guided by advanced IoU-based losses (such as CIoU or DIoU), which improve the alignment between predicted and ground-truth boxes by considering overlap, distance, and aspect ratio. Classification employs a Binary Cross-Entropy (BCE) formulation, while Distribution Focal Loss (DFL) is used to enhance the precision of bounding box boundary predictions by modeling the localization problem as a distributional task. Together, these losses ensure accurate box positioning, robust classification, and efficient optimization during training.

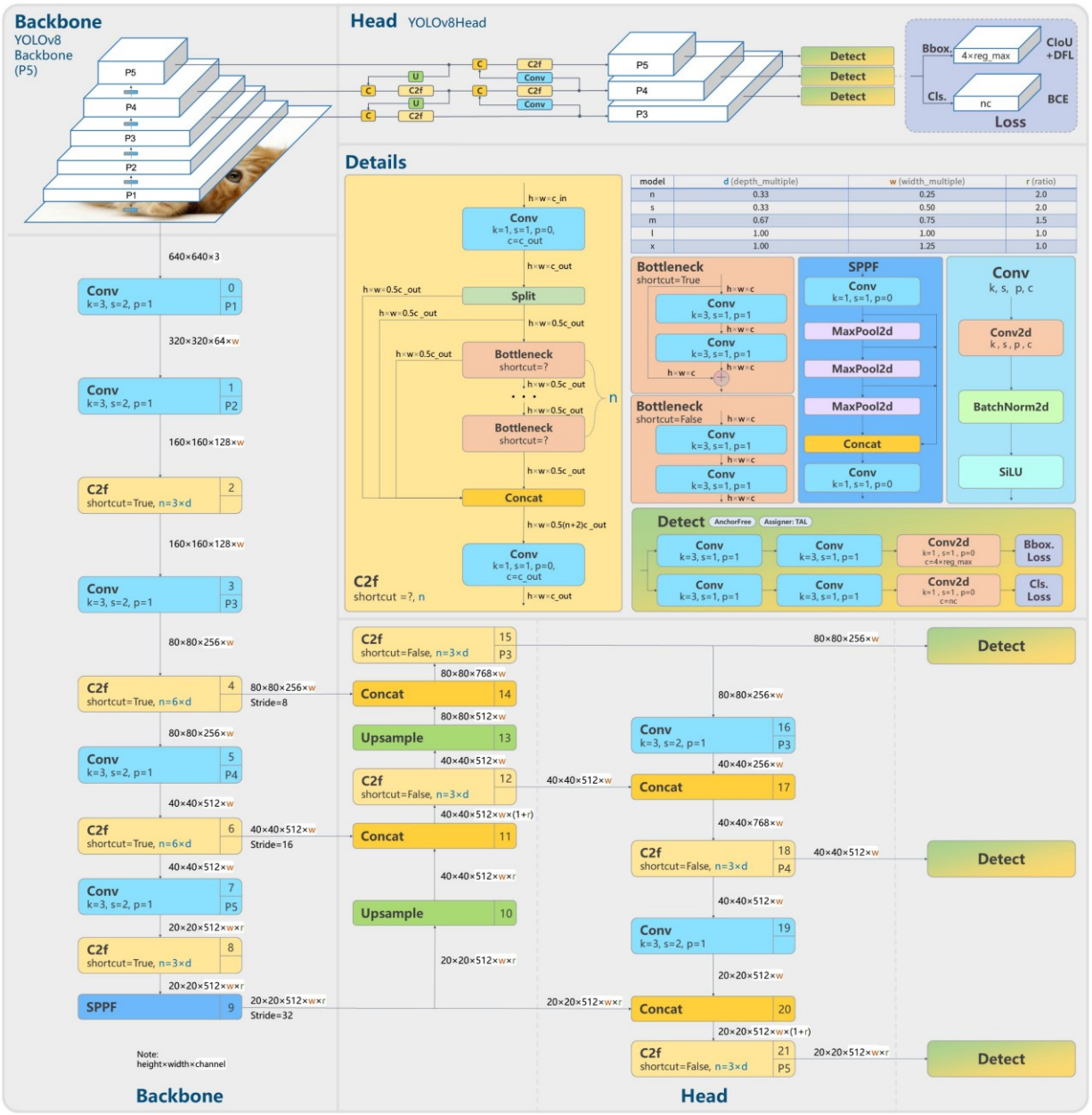

Figure 3-2 YOLOv8 architecture [30]

YOLOv8 is released in several model variants, ranging from lightweight to large-scale architectures (e.g., YOLOv8n, YOLOv8s, YOLOv8m, YOLOv8l, and YOLOv8x). These variants differ primarily in the number of parameters, computational cost, and detection accuracy, thereby offering a trade-off between inference speed and performance. For real-time applications on edge-computing devices such as the iCrest 2.0 platform, due to the limited computational capacity of this device, deploying heavier YOLOv8 variants would result in reduced inference speed and

hinder real-time detection. Therefore, among the available choices, YOLOv8n—the smallest and lightest variant—was selected as the baseline model.

3.4 Dual-stream Detection Model

3.4.1 Overview

While the single-stream detection model relies on only one modality of information (visible or infrared imagery), the DJI H20T camera mounted on the UAV is capable of capturing both visible (RGB) and Infrared (IR) imagery simultaneously. Leveraging only a single modality under such conditions would disregard the complementary information provided by the other. Therefore, to fully exploit the sensing capabilities of the H20T and enhance robustness under challenging environments such as smoke, low illumination, or nighttime operations, dual-stream detection frameworks are investigated and employed, in which RGB and IR features are fused for more reliable wildfire object detection.

To effectively integrate complementary information from both visible and infrared modalities, various fusion strategies have been explored. Fusion strategies are generally categorized into three levels: early fusion, mid-level fusion, and decision-level fusion [31]. In early fusion, the data from different modalities are combined at the input level, typically by concatenating the raw images or early feature maps, or fusing or overlaying different modal pictures at pixel levels [9]. And then, they are fed into one single backbone to allow the network to learn joint representations from the beginning. This approach is computationally efficient but can be sensitive to modality-specific noise and misalignment. Mid-level fusion (also known as feature-level fusion) involves extracting modality-specific features independently, usually by two backbones, and fusing them at an intermediate layer of the network. This strategy balances the preservation of modality-specific characteristics with joint learning and has been widely used in dual-stream architectures for object detection and semantic segmentation. Decision-level fusion combines the outputs (e.g., classification scores or bounding boxes) from modality-specific networks. This method allows each modality to be processed independently and can enhance robustness by leveraging complementary strengths, but it often lacks fine-grained interaction between features [32].

3.4.2 Visible–infrared Image Registration

To enable dual-stream detection, the visible and infrared images must first undergo a registration process to ensure spatial alignment. Through this procedure, each pixel in the two modalities corresponds to the same physical location in the scene, thereby allowing feature-level fusion to be performed in a geometrically consistent manner.

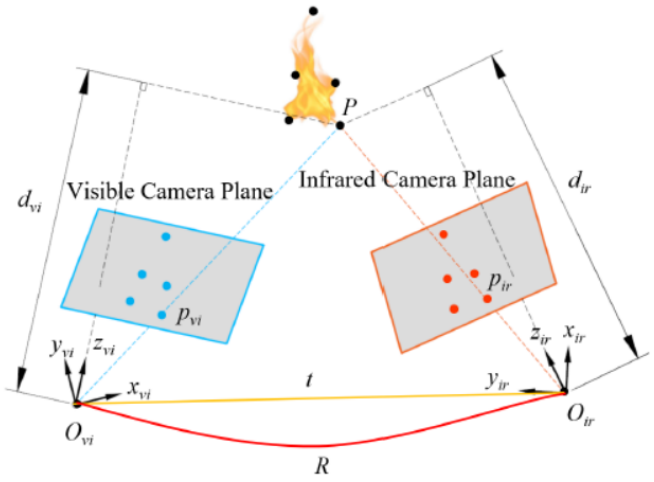

Figure 3-3 Geometric relationship of imaging

Figure 3-3 illustrates the fundamental geometric relationship among wildfire spots, the visible and infrared cameras, and their respective image planes. Suppose a wildfire corresponds to a 3D feature point $\boldsymbol{P}$. Its projection onto the two image coordinate systems can be denoted as $\boldsymbol{p}_{vi}$ and $\boldsymbol{p}_{ir}$, while $\boldsymbol{O}_{vi}$ and $\boldsymbol{O}_{ir}$ represent the optical centers of the visible and infrared cameras, respectively. The depths from the cameras to the feature point are denoted as d_{vi} and d_{ir}.

Let $\boldsymbol{O}_{vi} - \boldsymbol{x}_{vi}\boldsymbol{y}_{vi}\boldsymbol{z}_{vi}$ define the coordinate system of the visible camera and $\boldsymbol{O}_{ir} - \boldsymbol{x}_{ir}\boldsymbol{y}_{ir}\boldsymbol{z}_{ir}$ that of the infrared camera. The transformation between the two systems can be described by a rotation matrix $\boldsymbol{R}$ and a translation vector $\boldsymbol{t}$. For a 3D point $\boldsymbol{P}$, its projection onto a 2D image plane point $\boldsymbol{p}$ follows:

$$\boldsymbol{p} = \frac{1}{d}\boldsymbol{K}\boldsymbol{P}, \boldsymbol{p} = \begin{bmatrix} u \\ v \\ 1 \end{bmatrix}, \boldsymbol{P} = \begin{bmatrix} x \\ \mathrm{y} \\ d \end{bmatrix} \tag{3-1}$$

where $\boldsymbol{K}$ denotes the intrinsic matrix of the camera.

The transformation relationship between the two image coordinate systems of the two cameras is then:

$$\boldsymbol{P}_{vi} = \boldsymbol{R}\boldsymbol{P}_{ir} + \boldsymbol{t} \tag{3-2}$$

Substituting (3-1) into (3-2), one obtains

$$\boldsymbol{p}_{vi} = \frac{d_{ir}}{d_{vi}} \boldsymbol{K}_{vi}\boldsymbol{R}\boldsymbol{K}_{ir}^{-1}\, \boldsymbol{p}_{ir} + \frac{1}{d_{vi}} \boldsymbol{K}_{vi}\boldsymbol{t} \tag{3-3}$$

In practical wildfire detection scenarios, the difference between d_{vi} and d_{ir} is determined by the DJI H20T camera configuration and is typically on the order of centimeters. Since the absolute values of d_{vi} and d_{ir} are usually in the range of tens to hundreds of meters, this difference is negligible. Consequently, d_{vi} and d_{ir} can be considered approximately equal and uniformly represented as d. Therefore, (3-3) can be simplified as:

$$\boldsymbol{p}_{vi} = \boldsymbol{K}_{vi}\boldsymbol{R}\boldsymbol{K}_{ir}^{-1}\, \boldsymbol{p}_{ir} + \frac{1}{d} \boldsymbol{K}_{vi}\boldsymbol{t} \tag{3-4}$$

where the rotation matrix $\boldsymbol{R} \in \mathbb{R}^{3\times3}$, the two cameras' intrinsic matrix $\boldsymbol{K}_{vi}$, $\boldsymbol{K}_{ir} \in$, and the translation vector $\boldsymbol{t} \in \mathbb{R}^{3\times1}$ are unknown parameters from the above image pixel coordinate transformation equation.

To estimate these unknown parameters, (3-4) can be simplified as:

$$\boldsymbol{p}_{vi} = \boldsymbol{R}'\boldsymbol{p}_{ir} + \frac{1}{d}\boldsymbol{t}' \tag{3-5}$$

where $\boldsymbol{R}'=\boldsymbol{K}_{vi}\boldsymbol{R}\boldsymbol{K}_{ir}^{-1}$ and $\boldsymbol{t}' = \boldsymbol{K}_{vi}\boldsymbol{t}$.

Representing (3-5) in components form of matrix-vector format becomes:

$$\begin{bmatrix} u_{vi} \\ v_{vi} \\ 1 \end{bmatrix} = \begin{pmatrix} R_{11} & R_{12} & R_{13} \\ R_{21} & R_{22} & R_{23} \\ R_{31} & R_{32} & R_{33} \end{pmatrix} \begin{bmatrix} u_{ir} \\ v_{ir} \\ 1 \end{bmatrix} + \frac{1}{d} \begin{bmatrix} t_1 \\ t_2 \\ t_3 \end{bmatrix} \tag{3-6}$$

The least squares method is a widely adopted technique for parameter estimation [33]. In the case of equation (3-6), there are 12 unknown parameters in total, comprising 9 elements from the rotation matrix $\boldsymbol{R}'$ and 3 elements from the translation vector $\boldsymbol{t}'$. Each pair of corresponding points $\boldsymbol{p}_{vi}$ and $\boldsymbol{p}_{ir}$, provides two independent constraints. Consequently, a minimum of six such

pairs is required to estimate the 12 unknowns. In practice, however, incorporating additional pairs can significantly enhance the robustness and accuracy of the estimation.

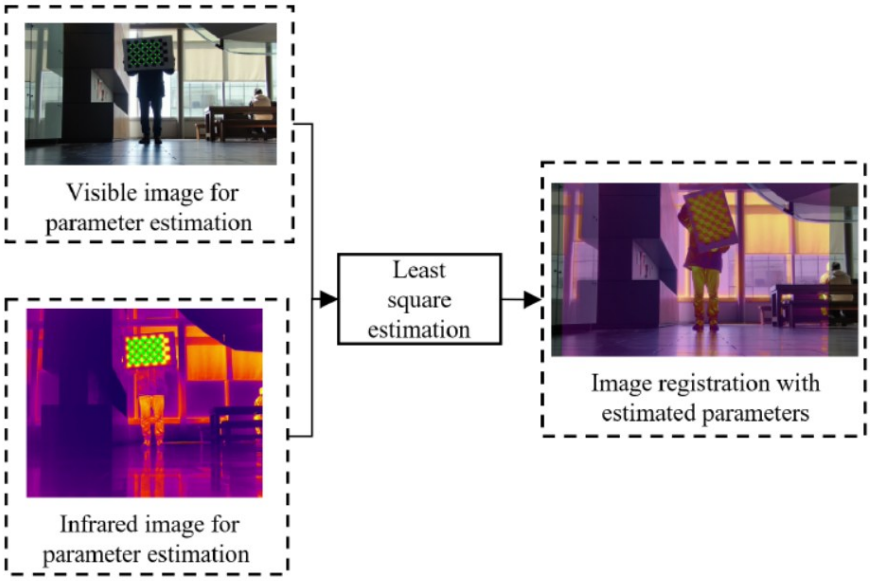

Figure 3-4 Visible–infrared image registration by estimation

As illustrated in Figure 3-4, a total of 48 pairs of corresponding pixels between the visible and infrared images were selected. Their coordinates in the respective images were extracted and used to form 48 data sets, which were then employed to estimate the 12 parameters in (3-6) through the least square method. The estimation was performed using MATLAB's "lsqnonlin" function. After obtaining the estimated transformation parameters, the visible and infrared images can be aligned according to (3-6). The registration result is presented in Figure 3-4, where the checkerboards in the visible and infrared images are shown to be precisely aligned.

All relevant source code is publicly available at: https://github.com/HuajunDong/Visible-infrared_Image_Registration.

With the estimated image registration parameters of the DJI H20T visible and infrared cameras, the wildfire dataset can be accurately aligned. Figure 3-5 presents sample registration results for visible–infrared image pairs captured at distances of 17.1 m, 22.7 m, 31.6 m, 41.2 m, and 51.2 m from the fire spots. To better illustrate the effectiveness of the registration, the original and aligned infrared images are blended with their corresponding visible images, highlighting the spatial consistency achieved through the registration process.

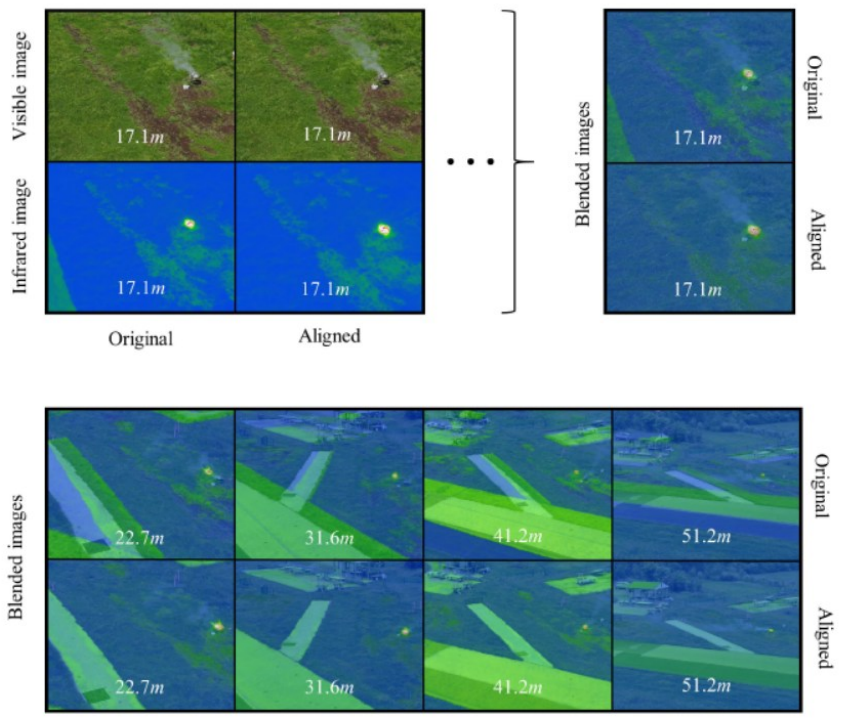

Figure 3-5 Registered visible–infrared dataset images

3.4.3 Dual-stream Detection Model Architecture

A. Early Fusion – Early Fusion Block

To achieve synergistic integration of dual-modal data, an Early Fusion Block is strategically designed to fuse registered visible and infrared images at the initial stage, as illustrated in Figure 3-6. The block takes registered visible and infrared images with a resolution of 640×640×3 as input. Each modality is first processed by a mask sub-block, in which a 1×1 convolution layer compresses all channels (that is, three channels from visible or infrared images) into a single-channel mask. This mask is then multiplied by the corresponding original feature map, and a skip connection is added to retain the original information. The refined feature map is subsequently passed through a 3×3 convolutional layer, where the number of output channels can be flexibly adjusted. Finally, the features from the two modalities are concatenated and fed into a Squeeze-and-Excitation (SE) block [34] for further refinement. After that, the refined feature maps will be sent into YOLO with the single backbone neural networks.

