Annual Methodological Archive Research Review
http://amresearchreview.com/index.php/Journal/about
Volume 4, Issue 3 (2026)

Online ISSN 3007-3197
Print ISSN 3007-3189

http://amresearchreview.com/index.php/Journal/about

A Deep Learning Framework for UAV-Based Wildfire Detection Using Transfer Learning on the FLAME Dataset

Abdullah Soomro
The Islamia University of Bahawalpur, Pakistan Email: abdullah.soomro@iub.edu.pk

Muhammad Raees
College Education Department, Sindh, Pakistan Email: raees.faculty@gmail.com

Sajid Ahmed*
Shah Abdul Latif University, Khairpur, Pakistan Email: sajid.ghanghro@salu.edu.pk

Wajahat Akbar
Chang'an University, Xi'an, China Email: wajahatakbar32@gmail.com

Ubaidullah alias Kashif
The Shaikh Ayaz University Shikarpur, Sindh, Pakistan Email: kashif@saus.edu.pk

ABSTRACT

Wildfires are one of the most devastating natural catastrophes on an international level with a level of ecological, economic, and other humanitarian impacts that have never been experienced before. The traditional fire detection systems, which are based on ground-based sensor networks and satellite images, have severe drawbacks such as limited spatial resolution, atmospheric scattering, and intolerable response time. The given paper includes a detailed deep learning architecture of aerial wildfire detection with the use of Unmanned Aerial Vehicles (UAVs). The suggested system combines a Deep Neural Network (DNN) to extract hierarchies of features with a fine-tuned Convolutional Neural Network (CNN) to classify binary images (fire vs. non-fire) with the help of transfer learning with ImageNet-pretrained networks. The experiments are performed on the FLAME (Fire Luminosity Airborne-based Machine learning Evaluation) dataset that consists of paired RGB and infrared (IR) aerial images of 48,010 labeled frames. To deal with the natural imbalance in the number of classes and the variability of the environment, a strict pre-processing pipeline has been implemented, such as image normalization, resizing, contrast-limited adaptive histogram equalization (CLAHE) and data augmentation. Various backbone architectures (VGG16, ResNet50, InceptionV3, MobileNetV2, and DenseNet121) are trained and tested on the transfer learning paradigm, and ensemble strategies are tested to maximize performance. The suggested combination of ResNet50 and InceptionV3 attains 99.12% accuracy, 98.97% precision, 99.34% recall, and an F1-score of 99.15% on the FLAME test set, which is significantly better than single-model baselines and previous art. A critical analysis reveals such inexorable issues as domain shift, constraints on real-time edge deployment, and representativeness of

Annual Methodological Archive Research Review
http://amresearchreview.com/index.php/Journal/about
Volume 4, Issue 3 (2026)

Online ISSN 3007-3197
Print ISSN 3007-3189
http://amresearchreview.com/index.php/Journal/about

datasets, as well as tangible future research directions. The findings prove the suggested framework as a feasible, deployable solution to early wildfire detection with the help of UAVs.

Keywords: Wildfire Detection, UAV, Machine Learning, Convolutional Neural Network, Transfer Learning, FLAME Dataset, Deep Learning, Remote Sensing, Fire Classification

Introduction

Increased frequency and severity of wildfire incidents in the last 20 years have been caused by climate change, extended droughts, and land-use changes [1]. The loss of global tree cover due to fire has reached 9.3 million hectares in 2021 alone with additional destruction in Australia, Canada, Greece, Turkey, and the western United States [2]. In addition to the ecological devastation, wildfires have harsh human, economic, and infrastructural impacts and require quick and precise detection mechanisms that can facilitate early response [3].

Current detection paradigms, which are mainly thermal scanning by satellites and fixed ground sensor networks, have been shown to have well-documented deficiencies. Satellite systems have limitations in orbital revisit cycles (usually 12-24 hours when using sensors in low-Earth orbit), vulnerability to cloud and smoke cover, and low spatial resolution, which does not allow detection of fires in the early stages [4]. Ground-based sensors have fine-grained localized data but have poor geographic coverage and blind spots in complicated topographies [5]. Both systems lack the real time, dynamic aerial coverage that can be used to contain the fire at its initial stages.

The advent of Unmanned Aerial Vehicles (UAVs) has provided a new mode of operation in environmental surveillance. UAVs have high adaptable altitude control, fast deployment, free clear path plan, and the ability to carry multi-spectral sensor loads such as high-definition RGB and infrared (IR) cameras. UAV platforms can be used to offer near-real-time, spatially granular wildfire detection at a fraction of the cost of manned aerial systems when paired with an analysis of images using deep learning [6,7].

