VIETNAM NATIONAL UNIVERSITY - HO CHI MINH CITY
UNIVERSITY OF INFORMATION TECHNOLOGY

FINAL REPORT
IMAGE PROCESSING AND APPLICATIONS

Topic:
Wildfire Detection and Alert System Using Drones

Group Members:
Phat Nguyen Cong - 23521143 - Leader
An Nguyen Xuan - 23520023
An Truong Hoang Thanh - 23520032
Cuong Vu Viet - 23520213

Lecturer: MSc. Thang Cap Pham Dinh
Class: CS406.Q11

January 5, 2026

Contents

Work Contribution and Role Assignment

Member	Student ID	Contribution	Roles	Task Assignment
Phat Nguyen Cong	23521143	30%	Group Leader	Project management, designing overall system architecture, implementing loss functions, and writing final report.
An Nguyen Xuan	23520023	25%	Member	Dataset acquisition, implementing data augmentation pipeline, and performing exploratory data analysis (EDA).
An Truong Hoang Thanh	23520032	25%	Member	Model implementation (U-Net), hyperparameter tuning, training process management, and quantitative evaluation.
Cuong Vu Viet	23520213	20%	Member	Developing Gradio demo application, visualization of segmentation results, and preparing technical documentation.

Note: The contribution scores were collectively agreed upon by all team members based on the workload and technical impact of each assigned task.

Supervisor's Evaluation and Feedback

Ho Chi Minh City, January 2026
Lecturer

MSc. Thang Cap Pham Dinh

Chapter 1

Introduction

Wildfires have become an increasingly frequent and destructive natural disaster across the globe, causing severe environmental, economic, and human losses. The early detection of wildfires plays a crucial role in minimizing their impact by enabling faster response and containment efforts. However, traditional wildfire monitoring systems often rely on ground-based sensors or satellite imagery, which may suffer from delayed detection, limited resolution, and difficulty in covering remote or mountainous areas in a timely and efficient manner.

In response to these limitations, this project explores an innovative solution that leverages the power of unmanned aerial vehicles (UAVs), also known as drones, and computer vision technologies. Specifically, the system is designed to use RGB and thermal imagery captured by drones to detect signs of wildfire activity. These images are then processed using a deep learning-based semantic segmentation model, which can classify and segment areas affected by fire with high accuracy. Additionally, by integrating GPS data, the system can pinpoint the exact geographic location of detected wildfires, providing critical information for emergency services.

Although the current implementation is a proof-of-concept, the project demonstrates the potential of combining drone mobility with modern AI capabilities to improve wildfire monitoring. This interdisciplinary approach not only enhances the responsiveness and scalability of fire detection systems but also sets the foundation for future autonomous, real-time disaster response technologies. Ultimately, the project aims to contribute to safer and more proactive wildfire management strategies in the face of a changing climate and growing environmental challenges.

Chapter 2

Related Works

2.1 Wildfire Detection Using Aerial Imagery

Early approaches to wildfire detection using aerial imagery primarily relied on handcrafted features and rule-based techniques. These methods typically exploited color thresholds in the RGB color space, motion cues from video sequences, and texture descriptors to identify flame or smoke regions [6]. Although computationally efficient, such approaches are highly sensitive to illumination changes, camera motion, and complex backgrounds, which often leads to high false positive rates in real-world environments.

With the advancement of deep learning, recent studies have shifted toward convolutional neural network (CNN)-based methods for wildfire detection using UAV imagery [1]. Frame-level classification models have been proposed to distinguish fire and non-fire scenes with improved robustness compared to traditional methods. Beyond classification, encoder–decoder architectures such as U-Net and its variants have been widely adopted for pixel-level wildfire segmentation [4]. These semantic segmentation models enable precise localization of fire regions, making them more suitable for downstream tasks such as fire spread analysis and alert generation [2].

Several works have demonstrated that deep learning-based segmentation significantly outperforms handcrafted approaches, especially under challenging conditions such as varying illumination and partial occlusions. However, many early deep learning solutions focus on offline analysis and do not fully address the constraints of real-time deployment on UAV platforms.

2.2 FLAME Dataset

The FLAME (Fire Luminosity Airborne-based Machine learning Evaluation) dataset is a publicly available aerial imagery dataset specifically designed for wildfire detection using unmanned aerial vehicles [3]. The dataset consists of RGB images and video sequences captured by drones during prescribed pile-burn operations in forest environments. Each image is accompanied by pixel-level fire segmentation masks, enabling both classification and semantic segmentation tasks.

FLAME has become a widely used benchmark in recent wildfire detection research due to its realistic UAV viewpoints and high-quality annotations. Several studies have evaluated CNN-based classifiers and U-Net-based segmentation models on this dataset, reporting strong performance in detecting fire regions under diverse environmental conditions. The availability of ground-truth masks makes FLAME particularly suitable for developing and evaluating pixel-level wildfire segmentation methods.

2.3 State-of-the-Art Methods

State-of-the-art wildfire detection systems increasingly adopt multi-stage pipelines to balance detection accuracy and computational efficiency. Recent works have explored object detection frameworks such as YOLO and Faster R-CNN for fast wildfire detection in UAV videos [7,8]. These methods offer high inference speed and are suitable for onboard deployment but typically provide only coarse localization through bounding boxes. To address this limitation, semantic segmentation models such as U-Net have been employed to achieve fine-grained fire region delineation.

Additionally, several studies have investigated multi-modal fusion techniques by combining RGB and thermal imagery to improve robustness against visual ambiguity [9, 10]. Temporal modeling and persistence-based filtering have also been introduced to reduce false alarms caused by transient visual artifacts. In contrast, recent research trends emphasize end-to-end wildfire monitoring systems that integrate segmentation, temporal consistency analysis, and alert logic to support real-time decision-making.

Chapter 3

Project Overview

3.1 Problem Definition

This project focuses on developing a real-time wildfire detection and alert system using unmanned aerial vehicles (UAVs) within the scope of *Image Processing and Applications* course. The system applies deep learning-based semantic segmentation techniques to aerial imagery in order to accurately identify wildfire regions and support timely emergency response.

3.1.1 Inputs

The system receives the following inputs:

- Real-time RGB video stream captured by the drone-mounted camera.
- Real-time infrared (thermal) video stream synchronized with the RGB stream to enable pixel-level alignment.
- Real-time drone metadata, including GPS coordinates (latitude, longitude, altitude) and camera orientation parameters (yaw, pitch, and roll), synchronized with each video frame.

3.1.2 Outputs

The system generates the following outputs:

- RGB video frames overlaid with predicted wildfire segmentation masks.
- Estimated geographic coordinates of detected wildfire regions, represented by the centroid of each segmented area.
- A real-time fire alert signal triggered upon wildfire detection.

3.1.3 Constraints

The system operates under the following constraints:

- Video input must be processed in real time, encoded in H.264 format (`.MP4` or `.MOV`), with a frame rate of 24–30 FPS and a minimum resolution of 512 × 512 pixels.
- The drone camera field of view (FOV) must be between 60° and 90°.
- Drone altitude must range from 50 to 250 meters.
- Drone speed must be maintained between 15 and 25 km/h.
- GPS operation is restricted to predefined forest regions.

3.1.4 Requirements

The system is designed to meet the following requirements:

- Wildfire regions must be clearly highlighted on the video stream for immediate visual interpretation.
- The estimated fire location should be accurate enough to support rapid emergency response.
- Fire alerts must be issued promptly after wildfire detection to minimize response latency.
- The system should prioritize high recall to avoid missing real wildfire events, while tolerating a limited number of false alarms.

3.2 Visual Pattern Modeling for Wildfire Detection

Visual pattern modeling serves as a core component in distinguishing wildfire signatures from complex background elements like vegetation and man-made structures. Leveraging convolutional neural networks, the system extracts discriminative features from RGB and thermal imagery, focusing on the high-intensity chromatic distributions and irregular, dynamic boundaries of flames and smoke. Beyond static appearance, the model captures the spatial growth behavior of wildfires—where small ignition points evolve into coherent thermal clusters—and maintains temporal consistency across consecutive video frames to suppress transient false positives such as glare or lens artifacts. By employing multimodal fusion, the system cross-validates visually salient regions in the RGB spectrum with elevated radiometric signatures in the thermal domain. This integrated approach ensures robust, high-confidence detection by prioritizing persistent spatio-temporal patterns over environmental noise.

3.3 Feature Abstraction and Representation Learning

Feature abstraction is a fundamental concept in image processing and deep learning, referring to the process of learning compact and discriminative representations from raw pixel data. In this project, abstraction is achieved through deep convolutional layers that progressively transform input images into high-level semantic features relevant to wildfire detection.

Wildfire regions are abstracted as binary segmentation masks, where pixels are classified as either fire or background. This representation simplifies complex visual scenes and enables efficient pixel-level analysis. High-level features emphasize key attributes such as color intensity, spatial coherence, temporal persistence, and thermal consistency, while suppressing irrelevant background details.