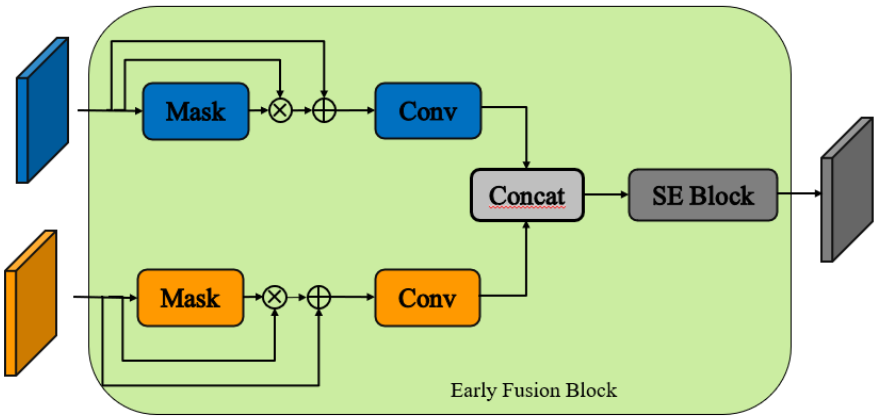

Figure 3-6 Early Fusion Block

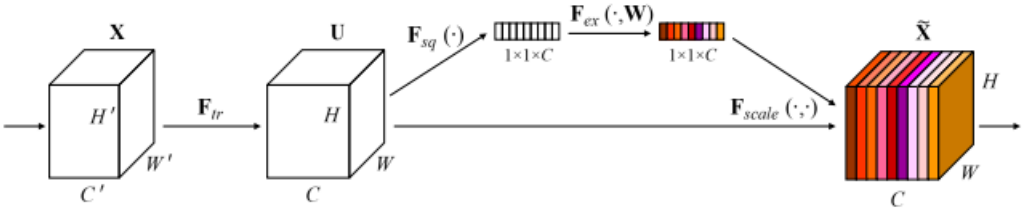

Figure 3-7 A Squeeze-and-Excitation block [34]

B. Mid-level Fusion – Direct Concatenation

The most straightforward approach to mid-level fusion is to employ two independent backbones to extract features from visible and infrared images, and then directly concatenate the resulting feature maps. Following this strategy, my model introduces an additional backbone for the infrared modality. The feature maps extracted from both modalities are concatenated and subsequently fed into Neck and Head modules to produce the final detection results. The overall architecture is illustrated in Figure 3-8.

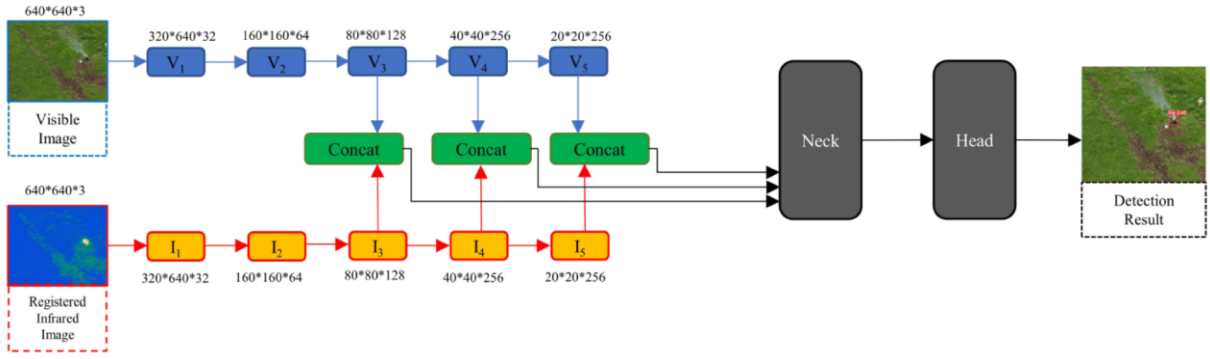

Figure 3-8 Direct concatenation dual-modality wildfire detection model

C. Mid-level Fusion - Attention-based Fusion

To further exploit both intra-modality and cross-modality feature fusion, a novel dual-modality wildfire detection model that integrates a Channel Prior Convolutional Attention (CPCA) [35] module with a Dual Modality Cross-attention Transformer Fusion (DMCTF) module is proposed, as shown in

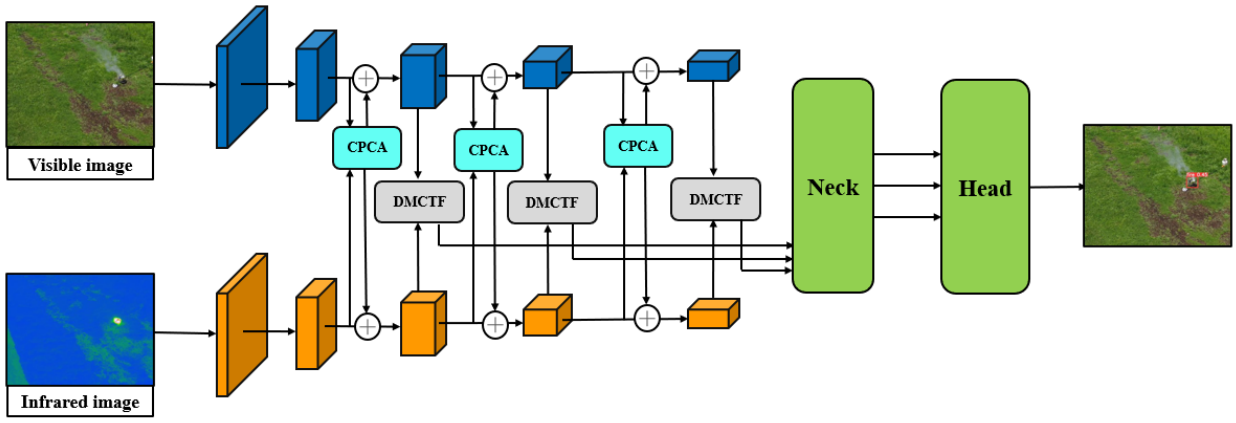

Figure 3-9 Attention-based fusion dual-modality wildfire detection model

1) Channel Prior Convolutional Attention (CPCA) Module

The CPCA module is used to improve the intra-modality feature extraction by combining channel attention and spatial attention in a sequential manner. Specifically, it first applies a channel attention mechanism, which aggregates spatial information using both average pooling and max pooling. These pooled features are then passed through a shared Multilayer Perceptron (MLP) to generate a channel attention map, enabling the network to emphasize informative channels while suppressing less relevant ones.

Following this, the refined feature maps are processed by a spatial attention mechanism built upon multi-scale depth-wise convolutions. Unlike conventional approaches that enforce identical spatial attention across channels, CPCA generates dynamically distributed spatial attention maps for each channel. This design preserves channel-specific information while effectively capturing spatial dependencies. Finally, a channel mixing step is applied through a 1×1 convolution to align the output dimensions with the input channels of the skip connections, thereby ensuring seamless feature fusion across network layers.

Through this sequential arrangement of channel and spatial attention, CPCA achieves more adaptive and discriminative feature extraction from the single-modality backbone.

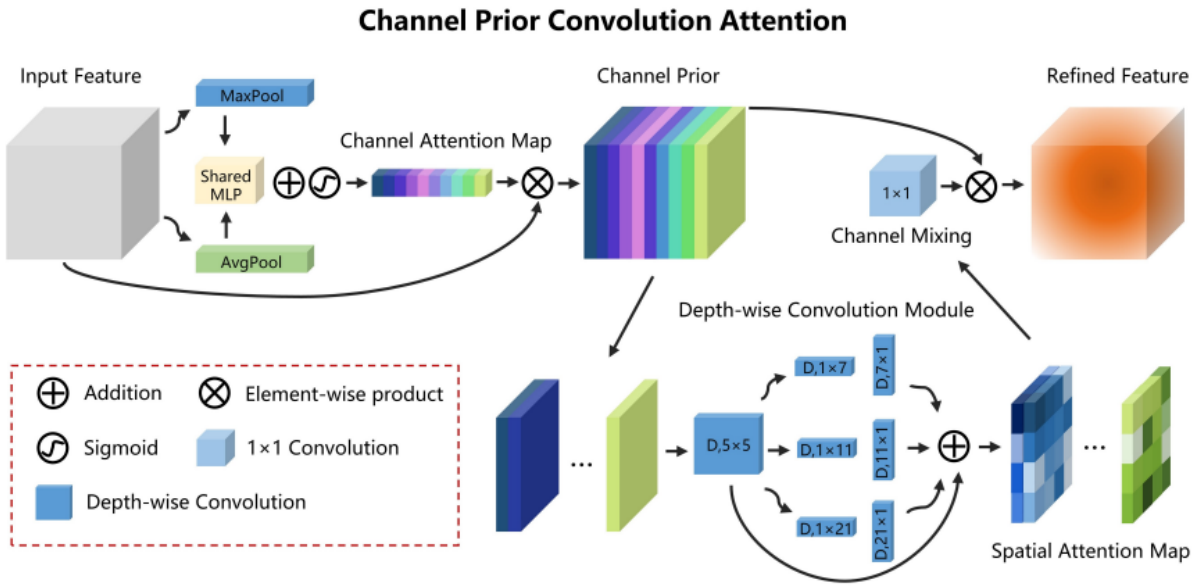

Figure 3-10 Channel Prior Convolutional Attention (CPCA) Module [35]

2) Dual Modality Cross-attention Transformer Fusion (DMCTF) Module

The DMCTF module is designed to improve both intra-modality and cross-modality feature extraction and fusion.

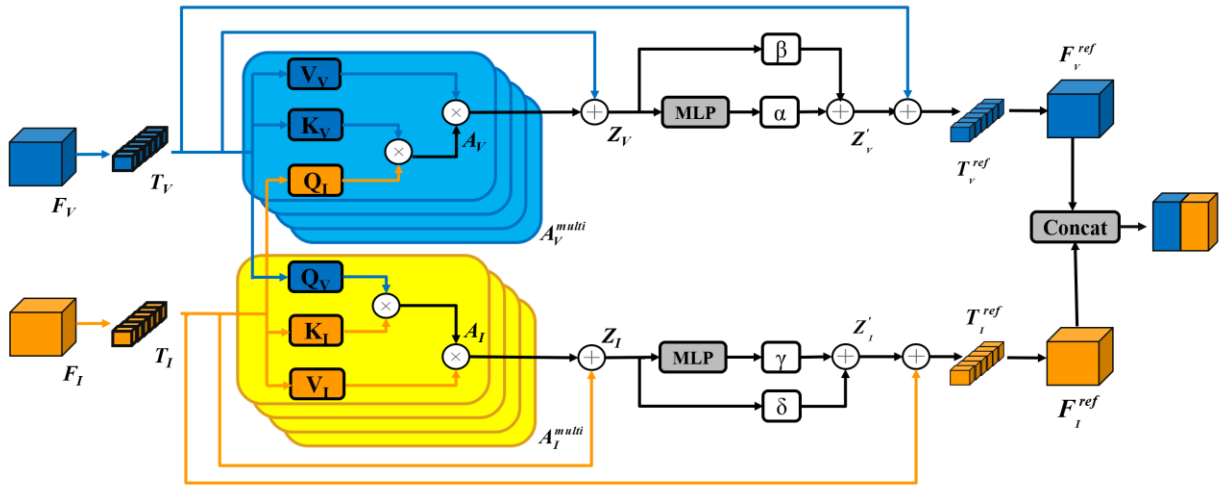

Figure 3-11 Dual Modality Cross-attention Transformer Fusion (DMCTF) Module

Given the input feature maps $\boldsymbol{F_V}$ and $\boldsymbol{F_I} \in \mathbb{R}^{H\times W\times C}$ from P3, P4, P5 layers of the visible and infrared backbones, the feature maps are firstly flattened into two sets of tokens (patches). A learnable positional embedding is then added, resulting in two sequences $\boldsymbol{T_V}$ and $\boldsymbol{T_I} \in \mathbb{R}^{HW\times C}$. A

layer norm layer is used to normalize. T_V and T_I are then projected into separate key, query, and value matrices K_v Q_v V_v and K_I K_Q K_V respectively, as follows:

$$V_I = T_I W_I^V,\ K_I = T_I W_I^K,\ Q_I = T_I W_I^Q \tag{3-7}$$

$$V_V = T_V W_V^V,\ K_V = T_V W_V^K,\ Q_V = T_V W_V^Q \tag{3-8}$$

where W_I^V, W_I^K, W_I^Q,W_V^V, W_V^K, and $W_V^Q \in \mathbb{R}^{C\times C}$ denote weight matrices with learnable parameters.

Secondly, a scaled dot-product operation is employed to compute the correlation between the query matrices from one modality and the key matrices from the other modality. The resulting similarity scores are passed through a softmax layer to form a probability distribution within $(0,1)$. These attention weights are then used to reweight the value matrices from the original modality, yielding the cross-attention outputs A_V and A_I.

$$A_V = \text{softmax}\left(\frac{Q_I K_V^T}{\sqrt{D_K}}\right)\cdot V_I \tag{3-9}$$

$$A_I = \text{softmax}\left(\frac{Q_V K_I^T}{\sqrt{D_K}}\right)\cdot V_I \tag{3-10}$$

To further enhance the representational capacity, a multi-head attention mechanism is employed, enabling multiple cross-attention maps to be exploited in parallel. In this work, the number of attention heads is set to 8. The outputs from different heads are concatenated and subsequently projected back to the original feature dimension through the projection matrices W_V^O and W_I^O, producing the aggregated representations as A_V^{multi} and A_I^{multi}. Finally, a residual (skip) connection is introduced to stabilize training and preserve the original feature information, yielding the final outputs Z_V and Z_I:

$$A_V^{multi} = \text{concat}(A_{V,1}, A_{V,2}, ..., A_{V,n}) W_V^O \tag{3-11}$$

$$A_I^{multi} = \text{concat}(A_{I,1}, A_{I,2}, ..., A_{I,n}) W_I^O \tag{3-12}$$

$$Z_V = A_V^{multi} + T_V \tag{3-13}$$

$$\boldsymbol{Z_I} = \boldsymbol{A_I^{multi}} + \boldsymbol{T_I} \tag{3-14}$$

After obtaining the cross-attention outputs $\boldsymbol{Z_V}$ and $\boldsymbol{Z_I}$, they are further processed by a Multilayer Perceptron (MLP)- to enhance feature representation and improve non-linear modeling capacity. In Transformer-based architectures, the MLP module typically adopts a two-layer feedforward design: the first fully connected layer projects the input features into a higher-dimensional space, followed by a non-linear activation function, and the second fully connected layer maps the features back to the original dimension. The MLP serves to refine the fused features by capturing higher-order interactions that cannot be fully exploited by the attention mechanism alone. To stabilize training and preserve the original feature information, skip connections with learnable coefficients $(\alpha, \beta, \gamma, \delta)$ are introduced. These coefficients act as adaptive weights that control the balance between the original features and the transformed outputs of the MLP. This design not only mitigates feature degradation but also enables the model to flexibly adjust the relative influence of different modalities during fusion.

As a result, the refined outputs of this stage are denoted as $\boldsymbol{Z_V^{'}}$ and $\boldsymbol{Z_I^{'}}$, which provide enriched and modality-aware feature representations for subsequent detection tasks. Additionally, a skip connection from the original token sequences $\boldsymbol{T_V}$ and $\boldsymbol{T_I}$ is integrated, yielding the final refined tokens $\boldsymbol{T_V^{ref}}$ and $\boldsymbol{T_I^{ref}}$, where the superscript ***ref*** indicates refinement. These tokens are subsequently reprojected back to the original spatial resolution, forming the refined feature maps $\boldsymbol{F_V^{ref}}$ and $\boldsymbol{F_I^{ref}}$. Finally, the refined visible and infrared features are concatenated and forwarded to Neck for joint processing.

$$\boldsymbol{Z_V^{'}} = \alpha \cdot \mathrm{MLP}(\boldsymbol{Z_V}) + \beta \cdot \boldsymbol{Z_V} \tag{3-15}$$

$$\boldsymbol{Z_I^{'}} = \gamma \cdot \mathrm{MLP}(\boldsymbol{Z_V}) + \delta \cdot \boldsymbol{Z_V} \tag{3-16}$$

$$\boldsymbol{T_V^{ref}} = \boldsymbol{Z_V^{'}} + \boldsymbol{T_V} \tag{3-17}$$

$$\boldsymbol{T_I^{ref}} = \boldsymbol{Z_I^{'}} + \boldsymbol{T_I} \tag{3-18}$$

3.5 Explainability - Gradient-weighted Class Activation Mapping

3.5.1 Motivation

Deep Convolutional Neural Networks (CNNs) have demonstrated remarkable capabilities in computer vision tasks. However, despite their high detection precision and accuracy, these deep learning models are fundamentally characterized as "black boxes". The high-dimensional matrix multiplications and complex, multi-layered non-linear transformations lack transparent theoretical support, making the internal feature extraction process highly opaque to researchers and end-users alike.