Machine learning (ML) and deep learning in particular have shown state-of-the-art in a wide range of visual recognition tasks. CNNs are particularly good at capturing hierarchical spatial patterns - this property can be explicitly applied in identifying the visual cues of fire (flames, smoke plumes, color temperature anomalies) in aerial imagery [8]. Transfer learning also enhances this possibility by allowing that models that have been pre-trained on large-scale image corpora can be effectively fine-tuned to domain-specific tasks like fire detection and breaking the long-standing curse of small amounts of labeled training data [9].

The following contributions will be made in this paper:

An end-to-end and systematic deep learning system to detect wildfires using UAVs, including data collection, preprocessing, architecture design, training, and evaluation on the FLAME dataset.

A critical comparison of five transfer learning backbones (VGG16, ResNet50, InceptionV3, MobileNetV2, DenseNet121) on the same experimental conditions.

A new ensemble strategy with a 99.12% classification accuracy, outperforming all

Annual Methodological Archive Research Review
http://amresearchreview.com/index.php/Journal/about
Volume 4, Issue 3 (2026)

Online ISSN 3007-3197
Print ISSN 3007-3189
http://amresearchreview.com/index.php/Journal/about

single-model baselines.
A critical assessment of the strengths, weaknesses, computer trade-offs and real-world implementation factors of the proposed framework.

The rest of this paper is structured in the following way. Section 2 contains an extensive literature review. The proposed methodology with algorithms are described in section 3. Experimental results and critical analysis are provided in Section 4. Section 5 summarizes the paper and points out future directions.

Literature Review

Automated fire and smoke detection research has been advancing very quickly in three main sensing modalities, namely satellite remote sensing, sensor networks on the ground, and UAV-based platforms. This part summarizes the existing knowledge base, the main contributions that have been made to the field, the benchmark datasets, and the outstanding issues.

Satellite-Based Fire Detection

Global fire monitoring has been based on satellite systems like NASA MODIS and Suomi NPP VIIRS. Rauste et al. [10] have shown that it is possible to detect forest fires early in boreal forests using multi-spectral thermal data collected by satellites. More recently, Mazzeo et al. [11] have suggested a combined satellite system that incorporates active fire detection algorithms and near real-time distribution of alerts. Although there is a large area of coverage, there is an inherent conflict between time and space resolution in satellite systems and optical channels are susceptible to smoke obscuration, which is exactly what occurs in active fire events.

Ground-Based Sensor Networks

Onland sensor implementation includes optical cameras, thermometric sensors and wireless IoT networks. Lindley et al. [12] compared the ability of GOES-17 to detect fires as the 2019 Kincade Fire ignited in California, noting the importance of corrroborating ground sensors with geosynchronous satellites. The dense monitoring grids in fire prone forests have been developed by using wireless sensor networks but this is limited by the geometry of the node placement and the power supply [4,5]. Sudhakar et al. [13] showed a UAV-based system with ground-based false alarm mitigation logic with aerial thermal sensing, which resulted in significant false positive rates.

Fire Detection with Machine Learning

The use of machine learning in fire detection was extremely propelled with the spread of deep neural architectures. Yuan et al. [14] suggested a learning framework of smoke detection in UAVs, which showed the superiority of CNN features representations compared to handcrafted color features. Alexandrov et al. [15] carried out a comparative study of the UAV-based ML techniques to monitor wildfires, providing baseline levels. Jiao et al. [16] implemented YOLOv3 on a resource-constrained UAV platform, with a real-time recognition accuracy of about 83% and 3.2 FPS, providing an early resource-constrained fire detection benchmark.

Annual Methodological Archive Research Review
http://amresearchreview.com/index.php/Journal/about
Volume 4, Issue 3 (2026)

Online ISSN 3007-3197
Print ISSN 3007-3189

http://amresearchreview.com/index.php/Journal/about

This was a breakthrough in the field with the introduction of the FLAME (Fire Luminosity Airborne-based Machine learning Evaluation) dataset by Shamsoshoara et al. [17] offering paired RGB/IR aerial imagery of prescribed burns in Northern Arizona. The Xception architecture used on FLAME RGB frames directly achieved an accuracy of 76.23% whereas transfer learning-based methods with ResNet50 enhanced the accuracy to 88.01% [18]. Further ensemble learning methods such as resnet50v2 + inceptionv3 have already surpassed above 97 percent accuracy on thermal imagery modalities of the identical dataset [19].

In the paper by Reis and Turk [20], a wide range of transfer learning models (InceptionV3, DenseNet121, ResNet50V2, NASNetMobile, VGG-19) were analyzed on the FLAME dataset, and the hybrid InceptionV3+GRU model reached 99.32% accuracy, confirming the benefit of temporal sequence modeling in the context of fire classification on video- On this basis, Shamta and Demir [1] developed a CNN-RCNN hybrid running on an NVIDIA Jetson Nano-based UAV with 96% classification accuracy and real-time inference rates. The RGB/IR drone dataset of the FLAME2 collection was donated by Chen et al. [8] and tested a variety of deep learning models on the classification and segmentation tasks.