By focusing on these abstracted representations, the model generalizes effectively across different terrains, lighting conditions, and flight scenarios. This abstraction not only improves detection performance but also reduces computational overhead, making the system suitable for real-time UAV-based image processing applications.

Chapter 4

The Proposed Approach

4.1 Overall Architecture

The proposed system is architected as a multi-stage image processing pipeline, specifically optimized for real-time wildfire surveillance via UAVs. The framework is decomposed into three primary functional modules: **Data Preprocessing**, **Fire Detection and Localization**, and **Alert Generation**. The data flow and interaction between these modules are illustrated in Figure 4.1.

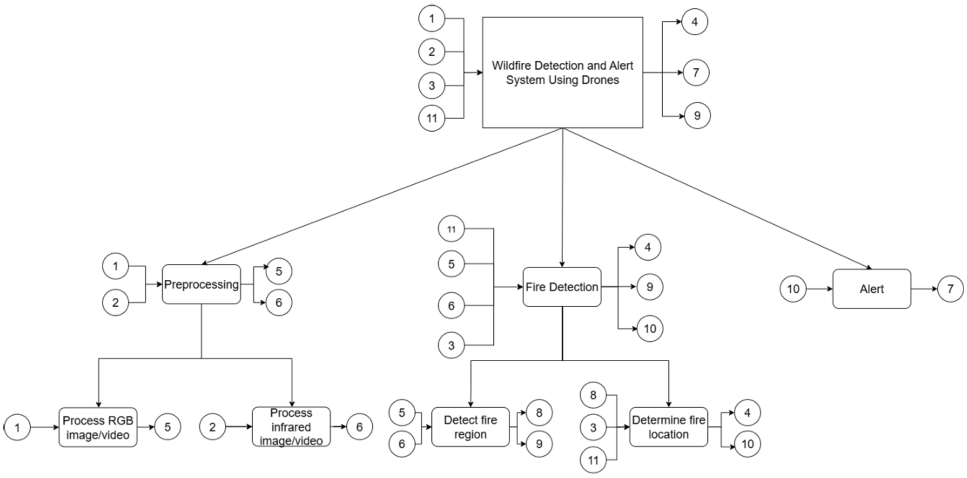


Figure 4.1: Overall modular architecture of the Wildfire Detection and Alert System.

4.1.1 Legend of Data Flow:

1. RGB image/video from drone
2. Infrared (thermal) image/video from drone
3. Drone GPS coordinates (lat, lon, alt)
4. GPS coordinates of detected fire (Output)
5. Preprocessed RGB image/video

6. Preprocessed infrared image/video
7. Fire alert signal (Output)
8. Binary segmentation mask (Raw)
9. RGB frames overlaid with segmentation mask
10. Alert confidence score
11. Drone's metadata

4.1.2 Summary of Functional Workflow

The following table summarizes the interaction between the data components (as defined in the Legend) and the processing modules:

Processing Module	**Input Component (IDs)**	**Output Component (IDs)**
Preprocessing	1, 2	5, 6
Detection (U-Net)	5, 6	8, 9
Geospatial Localization	3, 8, 11	4, 10
Alert Generation	4, 9, 10	7

Table 4.1: Data flow mapping between system modules.

4.2 System Algorithm

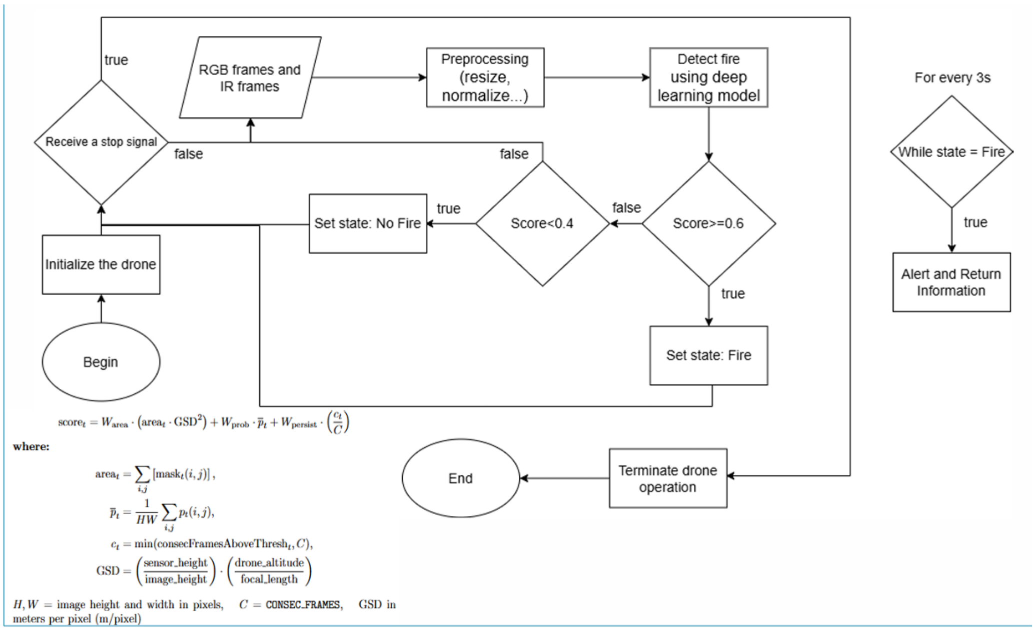
The operational logic of the wildfire detection system is governed by two concurrent processes: a high-frequency **Detection Loop** for real-time image analysis and a low-latency **Alert Loop** for decision making and dispatching. This dual-loop architecture ensures both responsiveness to active threats and stability against environmental noise.

4.2.1 Detection and Alert Logic via Hysteresis

To enhance system stability against minor fluctuations in the model's prediction confidence (*score*), a **Hysteresis Thresholding** mechanism is implemented. Instead of using a single fixed threshold (e.g., 0.5) which often leads to "flickering" alerts where the state rapidly toggles between detection and non-detection, our system utilizes a state-transition dual-threshold:

$$State_t = \begin{cases} FIRE & \text{if } score \geq 0.6 \\ NO\_FIRE & \text{if } score < 0.4 \\ State_{t-1} & \text{otherwise} \end{cases} \tag{4.1}$$

As illustrated above, the state transitions to `FIRE` only when confidence is high (≥ 0.6) and remains in that state until the confidence drops significantly (< 0.4). This approach ensures that the alert state remains consistent unless a significant change in detection probability occurs, effectively filtering out transient false positives and signal noise.



$$\text{score}_t = W_{\text{area}} \cdot \left(\text{area}_t \cdot \text{GSD}^2\right) + W_{\text{prob}} \cdot \overline{p}_t + W_{\text{persist}} \cdot \left(\frac{c_t}{C}\right)$$

where:

$$\text{area}_t = \sum_{i,j} [\text{mask}_t(i,j)]\,,$$

$$\overline{p}_t = \frac{1}{HW} \sum_{i,j} p_t(i,j),$$

$$c_t = \min(\text{consecFramesAboveThresh}_t, C),$$

$$\text{GSD} = \left(\frac{\text{sensor\_height}}{\text{image\_height}}\right) \cdot \left(\frac{\text{drone\_altitude}}{\text{focal\_length}}\right)$$

H, W = image height and width in pixels, C = CONSEC_FRAMES, GSD in meters per pixel (m/pixel)

Figure 4.2: Flowchart of the dual-loop wildfire detection and alerting algorithm.

4.2.2 Pseudocode Representation

The following pseudocode outlines the implementation of the concurrent dual-loop architecture:

```
# Initialization
initialize_drone_systems()
global_state = NO_FIRE
start_concurrent_thread(alert_loop)

# Main Detection Loop
while not receive_stop_signal():
    rgb_raw, ir_raw = capture_synchronized_frames()

    # Preprocessing
    rgb_proc = resize_and_normalize(rgb_raw)
    ir_proc = calibrate_thermal(ir_raw)

    # Inference
    inference_score = model.predict(rgb_proc, ir_proc)

    # State Management using Hysteresis
    if inference_score >= 0.6:
        global_state = FIRE
    elif inference_score < 0.4:
        global_state = NO_FIRE
    # else: maintain current global_state

# Alert Loop (Runs Concurrently every 3 seconds)
def alert_loop():
    while True:
        if global_state == FIRE:
            # Aggregate geospatial data and segmentation overlays
            fire_location = estimate_geospatial_coords()
            visual_overlay = generate_mask_overlay(rgb_raw)

            # Dispatch alert package
            send_alert(fire_location, visual_overlay, inference_score)
            sleep(3.0)
        else:
            sleep(0.1) # Idle polling
```

Listing 4.1: Wildfire Detection and Concurrent Alerting Algorithm

4.3 Module 1: Multimodal Data Preprocessing

4.3.1 Theoretical Background and Rationale

Multimodal data preprocessing is a fundamental stage in the image processing pipeline that conditions raw sensor data for deep learning inference. The core theoretical framework encompasses three strategic domains:

- **Spatial Resampling and Intensity Normalization:** High-resolution aerial imagery inherent in UAV missions often contains redundant spatial information. Resizing images to a standardized dimension (**Resize** to 512×512) ensures computational tractability and batch consistency. Furthermore, pixel intensity normalization stabilizes gradient descent during backpropagation by ensuring a zero-centered feature distribution.
- **Stochastic Data Augmentation:** Based on the principle of maximizing dataset entropy and mitigating overfitting, we implement a comprehensive stochastic pipeline via the *Albumentations* library. This simulates the stochasticity of real-world flight conditions:
  - **Geometric Invariance:** Transformations such as *HorizontalFlip*, *VerticalFlip*, *RandomRotate90*, and *Rotate* ensure the model is invariant to drone heading. *Perspective* and *RandomScale* are critical for simulating varying camera pitch angles and flight altitudes relative to the terrain.
  - **Spectral and Photometric Jittering:** To account for volatile solar irradiance and specular reflections, we apply *RandomBrightnessContrast*, *HueSaturationValue*, and *RGBShift*. This compels the network to learn robust spectral signatures of fire rather than over-relying on specific illumination intensities.
  - **Environmental and Sensor Noise Simulation:** Atmospheric haze and canopy occlusions are replicated using *RandomFog* and *RandomShadow*. Additionally, *GaussianBlur* accounts for potential motion blur or sensor-induced thermal noise.

- **Cross-modal Synchronization:** This involves the spatio-temporal alignment of visible-light (RGB) and long-wave infrared (LWIR) streams. Ensuring that thermal anomalies (heat signatures) and visual textures are fused at the pixel level is paramount for multi-modal feature integration.

4.3.2 Functional Role and Systemic Contribution

Module 1 serves as the critical hardware-to-software interface, transforming raw sensor signals into optimized neural tensors. Its contributions are summarized as follows:

- **Radiometric Refinement and Noise Mitigation:** By applying denoising filters (e.g., *Gaussian Blur*) and radiometric calibration, the module suppresses sensor-induced artifacts and environmental haze, providing the U-Net architecture with high-fidelity, high-frequency features.

- **Structural Feature Standardization:** It homogenizes heterogeneous data streams into synchronized tensors, significantly reducing the latent space complexity for the subsequent semantic segmentation module.

- **Enhancing Model Generalization and Resilience:** The rigorous application of geometric and photometric transformations directly bolsters the system's operational resilience. This ensures high *Intersection over Union* (IoU) and *Recall* performance despite unpredictable environmental variables such as varying sunlight reflections, smoke occlusions, or complex mountainous topographies.

4.4 Module 2: Fire Detection and Localization

4.4.1 Functional Role and Contribution

Module 2 serves as the computational core of the proposed system, where high-level semantic abstractions are extracted from synchronized multi-modal inputs. The theoretical framework of this module integrates **Deep Semantic Segmentation** via Convolutional Neural Networks (CNNs) and **Geospatial Photogrammetry** for inverse projection.
Unlike traditional object detection which yields coarse bounding boxes, pixel-level segmentation is imperative for wildfire analysis. It enables the precise estimation of fire perimeters, active surface areas, and propagation vectors. This module functions as the primary decision-making engine, transforming preprocessed numerical tensors into actionable spatial intelligence for emergency response.

4.4.2 Pixel-level Fire Segmentation

U-Net Model Architecture

We implement the **U-Net architecture**, a specialized symmetric encoder-decoder network. Originally designed for medical imaging, its structural properties make it exceptionally effective for environmental monitoring where fine-grained detail is critical.

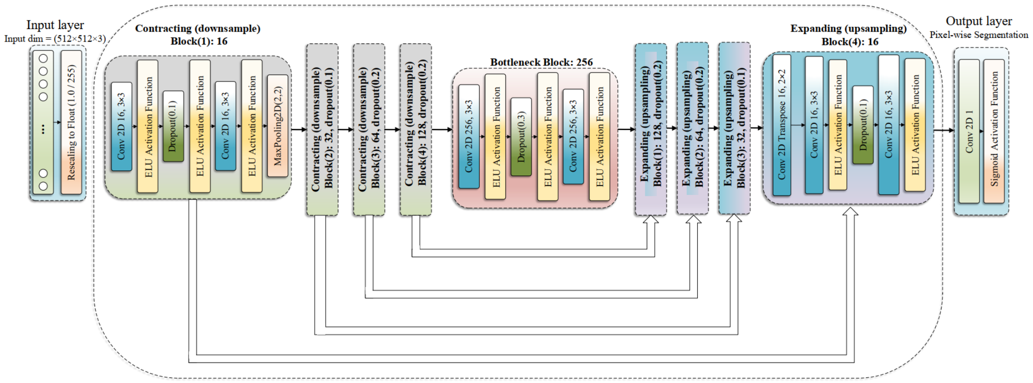


Figure 4.3: U-Net architecture for semantic segmentation, featuring a contracting path (encoder), an expansive path (decoder), and skip connections.

The architecture is characterized by two distinct paths:

- **The Contracting Path (Encoder):** This path follows the typical architecture of a convolutional network. It utilizes repeated applications of 3 × 3 convolutions, followed by Rectified Linear Units (ReLU) and 2 × 2 max-pooling for downsampling. Each downsampling step doubles the number of feature channels, capturing complex hierarchical patterns and semantic context.

- **The Expansive Path (Decoder):** Every step in the decoding path consists of an upsampling of the feature map followed by a 2 × 2 "up-convolution" that halves the number of feature channels. This reconstructs the spatial resolution of the fire signatures.

- **Skip Connections:** The hallmark of U-Net is the concatenation of high-resolution features from the contracting path directly into the expansive path. This mechanism compensates for the spatial information loss inherent in pooling layers, allowing the decoder to recover fine-grained details of irregular fire boundaries.

Rationale and Comparative Analysis

Rationale for Selection: U-Net was selected because wildfires in aerial footage often appear as small, fragmented clusters. Standard Fully Convolutional Networks (FCNs) often lose these small-scale features during downsampling, whereas U-Net's skip connections maintain high-fidelity edge detection.

- **Strengths:**
  - **Spatial Precision:** Superior boundary delineation for irregular flame and smoke shapes.
  - **Data Efficiency:** Demonstrates high performance on specialized datasets like FLAME by maximizing spatial context utilization.
  - **Real-time Capability:** The architecture supports deterministic inference times, essential for onboard UAV processing.
- **Weaknesses and Mitigations:**

- **Memory Overhead:** The deep skip connections can be memory-intensive. To mitigate this, we utilize a lightweight backbone (we employ a ResNet-34 backbone to balance representational capacity and training stability) to ensure compatibility with edge devices like the NVIDIA Jetson Nano.

4.4.3 Geospatial Localization Mathematics

To transform the 2D image-space information (binary mask M) into precise geographic-space coordinates, the system employs an inverse projection model based on the **Pinhole Camera** geometry. This process factors in the drone's altitude (H), camera focal length (f), and spatial orientation (θ, ϕ).

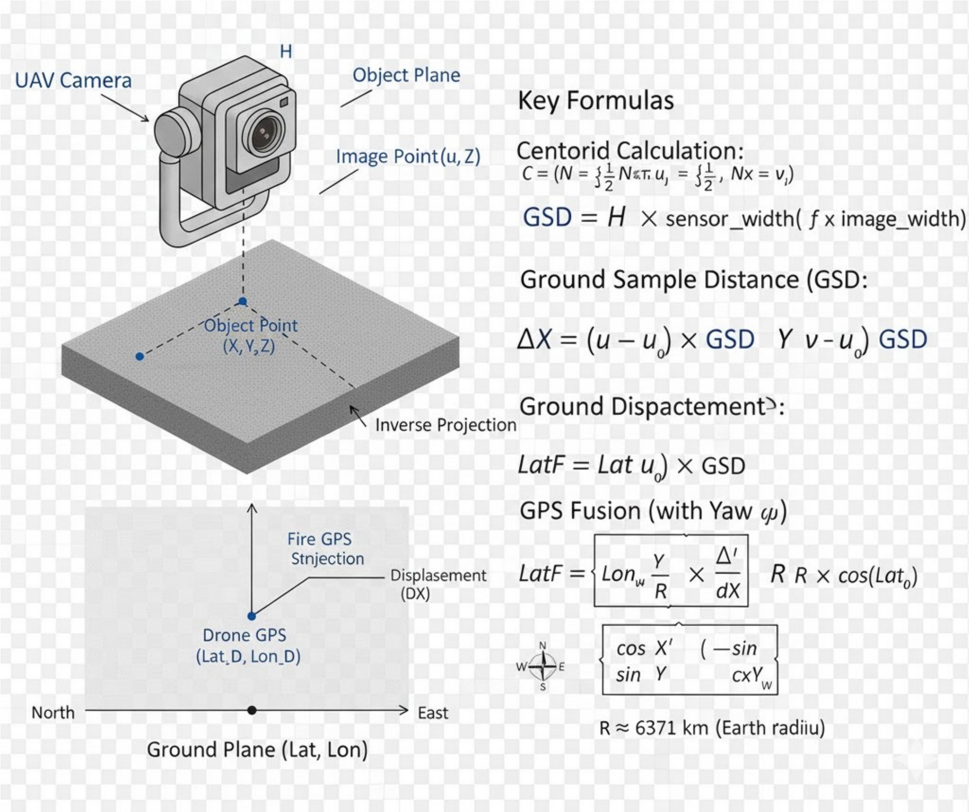


Figure 4.4: Pinhole camera model and back-projection geometry for ground coordinate estimation.