In this study, a sophisticated dual-stream architecture incorporating a novel cross-modal fusion mechanism was designed to aggregate information from both visible and infrared spectrums. While this architectural design aims to leverage the complementary strengths of different sensor modalities, its internal decision-making process remains unobservable. Specifically, without transparent visualization, it is inherently unclear which spatial regions or specific features the neural network is genuinely interested in during the perception phase. Furthermore, it is difficult to explicitly verify what tangible improvements the fusion mechanism brings to the network—whether it successfully suppresses background thermal noise, whether it resolves visual artifacts, or how effectively it aligns the cross-modal attention toward the actual wildfire targets.

Consequently, to validate the effectiveness of the proposed network architecture and to ensure the model is reliably "deployable" for critical wildfire management missions, introducing explainability and feature visualization is strictly necessary. A transparent model not only aids engineers in refining the network structure but also builds trust among firefighters and emergency responders who rely on the system for actionable decision-making.

3.5.2 Mechanism

To address the critical need for model interpretability, Gradient-weighted Class Activation Mapping (Grad-CAM) is employed in this research. Grad-CAM serves as a vital tool for visualizing the specific regions within an input image that significantly influences the decisions made by a CNN model.

The overall process of applying Grad-CAM to a network model, as originally proposed in [36], is illustrated in Figure 3-11. By computing the gradients flowing into a specific neural network layer, this technique effectively captures the importance of each feature map regarding the target class.

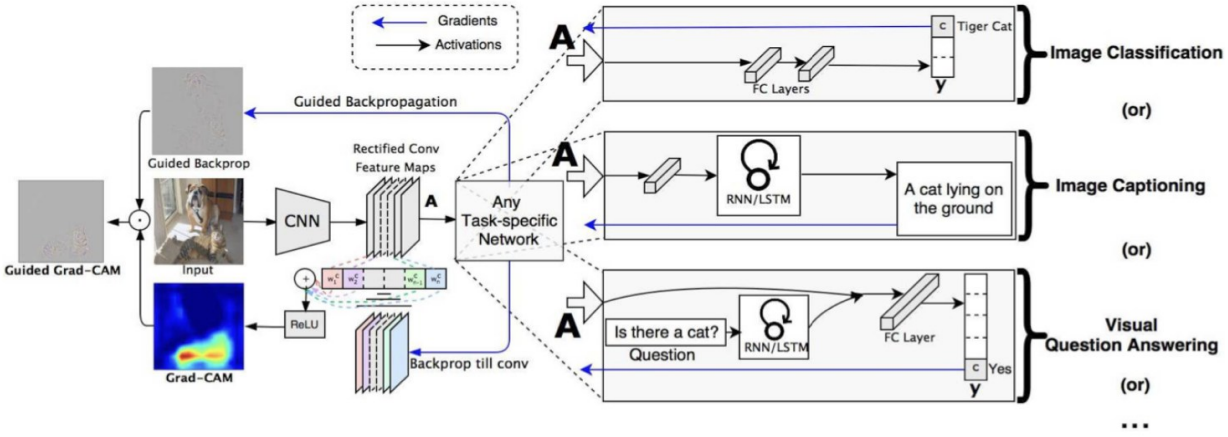

Figure 3-12 The overall process of Grad-CAM [36]

Unlike conventional attribution methods that often focus on individual neurons, Grad-CAM operates by generating a coarse, highly-discriminative localization map that highlights the pivotal regions in the input image without requiring architectural modifications or retraining. It leverages the gradients of any target concept flowing into the final (or intermediate) convolutional layers to capture the importance of each feature map.

Mathematically, let the class discriminative localization map (the Grad-CAM heat map) be denoted as $L^c_{Grad-CAM} \in \mathbb{R}^{u \times v}$, where u and v represent the width and height of the spatial frame, and c denotes the target class (e.g., wildfire). First, the gradient of the classification score for class c, denoted as y^c, is computed with respect to the convolutional feature map A^k. These gradients, represented as $\frac{\partial y^c}{\partial A^k}$ via backpropagation, are then global-average pooled to compute the neuron importance weights, α_k^c:

$$\alpha_k^c = \frac{1}{Z} \sum_i \sum_j \frac{\partial y^c}{\partial A_{ij}^k} \tag{3-19}$$

In this formulation, the neuron importance weights α_k^c represent a partial linearization of the deep network downstream from A, effectively capturing the specific importance of feature map k for the target class c.

Subsequently, the final Grad-CAM heat map is acquired by computing a linear combination of the forward-pass feature maps A^k and their corresponding importance weights α_k^c, followed by a Rectified Linear Unit (ReLU) activation:

$$L_{Grad-CAM}^c = ReLU(\sum_k \alpha_k^c A^k) \tag{3-20}$$

The application of the ReLU activation function is a critical step in this process. It ensures that only the features exerting a positive influence on the class of interest are retained. Negative pixels—which likely belong to other categories or represent background noise—are effectively filtered out, resulting in a clear and interpretable heat map.

By applying Grad-CAM to various target layers within the proposed YOLO-based network, the wildfire detection problem is pulled back into an interpretable feature classification domain.

Specifically, by extracting the Grad-CAM heat maps from layers immediately preceding and following the cross-modal fusion block, the transformation of the network's attention can be visually quantified. It allows us to explicitly observe the independent attention distribution of the visible and infrared branches, and most importantly, it visualizes how the fusion mechanism suppresses the individual noise from each modality and refines the spatial focus onto the true wildfire target. By overlaying these generated class activation maps onto the original input images, Grad-CAM provides profound insights into the model’s reasoning process, ultimately contributing to more proactive, informed, and trustable decision-making in wildfire-prone regions.

Chapter 4

4. Planning

4.1 Overview

After the perception stage, the geolocations of all detected fire spots (represented as 3D coordinates) are assumed to be available, and the UAV's current position is also known. One of the general objectives of the planning stage is to ensure that the UAVs cover all fire spots with minimal operational costs, such as traveling distance, fuel consumption, or battery usage. Based on the existing hardware platforms of our system, two scenarios are considered: a single-drone mission and a multi-drone mission.

For the single-drone case, the task of covering multiple fire spots can be modeled as a combinatorial optimization problem. When the number of fire spots is relatively small, Dynamic Programming (DP) is applied. In contrast, when the number of fire spots becomes large, Simulated Annealing (SA) is adopted to efficiently search for near-optimal solutions. In our group's hardware platform, the UAV's payload limits the number of fire spots per sortie, so the problem size remains small and DP is adopted as the primary method.

For the multi-drone case, the problem is inherently more complex, as both task allocation and path planning need to be addressed. To this end, a Genetic Algorithm (GA) is employed.

The overall categorization of the path planning algorithms is illustrated in Figure 4-1:

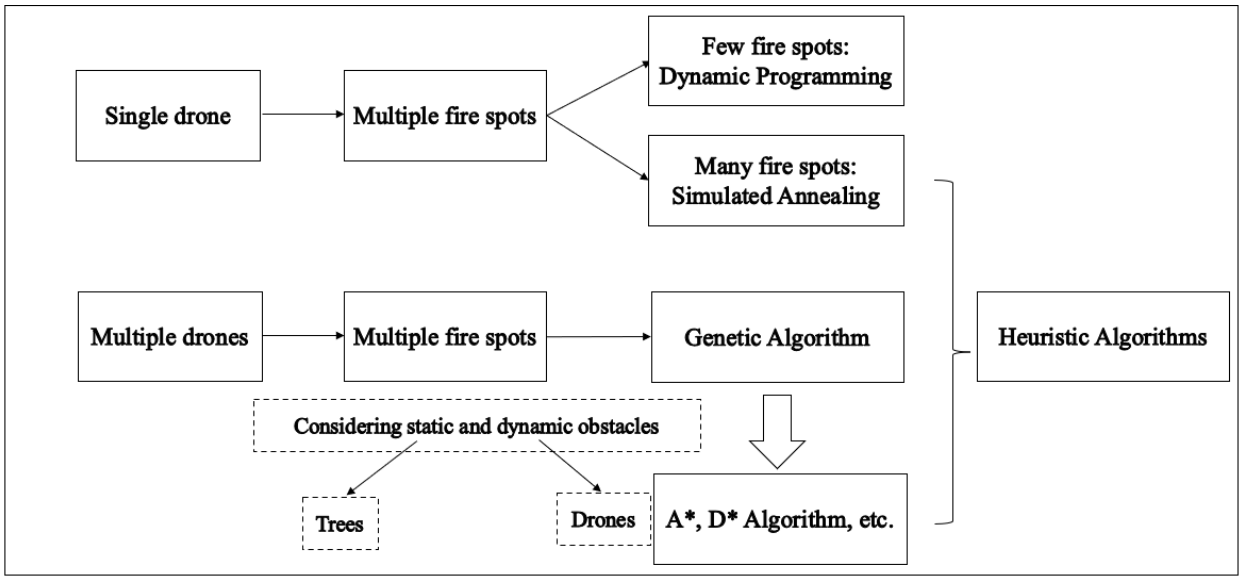

Figure 4-1 Path planning algorithms in UAV-based wildfire suppression mission

4.2 Dynamic Programming-based Planner

4.2.1 Objective

Figure 4-2 illustrates a scenario in which a single UAV is deployed to extinguish multiple wildfire spots. The UAV departs from an initial take-off position, and each wildfire spot is modeled as a waypoint that must be sequentially visited. At each waypoint, the UAV hovers in place to release fire suppressant onto the fire. Once all waypoints have been visited, the UAV returns to its original take-off position.

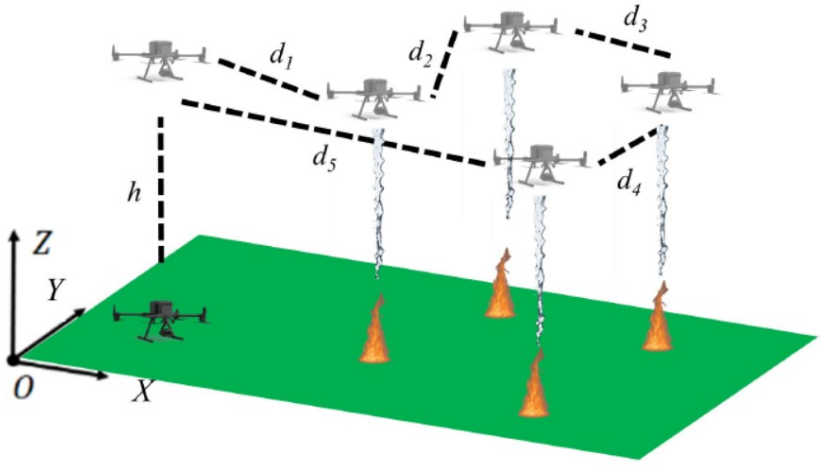

Figure 4-2 Schematic diagram of a single UAV for multiple wildfire spots suppression

To maximize endurance performance, the path planning cost function is defined as the total flight distance. The planning algorithm seeks to minimize this cost. Unlike ground-based vehicles, UAVs operate in relatively open airspace, where obstacles are negligible. Consequently, the distance between any two wildfire spots is approximated by their Euclidean (straight-line) distance, denoted as d_1, d_2, …, shown in Figure 4-2.

Formally, let the UAV's take-off position be denoted as

$$T(x_0, y_0, z_0) \tag{4-1}$$

and let there be n wildfire spots, each represented as

$$F_i(x_i, y_i, z_i), i = 1, 2, \ldots, n \tag{4-2}$$

The objective is to determine an optimal visiting sequence π, starting from T, traversing all wildfire spots $\{F_1, F_2, ..., F_n\}$, and finally returning to T, such that the total trajectory cost is minimized.

The trajectory cost function $f(\pi)$ is defined as the total Euclidean distance traveled along the visiting sequence π, plus an additional $2h$, which accounts for both the initial ascent and the final descent corresponding to the take-off height h.

$$f(\pi) = \sum_{i=0}^{n} \sqrt{(x_{i+1} - x_i)^2 + (y_{i+1} - y_i)^2 + (z_{i+1} - z_i)^2} + 2h \tag{4-3}$$

4.2.2 Algorithm

The above trajectory optimization problem can be reformulated as a classical Traveling Salesman Problem (TSP), in which the UAV must determine an optimal visiting sequence that covers all fire spots exactly once and returns to the initial take-off location. In UAV-based wildfire suppression missions, the problem size is inherently limited due to practical hardware constraints. Specifically, the payload capacity and the amount of fire retardant carried by a single UAV restrict the number of fire spots that can be effectively covered in a single sortie. In our group's hardware platform, the covered fire spots should be no more than 10 spots.

Given such a relatively small number of nodes, exact solutions to the trajectory optimization problem remain computationally feasible. Several classical approaches are available for solving TSP instances of this scale, including brute-force enumeration of all possible visiting sequences,

dynamic programming methods such as the Held–Karp algorithm, and branch-and-bound techniques. While brute-force search guarantees optimality, its factorial-time complexity makes it highly inefficient, even for moderate values of n. In contrast, the dynamic programming approach strikes a balance by achieving the same optimal solution with significantly reduced complexity, which makes it a suitable choice for real-time mission planning.

In this thesis, a dynamic programming-based trajectory planning algorithm is adopted. Formally, let $\boldsymbol{S}$ denote the set of all fire spots F. $D[\boldsymbol{M}][N]$ is defined as the minimum distance of a trajectory that begins at the UAV's take-off position T, visits all fire spots in the subset, and ends at a fire spot $N \in \boldsymbol{M}$.

The overall procedure is summarized in Algorithm 1 and can be outlined in four steps:

1. Compute Euclidean distances $d_{i,j}$ ($i \neq j$, $i < j$) for all node pairs.

2. Define initial state: $D[\{0, j\}][j] = d_{0,j}$ ($\forall j \neq 0$).

3. Define state transition equation: $D[S][j] = \min_{k \in S, k \neq j} [(D[S \setminus \{j\}][k] + d_{k,j}) + d_{j,0}]$.

4. Iterate to solve the optimal path π and minimal $f(\pi)$.

Algorithm 1: Dynamic Programming Algorithm for Multiple Wildfire Spots Suppression by Single UAV Trajectory Planning

---

Input: $T(x_0, y_0, z_0)$; $F_i(x_i, y_i, z_i)$, where $i = 1, 2, \ldots, n$

Output: π with $f(\pi)_{\min}$

Compute Euclidean distances $d_{i,j}$ for all node pairs $(i \neq j, i < j)$ where $i, j \in \{0, \ldots, n\}$

for i = 1 to n **do**

  Let $\boldsymbol{A}$ be the set of all subsets of size i from $\boldsymbol{S} = \{F_1, \ldots, F_n\}$

  while $\boldsymbol{A} \neq \varnothing$ **do**

    Select a subset $\boldsymbol{B} \subseteq \boldsymbol{A}$

    for each $F_k \in \boldsymbol{B}$ **do**

      if i=1 **then**

        $D[\boldsymbol{B}][F_k] \leftarrow d_{0,k}$

      else

10: $D[\boldsymbol{B}][F_k] \leftarrow \min_{F_{k'} \in \boldsymbol{B} \setminus F_k} (D[\boldsymbol{B} \setminus \{F_k\}][F_{k'}] + d_{k',k})$

11: **end if**

12 **end for**

13: $\boldsymbol{A} \leftarrow \boldsymbol{A} / \boldsymbol{B}$

14: **end while**

15: **end for**

16: $f(\pi)_{\min} \leftarrow \min_{F_l \in \boldsymbol{S}} (D[\boldsymbol{S}][F_l] + d_{l,0})$

17: **Set:** $min \leftarrow \infty, \hat{F} \leftarrow T, \pi \leftarrow T$

18: **while** $\boldsymbol{S} \neq \varnothing$ **do**

19: **for** j=1 to n **do**

20: **if** F_j is not visited **then**

21: **if** $min > D[\boldsymbol{S}][F_j] + d_{j,0}$ **then**

22: $min \leftarrow D[\boldsymbol{S}][F_j] + d_{j,0}$

23: $temp \leftarrow F_j$

24: **end if**

25: **end if**

26: **end for**

27: $min \leftarrow \infty$

28: $\hat{F} \leftarrow temp$

29: mark $\hat{F}$ as visited

30: $\pi \leftarrow \pi \cup \hat{F}$

31: $\boldsymbol{S} \leftarrow \boldsymbol{S} \setminus \hat{F}$

32: **end while**

33: **return** $\pi, f(\pi)_{\min}$

The time complexity of the dynamic programming approach is:

$$O(n^2 2^n) \tag{4-4}$$

4.3 Simulated Annealing-based Planner

4.3.1 Algorithm

Although Dynamic Programming (DP) is the preferred choice for the single-drone-multiple-fire-spot wildfire suppression mission, given the small number of fire spots, Simulated Annealing (SA) is also investigated here as an alternative for this TSP formulation. The full SA procedure is presented in Algorithm 2. The symbols $T(x_0, y_0, z_0)$; $F_i(x_i, y_i, z_i), i = 1, 2, \ldots, n$; π; $f(\pi)$ retain the same definitions as introduced in Chapter 4.2. The hyperparameters are defined as follows: P_0 is the initial acceptance probability; α is the cooling rate; L is the inner loop length; $T_{\min}$ is the minimum temperature; M is the maximum non-improvement iterations.