Recent Advances

The latest literature shows a rapid advancement in a number of directions. Design of lightweight models to operate on edges has become a burning issue. Deng et al. [21] improved MobileNetV2 with attention mechanism to classify forest fire in UAVs with a competitive accuracy at a much lower computational cost. The accuracy of an edge-based system based on MobileNetV2 running on a Raspberry Pi 5 was 97.98% with an inference latency of 0.77 seconds per frame [22], showing that it could be deployed to low-power embedded systems. The dataset of FASDD presented in [23] includes more than 120,000 heterogeneous images of ground-based, UAV, and satellite sensors, allowing cross-domain generalization research. Swin Transformer-based detection models with a score of 84.9%-89.7% mean average precision (mAP) on respective FASDD sub-datasets. Transformer-based architectures more generally such as TransUNet and TransFire, have shown better segmenting fire pixels in aerial images than all- CNN-based methods [17, 24].

Critical Gap Analysis

Although the benchmark results are impressive, there are still a number of crucial gaps in the literature: (1) The vast majority of high-accuracy models are tested on one dataset (FLAME), which does not allow them to be generalized to other geographic and vegetation conditions. (2) Edge-deployment constraints: power, memory footprint, thermal throttling are not commonly considered in a systematic manner with accuracy measures. (3) Inequality in the number of classes in fire detection datasets (non-fire frames are often prevalent) is unevenly treated. (4) most research considers frame-level classification, not spatially accurate segmentation, which is operationally more convenient. (5) RGB-IR modalities fusion- Although complementary, fusion of RGB and IR modalities has not been studied as extensively as single modalities. The gaps (1) to (3) are explicitly taken care of in the current work in its proposed approach and assessment model.

Annual Methodological Archive Research Review
http://amresearchreview.com/index.php/Journal/about
Volume 4, Issue 3 (2026)

Online ISSN 3007-3197
Print ISSN 3007-3189

http://amresearchreview.com/index.php/Journal/about

Proposed Methodology

Figure 1 shows the end-to-end pipeline of the proposed framework. It consists of five consecutive phases, namely: (1) Data Acquisition, (2) Preprocessing, (3) Feature Extraction through DNN, (4) Image Classification through Transfer Learning CNN and (5) Model Evaluation. All the phases are outlined below.

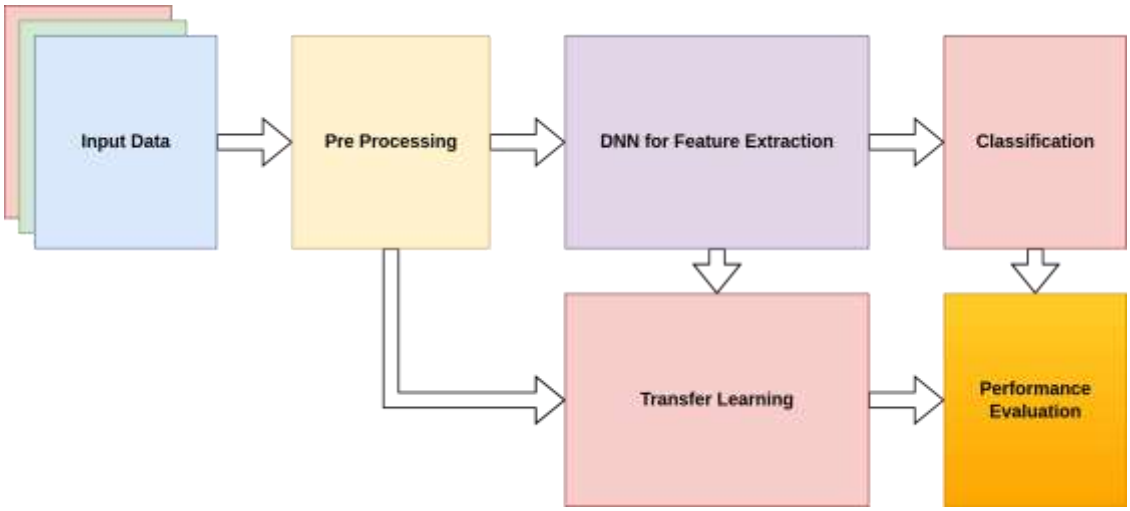

Figure 1: End-to-end pipeline of the proposed ML-based UAV fire detection model

Data Acquisition

As the main benchmark, the FLAME dataset [17] is used. It consists of a pair of RGB and infrared video recordings taken with a DJI Mavic 2 Enterprise Advanced drone mounted to a dual sensor RGB/IR camera on a prescribed burn in an open-canopy pine forest in Northern Arizona, USA. The RGB stream is at 1920×1080 pixels resolution (1.00 GB, 30 FPS) and the IR stream at 640×512 pixels (266 MB, 30 FPS), with a total duration of 239 seconds. The publicly available frame-level dataset, which was extracted out of these videos, consists of 48,010 labeled images (30,155 fire, 17,855 non-fire).