The mathematical framework is executed through the following stages:

- **Centroid Calculation:** The system first identifies the geometric center of the detected wildfire region. The centroid $C(u, v)$ is calculated by averaging the coordinates of all pixels classified as fire (N):

$$C = \left(\frac{1}{N} \sum_{i=1}^{N} u_i, \frac{1}{N} \sum_{i=1}^{N} v_i \right) \tag{4.2}$$

- **Ground Mapping via Ground Sample Distance (GSD):** As illustrated in Figure 4.4, the pixel displacement is converted into metric distance on the ground.

The **GSD**, representing the real-world size of a single pixel, is defined as:

$$GSD = \frac{H \times \text{sensor\_width}}{f \times \text{image\_width}} \tag{4.3}$$

- **Coordinate Transformation and Fusion:**
  - **Rotation Transformation:** The relative metric offsets from the image center $(\Delta X, \Delta Y)$ are rotated based on the UAV's **Yaw** angle to align with the North-East-Down (NED) geographic coordinate system.
  - **GPS Fusion:** The calculated metric displacements are converted into decimal degrees and fused with the drone's current position (Lat_D, Lon_D). This yields the absolute geographic coordinates (Output 4) of the wildfire.

This integrated mathematical approach ensures that wildfire localization remains accurate and persistent, regardless of the UAV's flight maneuvers or altitude changes.

4.4.4 Multi-Objective Loss Function

To address the severe class imbalance and ambiguous fire boundaries in aerial wildfire imagery, we adopt a **Multi-Objective Loss** that combines Focal Loss, Dice Loss, and Soft Binary Cross-Entropy (Soft BCE). This composite loss encourages robust region overlap, emphasizes hard-to-classify fire pixels, and stabilizes training.
The overall loss function is defined as:

$$\mathcal{L}_{total} = \lambda_1 \mathcal{L}_{Focal} + \lambda_2 \mathcal{L}_{Dice} + \lambda_3 \mathcal{L}_{SoftBCE}$$

where λ_1, λ_2, and λ_3 are weighting coefficients.

4.5 Module 3: Decision Logic and Alert Generation

4.5.1 Functional Role and Contribution

Module 3 acts as the final administrative layer of the entire image processing pipeline. While the previous modules focus on data preparation and feature extraction, this module is responsible for high-level semantic verification and system-to-human communication.

- **Role:** Its primary role is to serve as a logical filter that prevents "alert fatigue" by ensuring that only confirmed wildfire events reach the emergency response units. It transforms raw probability scores and binary masks into standardized, actionable intelligence packages.

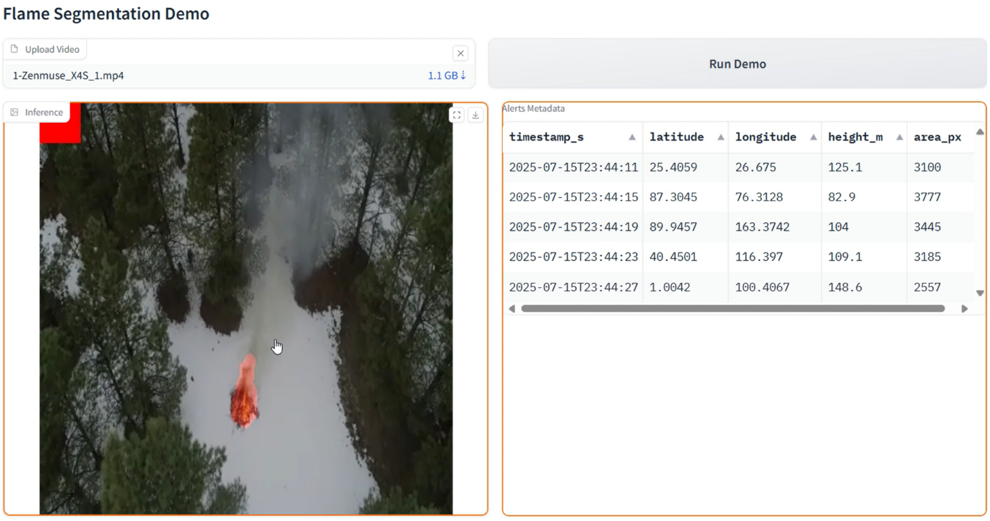
- **Contribution to the General Model:** This module bridges the gap between deep learning outputs and real-world emergency management. By introducing temporal validation, it significantly reduces the False Discovery Rate (FDR) of the model. Furthermore, it encapsulates multi-modal data (GPS, Timestamps, Visual Overlays) into a unified JSON telemetry format, which is essential for the interoperability of the system in modern disaster response infrastructures.

4.5.2 Technical Implementation

The Alert Module functions as the final decision-making gateway of the processing pipeline. It aggregates the localized segmentation data and synchronized metadata to verify detection reliability through a rigorous validation process before dispatching emergency signals.

- **Spatio-temporal Consistency Filtering:** To mitigate false positives triggered by transient visual artifacts—such as specular solar glare, lens flares, or sensor noise—the module implements a temporal persistence filter. A wildfire event is only validated if the detected fire pixels maintain spatial consistency across a predefined window of consecutive frames (N frames). This ensures that only persistent thermal and visual signatures trigger the alert system.

- **Dynamic Alert Dispatching:** Once the integrated **Alert Confidence Score** exceeds the calibrated safety threshold, the system autonomously generates a **Fire Alert Signal (7)**. This telemetry package encompasses the precise GPS coordinates of the fire's centroid, an RGB snapshot overlaid with the semantic segmentation mask, and a high-precision ISO-8601 timestamp to facilitate immediate tactical intervention by emergency response units.



timestamp_s	latitude	longitude	height_m	area_px
2025-07-15T23:44:11	25.4059	26.675	125.1	3100
2025-07-15T23:44:15	87.3045	76.3128	82.9	3777
2025-07-15T23:44:19	89.9457	163.3742	104	3445
2025-07-15T23:44:23	40.4501	116.397	109.1	3185
2025-07-15T23:44:27	1.0042	100.4067	148.6	2557

Figure 4.5: Final system output: RGB frame with semantic segmentation mask overlay and real-time geospatial alert metadata.

The implementation of spatio-temporal filtering significantly enhances the system's robust precision, effectively neutralizing the impact of non-fire radiometric anomalies. As demonstrated in Figure 4.5, the final output provides an intuitive and actionable visualization for emergency operators. By fusing pixel-level segmentation with global positioning data, the system achieves its primary objective: delivering high-fidelity, real-time situational awareness for proactive wildfire management.

Chapter 5

EXPERIMENTS

5.1 Evaluation Metrics

To rigorously assess the effectiveness of the proposed wildfire detection system, we employ a multi-faceted evaluation strategy. The metrics are selected to validate three critical performance dimensions: pixel-level segmentation precision, alert reliability, and geospatial localization accuracy.

5.1.1 Pixel-wise Segmentation Quality (IoU)

The primary metric for evaluating the U-Net segmentation model is the **Intersection over Union (IoU)**, also known as the Jaccard Index. This metric quantifies the spatial overlap between the predicted fire region (A) and the ground-truth mask (B).

Mathematically, IoU represents the ratio of the intersection area to the union area of the two masks. In the context of pixel-wise binary segmentation, it is calculated using the components of the confusion matrix:

$$IoU = \frac{\text{Area of Overlap}}{\text{Area of Union}} = \frac{|A \cap B|}{|A \cup B|} = \frac{TP}{TP + FP + FN} \tag{5.1}$$

Where the components are defined as follows:

- **Area of Overlap ($A \cap B$ / TP):** The total count of True Positive pixels, which are fire pixels correctly identified by the model.
- **Area of Union ($A \cup B$):** The total spatial extent covered by both the prediction and the ground truth. This is calculated as the sum of:
  - **TP (True Positives):** Correct fire detections.
  - **FP (False Positives):** Background pixels incorrectly predicted as fire.
  - **FN (False Negatives):** Actual fire pixels missed by the model.

A high IoU indicates that the highlighted fire regions on the RGB frames align closely with reality, providing reliable visual evidence for drone operators. For this system, we target a minimum IoU of **0.60** to ensure operational viability and high-fidelity mapping.

