Wildfire Detection with a CNN Ensemble

1INF52 Deep Learning

CPSquad

PUCP

March 3, 2025

Contents

Introduction and Problem Statement

- The increasing frequency of wildfires has led to a demand for **automated monitoring systems**.
- Traditional methods (satellites, thermal sensors) suffer from **delayed data retrieval**.
- **Deep Learning** can improve detection **speed and accuracy**.

Figure: Wildfire in Peru, 2024.

Research Motivation and Goals

Objective of this Study

- Develop an **ensemble of CNNs** for **wildfire detection** in aerial images.
- Evaluate the performance of **Xception, DenseNet121, and ResNet152**.
- Improve classification accuracy while keeping computational efficiency.

Traditional Approaches to Wildfire Detection

- **Sensor-based methods:** Use **temperature, smoke, and gas sensors**, but have **limited coverage**.
- **Classic Computer Vision methods:** Use **color-based segmentation**, but suffer from **high false positives**.
- **Machine Learning and Deep Learning approaches:**
  - CNN-based classification (e.g., Xception, DenseNet, ResNet).
  - Object detection with **YOLOv8**.
  - Vision Transformers (**ViTs**) for feature extraction.

Baseline Model: FireSight (Stanford)

- FireSight combines **CNNs and ViTs** for aerial wildfire detection.
- Achieves **82.28% accuracy** using **DenseNet + ResNet + ViT** ensemble.
- Our approach:
  - Improve CNN-only ensemble.
  - Optimize architecture for **real-time drone deployment**.

FLAME Dataset Overview

- **FLAME dataset** contains **drone-captured** images of wildfires.
- **Training Set:** 39,375 images **Test Set:** 8,617 images.
- **Class Distribution:**
  - Fire: **25,027 images (63.55%)**.
  - No-Fire: **14,357 images (36.45%)**.

Pipeline for Model Training

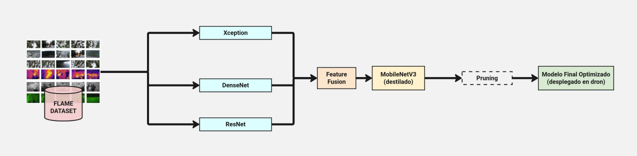

Figure: Model Architecture

- **Preprocessing:** Images resized to **224x224** and augmented.
- **Training Strategy:** Individual CNNs trained separately.
- **Evaluation:** Models compared based on **accuracy, precision, recall, and F1-score**.

Keras Tuner for Hyperparameter Search

- **Keras Tuner** used for optimizing **batch size, dropout, L2 regularization, and learning rate**.
- Best hyperparameters found:

Model	Batch Size	Dropout	L2 Factor	Learning Rate
Xception	10	0.45	0.001	0.00541
DenseNet121	64	0.35	0.001	0.00147
ResNet152	64	0.4	0.0005	0.00093

Why Use an Ensemble?

- **Goal:** Improve model **stability and accuracy**.
- We experimented with:
  - Majority Voting (Final Selection).
  - Weighted Averaging.
  - Stacking.

Final Approach: Voting

- Each model votes on the predicted class.
- The most frequent class is the **final prediction**.

Confusion Matrices

Final Model Performance

- **Baseline F1-score: 0.58**, Accuracy: **82.28%**
- **Our Final Model: F1-score: 0.61**, Accuracy: **86.50%**

Key Takeaways

- Our ensemble **outperforms the baseline** in **accuracy and F1-score**.
- CNN ensembles can achieve **real-time inference** on drones.
- Future work: **Dataset expansion, real-world testing, transfer learning**.

Thank You!

Questions?