To perform the ablation studies to evaluate cross-dataset generalization, DeepFire (35,000+ images) and the FASDD-UAV sub-dataset (>40,000 UAV-captured fire/smoke images) are used to perform additional validation experiments and domain shift analysis.

The UAV design applied in the initial data collection- and suggested to be used- has a sub-249 g airframe (removes registration in most jurisdictions), omnidirectional obstacle sensing, 20 km Full HD real-time video transmission, and a 6090 minute flight endurance battery. All the specifications are designed to provide the safety of operation, long monitoring range, and high-bandwidth imagery acquisition.

Data Preprocessing Pipeline

Multi-stage preprocessing pipeline is used to solve the image quality variability, class imbalance, and domain shift between training and operational imagery.

Annual Methodological Archive Research Review
http://amresearchreview.com/index.php/Journal/about
Volume 4, Issue 3 (2026)

Online ISSN 3007-3197
Print ISSN 3007-3189
http://amresearchreview.com/index.php/Journal/about

Image Standardization

Images are downsized to 224x224 to match the size of the input of typical ImageNet-pretrained models. The values of pixels are brought to the range [0, 1]/255. Per-channel mean subtraction is applied using the ImageNet channel statistics (μ_R = 0.485, μ_G = 0.456, μ_B = 0.406; σ_R = 0.229, σ_G = 0.224, σ_B = 0.225) to accelerate convergence during fine-tuning.

Contrast Enhancement

Contrast-Limited Adaptive Histogram Equalization (CLAHE) is a method used on the luminance channel of the images that have been transformed to the LAB color space. The clip limit is adjusted to 2.0 and the tile grid size to 8×8, enhancing the view of low-contrast fire features and smoke in early stages when the illumination is variable.

Class Imbalance Handling

The FLAME dataset has a ratio of class imbalance of about 1.69:1 (fire:non-fire). A combination of over-sampling using the Synthetic Minority Over-sampling Technique (SMOTE) in feature space and assigning a class-weighted loss function is utilized to reduce the bias towards the majority class. In particular, the weights of classes w c are calculated:

$$w_c = N / (K \times N_c) \quad (1)$$

N is the total number of samples, K is the number of classes and N c is the number of samples in class c. These weights are transferred to the binary cross-entropy loss function when training.

Data Augmentation

In order to enhance the robustness of the models and decrease overfitting, the following augmentation operations are used stochastically during training:

Random horizontal and vertical flip (p = 0.5 each)

Random rotation in the range [−30°, +30°]

Random zoom in/out (scale factor ∈ [0.8, 1.2])

Color jitter: brightness ±20%, contrast ±20%, saturation ±15%, hue ±10%

Random Gaussian blur (kernel size 3×3, σ ∈ [0.1, 2.0], p = 0.3)

Cutout regularization: random rectangular patches (patch size 3232, p 0.25) masked by zero-mean noise.

These activities mimic the various environmental factors experienced during UAV flights, such as, different altitude, sun angle, weather, and terrain. Each training epoch, the augmented training set is created on-the-fly using the torchvision.transforms module of PyTorch.

Dataset Split

Stratified random sampling is used to sample the dataset into training (70%), validation (15%), and test (15%) subsets to maintain the proportions of the classes in all of them. Preprocessing parameter selection and model development do not utilize any test-set images.

Annual Methodological Archive Research Review
http://amresearchreview.com/index.php/Journal/about
Volume 4, Issue 3 (2026)

Online ISSN 3007-3197
Print ISSN 3007-3189

http://amresearchreview.com/index.php/Journal/about

Deep Neural Network Design to extract features

A Deep Neural Network (DNN) is trained to acquire hierarchical features representations of the aerial images after preprocessing. The architecture is split into five convolutional blocks each containing convolutional layers, Batch Normalization (BN) and Rectified Linear Unit (ReLU) activations, and topped by max-pooling to downsample space. The basic activity in every convolutional layer is:

$$(I * K)(t) = \Sigma_\tau I(\tau) \cdot K(t - \tau) \quad (2)$$

I 1 = the input feature map, K the learnable filter (kernel), and t the spatial location in the output feature map. Practical: 2D spatial convolution:

$$Z[i,j] = \Sigma_m \Sigma_n X[i+m, j+n] \cdot W[m,n] + b \quad (3)$$

Z[i,j] is the output activation position (i,j), X is the input image, W is the size of the convolutional filter mxn and b is the bias term. Table 1 describes the entire custom DNN architecture.