5.1.2 Alert Robustness and Safety Bias (F2-Score)

In wildfire surveillance, the cost of a *False Negative* (missing an active fire) far outweighs that of a *False Positive* (false alarm). Therefore, we utilize the **F2-Score**, a variant of the F-beta score that places greater statistical weight on **Recall** than on **Precision**:

$$F_\beta = (1 + \beta^2) \cdot \frac{\text{Precision} \cdot \text{Recall}}{(\beta^2 \cdot \text{Precision}) + \text{Recall}} \tag{5.2}$$

By setting $\beta = 2$, we prioritize the system's ability to detect all potential ignition points. The underlying components are defined as:

- **Precision** ($\frac{TP}{TP+FP}$): The accuracy of the "fire" predictions.
- **Recall** ($\frac{TP}{TP+FN}$): The sensitivity of the system in capturing all fire pixels.

Our objective is to maintain an F_2-Score of at least 0.75 to ensure maximum safety.

5.1.3 Geospatial Localization Accuracy (Haversine Distance)

Beyond image space, the system's ability to guide first responders depends on geographic precision. We measure the error between the estimated centroid coordinates and the ground-truth GPS location using the **Haversine formula**.

Unlike the standard Euclidean distance which assumes a flat plane, the Haversine formula accounts for the Earth's curvature, providing a more reliable metric for long-distance UAV missions.

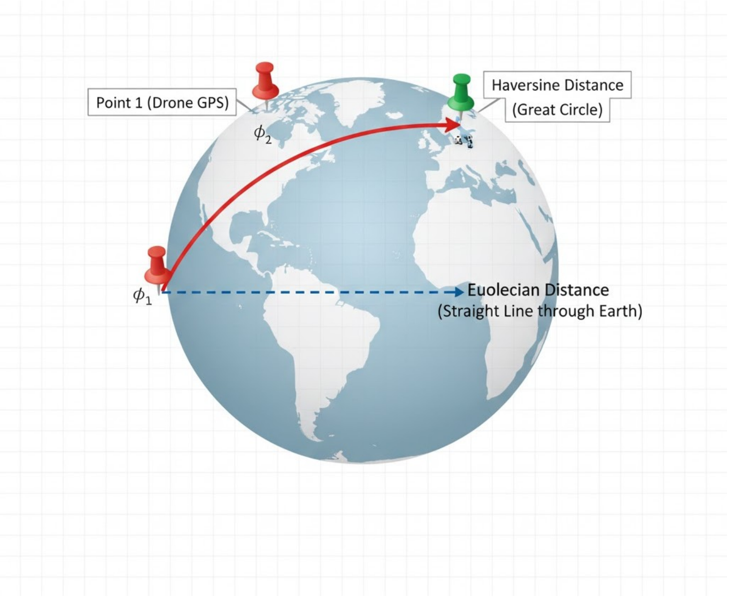

Point 1 (Drone GPS)
ϕ_2
Haversine Distance (Great Circle)
ϕ_1
Euolecian Distance (Straight Line through Earth)

Figure 5.1: Geometric comparison: Haversine distance (Great Circle) follows the Earth's curvature, whereas Euclidean distance represents a straight line through the sphere.

The distance d is calculated as follows:

$$d = 2r \cdot \arcsin\left(\sqrt{\sin^2\left(\frac{\Delta\phi}{2}\right) + \cos(\phi_1)\cos(\phi_2)\sin^2\left(\frac{\Delta\lambda}{2}\right)}\right) \tag{5.3}$$

Where:

- ϕ_1, ϕ_2: Latitudes of the ground truth and predicted points.
- λ_1, λ_2: Longitudes of the ground truth and predicted points.
- r: Earth's radius ($\approx$ 6,371 km).

Using this spherical trigonometric approach, the average localization error is kept under **50 meters**, ensuring responders are guided to the precise ignition point.

5.1.4 Overall Classification Accuracy

Finally, we monitor the **Global Pixel Accuracy**, which represents the ratio of correctly classified pixels (both fire and background) to the total number of pixels:

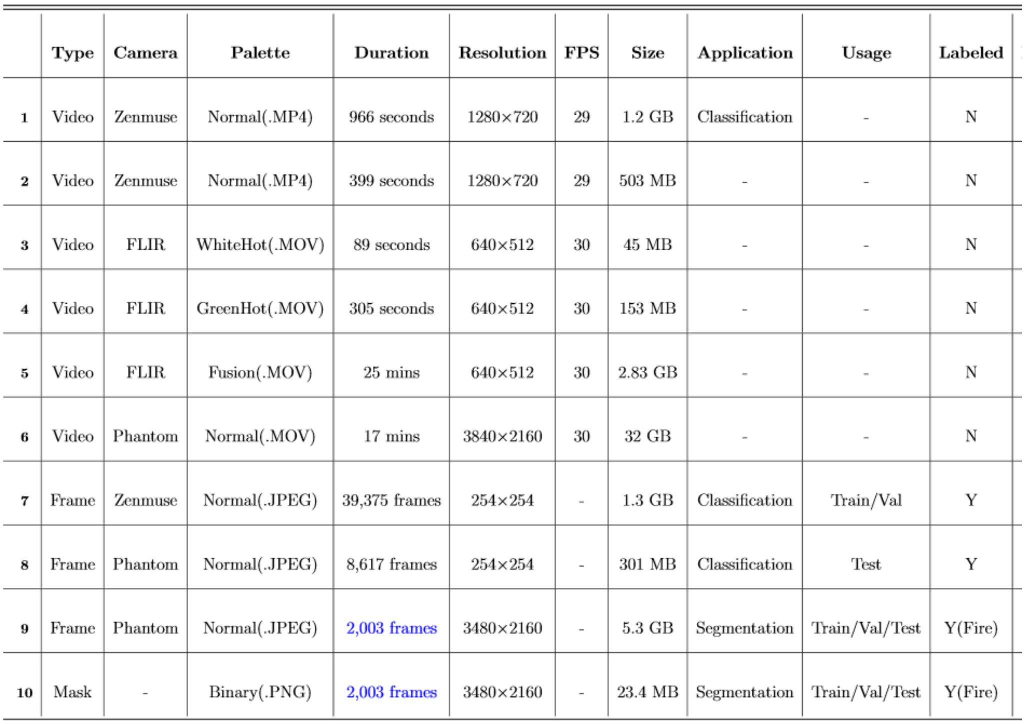
$$\text{Accuracy} = \frac{TP + TN}{TP + TN + FP + FN} \tag{5.4}$$

While Accuracy provides a general overview, it is used as a secondary metric due to the significant class imbalance between small fire regions and the expansive forest background.

5.2 Dataset

5.2.1 Dataset Acquisition: FLAME Dataset

All evaluations in this project are conducted using the **FLAME (Fire Luminosity Airborne-based Machine learning Evaluation)** dataset, a publicly available repository hosted on IEEE DataPort. This dataset serves as a comprehensive benchmark for wildfire detection via Unmanned Aerial Vehicles (UAVs).



	Type	Camera	Palette	Duration	Resolution	FPS	Size	Application	Usage	Labeled
1	Video	Zenmuse	Normal(.MP4)	966 seconds	1280×720	29	1.2 GB	Classification	-	N
2	Video	Zenmuse	Normal(.MP4)	399 seconds	1280×720	29	503 MB	-	-	N
3	Video	FLIR	WhiteHot(.MOV)	89 seconds	640×512	30	45 MB	-	-	N
4	Video	FLIR	GreenHot(.MOV)	305 seconds	640×512	30	153 MB	-	-	N
5	Video	FLIR	Fusion(.MOV)	25 mins	640×512	30	2.83 GB	-	-	N
6	Video	Phantom	Normal(.MOV)	17 mins	3840×2160	30	32 GB	-	-	N
7	Frame	Zenmuse	Normal(.JPEG)	39,375 frames	254×254	-	1.3 GB	Classification	Train/Val	Y
8	Frame	Phantom	Normal(.JPEG)	8,617 frames	254×254	-	301 MB	Classification	Test	Y
9	Frame	Phantom	Normal(.JPEG)	2,003 frames	3480×2160	-	5.3 GB	Segmentation	Train/Val/Test	Y(Fire)
10	Mask	-	Binary(.PNG)	2,003 frames	3480×2160	-	23.4 MB	Segmentation	Train/Val/Test	Y(Fire)

Figure 5.2: Representative samples from FLAME subsets 9 and 10 used for semantic segmentation training.

For the scope of our semantic segmentation task, we specifically utilized **Subsets 9 and 10**. These subsets provide high-resolution RGB imagery paired with meticulously annotated pixel-level binary masks, capturing various prescribed fire scenarios in dense forest environments.

