AI-FireGuard

THERMAL-RGB FUSION FOR UAV-BASED WILDFIRE MONITORING

Agenda

- Objective & environment
- YOLO workflow & Architecture
- U-Net workflow & Architecture
- Results & Future Vision
- Argument
- Summary




Objective & Environment

The goal of the Fire Detection Project is to build an intelligent system that automatically detects fire or smoke from images or video streams at an early stage, helping to reduce damage, save lives, and support fast emergency response using computer vision and machine learning techniques.

Generalization:

The model is designed to generalize well to unseen data by accurately detecting fire under different lighting conditions, environments (indoor/outdoor), camera angles, and fire types, making it reliable for real-world deployment.

Why Google Colab :

Google Colab was chosen as the working environment because it provides free access to GPUs, requires no local setup, supports easy collaboration, and integrates smoothly with popular machine learning libraries such as TensorFlow, PyTorch, and OpenCV.

YOLO v8 workflow

- **1. Data Preparation & Preprocessing**
- Objective: Convert raw dataset into YOLO-compatible format for training.

Steps:

- **Dataset Organization:**
- Images: dataset/images/
- Masks: dataset/masks/
- Labels output: dataset/labels/
-
- **Mask Conversion:**
- Each mask is converted to YOLO .txt format.
- Multi-class handling using class_map
- Bounding boxes extracted from contours of non-zero regions.
- Minimum area filtering to remove noise.
- Morphological operations applied to refine masks.

Dataset Split:

Train/Validation split ratio: 80/20.

YAML file (dataset.yaml) automatically generated for YOLO.

Output:

.txt label files aligned with each image.

Cleaned, preprocessed dataset ready for YOLOv8 training.

YOLO v8 workflow

- **2. Model Definition**
- Objective: Initialize YOLOv8 model for fire detection.
- Details:
- Pretrained model used: yolov8n.pt (nano version for speed).
- Single-class detection (single_cls=True) targeting fire regions.
- Input image size: 640x640 pixels.

3. Training Loop

Objective: Train YOLOv8 model with early stopping and monitoring.

Key Parameters:

Epochs: 100

Batch Size: 16

Learning Rate: Initial = 0.001, Final = 0.01

Optimizer: Auto

Patience: 25 (early stopping)

Loss weights: Box = 7.5, Classification = 0.5, DFL = 1.5

Warmup Epochs: 3

YOLO v8 workflow

4 Loss Functions & Metrics

- **YOLOv8 Loss Components:**
- 1. Box Loss: Measures localization error of predicted bounding boxes.
- 2. Classification Loss: Cross-entropy between predicted class and ground truth.
- 3. DFL Loss: Distribution Focal Loss for bounding box refinement.

Evaluation Metrics:

- Precision: Fraction of correct positive predictions.
- Recall: Fraction of true positives detected.

mAP50: Mean Average Precision at 0.5 IoU threshold.

mAP50-95: mAP averaged across IoU thresholds 0.5–0.95.

F1-score: Harmonic mean of Precision and Recall.

Model Box-plot

- Raw Dataset (Images + Masks)

|

▼

- Preprocessing & Mask Conversion

|

▼

- Dataset Split (Train/Validation)

|

▼

- YOLOv8\UNet Model Initialization

|

▼

- Model Training Loop
- (Epochs, Early Stopping, Checkpoints)