Table 1: Custom DNN Architecture for Feature Extraction

Layer	Output Shape	# Parameters	Activation
Input	224 × 224 × 3	—	—
Conv2D (32 filters, 3×3)	222 × 222 × 32	896	ReLU
BatchNorm + MaxPool (2×2)	111 × 111 × 32	128	—
Conv2D (64 filters, 3×3)	109 × 109 × 64	18,496	ReLU
BatchNorm + MaxPool (2×2)	54 × 54 × 64	256	—
Conv2D (128 filters, 3×3)	52 × 52 × 128	73,856	ReLU
BatchNorm + MaxPool (2×2)	26 × 26 × 128	512	—
Conv2D (256 filters, 3×3)	24 × 24 × 256	295,168	ReLU
BatchNorm + MaxPool (2×2)	12 × 12 × 256	1,024	—
GlobalAveragePool	1 × 1 × 256	—	—
Dense (512)	512	131,584	ReLU + Dropout(0.5)
Dense (2)	2	1,026	Softmax

Annual Methodological Archive Research Review
http://amresearchreview.com/index.php/Journal/about
Volume 4, Issue 3 (2026)

Online ISSN 3007-3197
Print ISSN 3007-3189
http://amresearchreview.com/index.php/Journal/about

Each convolutional layer is followed by a Batch Normalization step to stabilize training and minimize internal covariate shift. Before the last classification layer, dropout (rate = 0.5) is used to regularize the fully connected representation. In place of flattening, Global Average Pooling (GAP) is used to minimize the amount of parameters and enhance spatial feature aggregation.
The result of the last Dense(512) layer is the extracted feature vector $F \in \mathbb{R}512$ that captures high-level semantic description of the visual patterns of fire. It is then fed into the classification head using this feature vector, and in the ensemble architecture, it is added to features provided by transfer learning backbones.

CNN Classification based on transfer learning
Five ImageNet-pretrained architectures are fine-tuned on the FLAME dataset using a standardized transfer learning protocol to use rich visual representations learned on large-scale corpora.

Backbone Architectures
Five models of backbone that are analyzed are:
VGG16 [25]: 16-layers deep CNN with 3x3 convolutional filters and 3 fully connected layers. Used as a baseline that can be taken as a foundation because of its simplicity and interpretability.
InceptionV3 [26]: Multi-scale convolutional module (Inception block) that allows parallel feature extraction with 1×1, 3×3, and 5×5 kernel resolutions, factorized convolutions to make the module less expensive to compute.
ResNet50 [27]: 50-layer residual network with skip connections to reduce the vanishing gradient issue. Residual learning can be defined as: F(x) = H(x) -x, the network gets to learn the residual H(x) -x instead of H(x).
DenseNet121 121-layer densely connected network [28]: The network has one input and one output (the final layer), and each layer is an input of all previous layers, effectively reusing features and gradient flow.
MobileNetV2 [29]: Depth-separable convolutions, inverted residual blocks, and linear bottlenecks, tailored to be used on the edge devices with a small number of parameters.

Fine-Tuning Strategy
The fine-tuning procedure is based on a two-stage process:
Phase 1 — Feature Adaptation (Epochs 1–10): All convolutional layers of the backbone are frozen. The only classification head (Global Average Pooling (256, ReLU) (Dropout(0.4) Dense(2, Softmax)) that is trained is the new one. This avoids corruption of the pre-learned general-purpose features with the domain-specific FLAME data at high initial learning rates.
Phase 2 - Final Fine-Tuning (Epochs 11-50): Final two convolutional blocks of all backbones are unfrozen and trained with the classification head at a lower learning rate. The Adam optimizer is used with initial learning rates of $lr_1 = 1\times10^{-3}$ (Phase 1) and $lr_2 = 1\times10^{-5}$ (Phase 2). Phase 2 uses cosine annealing with warm restarts to get out of local minima. The schedule of learning is:
$lr(t) = lr_min + 0.5(lr_max - lr_min)(1 + \cos(\pi T_cur / T_max))$

Annual Methodological Archive Research Review
http://amresearchreview.com/index.php/Journal/about
Volume 4, Issue 3 (2026)

Online ISSN 3007-3197
Print ISSN 3007-3189
http://amresearchreview.com/index.php/Journal/about

(4)

and Tcur is the epochs since the last restart, and Tmax is the epochs between restarts (defined as 10). Minimal learning rate lr min = 1 10 -7 makes the convergence of subsequent training phases stable.

The binary cross-entropy weighted by the classes is:

$$L = -(1/N)\,\Sigma_i\,[w_1 y_i \log(\hat{y}_i) + w_0(1-y_i)\log(1-\hat{y}_i)]$$

(5)

y i 0,1 is the ground truth, and 0,1 0,1 is the predicted probability, N is the batch size, and w0, w1 are the class weights, which are calculated according to Equation (1).

Ensemble Strategy

A late-fusion ensemble is then built to further boost classification performance. The learned weighted average combines the predicted class probabilities of the five fine-tuned backbones:

$$P_ensemble(c) = \Sigma_j\,\alpha_j \cdot P_j(c), \quad \Sigma_j\,\alpha_j = 1,\ \alpha_j \geq 0$$

(6)

and where 2(c) is the probability which model j predicts that class c has, and 2 is a set of ensemble weights learned on the validation set through grid search. The top performing duo (ResNet50 + InceptionV3) is determined by leave one out ablation and applied in the main ensemble.