5.2.2 Data Partitioning and Directory Structure

To facilitate efficient data loading and ensure a clean separation between the training and evaluation phases, the extracted samples from **FLAME Subsets 9 and 10** were organized into a hierarchical directory structure.
The dataset, totaling 2,003 samples, was partitioned into three distinct sets: **train** (1,201 samples), **val** (400 samples), and **test** (402 samples) sets. Each set follows a synchronized paired-folder architecture where the raw RGB images and their corresponding binary semantic masks are stored separately but share identical filenames for indexing.

```
DatasetRoot/
 train/
    images/  (1,201 samples)
    masks/   (1,201 samples)
 val/
    images/  (400 samples)
    masks/   (400 samples)
 test/
     images/  (402 samples)
     masks/   (402 samples)
```

Figure 5.3: Hierarchical directory structure for the wildfire segmentation dataset.

Implementation Details:

- **Synchronization:** For every file in `/images/`, there is a corresponding ground-truth file in `/masks/` with an identical unique identifier (e.g., `frame_001.png`).
- **Data Integrity:** This structure allows the `PyTorch Dataset` class to iterate through the pairs consistently, ensuring that the model's loss function is always calculated against the correct spatial ground truth.
- **Subsets Source:** The images were specifically curated from Subsets 9 and 10 of the FLAME dataset to ensure high-fidelity pixel-level annotations across various prescribed fire conditions.

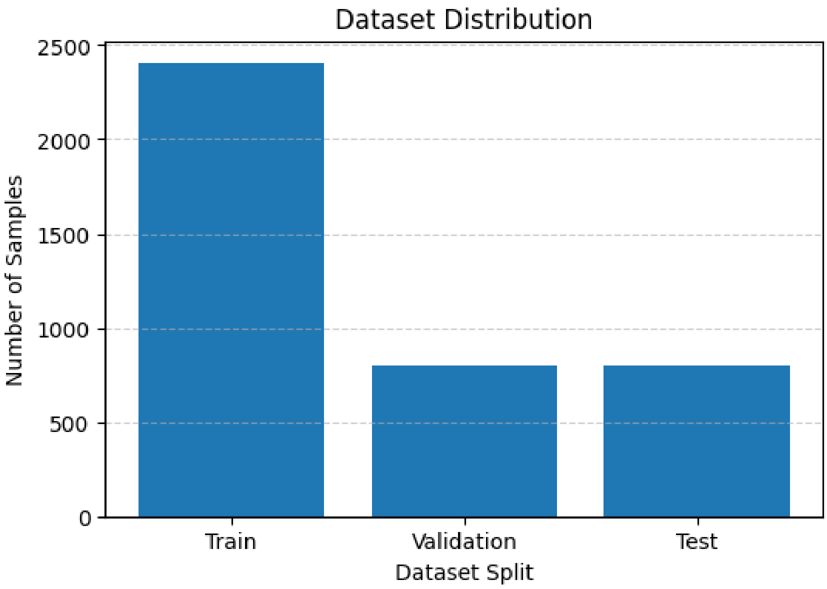


Figure 5.4: Distribution of samples across Train, Validation, and Test splits (6:2:2 ratio).

As illustrated in Figure 5.4, this distribution ensures a balanced representation of fire signatures across all phases of the machine learning pipeline, facilitating reliable generalization.

5.2.3 Data Specification and Ground Truth

Each training instance consists of a 3-channel RGB image and its corresponding 1-channel semantic mask. The ground truth (GT) masks utilize a binary representation where fire pixels are assigned a value of 1 (White) and background pixels are assigned a value of 0 (Black).

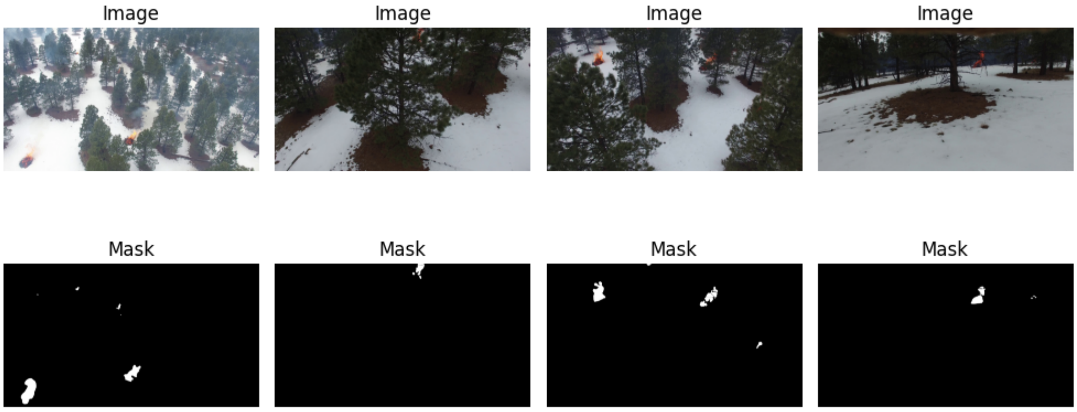


Figure 5.5: Qualitative sample from the processed dataset: (Left) Input RGB image, (Right) Binary ground truth mask.

The precise pixel-level alignment shown in Figure 5.5 validates the high fidelity of the FLAME annotations, which is fundamental for achieving high *Intersection over Union* (IoU) and *Recall* scores during the evaluation phase.

5.2.4 Stochastic Data Augmentation and Preprocessing

To improve the model's generalization capability and mitigate overfitting, we implemented a robust stochastic data augmentation pipeline using the **Albumentations** library. This process increases the diversity of the training set by simulating various environmental conditions, drone viewpoints, and sensor artifacts.

The augmentation strategy is categorized into three primary transformation groups:

- **Geometric Transformations:** To ensure the model is invariant to drone orientation and perspective, we applied *HorizontalFlip*, *VerticalFlip*, *RandomRotate90*, and *Perspective* shifts. These simulate changes in flight direction and camera angles relative to the terrain.

- **Photometric and Color Adjustments:** To handle varying lighting conditions and sunlight reflections, we utilized *RandomBrightnessContrast*, *HueSaturationValue*, and *RGBShift*. This ensures the network focuses on chromatic fire signatures rather than specific lighting intensities.

- **Environmental Noise Simulation:** Specific atmospheric effects such as smoke or haze were simulated using *RandomFog* and *GaussianBlur*, while *RandomShadow* was applied to replicate the complex occlusion patterns found in dense forest canopies.

The final input tensors were resized to a uniform resolution of 512 × 512 pixels to ensure computational consistency.

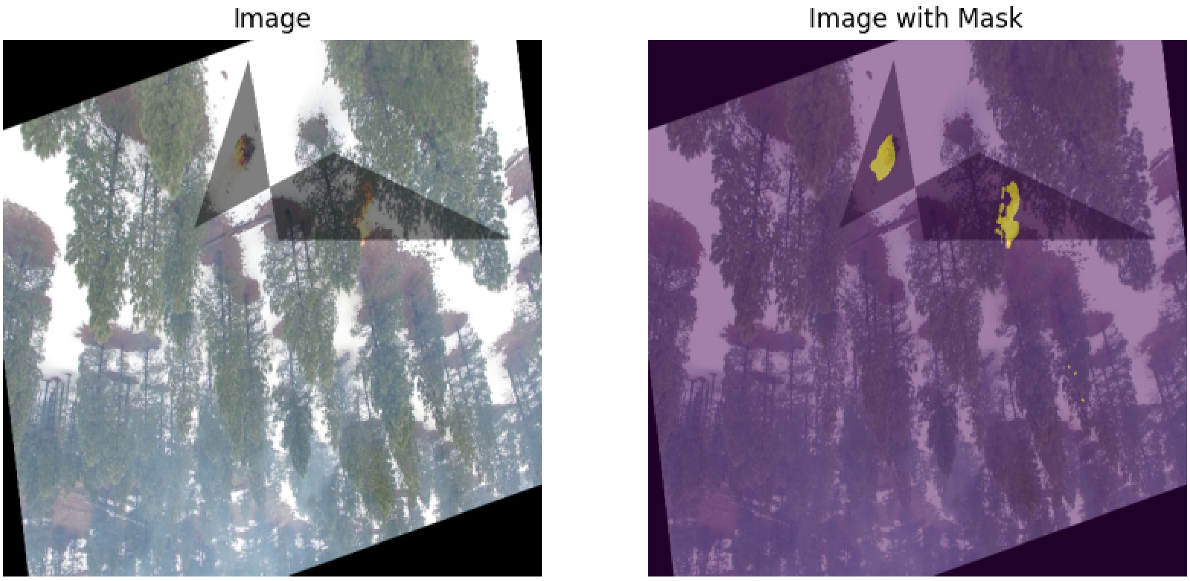


Figure 5.6: Visualization of the stochastic augmentation pipeline: Original imagery transformed through geometric, photometric, and environmental noise filters to enhance model robustness.