Algorithm 2: Simulated Annealing Algorithm for Multiple Wildfire Spots Suppression by Single UAV Trajectory Planning

---

Input: $T(x_0, y_0, z_0)$; $F_i(x_i, y_i, z_i)$, where $i = 1, 2, \ldots, n$; $P_0; \alpha; L; T_{\min}; M$

Output: $\pi_{\min}$ with $f(\pi)_{\min}$

1: Compute Euclidean distances $d_{i,j}$ for all node pairs $(i \neq j, i < j)$ where $i, j \in \{0, \ldots, n\}$

2: Construct an initial visiting sequence π_0 by Nearest-Neighbor method

3: Set $\pi_{\min} = \pi_0$, $f(\pi)_{\min} = f(\pi)$

4: Estimate the average uphill cost $\bar{\Delta}$

5: Initialize temperature $T_0 = -\dfrac{\bar{\Delta}}{\ln p_0}$

6: Set $T = T_0$, counter = 0

7: **While** $T > T_{\min}$ **and** counter < M **do**

8: **for** iter = 1 to L **do**

9: Generate a neighbor sequence π' from π by choosing uniformly one of the following:

10: a) Swap: swap two spots F_i, F_j

11: b) 2-opt: reverse the sequence π

12 c) Insert: remove F_i and re-insert after F_j

13: Compute $\Delta f = f(\pi') - f(\pi)$

14: **if** $\Delta f \le 0$ **then** $\pi = \pi'$, $f(\pi)_{\min} = f(\pi)$ **else** accept with probability $e^{\frac{-\Delta f}{T}}$

15: **if** $f(\pi) < f(\pi)_{\min}$ **then** $\pi = \pi$, $f(\pi)_{\min} = f(\pi)$, mark = true

16: **end for**

17: **if** mark = true **then** counter = 0 **else** counter = counter +1

18: Cool down: $T = \alpha T$

19: Reset mark = false

20: **end while**

21: **return** $\pi_{\min}$ and $f(\pi)_{\min}$

4.3.2 Comparison with Dynamic Programming

A. Optimality

DP: Guarantees an exact optimal solution, since it exhaustively explores all possible subsets of fire spots using a recursive state transition formulation.

SA: Provides a near-optimal solution by probabilistically exploring the search space and allowing occasional acceptance of worse solutions to escape local minima.

B. Computational Complexity

DP: Has exponential time complexity $O(n^2 2^n)$, which grows rapidly with the number of fire spots. It is computationally feasible only for a relatively small number of fire spots (e.g., $n \leqslant 20$).

SA: Has polynomial-time complexity depending on the chosen parameters (inner loop length L, cooling rate α, termination criteria). It is computationally more efficient and can handle larger problem sizes, although at the cost of solution accuracy.

C. Implementation Characteristics

DP: Requires explicit state definition, distance precomputation, and Dynamic Programming recursion. It guarantees reproducibility and deterministic results.

SA: Relies on stochastic operations such as swap, 2-opt, and insert. Its performance depends on hyperparameters (initial temperature, cooling schedule, iterations), and results may vary across runs.

D. Practical Applicability in UAV Fire Suppression

DP: Suitable when the number of fire spots is limited, ensuring the UAV follows the exact shortest trajectory and maximizes endurance efficiency.

SA: More practical when dealing with a larger number of fire spots or when real-time computation is required, as it provides high-quality feasible solutions within a reasonable computational time.

4.4 Genetic Algorithm-based Planner

While Chapters 4.2 and 4.3 introduced two trajectory planners (DP and SA) for a single UAV tasked with extinguishing multiple wildfire spots, this subsection extends the planning framework to a multi-UAV scenario. In practice, wildfire suppression missions can involve the collaboration of multiple UAVs, each carrying a limited payload of fire suppressants. Consequently, the mission can be formulated as a Multiple Traveling Salesman Problem (MTSP), where several UAVs depart from a common depot (take-off location), cooperatively visit all wildfire spots, and return to the depot after completing their assigned tasks.

The objective of the MTSP formulation is to minimize the total flight distance, while ensuring that:

1. Each wildfire spot is visited by exactly one UAV.
2. The workload among UAVs is balanced, avoiding overloading any single UAV.
3. All UAVs return to the depot after completing their assigned tours.

To solve this problem, a Genetic Algorithm (GA) is adopted. GA is a population-based metaheuristic inspired by the principles of natural evolution. It iteratively evolves a population of candidate solutions (chromosomes), where each chromosome encodes a feasible assignment and

visiting sequence of wildfire spots to multiple UAVs. The quality of each chromosome is evaluated by a fitness function defined as the total trajectory cost. By applying genetic operators such as selection, crossover, and mutation, the population gradually evolves toward high-quality solutions.

The overall procedure is summarized in Algorithm 3. Specifically, m is the number of UAVs, P is the population size, p_c is the crossover probability, p_m is the mutation probability, and G is the maximum generations.

Algorithm 3: Genetic Algorithm for Multiple Wildfire Spots Suppression by Multiple UAV Trajectory Planning

Input: $T(x_0, y_0, z_0)$; $F_i(x_i, y_i, z_i)$, where $i = 1, 2, \ldots, n$; m; P; p_c; p_m; G

Output: $\pi_{\min}$ with $f(\pi)_{\min}$

Compute Euclidean distances $d_{i,j}$ for all node pairs $(i \neq j, i < j)$ where $i, j \in \{0, \ldots, n\}$

Encode each solution as a chromosome by:

a) Construct a permutation of all wildfire spots.

b) Insert partition markers into the permutation to divide it into routes assigned to m UAVs.

for generation = 1 to G **do**

Evaluate the fitness of each chromosome as the sum of the total flight distance

Select parent chromosomes using tournament selection

Apply crossover with probability p_c to generate offspring

Apply mutation with probability p_m

Repair infeasible chromosomes

Update the population by replacing the least-fit individuals with new offspring

Calculate $f(\pi)$

Update $f(\pi)_{\min}$ and $\pi_{\min}$

end for

13: **return** $\pi_{\min}$ and $f(\pi)_{\min}$

Chapter 5

5. Control

5.1 Overview

After the trajectory planner determines the optimal path, a controller should be designed to ensure that the UAV accurately tracks this path. In this chapter, a Linear Quadratic Tracker (LQT) is developed to address the optimal trajectory tracking problem.

5.2 Linear Quadratic Tracker

As LQT is a model-based control method, it is first necessary to analyze the dynamic model of the UAV. In this study, the DJI Matrice 300 RTK quadrotor is employed as the experimental platform for outdoor validation. The system modeling and controller design are therefore developed on the basis of the physical and dynamic characteristics of this quadrotor.

5.2.1 System Modeling

A. Kinematics

As illustrated in Figure 5-1, the inertial frame is defined as $O-X_iY_iZ_i$, while the body frame is defined as $C-X_bY_bZ_b$, where C represents the quadrotor's center of mass. The distance

between the center of mass C and each motor axis is denoted by L. The quadrotor's orientation with respect to the inertial frame is described using the $Z-X-Y$ Euler angles, and the corresponding rotation matrix $\boldsymbol{R}$ is given in (5-1).

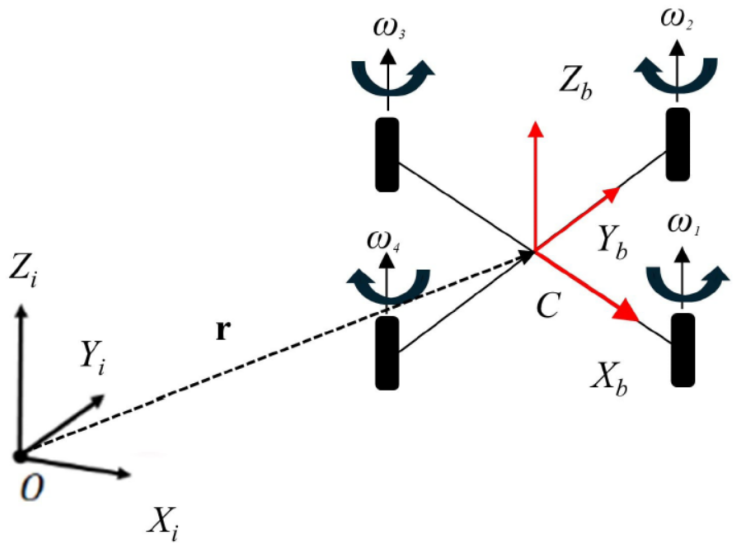

Figure 5-1 Quadrotor coordinate systems.

$$\boldsymbol{R}=\begin{pmatrix} c_\psi c_\theta - s_\phi s_\psi s_\theta & -c_\phi s_\psi & c_\psi s_\theta + c_\theta s_\phi s_\psi \\ c_\theta s_\psi + c_\psi s_\phi s_\theta & c_\phi c_\psi & s_\psi s_\theta - c_\psi c_\theta s_\phi \\ -c_\phi s_\theta & s_\psi & c_\phi c_\theta \end{pmatrix} \tag{5-1}$$

where $c = \cos(\cdot)$ and $s = \sin(\cdot)$, ϕ,θ,ψ are the roll, pitch, and yaw angles.

The relationships between the components of angular velocity in the quadrotor body frame p, q, r and the derivatives of the roll, pitch, yaw angles $\dot{\phi},\dot{\theta},\dot{\psi}$ are described in (5-2).

$$\begin{bmatrix} p \\ q \\ r \end{bmatrix}=\begin{pmatrix} c_\theta & 0 & -c_\phi s_\theta \\ 0 & 1 & s_\phi \\ s_\theta & 0 & c_\phi c_\theta \end{pmatrix}\begin{bmatrix} \dot{\phi} \\ \dot{\theta} \\ \dot{\psi} \end{bmatrix} \tag{5-2}$$

B. Dynamics

Let $\boldsymbol{r} = (x, y, z)$ represent the position vector of point C in the inertial frame $O-X_iY_iZ_i$. The system is subject to gravitational force as well as the thrust forces generated by the individual rotors, denoted as F_i. Based on these forces, the equations governing the translational acceleration of the quadrotor can be expressed as follows:

$$m\ddot{\boldsymbol{r}} = \begin{bmatrix} 0 \\ 0 \\ -mg \end{bmatrix} + \boldsymbol{R} \begin{bmatrix} 0 \\ 0 \\ F_1 + F_2 + F_3 + F_4 \end{bmatrix} \tag{5-3}$$

where m is the mass of the quadrotor.

In addition to forces, each rotor produces a moment perpendicular to the plane of rotation of the blade, M_i. The angular acceleration of the quadrotor is:

$$\boldsymbol{I} \begin{bmatrix} \dot{p} \\ \dot{q} \\ \dot{r} \end{bmatrix} = \begin{bmatrix} L(F_2 - F_4) \\ L(F_3 - F_1) \\ M_1 - M_2 + M_3 - M_4 \end{bmatrix} - \begin{bmatrix} p \\ q \\ r \end{bmatrix} \times \boldsymbol{I} \begin{bmatrix} p \\ q \\ r \end{bmatrix} \tag{5-4}$$

where $\boldsymbol{I}$ is the inertial matrix of the quadrotor.

$$\boldsymbol{I} = \begin{pmatrix} I_{xx} & 0 & 0 \\ 0 & I_{yy} & 0 \\ 0 & 0 & I_{zz} \end{pmatrix} \tag{5-5}$$

C. Motor Model

Assume each motor rotates with an angular speed ω_i, and the generated vertical force F_i is

$$F_i = k_F \omega_i^2 \tag{5-6}$$

and the generated moment is

$$M_i = k_M \omega_i^2 \tag{5-7}$$

Define $\gamma = \dfrac{k_M}{k_F}$, and (6) can be substituted as

$$\boldsymbol{I} \begin{bmatrix} \dot{p} \\ \dot{q} \\ \dot{r} \end{bmatrix} = \begin{pmatrix} 0 & L & 0 & -L \\ -L & 0 & L & 0 \\ \gamma & -\gamma & \gamma & -\gamma \end{pmatrix} \begin{bmatrix} F_1 \\ F_2 \\ F_3 \\ F_4 \end{bmatrix} - \begin{bmatrix} p \\ q \\ r \end{bmatrix} \times \boldsymbol{I} \begin{bmatrix} p \\ q \\ r \end{bmatrix} \tag{5-8}$$

The system input $\boldsymbol{u}$ is defined as

$$\boldsymbol{u} = \left[u_1, \boldsymbol{u}_2 \right] \tag{5-9}$$

where

$$u_1 = \sum_{i=1}^{4} F_i \tag{5-10}$$

$$\boldsymbol{u}_2 = \begin{pmatrix} 0 & L & 0 & -L \\ -L & 0 & L & 0 \\ \gamma & -\gamma & \gamma & -\gamma \end{pmatrix} \begin{bmatrix} F_1 \\ F_2 \\ F_3 \\ F_4 \end{bmatrix} \tag{5-11}$$

5.2.2 LQT Design

A. Model Linearization

The model is linearized at an operating point which corresponds to the nominal hovering state, $r = r_0, \theta = \phi = 0, \psi = \psi_0, \ddot{r} = 0$, and $\dot{\phi} = \dot{\theta} = \dot{\psi} = 0$.

By linearizing (5-3), it derives

$$\ddot{r} = \begin{bmatrix} g(\theta \cos\psi_0 + \phi \sin\psi_0) \\ g(\theta \sin\psi_0 - \phi \cos\psi_0) \\ \frac{1}{m} u_1 - g \end{bmatrix} \tag{5-12}$$

By linearizing (5-8), it derives

$$\begin{bmatrix} \dot{p} \\ \dot{q} \\ \dot{r} \end{bmatrix} = \boldsymbol{I}^{-1} \begin{pmatrix} 0 & L & 0 & -L \\ -L & 0 & L & 0 \\ \gamma & -\gamma & \gamma & -\gamma \end{pmatrix} \begin{bmatrix} F_1 \\ F_2 \\ F_3 \\ F_4 \end{bmatrix} \tag{5-13}$$

B. State-Space Model

Given that the primary objective of this study is trajectory tracking, the LQT design only concentrates on the positional control. The inner-loop attitude control is implemented using a simplified PID scheme. For the positional control purpose, the state variable is selected as $\boldsymbol{x} = [x, y, z, \dot{x}, \dot{y}, \dot{z}]^T$, and the input variable is $\boldsymbol{u} = [u_1, \boldsymbol{u}_2]^T$.

Combining (5-2), (5-8), (5-12) & (5-13), the system's state-space model can be described as

$$\dot{\boldsymbol{x}} = \boldsymbol{A}\boldsymbol{x} + \boldsymbol{B}\boldsymbol{u} \tag{5-14}$$

LQT is a controller design method from optimal control theory. It aims to force the system state $\boldsymbol{x}$ to track a time-varying reference trajectory $\boldsymbol{r}$. By defining the tracking error as $\boldsymbol{e} = \boldsymbol{x} - \boldsymbol{r}$, the LQT algorithm aims to minimize the cost function:

It aims to minimize the cost function as

$$J = \frac{1}{2}\int_0^{\infty} (\boldsymbol{e}^T \boldsymbol{Q}\boldsymbol{e} + \boldsymbol{u}^T \boldsymbol{R}\boldsymbol{u})\, dt \tag{5-15}$$

where $\boldsymbol{Q}$ and $\boldsymbol{R}$ are the weighted positive semi-definite matrices penalizing the tracking error vector and the input variable vector.

At the minimum cost J, the optimal control law solution consists of a state feedback term and a feedforward tracking term $\boldsymbol{u}_{ff}$:

$$\boldsymbol{u} = -\boldsymbol{K}\boldsymbol{x} + \boldsymbol{u}_{ff} \tag{5-16}$$

where the optimal feedback gain matrix is:

$$\boldsymbol{K} = \boldsymbol{R}^{-1}\boldsymbol{B}^T \boldsymbol{P} \tag{5-17}$$

and $\boldsymbol{P}$ is the solution of the Algebraic Riccati Equation

$$\boldsymbol{P}\boldsymbol{A} + \boldsymbol{A}^T \boldsymbol{P} - \boldsymbol{P}\boldsymbol{B}\boldsymbol{R}^{-1}\boldsymbol{B}^T \boldsymbol{P} + \boldsymbol{Q} = 0 \tag{5-18}$$

The feedforward term $\boldsymbol{u}_{ff}$, which anticipates and compensates for the dynamics of the desired reference trajectory, is given by:

$$\boldsymbol{u}_{ff} = \boldsymbol{R}^{-1}\boldsymbol{B}^T \boldsymbol{g} \tag{5-19}$$

where $\boldsymbol{g}$ is the auxiliary adjoint vector. The dynamics of $\boldsymbol{g}$ are governed by the following backward differential equation:

$$\dot{\boldsymbol{g}} = -(\boldsymbol{A} - \boldsymbol{B}\boldsymbol{K})^T \boldsymbol{g} - \boldsymbol{Q}\boldsymbol{r} \tag{5-20}$$

Chapter 6

6. Experimental Design and Analysis

6.1 Experimental Platform

6.1.1 Hardware

A. DJI M300 RTK and H20T Camera

The DJI Matrice 300 RTK (M300 RTK) is a professional-grade industrial quadrotor UAV widely adopted in research and commercial applications. It features a maximum payload capacity of approximately 2.7 kg and a maximum takeoff weight of 9 kg, allowing the integration of multiple sensors and customized hardware. With a maximum flight time of up to 55 minutes under standard conditions, the M300 RTK offers extended mission endurance compared with typical UAV platforms.