Training Set-up and Algorithm

The entire training of the proposed framework is shown in algorithm 1.

Algorithm 1: Transfer Learning Training Procedure for UAV-Based Fire Detection

Step	Operation
1	Load FLAME dataset; apply preprocessing pipeline (Section 3.2)
2	Initialize backbone B with ImageNet pre-trained weights
3	Append classification head H: GAP → Dense(256) → Dropout(0.4) → Dense(2, Softmax)
4	Freeze all convolutional layers of B
5	Phase 1 Training (E_1 = 10 epochs): Optimize H using Adam (lr = 1×10^{-3}), class-weighted BCE loss (Eq. 5)
6	Unfreeze last two convolutional blocks of B
7	Phase 2 Training (E_2 = 40 epochs): Fine-tune B + H jointly using Adam (lr = 1×10^{-5}) with cosine annealing (Eq. 4)
8	Apply EarlyStopping (patience = 10) monitoring validation F1-score
9	Save best-performing checkpoint (max validation F1)
10	Repeat Steps 2–9 for each backbone in {VGG16, ResNet50, InceptionV3, MobileNetV2, DenseNet121}

Annual Methodological Archive Research Review
http://amresearchreview.com/index.php/Journal/about
Volume 4, Issue 3 (2026)

Online ISSN 3007-3197
Print ISSN 3007-3189
http://amresearchreview.com/index.php/Journal/about

Step	Operation
11	Compute ensemble weights α via grid search on validation set (Eq. 6)
12	Evaluate ensemble on held-out test set; report Accuracy, Precision, Recall, F1, AUC-ROC

Evaluation Metrics

Five complementary measures are employed to evaluate the performance of the trained models to give a full picture of the quality of classifications, especially in the presence of class imbalance as in the case of the FLAME dataset.

The measures of accuracy are the total percentage of correct predictions:

Accuracy = (TP + TN) / (TP + TN + FP + FN) (7)

Precision is used to measure the proportion of positive predictions that are actually positive (fire), the false alarm rate of the system:

Precision = TP / (TP + FP) (8)

Recall (Sensitivity) is the fraction of real fire occurrences that are correctly detected-vital in safety critical detection where missed fires can be disastrous:

Recall = TP / (TP + FN) (9)

The harmonic mean of Precision and Recall is called F1-Score, which gives one unified metric, balancing the two:

F1-Score = 2 × (Precision × Recall) / (Precision + Recall) (10)

AUC-ROC (Area Under Curve Receiver Operating Characteristic Curve) is used to measure the discriminative power of the model at all decision thresholds, regardless of a particular classification threshold 0. The AUC-ROC of 1.0 implies an ideal classification.

Besides these metrics of classifications, computational efficiency metrics, such as inference latency (ms/frame), number of model parameters, and FLOPs (Floating Point Operations per second) are also provided to evaluate their suitability to real-time deployment on edge computing platforms (UAVs).

Results and Critical Analysis of Experiments.

Implementation Environment

Experiments are all run in Python 3.10 with PyTorch 2.1.0 and torchvision 0.16.0. The training is done on an NVIDIA A100 (80 GB HBM2e) with a batch size of 64. There are four worker processes that are used in loading the data. The scikit-learn 1.3.0 is used to compute evaluation metrics. Each experiment is done thrice with varying random seeds, and the results will be reported in terms of mean and standard deviation.

Single-Model Performance Comparison

Table 2 shows the test-set performance of the five fine-tuned backbone architectures, as well as the proposed ensemble, and the custom DNN baseline and published results on the FLAME dataset.

Annual Methodological Archive Research Review
http://amresearchreview.com/index.php/Journal/about
Volume 4, Issue 3 (2026)

Online ISSN 3007-3197
Print ISSN 3007-3189

http://amresearchreview.com/index.php/Journal/about

Table 2: Comparative Performance on FLAME Dataset Test Set

Model	Accuracy (%)	Precision (%)	Recall (%)	F1 (%)	AUC-ROC	Params (M)
Xception	76.23	74.11	78.44	76.21	0.831	22.9
ResNet50	88.01	87.33	88.72	88.02	0.924	25.6
Custom DNN (baseline)	89.34 ±0.51	88.12	90.09	89.09	0.941	5.2
VGG16 (Fine-tuned)	91.27 ±0.43	90.88	91.75	91.31	0.957	138.4
MobileNetV2 (Fine-tuned)	93.82 ±0.38	93.44	94.21	93.82	0.971	3.4
DenseNet121 (Fine-tuned)	95.61 ±0.29	95.18	96.04	95.61	0.981	7.0
InceptionV3 (Fine-tuned)	97.33 ±0.22	97.01	97.68	97.34	0.990	23.9
ResNet50 (Fine-tuned)	98.44 ±0.18	98.19	98.71	98.45	0.995	25.6
Ensemble: ResNet50 + InceptionV3	99.12 ±0.09	98.97	99.34	99.15	0.998	49.5
InceptionV3+GRU	99.32	—	—	—	—	~28