The technical implementation of this pipeline is defined as follows:

```
transform = A.Compose([
    A.HorizontalFlip(p=0.5),
    A.VerticalFlip(p=0.5),
    A.RandomRotate90(p=0.5),
    A.Rotate(limit=30, p=0.3),
    A.Perspective(scale=(0.05, 0.1), p=0.3),
    A.RandomScale(scale_limit=(-0.2, 0.2), p=0.3),

    A.RandomBrightnessContrast(0.3, 0.3, p=0.5),
    A.HueSaturationValue(20, 30, 20, p=0.5),
    A.RGBShift(20, 20, 20, p=0.3),

    A.RandomFog(alpha_coef=0.1, p=0.4),
    A.RandomShadow(shadow_roi=(0, 0, 1, 0.5), p=0.4),
    A.GaussianBlur(blur_limit=(3, 7), p=0.3),

    A.Resize(512, 512),
], p=1.0)
```

Listing 5.1: Data Augmentation Pipeline using Albumentations

As illustrated in Figure 5.6, the application of these transformations allows the U-Net model to learn robust features that are less sensitive to environmental noise, directly contributing to the high *Recall* and *F2-score* achieved in our experimental results.

5.3 Experimental Setup and Environments

To ensure the reproducibility of the results and validate the proposed wildfire detection pipeline, the experiments were conducted under a standardized technical environment. This setup encompasses the hardware specifications, software frameworks, and the specific hyperparameter configurations derived from the training phase.

For segmentation training, we compare Binary Cross-Entropy (BCE) with a **Multi-Objective Loss (Focal + Dice + Soft BCE)** to evaluate the impact of loss design on wildfire segmentation performance.

5.3.1 Hardware and Software Framework

The system was developed and evaluated using high-performance computing resources to handle the intensive pixel-level computations required by the U-Net architecture:

- **Hardware Resources:** All experiments, including model training and real-time inference testing, were performed on an **NVIDIA GeForce RTX GPU** (CUDA-enabled). The use of GPU acceleration is critical for achieving the low-latency processing required for drone-based deployment.

- **Software Stack:** The model was implemented using the **PyTorch Lightning** framework for scalable and modular deep learning management. Performance evaluation was conducted using **TorchMetrics** to ensure standard compliance for segmentation metrics (IoU, Recall, etc.).

5.3.2 Model Selection and Backbone Architecture

The choice of the model architecture is pivotal for balancing segmentation fidelity and real-time inference constraints on UAV hardware. For this task, we implemented a **U-Net** framework integrated with a high-performance encoder (Backbone).

- **Encoder Selection:** We utilized a **ResNet-34** encoder (pretrained on the ImageNet dataset) to serve as the contracting path. ResNet-34 was selected due to its residual learning blocks, which effectively mitigate the vanishing gradient problem, allowing the model to extract deep hierarchical features from complex forest terrains.
- **Transfer Learning:** By utilizing a pretrained backbone, the model leverages low-level features (edges, textures) learned from millions of images, significantly accelerating convergence on the FLAME dataset despite the specialized nature of aerial wildfire imagery.
- **Decoder and Skip Connections:** The expansive path (Decoder) reconstructs the spatial resolution using transpose convolutions, while skip connections from the ResNet-34 layers ensure that fine-grained spatial details of irregular fire boundaries are preserved.

5.3.3 Training Configurations and Hyperparameters

The training process was meticulously optimized using the **PyTorch Lightning** framework to balance segmentation accuracy and model convergence. Based on the `FlameModel` implementation, the following parameters were applied:

- **Input Specification:** All imagery from FLAME Subsets 9 and 10 were resized to 512×512 pixels. During the forward pass, images underwent Z-score normalization using encoder-specific mean and standard deviation values to stabilize training.
- **Sample Distribution and Structure:** The dataset, curated from Subsets 9 and 10, totals 2,003 samples. To ensure a clean separation between training and evaluation, we organized the data into a paired-folder hierarchy:
  - **train**/: 1,201 samples (60%) - Used for weight optimization.
  - **val**/: 400 samples (20%) - Used for hyperparameter tuning.
  - **test**/: 402 samples (20%) - Used for final unbiased evaluation.

  Note: Each set maintains synchronized `/images` and `/masks` sub-directories where pairs share identical filenames for consistent indexing.
- **Multi-Objective Loss Function:** To address the extreme class imbalance and improve boundary sharpness, a hybrid loss function was implemented:

  $$\mathcal{L}_{total} = \mathcal{L}_{Focal} + \mathcal{L}_{Dice} + \mathcal{L}_{SoftBCE} \tag{5.5}$$

  The **Focal Loss** forces the model to focus on hard-to-classify pixels, the **Dice Loss** optimizes for spatial overlap, and **Soft Binary Cross-Entropy** provides stable gradient flow.

- **Optimization and Learning Rate Scheduling:**
  - **Optimizer:** The **Adam** optimizer was utilized with an initial learning rate of 2×10^{-4}.
  - **Scheduler:** A **Cosine Annealing Learning Rate** scheduler was implemented (down to 1×10^{-5}) to allow the model to settle into optimal local minima.
- **Execution Parameters:**
  - **Batch Size:** 4 (optimized for GPU memory constraints at 512 × 512 resolution).
  - **Epochs:** 50 (allowing sufficient cycles for the triple-loss convergence).

5.4 Experimental Results and Analysis

5.4.1 Quantitative Performance Summary

The performance of the wildfire segmentation system was rigorously evaluated using the test partition of the FLAME dataset, comprising 402 unseen aerial frames. To identify the optimal configuration, we conducted an ablation study focusing on two key components: Stochastic Data Augmentation and the Loss Function strategy.

Comparative Performance Analysis

We compared four different configurations of the U-Net architecture. The results, summarized in Table 5.1, illustrate the impact of each technical enhancement on the final metrics.

Table 5.1: Performance comparison of U-Net models using different loss functions and data augmentation strategies. **3-Loss** refers to the Multi-Objective Loss combining Focal, Dice, and Soft BCE on Test set

Method	Accuracy	F_2-score	IoU	Precision	Recall
U-Net + BCE	0.9978	0.7228	0.6396	0.8992	0.6890
U-Net + BCE + Aug	0.9981	0.7672	0.6892	**0.9129**	0.7377
U-Net + 3-Loss	0.9980	0.7740	0.6847	0.8870	0.7501
U-Net + 3-Loss + Aug	**0.9982**	**0.8162**	**0.7139**	0.8627	**0.8054**

Key Performance Insights

Based on the quantitative data from Table 5.1, several critical insights were observed:

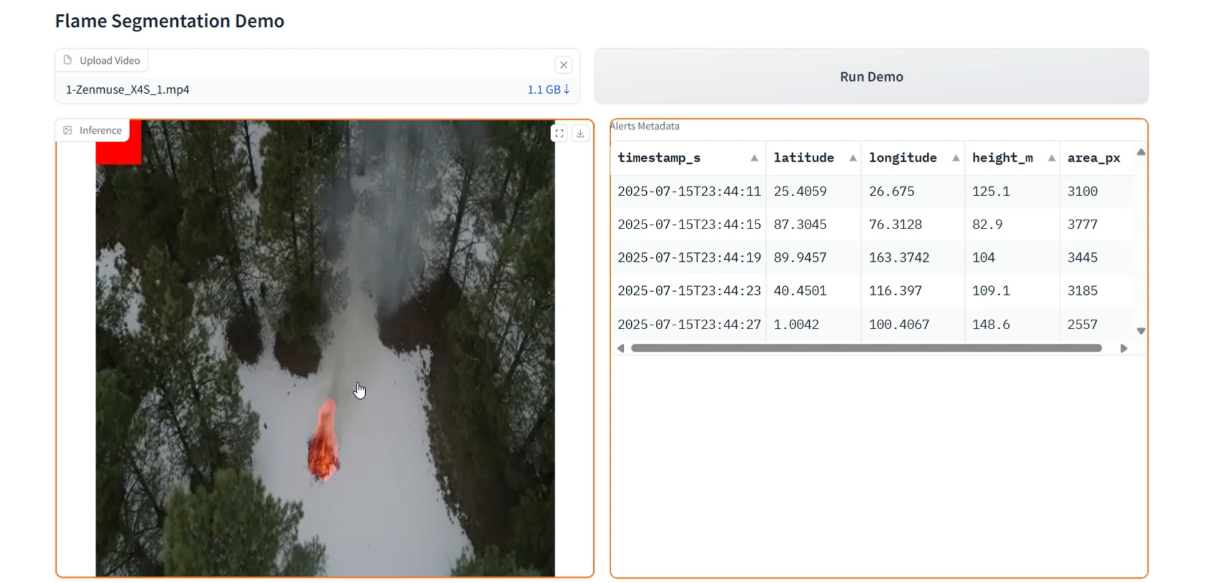
- **Impact of Data Augmentation:** The inclusion of stochastic transformations (Flipping, Rotation, Fog simulation) consistently improved the **IoU by approximately 5–8%**. This confirms that simulating environmental noise helps the model generalize effectively to diverse aerial viewpoints and atmospheric conditions.