|

▼

- Metrics Computation & Loss Monitoring

|

▼

- Training Curves & Progress Analysis

|

▼

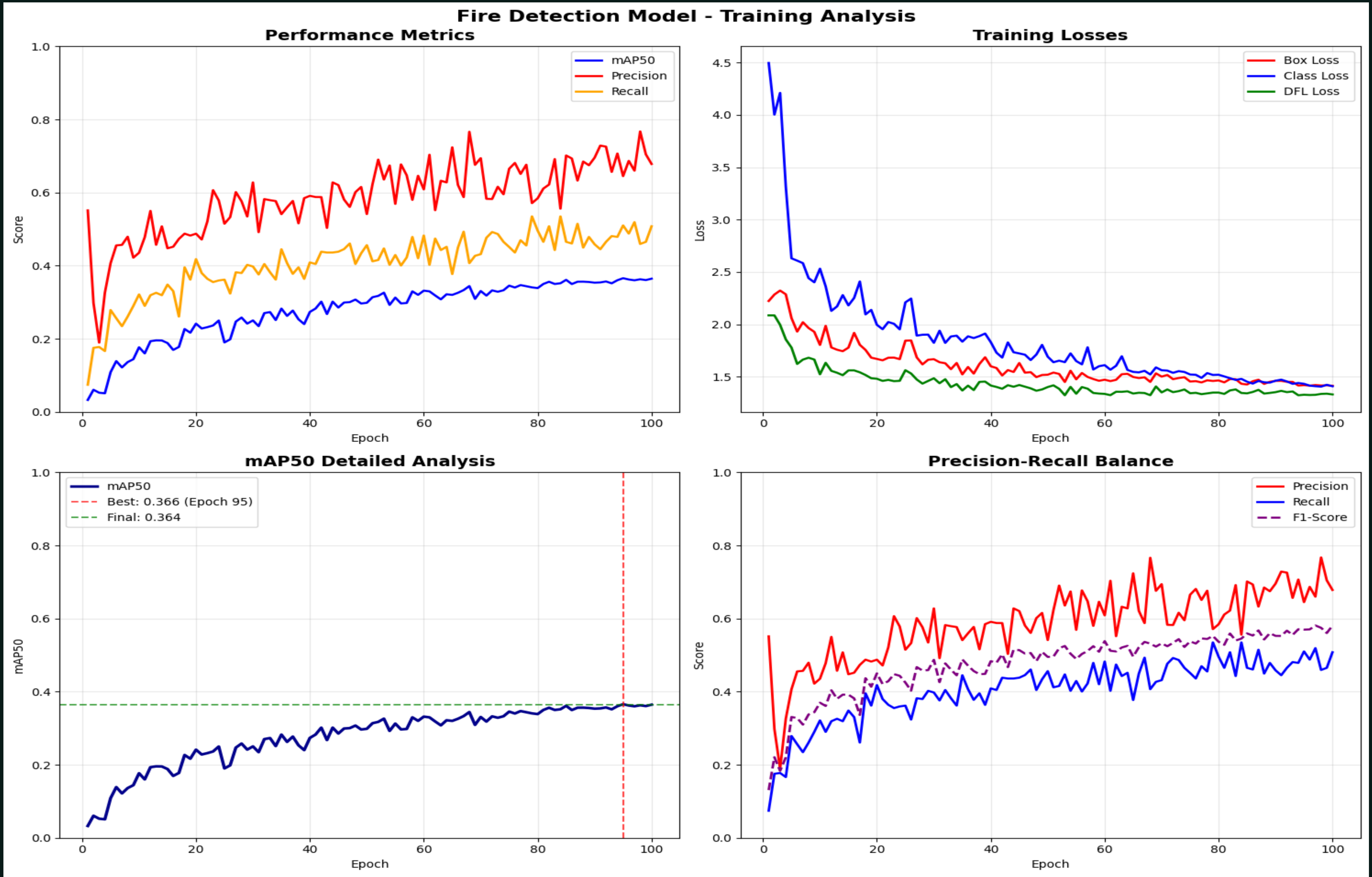
- Final Results & Performance Analysis

|

▼

- Recommendations & Deployment

5 Training Progress & Visualization



YOLOv8n Model Summary (Nano Version)

- Component Output Channels Purpose
- Focus Layer 32 Downsampling and initial feature extraction
- C2f Bottleneck × n 64, 128, 256 Deep hierarchical feature extraction
- FPN/PAN Neck 256, 128, 64 Multi-scale feature aggregation
- Detection Head 1 (single class) Box + class probability
- **Parameters:** ~3.2M for YOLOv8n
- **Model Size:** ~6.3 MB
- **Inference Speed:** Very fast; suitable for real-time detection

U-Net Pipeline

- **Data Preparation and Preprocessing**
- **1.1 Data Loading**
- Each sample consists of:
- RGB image (3 channels)
- Thermal image (1 channel)
- Binary segmentation mask
- RGB and thermal images are fused to form a **4-channel input tensor**:
- Input shape $= [4, H, W]$