The UAV incorporates multiple redundancy designs, including dual batteries, dual IMUs, dual barometers, and dual RTK antennas, which enhance operational safety and reliability. In terms of perception, the platform is equipped with a six-directional sensing and positioning system (vision sensors and ToF sensors in the front, rear, left, right, top, and bottom directions) that improves navigation safety in complex outdoor environments. With centimeter-level accuracy provided by the RTK positioning system (approximately 1 cm + 1 ppm horizontally and 1.5 cm + 1 ppm

vertically), the M300 RTK establishes a robust and reliable experimental platform. These features make it particularly suitable for UAV-based wildfire detection and suppression tasks, where both high payload capacity and precise navigation are essential.

The DJI Zenmuse H20T is a multi-sensor payload specifically designed for integration with the M300 RTK platform. It combines four sensing modules into a single gimbal: a 20 MP zoom RGB camera, a 12 MP wide-angle RGB camera, a radiometric thermal infrared camera, and a laser rangefinder with a detection range of up to 1200 m. This configuration provides complementary visual and thermal information, enabling robust perception under diverse environmental conditions, including smoke, low visibility, and nighttime operations.

The thermal camera supports real-time temperature measurement and provides radiometric data, which is particularly valuable for wildfire detection and hotspot localization. The integrated zoom and wide-angle cameras allow both detailed inspection of specific regions and wide-area situational awareness, while the laser rangefinder enables accurate georeferencing of detected targets.

Mounted on a stabilized three-axis gimbal, the H20T ensures high-quality imagery and reliable data acquisition during UAV flight. When combined with the precise navigation and payload capacity of the M300 RTK, the H20T establishes a powerful sensing platform for real-time wildfire detection, monitoring, and suppression guidance.

B. iCrest 2.0 Onboard Computer

The iCrest 2.0 onboard computer, developed by GEOAI, is employed as the onboard processing unit for the experimental platform. It is powered by the NVIDIA Jetson Xavier NX module, which provides high-performance GPU computing capability for real-time perception, planning, and control tasks. The onboard computer integrates essential interfaces such as USB, CAN, UART, and Ethernet, enabling seamless communication with the UAV flight controller, sensors, and external hardware modules. By serving as the central hub of computation and coordination, the iCrest 2.0 ensures that the UAV can perform autonomous wildfire detection and suppression missions in real-world environments.

C. Ground Station

The ground station is used to send commands to the onboard computer. It communicates with the onboard computer via SSH over a Wi-Fi connection. Both computers are equipped with

wireless network cards and are connected to the same Local Area Network (LAN) established by a router.

D. Multi-drop Mechanism

To support the UAV-based wildfire suppression mission, a customized water-dropping mechanism was designed and connected to an existing 3D-printed water tank built by a group member, Erfan Dilfanian. To enable multiple-spot wildfire suppression operations within a single mission, the water release mechanism must be capable of repeated opening and closing. Therefore, a solenoid valve is employed to realize the multi-drop functionality. The solenoid valve is powered by a 12V battery, and an Arduino Nano is used to send the controller signal. A relay is used between the battery and Arduino Nano to allow 5V low-voltage control signals to control 12V high-voltage power signals. The solenoid valve is powered by a 12V battery, while an Arduino Nano is employed to generate the control signals. A relay is used between the Arduino Nano and the battery to enable the 5V low-voltage control signals to switch the 12V high-voltage power required for valve actuation. The overall design of the water-dropping system is illustrated in Figure 6-2.

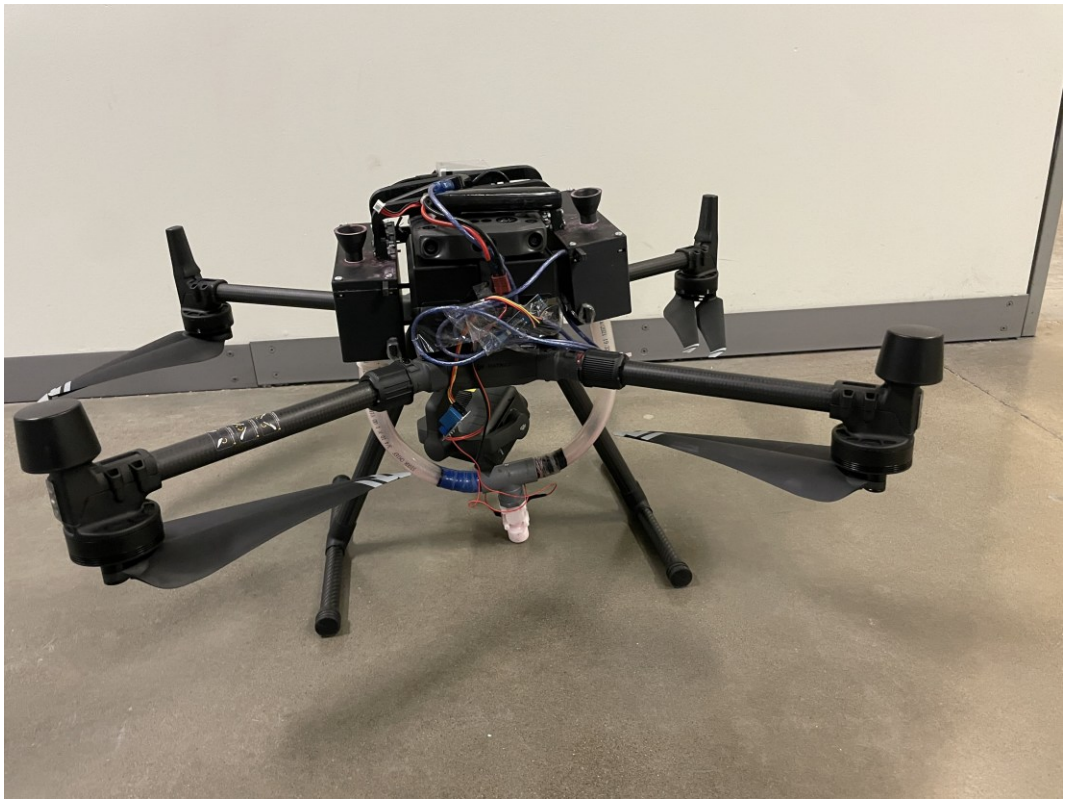

Figure 6-1 Overall hardware

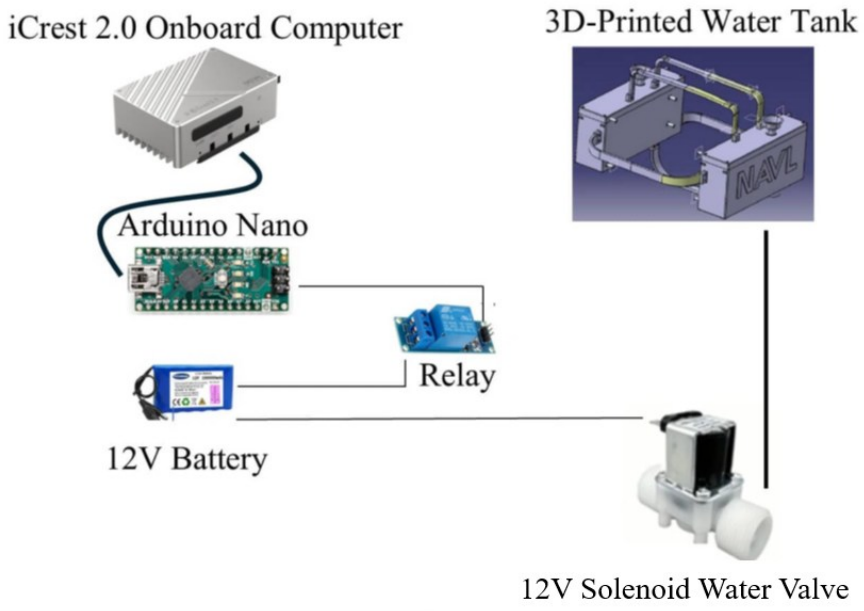

Figure 6-2 Schematic of the multi-drop mechanism

6.1.2 Software

A. DJI OSDK

The DJI Onboard Software Development Kit (OSDK) is a software framework that provides access to the internal functionalities of DJI enterprise UAV platforms, including the Matrice 300 RTK. Through the DJI OSDK, the onboard computer can communicate directly with the UAV flight controller, enabling the implementation of customized control, navigation, and perception tasks. The DJI OSDK offers a comprehensive set of APIs that support functionalities such as flight control (e.g., waypoint navigation, takeoff, landing, velocity, and attitude control), payload management, gimbal control, and real-time telemetry data acquisition.

B. DJI Assistant 2 Simulator

DJI Assistant 2 software suite incorporates a flight simulator that enables verification of UAV operations in a virtual environment prior to real-world deployment. By connecting the UAV to a computer, the simulator reproduces the dynamics of flight, including position, velocity, and GPS feedback, within a three-dimensional environment. This functionality allows researchers to evaluate control algorithms, mission planning strategies, and onboard software in a safe and controlled manner, thereby reducing the risk of hardware damage during preliminary testing.

In the context of this thesis, the simulator serves as an effective platform for debugging, algorithm validation, and operator familiarization. It facilitates the assessment of trajectory planning, sensor integration, and system performance without the constraints of outdoor flight conditions.

C. Robotic Operating System (ROS)

The Robot Operating System (ROS) is an open-source middleware framework widely used in robotics research and development. It provides a structured communication layer for managing distributed processes across robotic systems, enabling seamless integration of perception, planning, and control modules. ROS adopts a publish–subscribe architecture, where data is exchanged through topics, services, and actions, allowing modular and reusable software design.

The DJI-OSDK-ROS is a middleware package that integrates the DJI Onboard Software Development Kit (OSDK) with the Robot Operating System (ROS). It provides a set of ROS nodes, topics, and services that expose the functionalities of DJI enterprise UAV platforms, such as flight control, telemetry data acquisition, gimbal operation, and payload management. By bridging the low-level OSDK interfaces with the ROS ecosystem, DJI-OSDK-ROS enables us to access UAV control and state information through standardized ROS communication mechanisms. This integration allows seamless interoperability with other ROS modules, including perception, mapping, planning, and control, thereby facilitating the development of advanced autonomy and multi-sensor applications.

In this thesis, DJI-OSDK-ROS is used to acquire the images from the camera, set optimal waypoints for the UAV, and call the low-level flight controller to achieve perception-planning-control function.

D. MATLAB/Simulink

MATLAB/Simulink is a widely used software environment for numerical computation, modeling, and simulation. MATLAB provides a high-level programming language and extensive libraries for matrix computation, data analysis, and visualization, while Simulink offers a graphical environment for modeling dynamic systems and designing control algorithms through block diagrams. The tight integration between MATLAB and Simulink enables rapid prototyping, system-level simulation, and automatic code generation for real-time implementation.

In this thesis, MATLAB/Simulink is employed for the simulation of the dynamic programming-based trajectory planner and linear quadratic controller.

E. PyTorch

PyTorch is an open-source deep learning framework widely adopted in both academic research and industry applications. Developed by Meta AI, it provides a flexible and efficient platform for building and training neural networks. PyTorch is built around a dynamic computation graph (define-by-run paradigm), which allows intuitive debugging and flexible model development compared to static graph frameworks. It also includes optimized tensor operations on both CPUs and GPUs, making it highly suitable for large-scale machine learning tasks.

In this thesis, PyTorch is employed to develop and train deep learning neural network models for wildfire detection.

6.2 Experimental Results and Analysis

6.2.1 Experimental Results and Analysis on Perception (Wildfire Detection)

A. Implementation Details

From Chapters 3.3 and 3.4, baseline single-stream detection model YOLOv8n and dual-stream models with various fusion mechanisms, including early fusion, mid-level fusion by direct concatenation, and mid-level fusion by attention-based fusion, were introduced. These deep neural network models are trained on the dataset introduced in Chapter 3.2.

1) Hardware & Software Environment

The training and inference of models are conducted on the ground station introduced in Chapter 6.1.1 *C. Ground Station*, with specifications shown in Table 6-1. Moreover, the software environment for the experiment is also provided in the Table 6-1.

Table 6-1 Hardware & software environment

	Description
CPU	Intel Core i9-12900H
GPU	NVIDIA GeForce RTX 4060 Laptop

RAM	16 GB
CUDA version	12.0
Operating system	Windows 11 Home 25H2
	Python 3.9
Package version	Torch 2.6
	Opencv-python 4.11

2) Hyperparameter Setting

The hyperparameters used for training are listed in Table 6-2.

Table 6-2 Hyperparameter setting

	Setting
Epochs	100
Batch size	8
Optimizer	SGD + Cosine Learning Rate Scheduler
Learning rate	0.01

B. Experiment

1) Evaluation Metrics

Four commonly used metrics in object detection tasks are adopted, which are Precision, Recall, mAP_{50}, mAP_{50-95}.

$$\text{Precision} = \frac{TP}{TP + FP} \tag{6-1}$$

$$\text{Recall} = \frac{TP}{TP + FN} \tag{6-2}$$

where *TP*, *FP*, and *FN* represent true positives, false positives, and false negatives of the fire and background classes in a binary classification object detection task.

The Average Precision (AP) is calculated as:

$$AP = \int_0^1 P(R)\, dR \tag{6-3}$$

where R represents the Recall metric. $P(R)$ denotes the Precision-Recall (P-R) curve, which expresses Precision as a function of Recall across various confidence thresholds. The integral computes the area under this P-R curve from a Recall of 0 to 1, yielding the Average Precision (AP) for the object detection task.

The mean Average Precision at IoU (Intersection over Union) = 0.5 (mAP_{50}) is computed as:

$$mAP_{50} = \frac{1}{N}\sum_{i=1}^{N} AP_i \tag{6-4}$$

where N is the number of classes and AP_i is the Average Precision for the i-th class.

The mean Average Precision over multiple IoU thresholds is computed as:

$$mAP_{50-95} = \frac{1}{10}\sum_{i=1}^{10} AP_{IoU_i} \tag{6-5}$$

where IoU_i corresponds to the thresholds {0.50, 0.55, ...,0.95}.

2) Experimental Results and Analysis

2.1) Model Performance Results and Analysis

Table 6-3 presents a performance comparison for wildfire detection between the proposed model and several baselines. The baseline models include single-stream YOLOv8n models trained and evaluated on visible and infrared images individually, as well as dual-stream models employing early fusion, direct concatenation, and attention-based fusion mechanisms using both visible and infrared images as inputs.

Table 6-3 Wildfire detection performance comparison of different models

	Precision/%	Recall/%	mAP_{50}/%	mAP_{50-95}/%
Single-stream (visible)	94.4	93.0	96.0	55.6
Single-stream (IR)	90.2	87.3	93.3	50.4
Dual-stream (early fusion)	93.4	93.4	96.5	56.5
Dual-stream (direct concat)	94.9	93.7	**97.6**	57.3

Dual-stream (attention-based)	**95.7**	**94.5**	97.2	**57.4**

Bold: The best.

Based on the quantitative results presented in Table 6-3, several key observations and in-depth analyses can be made regarding the performance of different network architectures.

First, regarding the single-modality models, the visible imagery input clearly outperforms the Infrared (IR) imagery input across all evaluation metrics. Specifically, the visible single-stream model exceeds the IR model by 4.2% in Precision and 5.2% in $mAP_{50\text{-}95}$. This significant performance gap can be attributed to the inherent physical and visual characteristics of the input data. Visible images inherently possess higher spatial resolution and richer textural and color features, which are crucial for early wildfire detection where visual cues like smoke plumes and flame colors are primary indicators. Furthermore, state-of-the-art object detection models like YOLOv8 are typically pre-trained on massive visible-spectrum datasets (e.g., COCO), allowing the visible-stream model to leverage robust, pre-learned feature extraction capabilities. In contrast, infrared images, while highly effective at capturing thermal signatures, often lack detailed semantic background information and suffer from lower resolution, leading to a relatively lower detection accuracy when used in isolation.

Second, it is evident that all three dual-modality models consistently surpass the performance of their single-modality counterparts. This strongly demonstrates the complementary nature of visible and infrared spectra in complex environments. While visible imagery provides detailed structural and contextual information (e.g., the shape and density of smoke), it is highly susceptible to variations in illumination, shadows, or visual obstructions. Infrared imagery compensates for these limitations by highlighting high-temperature regions regardless of lighting conditions, effectively penetrating thick smoke or darkness. By intelligently leveraging this multi-modal information, the dual-stream deep neural networks can learn more comprehensive and robust feature representations. This complementarity significantly reduces false positives (such as clouds resembling smoke) and false negatives (such as hidden flames obscured by obstacles), ultimately leading to enhanced and more stable inference performance.