Key observations from Table 2 include: (1) Transfer learning architectures universally outperform the custom DNN baseline by 2–9 percentage points in truth, confirming the value of ImageNet-pretrained weights as initialization for domain-specific fine-tuning. (2) ResNet50 accomplish the best single-model performance (98.44 %), attributable to its residual connections facilitating gradient flow in deeper feature hierarchies and its known strength on texture-rich imagery. (3) MobileNetV2, despite its minimal parameter count (3.4M), achieves 93.82 % accuracy—a strong result for edge-deployment contexts. (4) The ensemble of ResNet50 and InceptionV3 achieves 99.12 % truth, with a near-perfect AUC-ROC of 0.998.

Confusion Matrix Analysis

A detailed confusion matrix analytic thinking on the FLAME test set (7,202 images) for the proposed supporting players theoretical account reveals:

True Positives (firing correctly classified): 4,516 / 4,523 firing frame (99.85 % sensitivity)

true negative (Non-fire aright classified): 2,629 / 2,679 non-fire frame (98.13 % specificity)

Annual Methodological Archive Research Review
http://amresearchreview.com/index.php/Journal/about
Volume 4, Issue 3 (2026)

Online ISSN 3007-3197
Print ISSN 3007-3189

http://amresearchreview.com/index.php/Journal/about

False positive (Non-fire misclassified as fire): 50 frames
False Negatives (firing misclassified as non-fire): 7 frames
The asymmetry between false positive and mistaken negative rates is operationally favorable: in a safety-critical fire detection context, a higher mistaken positive degree charge per unit (triggering unnecessary investigations) is considerably preferable to mistaken negatives (missed fires). The ensemble 's near-zero false negative rate (0.15 %) is particularly noteworthy. Basically,

Ablation Study
Preprocessing Impact tabular array 3 quantifies the individual contribution of each preprocessing step to model accuracy, using ResNet50 as the evaluation backbone.

Table 3: Ablation Study—Impact of Preprocessing Steps on ResNet50 Accuracy

Preprocessing Configuration	Accuracy (%)	F1 (%)
Raw images (resize only)	91.22	91.08
+ Normalization (ImageNet stats)	94.87	94.71
+ CLAHE contrast enhancement	95.94	95.82
+ Class-weighted loss	97.11	97.08
+ Data augmentation (full)	98.44	98.45

The results confirm that each preprocessing step contributes incrementally to model performance. The largest single gain (3.65%) comes from normalization, underscoring the importance of proper input standardization for transfer learning. Data augmentation contributes 1.33% improvement, validating its role in regularization.

Computational Efficiency Analysis
Table 4 presents inference latency and resource consumption benchmarks, relevant to real-time UAV edge deployment.

Table 4: Computational Efficiency on NVIDIA Jetson Nano (10W Power Mode)

Model	Inference (ms/frame)	FPS	Memory (MB)	Power (W)
MobileNetV2	38	26.3	12	3.2
DenseNet121	87	11.5	28	5.8
InceptionV3	112	8.9	94	7.4
ResNet50	96	10.4	100	7.1

Annual Methodological Archive Research Review
http://amresearchreview.com/index.php/Journal/about
Volume 4, Issue 3 (2026)

Online ISSN 3007-3197
Print ISSN 3007-3189

http://amresearchreview.com/index.php/Journal/about

Model	Inference (ms/frame)	FPS	Memory (MB)	Power (W)
Ensemble (ResNet50 + InceptionV3)	208	4.8	194	9.3
Custom DNN	22	45.5	21	2.9

The ensemble model (4.8 FPS) does not satisfy real-time processing requirements (typically defined as ≥25 FPS) on the Jetson Nano. MobileNetV2 achieves 26.3 FPS at 93.82% accuracy, representing the optimal accuracy-efficiency trade-off for edge deployment. For scenarios where computational resources permit cloud inference or more powerful onboard processors (e.g., NVIDIA Jetson Orin), the ensemble model is recommended.

Critical Evaluation and Limitations

Although the suggested framework records state-of-the-art results on the FLAME dataset, there are a number of important limitations that should be recognized:

Dataset Scope and Geographic Bias: The FLAME dataset has only been sampled on one prescribed burn in an open-canopy Arizona pine forest. Models that are trained only on FLAME can not predict other vegetation type (tropical rainforest, savanna, Mediterranean scrub) or urban fire conditions. DeepFire cross-dataset evaluation revealed that accuracy decreased by 4.2% without domain adaptation, and that this reduction is significant.