- **1.2 Data Augmentation (Albumentations)**
- **Training augmentations:**
- Resize to 512 × 640
- Normalization : std & mean
- **Validation & Test:**
- Resize + normalization only (no augmentation)

1.3 Dataset Split and Loading

- **Training:** 80%
- **Validation:** 10%
- **Testing:** 10%

Batch sizes:

- Train: 4
- Validation/Test: 2

Model Architecture

U-Net is specifically designed for **pixel-level segmentation** tasks and performs well even with limited data.
Include total 19 layers :

ENCODER (DOWNSAMPLING PATH)

Block	Input Channels	Output Channels
Down 1	4	64
Down 2	64	128
Down 3	128	256
Down 4	256	512

Each block uses:

- 2 × Conv2D (3×3)
- ReLU activations
- MaxPooling (2×2) between blocks

SKIP CONNECTIONS

preserve spatial details

Final Layer:

- 1×1 convolution → single-channel output
- Sigmoid activation applied during loss/metric computation

DECODER (UPSAMPLING PATH)

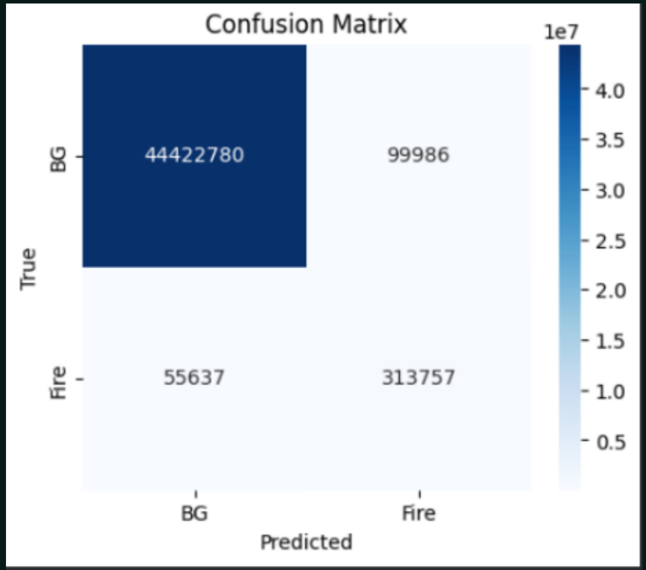
Block	Input Channels	Output Channels
Up 3	512 + 256	256
Up 2	256 + 128	128
Up 1	128 + 64	64

- Bilinear upsampling
- Skip connections from encoder layers
- Double convolution blocks