Third, among the three investigated dual-modality architectures, the attention-based fusion mechanism achieves the most optimal overall performance, securing the highest scores in

Precision (95.7%), Recall (94.5%), and the strict mAP_{50-95} metric (57.4%). The early fusion approach performs the worst among the three, likely because prematurely combining fundamentally different modalities at the input level can disrupt the distinct low-level feature extraction processes. While the direct concatenation approach preserves these features and slightly outperforms the attention-based mechanism only in terms of mAP_{50} (97.6% vs 97.2%), it simply treats all channels equally. This naive fusion often introduces feature redundancy and background noise, which negatively impacts precise bounding box regression. The attention-based fusion mechanism addresses this bottleneck by dynamically evaluating the importance of different spatial and channel features. It selectively emphasizes the most informative features from both modalities—such as focusing on the IR thermal signature when a suspicious anomaly is detected on the RGB channel—while suppressing irrelevant background noise. This refined feature selection capability is explicitly reflected in its superior mAP_{50-95} score, indicating that the attention mechanism not only detects the fire more accurately but also localizes the bounding boxes with significantly higher precision. Overall, these comprehensive findings thoroughly validate the superiority and effectiveness of the proposed dual-stream attention-based method for reliable wildfire detection.

2.2) Model Complexity Analysis

Table 6-4 and Table 6-5 present the model complexity in terms of layer depth and parameter count during the training and inference phases, respectively.

Table 6-4 Model complexity of different network architectures during the training phase

	Layer number	Parameter number
Single-stream (visible)	225	3011043
Single-stream (IR)	225	3011043
Dual-stream (early fusion)	238	3013387
Dual-stream (direct concat)	328	2377443
Dual-stream (attention-based)	526	3394743

Table 6-5 Model complexity of different network architectures during the inference phase

	Layer number	Parameter number
Single-stream (visible)	168	3005843
Single-stream (IR)	168	3005843
Dual-stream (early fusion)	181	3008187
Dual-stream (direct concat)	244	2372243
Dual-stream (attention-based)	439	3389319

During the training phase (Table 6-4), the single-stream baselines consist of 225 layers and approximately 3.01 million (M) parameters. The introduction of early fusion minimally alters the architecture, resulting in 238 layers with a negligible parameter increase. In contrast, the direct concatenation approach deepens the network to 328 layers but paradoxically reduces the overall parameter count to approximately 2.37M. This reduction is attributed to the necessary channel

reduction and structural downscaling implemented within the parallel backbone branches prior to feature concatenation, designed to prevent memory overload.

Notably, the proposed dual-stream attention-based model significantly deepens the architecture to 526 layers during training, with a total parameter count of 3.39M. The dramatic increase in layer depth is a fundamental characteristic of attention mechanisms. These modules typically incorporate numerous lightweight operational layers—such as global average pooling, Multilayer Perceptrons (MLPs), and activations—to compute dynamic spatial and channel weights. Despite the extensive depth, the actual increase in trainable parameters remains modest (a mere ~12.6% increase compared to the single-stream baselines), highlighting the parameter efficiency of the designed attention module.

A critical structural optimization emerges when transitioning from the training phase to the inference phase (Table 6-5). As the model prepares for deployment, the framework executes a structural reparameterization technique, specifically Convolutional and Batch Normalization (Conv-BN) fusion. In this process, the independent parameters of the Batch Normalization layers are mathematically absorbed into the weights and biases of their preceding Convolutional layers. This equivalent transformation safely reduces the proposed model's depth from 526 to 439 layers and slightly drops the parameter count to 3.38M. Crucially, this fusion eliminates the computational overhead of BN layers during forward passes, condensing the network structure without compromising any learned representations.

2.3) Computational Efficiency and Real-Time Feasibility

Beyond structural complexity, the actual inference speed is the ultimate bottleneck for UAV applications. Table 6-6 details the per-frame computational latency, broken down into preprocessing, inference, and postprocessing stages.

Table 6-6 Processing latency of different models

	Preprocess/ms	Inference/ms	Postprocess/ms
Single-stream (visible)	0.2	1.5	1.0
Single-stream (IR)	0.2	1.4	0.9
Dual-stream (early fusion)	0.4	2.0	1.5
Dual-stream (direct concat)	0.7	3.5	2.4
Dual-stream (attention-based)	0.4	3.7	1.3

The single-stream models are naturally the fastest, requiring merely 2.5 to 2.7 milliseconds (ms) of total processing time. Processing dual modalities inevitably introduces computational overhead. However, the proposed attention-based model maintains a highly efficient profile, recording an inference time of 3.7 ms and a postprocessing time of 1.3 ms, resulting in a total latency of only 5.4 ms. Interestingly, the direct concatenation model exhibits a higher total latency of 6.6 ms, heavily burdened by a slower postprocessing stage (2.4 ms). This bottleneck is likely due to the massive, unrefined concatenated feature maps passed to the detection head, which complicate the bounding box decoding and Non-Maximum Suppression (NMS) processes.

From a practical UAV deployment perspective, a total processing time of 5.4 ms translates to a theoretical processing speed of approximately 185 frames per second (fps). This performance drastically exceeds the standard real-time video processing requirement of 30 fps. By combining this real-time processing capability with an extremely lightweight footprint of ~3.38M parameters, the comprehensive analysis validates that the proposed dual-stream attention-based method strikes an optimal balance. The fractional increases in parameter count and latency compared to single-stream models are highly justified compromises, yielding significant improvements in wildfire detection accuracy within complex environments.

2.4) Visualization of Feature Activation via Grad-CAM

To further comprehend the internal feature extraction process and visually validate the effectiveness of the proposed dual-stream fusion architecture, Gradient-weighted Class Activation Mapping (Grad-CAM) introduced in Chapter 3.5 was employed. Grad-CAM provides a coarse localization map that highlights the pivotal regions in the input image strongly influencing the model's prediction. By visualizing the feature maps at different depths, the "black box" nature of the neural network becomes transparent, allowing for an intuitive evaluation of how the model manages cross-modal information.

Figure 6-3 illustrates the Grad-CAM results extracted from critical intermediate layers of the proposed dual-stream attention-based fusion model during a wildfire detection inference.

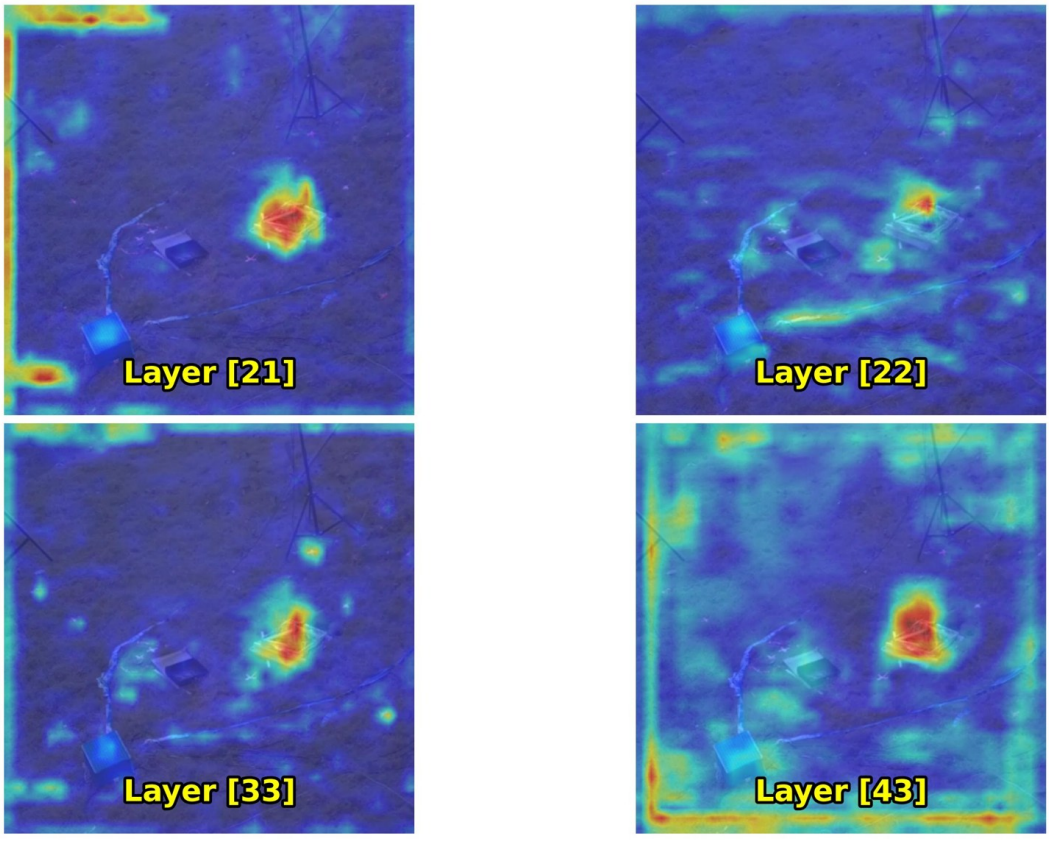

Figure 6-3 Grad-CAM results of different target layers

- Single-Modality Feature Extraction (Layers [21] & [22]): Layers [21] and [22] represent the deep feature maps from the independent visible and infrared backbones, respectively. The visible branch (Layer [21]) successfully identifies the central fire source but exhibits significant dispersed attention along the image boundaries and background textures, indicating a susceptibility to visual noise. Conversely, the infrared branch (Layer [22])

demonstrates extreme sensitivity to thermal signatures. While the core fire is intensely activated, the infrared stream also erroneously highlights a long, horizontal non-target thermal object (likely a heated pipeline or ground reflection) in the lower-middle section of the frame. This visualizes the inherent limitations discussed in Section 6.2.1: single modalities are prone to modality-specific background interference.

- Cross-Modal Feature Refinement (Layer [33]): Layer [33] visualizes the feature activation immediately following the proposed Channel Prior Convolutional Attention (CPCA) module and Dual Modality Cross-attention Transformer Fusion (DMCTF) module. A profound transformation is observable: the fusion mechanism successfully synthesizes the complementary strengths of both modalities while actively suppressing their respective weaknesses. The horizontal thermal noise prevalent in the infrared branch (Layer [22]) has been almost entirely eliminated, and the peripheral visual artifacts from the visible branch (Layer [21]) are significantly dampened. The network's attention is now sharply focused and highly concentrated exclusively on the genuine wildfire target.
- Final Detection Confidence (Layer [43]): Layer [43] represents the ultimate feature representation fed into the P4 detection head. The Grad-CAM heat map reveals an exceptionally high activation intensity (deep red) precisely centered on the fire. The surrounding environment remains largely ignored (blue), confirming that the network has made a highly confident and accurate localization decision based on the refined dual-stream features.

In conclusion, the Grad-CAM visualization provides compelling evidence supporting the quantitative findings in Table 6-3. The proposed attention-based fusion module effectively acts as a spatial filter, suppressing modality-specific noise and aligning the cross-modal focus. This ensures high-quality feature representations are delivered to the detection heads, directly contributing to the model's superior Recall and robust perception capabilities in complex aerial scenarios.

6.2.2 Experimental Results and Analysis on Planning and Control

A. Dynamic Programming-based Planner and Linear Quadratic Tracker

1) Simulation Results and Analysis

To validate the proposed dynamic programming-based single-drone multiple-fire-spot planner and linear quadratic tracker framework, MATLAB/Simulink is first employed to simulate both the trajectory planning algorithm and the trajectory tracking controller.

The quadrotor's take-off point is set as T (0, 0, 0) m, with a designated take-off altitude of h = 10 m. Four random wildfire locations are generated as F_1 (30, 45, 0) m, F_2 (5, 10, 0) m, F_3 (15, 25, 0) m, and F_4 (35, 10, 0) m. Based on publicly available specifications and measurements of the DJI Matrice 300 (M300), the simulation parameters adopted in this study are summarized in Table 6-7.

Table 6-7 Simulation parameters of the M300 quadrotor system

Parameter	Value
m	9 kg
g	9.81m/s^2
$\boldsymbol{I}$	$\begin{pmatrix} 0.449 & 0 & 0 \\ 0 & 0.449 & 0 \\ 0 & 0 & 0.899 \end{pmatrix}$ $\text{kg}\cdot\text{m}^2$
L	0.317 m

In the LQT controller, the weighted matrices $\boldsymbol{Q}$ and $\boldsymbol{R}$ are selected as follows:

$$\boldsymbol{Q}=\begin{bmatrix} 20 & 0 & 0 & 0 & 0 & 0 \\ 0 & 20 & 0 & 0 & 0 & 0 \\ 0 & 0 & 100 & 0 & 0 & 0 \\ 0 & 0 & 0 & 10 & 0 & 0 \\ 0 & 0 & 0 & 0 & 10 & 0 \\ 0 & 0 & 0 & 0 & 0 & 20 \end{bmatrix}, \boldsymbol{R}=0.1\boldsymbol{I}$$

The simulation results are shown in Figure 6-4, Figure 6-5, and Figure 6-6.