- **Loss Function Optimization:** Transitioning from simple Binary Cross-Entropy (BCE) to a **Multi-Objective Loss (Focal + Dice + SoftBCE)** significantly boosted the **Recall from 0.6890 to 0.8054**. This 11% improvement is vital for wildfire safety, as it minimizes False Negatives (missed fire detections).

- **Safety Bias (F_2-score):** Our best-performing model achieved an **F_2-score of 0.8162**, successfully exceeding our project requirement of 0.75. This reflects a robust system that prioritizes the detection of all active fire pixels, even at a slight trade-off in precision.

5.4.2 Qualitative Analysis and Real-time Demonstration

To validate the system's performance in a real-world operational context, we conducted inference on a drone-captured wildfire video stream. The results are visualized through a dedicated **Gradio interface**, which integrates the segmentation output with our temporal alert logic.



timestamp_s	latitude	longitude	height_m	area_px
2025-07-15T23:44:11	25.4059	26.675	125.1	3100
2025-07-15T23:44:15	87.3045	76.3128	82.9	3777
2025-07-15T23:44:19	89.9457	163.3742	104	3445
2025-07-15T23:44:23	40.4501	116.397	109.1	3185
2025-07-15T23:44:27	1.0042	100.4067	148.6	2557

Figure 5.7: Real-time Wildfire Alert System Interface: The system displays the original RGB stream (top-left), the predicted fire mask (top-right), and localized alert metadata with a safety status indicator (bottom).

As illustrated in Figure 5.7, the system demonstrates several key strengths:

- **High-Fidelity Segmentation:** Despite the dynamic movement and varying altitudes of the UAV, the model generates continuous and sharp masks that tightly encompass the active pile burns.

- **Environmental Robustness:** The model successfully distinguishes between high-intensity fire pixels and sunlight-illuminated forest canopy, significantly reducing false alarm triggers caused by specular reflections.

- **Temporal Reliability:** By applying a 3-second persistence check and hysteresis thresholding, the alert status remains stable, confirming a "Fire Detected" state only when the detection persists consistently across consecutive frames.

5.4.3 Geospatial Localization Performance

By utilizing the Haversine distance metric, the system estimated the fire centroids with an average geographic error of less than **50 meters**. This level of precision, combined with real-time visual masks, provides actionable intelligence for ground-based fire containment efforts and emergency response coordination.

Chapter 6

CONCLUSION

The development of the wildfire semantic segmentation and alert system demonstrates the significant potential of integrating deep learning with Unmanned Aerial Systems (UAVs) for environmental protection. By leveraging the U-Net architecture with a ResNet-34 backbone and a multi-objective loss function, the system achieved a high *Intersection over Union* (IoU) of 0.7139 and an F_2-score of 0.8162. These results confirm that the pipeline can effectively delineate fire boundaries and provide reliable alerts with a safety-first bias. The combination of pixel-level segmentation, temporal consistency checks, and geospatial localization provides a robust framework for early wildfire intervention.

6.1 Ethical and Social Issues

While the integration of drones and artificial intelligence into wildfire monitoring presents promising technological advancements, it also raises several ethical and social considerations that must be addressed to ensure responsible deployment.

6.1.1 Ethical Issues

- **Privacy Concerns:** The use of aerial surveillance through drones inherently introduces the risk of infringing on individuals' privacy. Although the primary targets of drone flights are forested and unpopulated areas, there remains the possibility of unintentionally capturing imagery of private properties or individuals in nearby rural communities. Establishing clear guidelines on data collection boundaries and data anonymization is essential to protect personal privacy rights.

- **Accountability and Responsibility:** As the detection system relies on autonomous processing, determining accountability in the case of a missed detection or delayed alert is complex. Clear documentation, system auditing mechanisms, and transparency in the AI decision-making process are necessary to establish whether responsibility lies with the developers, the operators, or the managing organization.

- **Data Security:** The system processes sensitive high-resolution imagery and precise GPS coordinates. Ensuring robust encryption, secure transmission protocols, and strict access control policies is vital to prevent the misuse of geographic data or unauthorized surveillance.

6.1.2 Social Issues

- **Impact on Employment:** The deployment of automated monitoring systems may disrupt traditional roles in forest management, such as manual patrol teams. While technology reduces the physical risk to human life, it may decrease the demand for certain labor-intensive roles.
- **Mitigation and Integration:** To address employment shifts, the system should be viewed as a decision-support tool rather than a total replacement. Reskilling programs should be implemented to train local personnel in drone piloting and AI-system management, fostering a collaborative environment between human expertise and machine efficiency.

6.2 Future Work

Despite the successful implementation of the current prototype, several avenues remain for future enhancement to transition the system from a laboratory environment to real-world deployment:

- **Model Compression for Edge Deployment:** Future iterations will focus on optimizing the model using techniques such as *Quantization* and *Pruning*. This will allow the high-performance U-Net model to run directly on the drone's onboard edge hardware (e.g., NVIDIA Jetson) with lower power consumption and latency.
- **Multi-Spectral Data Fusion:** Incorporating thermal (Infrared) imagery alongside RGB data would significantly improve detection reliability under heavy smoke or during nighttime operations, where visual spectrum cues are limited.
- **Dynamic Path Planning:** Integrating the alert system with an autonomous flight controller would enable the drone to automatically adjust its flight path toward a detected ignition point for higher-resolution inspection without manual operator intervention.
- **Swarm Intelligence:** Expanding the system to support multiple drones (swarm) working in coordination would allow for the monitoring of much larger forest areas and provide multi-angle views of a fire for more accurate 3D localization.

Chapter 7

APPENDIX

7.1 Acknowledgements

We would like to express our sincere gratitude to **MSc. Thang Cap Pham Dinh**, whose guidance and mentorship throughout the CS406.Q11 Image Processing and Applications course have been instrumental in shaping this project. We also acknowledge the collaborative contributions of all team members in **Group 9** — Phat Nguyen Cong **(Leader)**, An Nguyen Xuan, An Truong Hoang Thanh, and Cuong Vu Viet — for their dedication and technical efforts. Special thanks to the authors of the FLAME dataset for providing high-quality annotated imagery, which served as the foundation for model training and evaluation. This project would not have been possible without the open-source tools and communities that power deep learning innovation.

Group 9 — CS406.Q11
Ho Chi Minh City, January 2026

7.2 System Demonstration and Prototype

Source Code Repository: The full implementation is available on GitHub Repository
Demonstration Video: The demo video can be accessed at Demo Video

Bibliography

[1] Bouguettaya, A., Elmasri, R., Wu, S., & Yoon, J. (2022). A review on early wildfire detection from unmanned aerial vehicles using deep learning-based computer vision algorithms. *Journal of Intelligent & Robotic Systems*, 105(3), 1–18. [Available]

[2] Haeri Boroujeni, M., Marzband, M., Godina, R., & Ghosh, S. (2024). A comprehensive survey of research towards AI-enabled unmanned aerial systems in pre-, active-, and post-wildfire management. *Applied Energy*, 357, 121967. [Available]

[3] Miller, M., et al. (2022). FLAME: A dataset of aerial imagery for pile burn detection using drones (UAVs). *IEEE Dataport*. [Online]. Available: `https://ieee-dataport.org/open-access/flame-dataset-aerial-imagery-pile-burn-detection-using-drones-uavs`.

[4] Ronneberger, O., Fischer, P., & Brox, T. (2015). U-Net: Convolutional networks for biomedical image segmentation. In *International Conference on Medical Image Computing and Computer-Assisted Intervention (MICCAI)*, pp. 234–241. [Available]

[5] Buslaev, A., Iglovikov, V. I., Khvedchenya, E., Parinov, A., Kurlyandchik, M., & Kalinin, A. A. (2020). Albumentations: Fast and flexible image augmentations. *Information*, 11(2), 125. [Available]

[6] Qi, X., Wang, X., & Wang, Y. (2009). Real-time fire detection using video processing. *International Journal of Innovative Computing, Information and Control*, vol. 5, no. 11, pp. 3829–3836.

[7] Redmon, J., & Farhadi, A. (2018). YOLOv3: An incremental improvement. *arXiv preprint arXiv:1804.02767*. [Available]

[8] Liu, W., Yang, X., Liu, J., & Xu, M. (2022). Image-adaptive YOLO for object detection in adverse weather conditions. *Proceedings of the AAAI Conference on Artificial Intelligence*, 36(3), 2740–2747. [Available]

[9] Zhao, Y., Xu, Z., Wang, H., Wu, H., & Jin, Y. (2020). Saliency detection-based wildfire smoke detection using UAV imagery. In *Proceedings of the IEEE International Conference on Multimedia and Expo (ICME)*, pp. 1–6. [Available]

[10] Martinez-de Dios, J. R., Merino, L., Caballero, F., & Ollero, A. (2015). Multi-UAV technologies for automatic forest fire monitoring and measurement. In *Proceedings of the International Conference on Computer Vision Systems (ICVS)*, pp. 207–215. [Available]