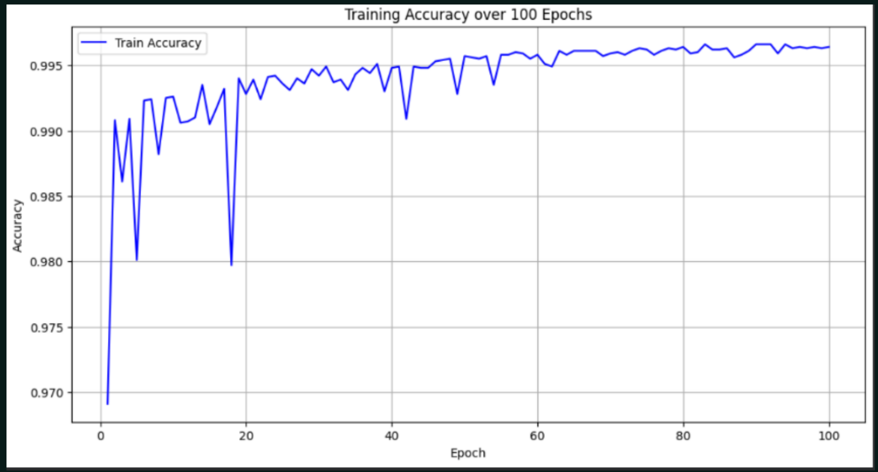
Visualization

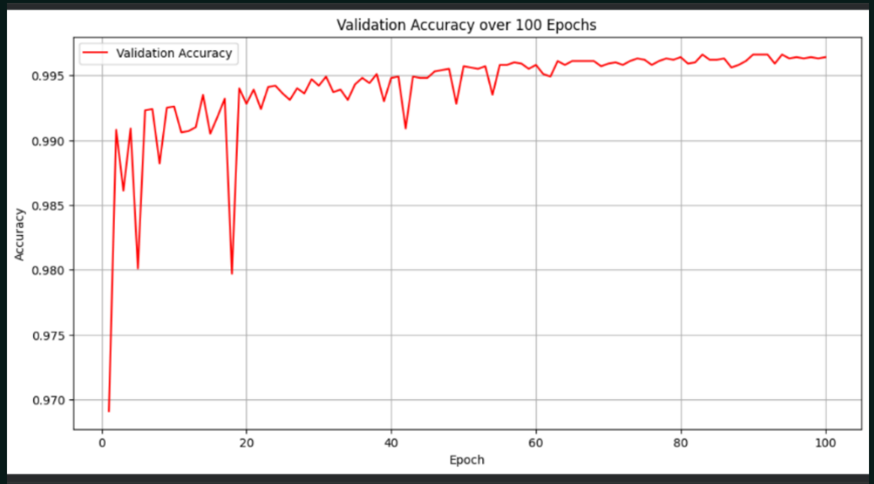
•Training & validation loss curves:

Confusion Matrix:



- Training & validation Accuracy:





Results & Future Vision

RESULTS:

Model	Accuracy	Precision	Recall	F1-score	IoU
U-Net	0.9965	0.7583	0.8494	0.8013	0.6684

Metric	Initial	Final	Best
Precision	0.5513	0.6781	0.7673
Recall	0.0742	0.5079	0.5348
mAP@50	0.0326	0.3644	0.3657

MODELS COMPARISON:

U-Net demonstrates superior overall performance. It achieves higher precision, F1-score, and Intersection over Union (IoU), indicating more accurate fire boundary delineation and fewer false positives.

YOLOv8 shows substantial improvement throughout training, particularly in recall and mAP@50. This indicates strong learning capability and suitability for real-time fire localization tasks. Nevertheless, its performance remains inherently less precise than segmentation-based approaches due to the complexity of object-level detection.

Most Important Metrics for Fire Applications

- For fire segmentation, recall is the most critical metric, as failing to detect fire pixels poses a significant safety risk. IoU or Dice score follows in importance, as these metrics evaluate the spatial accuracy of fire regions. The F1-score provides a balanced measure between false alarms and missed detections.

- For fire detection, recall remains the top priority, ensuring fires are not overlooked. Mean Average Precision (mAP@50) reflects overall detection robustness, while precision is essential to minimize false fire alerts.

Final Conclusion

- Based on the presented results, U-Net is the most effective model for fire segmentation, offering the best balance between accuracy and reliability. YOLOv8 is well-suited for real-time fire detection scenarios where rapid localization is required. For safety-critical fire monitoring systems, prioritizing recall alongside region-based metrics such as IoU or Dice is essential.

How to improve results

- **U-Net:** adding BatchNorm, residual connections, and learning-rate schedulers.
- **YOLOv8:** increasing image resolution, using stronger augmentation, and trying larger backbones.

Argument

The challenge of accurate fire detection using UAV imagery arises from environmental complexity, low visibility, and significant variation in fire appearance across RGB and thermal modalities. Traditional single-modal systems struggle to maintain high detection performance in real-world conditions such as smoke, darkness, and occlusions.

By leveraging a dual-modal RGB–Thermal dataset and a deep learning model trained on aligned multi-spectral inputs, the system provides more reliable detection and segmentation, improving safety, monitoring efficiency, and early-response capabilities.

Summary

This project developed a UAV-based fire detection system using RGB-Thermal imagery and a deep-learning segmentation model. Through multi-modal fusion, data preprocessing, model training, and evaluation using accuracy, precision, recall, F1 score, and Dice/IoU metrics, the system achieved strong performance in identifying fire regions under challenging conditions. The results demonstrate the effectiveness of combining RGB and thermal data for robust wildfire detection and highlight the potential for future improvements such as real-time onboard deployment and advanced multi-modal architectures.

Thank you

KHAWLA AL-SHAIKH

HANIN SHAHIN

JUDY DUKMAK

BASHAR ALSALEH

Supervised By:

Dr. RIAD SONBOL

Eng. AYA ALASWAD