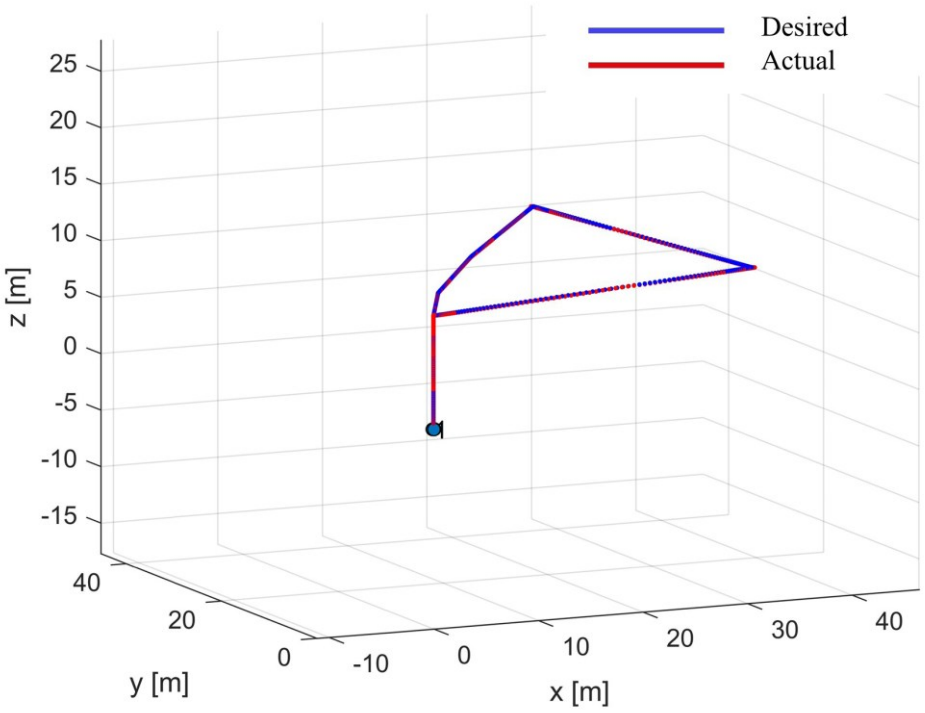

Figure 6-4 Simulated desired and actual quadrotor trajectory

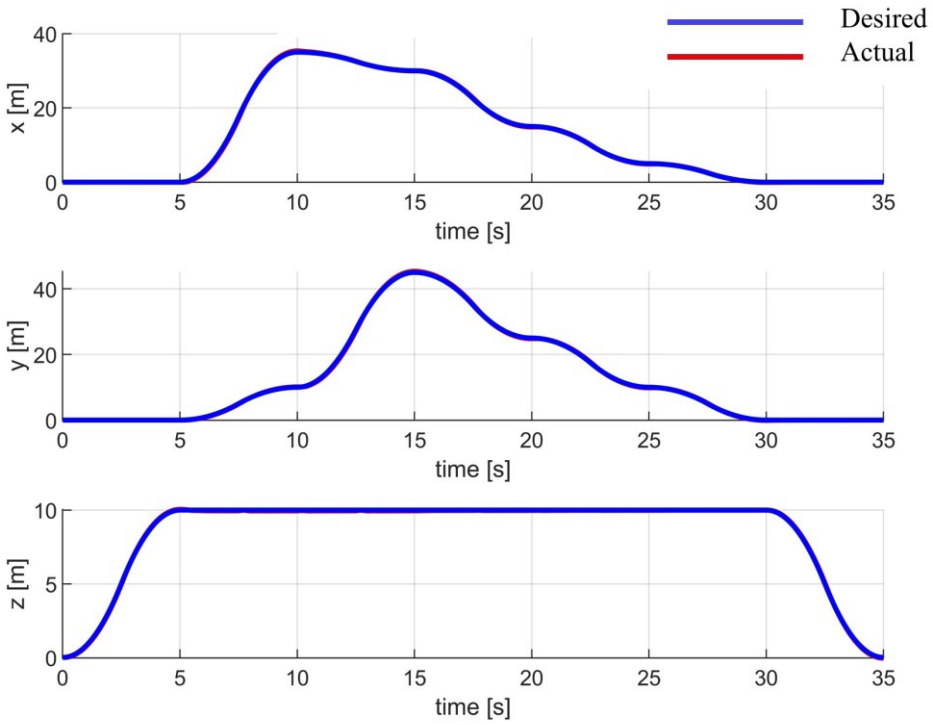

Figure 6-5 Position response of the quadrotor

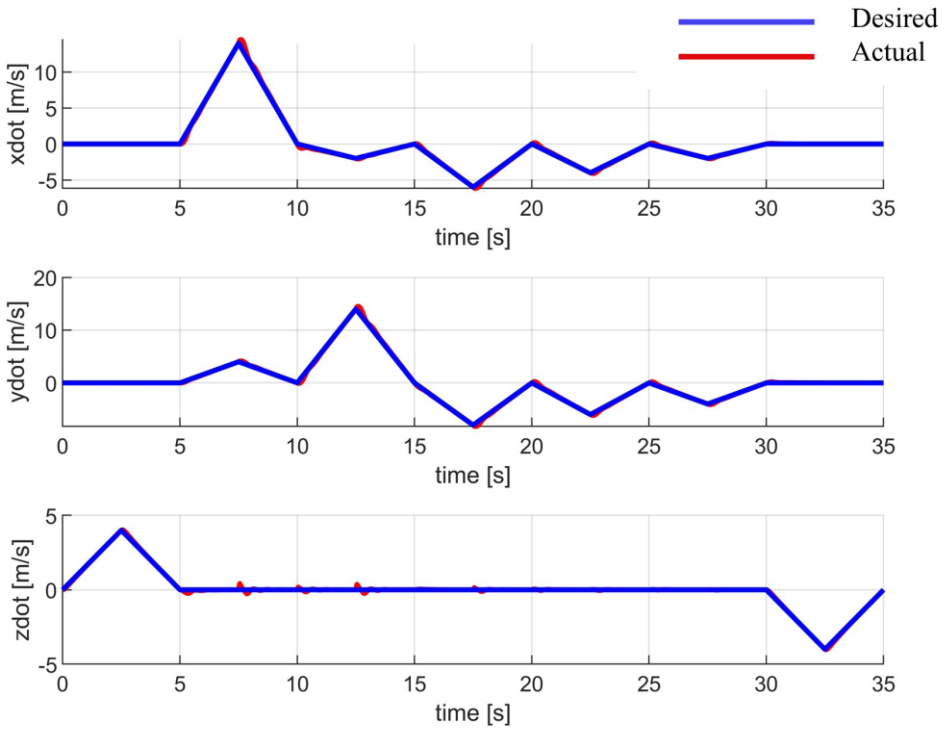

Figure 6-6 Velocity response of the quadrotor

As the above figures show, the quadrotor UAV takes off from T (0, 0, 0) m, sequentially visits all fire spots F_i, and finally returns to the take-off point T. Using the proposed dynamic programming-based trajectory planner, the desired position of the UAV over time is generated, while the corresponding desired velocity is obtained as the first derivative of the position. Subsequently, the designed LQT controller ensures that the UAV accurately tracks the planned trajectory, achieving fast response and low tracking error.

Secondly, prior to conducting outdoor experiments, the DJI Assistant 2 simulator is employed to reproduce the real flight trajectory of the DJI M300 UAV. The simulation environment is configured to match the actual outdoor test site, ensuring consistency between virtual and physical conditions. The corresponding simulation results are presented in Figure 6-7 and Figure 6-8. As can be observed, the proposed algorithm successfully enables the single M300 UAV to reach all predefined wildfire locations in the simulated environment.

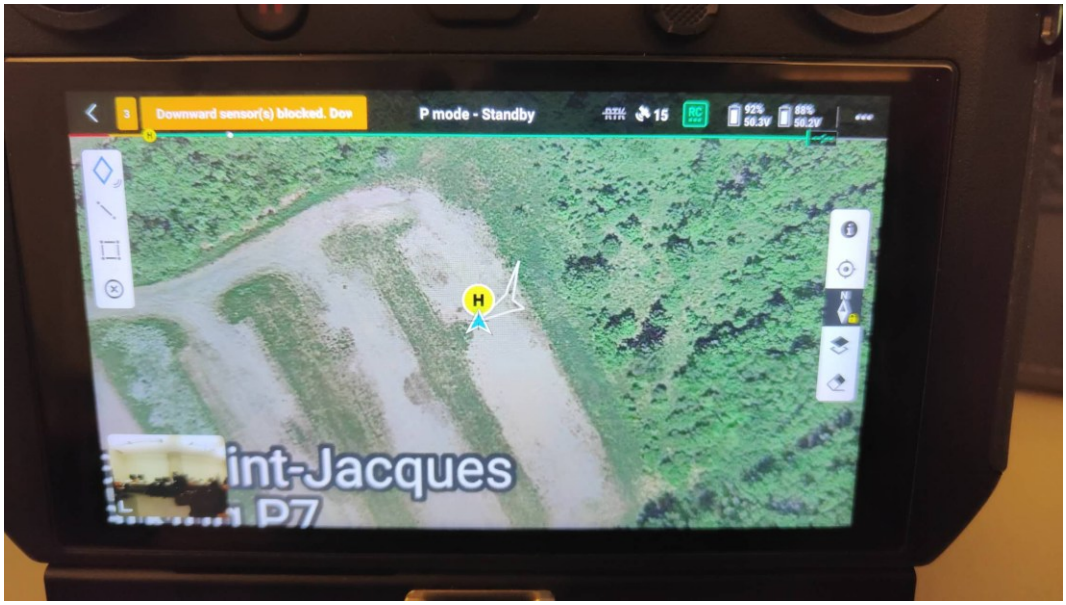

Figure 6-7 Simulated trajectory in DJI remote controller

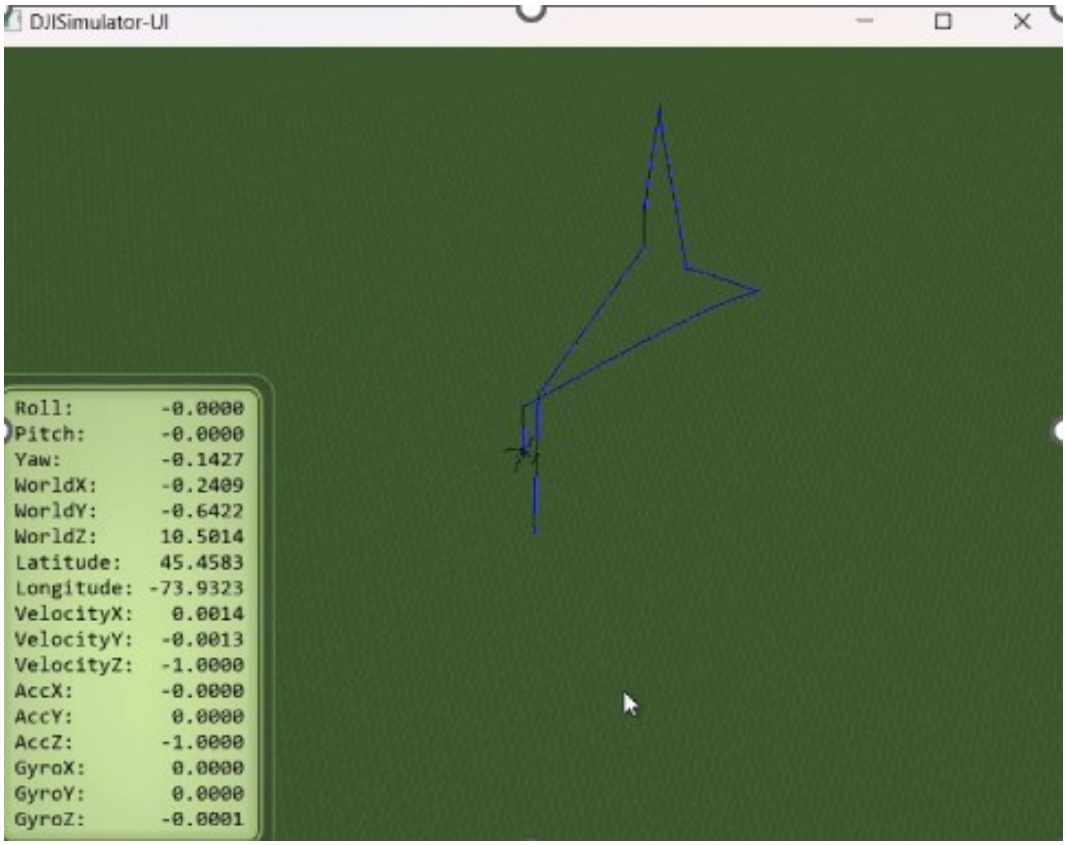

Figure 6-8 Simulated trajectory in DJI Assistant 2 Simulator

2) Outdoor Experimental Results and Analysis

With the support of the simulation results, outdoor experiments were subsequently carried out. At the experimental site, four mimic wildfire spots were created using fire pots, and their three-dimensional coordinates were acquired. The proposed algorithm was deployed on the onboard computer (iCrest 2.0), where the optimal trajectory was generated based on the acquired positional information. Through the DJI OSDK–ROS interface, the computed trajectory was transmitted to the UAV as the flight command. DJI M300 UAV then took off to a predefined altitude and navigated toward the first fire spot following the planned trajectory determined by the dynamic programming algorithm. Upon reaching the target location, the UAV hovered in position while the firefighting mechanism was activated. The solenoid valve of the onboard multi-drop mechanism was opened, allowing water from the tank to be discharged onto the designated fire spot. After a preset release duration, the valve was closed, and the UAV proceeded to the subsequent fire spots sequentially. Once all four fire spots were covered, the UAV returned to its initial take-off position and landed. The outdoor experiment video can be viewed at https://www.youtube.com/watch?v=1xUFH-iEu0k.

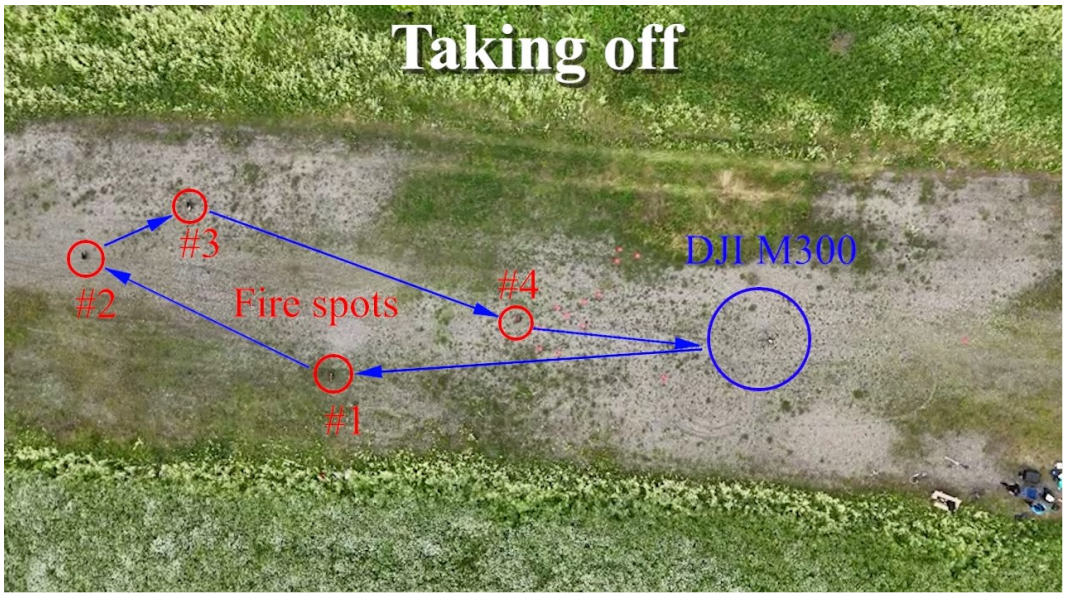

Figure 6-9 Schematic of the outdoor experiment (from the video)

However, as observed in the recorded experiment, the water released at the fire spots deviated from the target location, with a maximum positional error of approximately 2 meters. This deviation is primarily attributed to localization inaccuracies of the UAV and external environmental disturbances such as wind. In addition, due to the limited water capacity of the

onboard tank, the UAV was unable to conduct water release at the fourth fire spot, as the water supply had been depleted beforehand. Future work should therefore investigate enhanced localization methods and wind field modeling to improve the accuracy and reliability of the proposed framework.

B. Genetic Algorithm-based Planner

In the complex operational scenario of multiple-drone and multiple-fire-spot management, efficient task allocation and path planning are critical to minimizing the emergency response time and preserving the limited battery life of the UAV fleet. The proposed Genetic Algorithm (GA) is specifically designed to address this multi-agent routing problem, aiming to minimize the summation of the total flight distance across all deployed drones. To validate the computational feasibility and operational effectiveness of this approach, a numerical simulation was constructed and executed using MATLAB.

The simulation environment is initialized with a highly randomized scenario to accurately emulate unpredictable wildfire outbreaks. As depicted in Figure 6-10, a set of target fire spots (represented by blue dots) is randomly generated and scattered within a predefined 2D operational square region, ranging from -20 meters to 20 meters on both the X and Y spatial axes. A firefighting fleet consisting of three UAVs is stationed at a centralized depot, marked by a red asterisk at the origin coordinate (0,0). This origin serves as the common take-off and landing point for all UAV deployment missions.

Through iterative evolution, the GA-based planner simultaneously resolves two major challenges: task allocation (assigning specific fire spots to specific drones) and routing (determining the optimal visiting sequence). The resulting optimal flight trajectories for the three UAVs are clearly demarcated by red, blue, and green lines in Figure 6-11. A detailed spatial analysis of this result reveals that the algorithm successfully divided the operational area into logical, distinct sub-regions. Each drone efficiently visits its assigned cluster of fire spots in a continuous, closed-loop trajectory before seamlessly returning to the central depot. The smooth routing paths and the absence of heavily entangled or redundant travel lines signify a highly optimized multi-agent coordination strategy that avoids unnecessary energy consumption.

Furthermore, the computational efficiency and evolutionary stability of the proposed algorithm are validated by the objective function convergence curve, as presented in Figure 6-12.

In the initial generations, the randomly initialized population exhibits a high total flight distance, exceeding 310 meters. However, driven by robust selection, crossover, and mutation operators, the algorithm rapidly discards inefficient routes. A steep descent in the travel cost is observed within the first 20 to 30 iterations. The total flight distance steadily converges to a near-optimal global minimum of approximately 228 meters well before the 100th iteration, after which the curve completely flattens into a stable state. This rapid and stable convergence demonstrates that the GA-based planner is not only capable of finding high-quality routing solutions but is also computationally efficient enough to support rapid decision-making in time-sensitive wildfire emergencies.

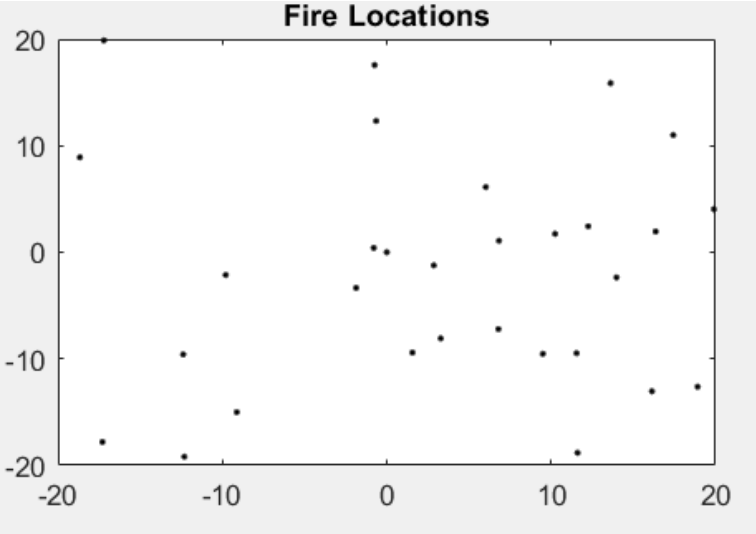

Figure 6-10 Randomly generated fire spot locations

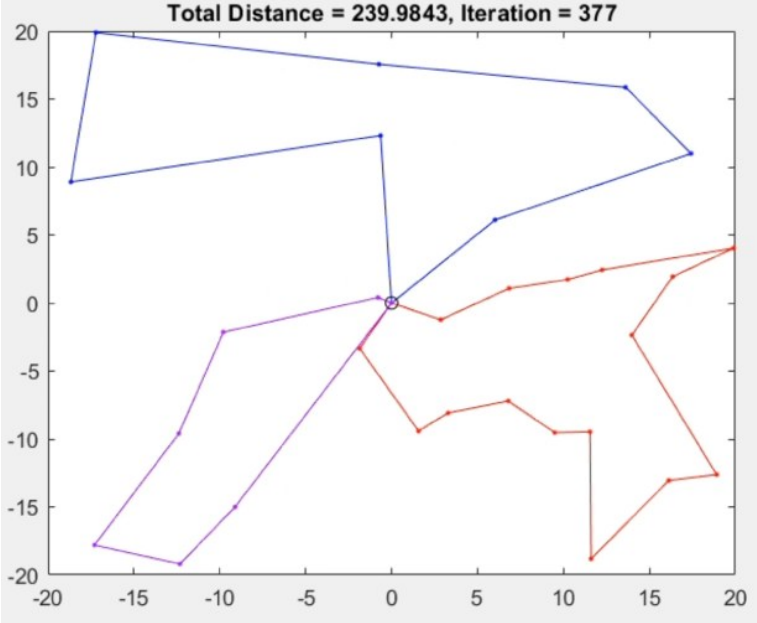

Figure 6-11 Optimal flight trajectory after genetic algorithm iterations

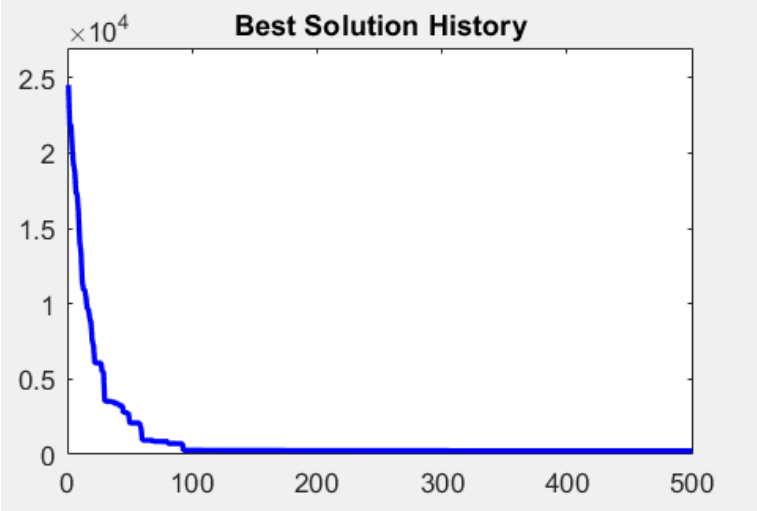

Figure 6-12 Optimization of the total flight distance over iterations

Chapter 7

7. Conclusion and Future Work

7.1 Conclusion

This thesis proposed and implemented a comprehensive, autonomous framework for early-stage wildfire detection and suppression using Unmanned Aerial Vehicles (UAVs). By integrating advanced computer vision, mathematical optimization, and modern control theory, the system empowers a UAV to perceive its environment, formulate optimal flight plans, and execute targeted firefighting actions within a single mission. The major contributions and findings of this research are summarized as follows:

1. **Robust Dual-Stream Wildfire Perception:** To overcome the limitations of single-modality sensors in complex environments (e.g., smoke, low illumination), a visible–infrared wildfire image dataset was constructed and aligned using a least-squares image registration technique. A novel dual-stream detection model based on the lightweight YOLOv8n architecture was developed. By integrating the Channel Prior Convolutional Attention (CPCA) and Dual Modality Cross-attention Transformer Fusion (DMCTF) modules, the model dynamically refined intra-modality and cross-modality features. The proposed attention-based fusion model achieved superior performance with a Precision of

95.7%, a Recall of 94.5%, and an $mAP_{50\text{-}95}$ of 57.4%, while maintaining a real-time processing latency of only 5.4 ms. Furthermore, the application of Grad-CAM provided critical explainability, visually proving that the attention mechanism effectively suppresses background thermal noise and visual artifacts.

2. **Efficient Mission Planning:** For the trajectory planning phase, the mission was formulated to minimize the UAV's total flight distance. For the single-UAV, multiple-fire-spot scenario, the Dynamic Programming (DP) algorithm and Simulated Annealing (SA) were implemented to guarantee an exact optimal visiting sequence, which is highly suitable given the UAV's limited payload capacity. To ensure the scalability of the framework, a Genetic Algorithm (GA)-based planner was developed to solve the Multiple Traveling Salesman Problem (MTSP) for multi-UAV cooperative missions, demonstrating rapid convergence and balanced task allocation in simulations.
3. **Precise Control and Real-World Validation:** A Linear Quadratic Tracker (LQT) was designed based on the kinematics and dynamics of the DJI M300 quadrotor to ensure accurate trajectory tracking. The entire perception-planning-control pipeline was verified through MATLAB/Simulink and DJI Assistant 2 simulations, followed by a real-world outdoor flight test. Equipped with a customized 3D-printed water tank and a solenoid valve-based multi-drop mechanism, the UAV autonomously navigated to multiple predefined fire spots and successfully deployed the fire retardant.

7.2 Future Work

While the proposed framework establishes a solid foundation for autonomous UAV-based firefighting, the outdoor experiments highlighted several practical limitations that provide clear directions for future research:

1. **Enhanced Localization and Disturbance Rejection:** During the outdoor flight test, the deployed water deviated from the target fire spots by a maximum of approximately 2 meters. This discrepancy was primarily caused by the UAV's localization inaccuracies and external environmental disturbances, particularly wind. Future work should focus on integrating advanced visual servoing techniques during the terminal descending phase to visually lock onto the fire target. Additionally, incorporating robust control strategies or

wind field modeling into the LQT controller would significantly improve tracking accuracy and retardant drop precision under adverse weather conditions.

2. **Multi-Agent Coordination and Continuous Operation in the Real World:** The current physical experiments were constrained by the limited capacity of the onboard water tank, which depleted before the UAV could suppress the fourth fire spot. To address the payload bottleneck, future research should transition the multi-UAV Genetic Algorithm (GA) planner from simulation to real-world deployment. Developing a cooperative UAV swarm system where multiple drones share the firefighting workload will drastically improve suppression efficiency. Furthermore, investigating automated refilling stations would allow a UAV to return to base, refill its water tank autonomously, and initiate the next mission without manual interference.
3. **Expanding the Perception Generalization:** Although the dual-stream attention model performed exceptionally well, the current dataset is relatively constrained. Future iterations should expand the visible–infrared dataset to include more extreme scenarios, such as dense canopy occlusions, various weather conditions (e.g., heavy rain, fog), and massive-scale crown fires. Deploying the model onto even lower-power edge devices via model quantization and pruning techniques will also be explored to further reduce power consumption on the UAV.

Bibliography

[1] A. Kumar, T. Ray, T. Mohanasundari, S. S. Jatav, U. Chatterjee, S. Shekhar, E. Alam, and M. K. Islam, "Forest fires and climate change in India: impacts, adaptive strategies, and pathways for climate action (Sustainable Development Goal-13)," *Environmental Sciences Europe*, vol. 37, no. 1, p. 147, Sep. 2025.

[2] E. Ramberg, M. Edman, G. Granath, J. Sjögren, and J. Strengbom, "Prescribed burning for boreal forest restoration: Evaluating challenges and conservation outcomes," *Ambio*, Sep. 2025.

[3] N. R. Canada, "Canadian Wildland Fire Information System | Canadian National Fire Database (CNFDB)." Accessed: Oct. 02, 2025. [Online]. Available: https://cwfis.cfs.nrcan.gc.ca/ha/nfdb

[4] A. Mohapatra and T. Trinh, "Early Wildfire Detection Technologies in Practice—A Review," *Sustainability*, vol. 14, no. 19, p. 12270, Sep. 2022.

[5] C. Yuan, Y. Zhang, and Z. Liu, "A survey on technologies for automatic forest fire monitoring, detection, and fighting using unmanned aerial vehicles and remote sensing techniques," *Canadian Journal of Forest Research*, vol. 45, no. 7, pp. 783–792, Jul. 2015.

[6] M. A. Akhloufi, A. Couturier, and N. A. Castro, "Unmanned Aerial Vehicles for Wildland Fires: Sensing, Perception, Cooperation and Assistance," *Drones*, vol. 5, no. 1, p. 15, Feb. 2021.

[7] E. A. Yfantis, "A UAV with autonomy, pattern recognition for forest fire prevention, and AI for providing advice to firefighters fighting forest fires," in *2019 IEEE 9th Annual Computing and Communication Workshop and Conference (CCWC)*, IEEE, 2019, pp. 0409–0413.

[8] E. Seraj and M. Gombolay, "Coordinated control of UAVs for human-centered active sensing of wildfires," in *2020 American control conference (ACC)*, IEEE, 2020, pp. 1845–1852.

[9] L. Mu, Y. Yang, B. Wang, Y. Zhang, N. Feng, and X. Xie, "Edge Computing-based Real-time Forest Fire Detection using UAV Thermal and Color Images," *IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing*, pp. 1–12, 2025.

[10] R. Bailon-Ruiz, A. Bit-Monnot, and S. Lacroix, "Real-time wildfire monitoring with a fleet of UAVs," *Robotics and Autonomous Systems*, vol. 152, p. 104071, Jun. 2022.

[11]M. Kumar, K. Cohen, and B. HomChaudhuri, “Cooperative Control of Multiple Uninhabited Aerial Vehicles for Monitoring and Fighting Wildfires,” *Journal of Aerospace Computing, Information, and Communication*, vol. 8, no. 1, pp. 1–16, Jan. 2011.

[12]E. Seraj, A. Silva, and M. Gombolay, “Multi-UAV planning for cooperative wildfire coverage and tracking with quality-of-service guarantees,” *Autonomous Agents and Multi-Agent Systems*, vol. 36, no. 2, p. 39, Oct. 2022.

[13]R. B. Zadeh, A. Elmi, V. Moghaddam, and S. Mahmoudzadeh, “A conceptual high level multi-agent system for wildfire management,” *IEEE Transactions on Geoscience and Remote Sensing*, 2025.

[14]S. Partheepan, F. Sanati, and J. Hassan, “Autonomous unmanned aerial vehicles in bushfire management: Challenges and opportunities,” *Drones*, vol. 7, no. 1, p. 47, 2023.

[15]S.-J. Chung, A. A. Paranjape, P. Dames, S. Shen, and V. Kumar, “A survey on aerial swarm robotics,” *IEEE Transactions on Robotics*, vol. 34, no. 4, pp. 837–855, 2018.

[16]D. G. Lowe, “Distinctive Image Features from Scale-Invariant Keypoints,” *International Journal of Computer Vision*, vol. 60, no. 2, pp. 91–110, Nov. 2004.

[17]N. Dalal and B. Triggs, “Histograms of oriented gradients for human detection,” in *2005 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR 2005)*, IEEE, 2005, pp. 886–893.

[18]H. Bay, A. Ess, T. Tuytelaars, and L. Van Gool, “Speeded-up robust features (SURF),” *Computer Vision and Image Understanding*, vol. 110, no. 3, pp. 346–359, 2008.

[19]Y. LeCun, Y. Bengio, and G. Hinton, “Deep learning,” *Nature*, vol. 521, no. 7553, pp. 436–444, 2015.

[20]H. Liu, Y. Chen, R. Wang, M. Li, and Z. Li, “MFA-Deeplabv3+: an improved lightweight semantic segmentation algorithm based on Deeplabv3+,” *Complex Intelligent Systems*, vol. 11, no. 10, p. 424, Oct. 2025.

[21]S. Liang *et al.*, “Edge YOLO: Real-time intelligent object detection system based on edge-cloud cooperation in autonomous vehicles,” *IEEE Transactions on Intelligent Transportation Systems*, vol. 23, no. 12, pp. 25345–25360, 2022.

[22]X. Rui, Z. Li, X. Zhang, Z. Li, and W. Song, “A RGB-Thermal based adaptive modality learning network for day–night wildfire identification,” *International Journal of Applied Earth Observation and Geoinformation*, vol. 125, p. 103554, Dec. 2023.

[23]T. Toulouse, L. Rossi, A. Campana, T. Celik, and M. A. Akhloufi, “Computer vision for wildfire research: An evolving image dataset for processing and analysis,” *Fire Safety Journal*, vol. 92, pp. 188–194, Sep. 2017.

[24]R. Girshick, J. Donahue, T. Darrell, and J. Malik, “Rich feature hierarchies for accurate object detection and semantic segmentation,” in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2014, pp. 580–587.

[25]R. Girshick, “Fast R-CNN,” in *Proceedings of the IEEE International Conference on Computer Vision*, 2015, pp. 1440–1448.

[26]S. Ren, K. He, R. Girshick, and J. Sun, “Faster R-CNN: Towards Real-Time Object Detection with Region Proposal Networks,” Jan. 06, 2016, *arXiv*: arXiv:1506.01497.

[27]J. Redmon, S. Divvala, R. Girshick, and A. Farhadi, “You only look once: Unified, real-time object detection,” in *Proceedings of the IEEE Conference on Computer vision and Pattern Recognition*, 2016, pp. 779–788.

[28]W. Liu *et al.*, “SSD: Single Shot MultiBox Detector,” in *Computer Vision – ECCV 2016*, vol. 9905, B. Leibe, J. Matas, N. Sebe, and M. Welling, Eds., in Lecture Notes in Computer Science, vol. 9905 , Cham: Springer International Publishing, 2016, pp. 21–37.

[29]N. Carion, F. Massa, G. Synnaeve, N. Usunier, A. Kirillov, and S. Zagoruyko, “End-to-End Object Detection with Transformers,” in *Computer Vision – ECCV 2020*, vol. 12346, A. Vedaldi, H. Bischof, T. Brox, and J.-M. Frahm, Eds., in Lecture Notes in Computer Science, vol. 12346. , Cham: Springer International Publishing, 2020, pp. 213–229.

[30]“Brief summary of YOLOv8 model structure · Issue #189 · ultralytics/ultralytics · GitHub.” Accessed: Apr. 19, 2025. [Online].

[31]J. Shen, Y. Chen, Y. Liu, X. Zuo, H. Fan, and W. Yang, “ICAFusion: Iterative cross-attention guided feature fusion for multispectral object detection,” *Pattern Recognition*, vol. 145, p. 109913, Jan. 2024.

[32]T. Baltrušaitis, C. Ahuja, and L.-P. Morency, “Multimodal machine learning: A survey and taxonomy,” *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 41, no. 2, pp. 423–443, 2018.

[33]R. Koenker, *Quantile Regression*. Cambridge University Press, 2005.

[34]J. Hu, L. Shen, and G. Sun, “Squeeze-and-excitation networks,” in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2018, pp. 7132–7141.

[35]H. Huang *et al.*, “Channel prior convolutional attention for medical image segmentation,” *Computers in Biology and Medicine*, vol. 178, p. 108784, Aug. 2024.

[36]R. R. Selvaraju, M. Cogswell, A. Das, R. Vedantam, D. Parikh, and D. Batra, “Grad-CAM: Visual Explanations from Deep Networks via Gradient-Based Localization,” presented at the *Proceedings of the IEEE International Conference on Computer Vision*, 2017, pp. 618–626.

Appendix

GitHub link of the image registration algorithm:

https://github.com/HuajunDong/Visible-infrared_Image_Registration

YouTube link of the outdoor experimental video:

https://www.youtube.com/watch?v=1xUFH-iEu0k

Publications

The following is a list of academic papers I have authored or co-authored during my Master's program.

- **H. Dong**, Q. Qin, E. Dilfanian, Y. Fu, X. Wu, Y. Zhang, "Dynamic Programming-Based Multi-Spot Path Planning and LQR Control for Autonomous UAV Firefighting," in *IECON 2025 - 51st Annual Conference of the IEEE Industrial Electronics Society*, Madrid, Spain: IEEE, Oct. 2025, pp. 1-6.
- **H. Dong**, Y. Fu, Y Zhang, L Qiao, E. Dilfanian, X. Wu, Q. Qin, "Dual-Modality Wildfire Detection with Visible and Infrared Images from UAVs," in *2025 IEEE 19th International Conference on Control & Automation (ICCA)*, Tallinn, Estonia, 2025, pp. 841-846.
- E. Dilfanian, **H. Dong**, Y. Zhang, H. Benzerrouk, H. Guiddir, "Depth-Homography Registration Framework and YOLOv8n-Coordinate Attention Forest Fire Detection for Visible–Infrared UAV Imagery," in *IECON 2025 - 51st Annual Conference of the IEEE Industrial Electronics Society*, Spain: IEEE, Oct. 2025, pp. 1-6.
- X. Wu, Y. Fu, L. Qiao, **H. Dong**, Q. Qin, E. Dilfanian, A. Taherzadeh, Y. Zhang, H. Benzerrouk, H. Guiddir, "An Intelligent Algorithm for Determining Optimal Wildfire Suppression Zone Using UAVs," in *2025 IEEE 19th International Conference on Control & Automation (ICCA)*, Tallinn, Estonia, 2025, pp. 704--709.
- E. Dilfanian, Y. Zhang, **H. Dong**, L. Qiao, A. Taherzadeh, X. Wu, H. Benzerrouk, and H. Guiddir, "Autonomous Vision-Guided High-Precision Firefighting using Unmanned Aerial Vehicles," in *IECON 2024 - 50th Annual Conference of the IEEE Industrial Electronics Society*, Chicago, IL, USA: IEEE, Nov. 2024, pp. 1--6.
- L. Qiao, Y. Fu, **H. Dong**, Q. Qin, Y. Zhang, X. Wu, E. Dilfanian, A. Taherzadeh, H. Benzerrouk, H. Guiddir, and H. Murray, "Grad-CAM for Network Models: To Support Aerial Vision Based Wildfire Perception," in *IECON 2024 - 50th Annual Conference of the IEEE Industrial Electronics Society*, Chicago, IL, USA: IEEE, Nov. 2024, pp. 1-6.