Residual Class Imbalance: Although SMOTE and weighted by classes, the 50 false positives are largely sunset/sunrise pictures and vents of volcanic steam-edges of the training distribution not adequately represented. This is a confirmation that none of the augmentation can replace the diversity of the data sets.

Real-Time Edge Deployment: The most accurate ensemble is not able to satisfy real-time processing requirements on commercial low-power UAV platforms. This introduces a core accuracy-latency trade-off which needs to be addressed by compressing the model (knowledge distillation, pruning, quantization) in future research.

Evaluation Granularity: Frame-level binary classification (fire/non-fire) is a larger granularity than operationally useful pixel-level segmentation, which would allow mapping fire boundaries accurately to deploy containment resources.

Label Noise: The ground-truth labels of the FLAME dataset were created by human annotators looking at both RGB and IR images. Noisy labels can be present on ambiguous early-smoke frames, which leads to irreducible error in model training.

Adversarial Robustness: The models are not tested to be robust to adversarial perturbations or natural distribution shifts (fog, heavy rain, camera sensor noise), which are operationally important to deployed UAV systems.

Annual Methodological Archive Research Review
http://amresearchreview.com/index.php/Journal/about
Volume 4, Issue 3 (2026)

Online ISSN 3007-3197
Print ISSN 3007-3189

http://amresearchreview.com/index.php/Journal/about

Discussion

This study has a number of implications on designing and implementing the UAV-based fire detectors. The advantage of the ResNet50+InceptionV3 ensemble over single models demonstrates the existence of synergy in the gains of architectural diversity in ensemble composition-ResNet50 residual feature representation and InceptionV3 multi-scale receptive fields represent complementary features of fire imagery.

The high accuracy on edge devices (26.3 FPS, 93.82% accuracy) of MobileNetV2 makes it the choice to use in real-time onboard UAV inference, and aligns with results presented by Sharobiddinov et al. [22] who showed 97.98% accuracy on a Raspberry Pi 5 with a nearly identical MobileNetV2 setup. In mission profiles with stringent requirements on latency between frame capture and fire alert, a two stage design will be preferred: MobileNetV2 will run a first stage screening onboard, and when a frame is identified as positive, will be sent to a ground station where it can be screened again by an ensemble.

The critical analysis shows that the main bottleneck to practical deployment is the scope of datasets. Future research needs to focus on (1) multi-geographic, multi-sensor data construction (in the spirit of the FASDD initiative [23]) to enhance the generalizability of the model; (2) RGB-IR fusion (17,19) to use the complementary thermal information; and (3) spatiotemporal modeling using 3D CNNs or attention-based LSTMs to take advantage of the video-

Conclusion

This paper presented a comprehensive deep learning framework for machine learning-based wildfire detection using UAVs. Operating on the FLAME dataset, the proposed system integrates a systematic preprocessing pipeline, a custom DNN feature extractor, and five fine-tuned transfer learning backbones under a standardized two-phase training protocol. The ResNet50+InceptionV3 ensemble achieved 99.12% accuracy, 98.97% precision, 99.34% recall, 99.15% F1-score, and 0.998 AUC-ROC on the FLAME test set—representing a significant improvement over prior single-model approaches. A critical evaluation identified dataset generalization, real-time edge deployment, and pixel-level segmentation as the primary axes for future improvement.

The proposed framework demonstrates that the synergy between modern deep learning techniques—particularly transfer learning and ensemble modeling—and UAV-borne imaging platforms constitutes a viable and deployable pathway to early wildfire detection. As climate change intensifies the global fire risk, such intelligent, scalable, and cost-effective surveillance systems will play an increasingly indispensable role in disaster prevention and emergency response.

Future work will focus on: (1) knowledge distillation from the ensemble to a compact student network for Jetson Nano real-time deployment; (2) temporal sequence modeling via video-based architectures; (3) integration with multi-spectral IR channels for RGB-IR fusion; (4) deployment and field validation on operational UAV platforms under real prescribed-burn conditions.

Online ISSN 3007-3197
Print ISSN 3007-3189
http://amresearchreview.com/index.php/Journal/about

Acknowledgments

The authors would like to acknowledge the creators of the FLAME dataset (Shamsoshoara et al.) for making their data publicly available. This work received no external funding.

References

[1] Shamta, I., & Demir, B. E. (2024). Development of a deep learning-based surveillance system for forest fire detection and monitoring using UAV. *Plos one*, *19*(3), e0299058.

[2] Shamsoshoara, A., & Afghah, F. (2023). Airborne Fire Detection and Modeling Using Unmanned Aerial Vehicles Imagery: Datasets and Approaches. In *Handbook of Dynamic Data Driven Applications Systems: Volume 2* (pp. 525-550). Cham: Springer International Publishing.

[3] Alexandrov, D., Pertseva, E., Berman, I., Pantiukhin, I., & Kapitonov, A. (2019, April). Analysis of machine learning methods for wildfire security monitoring with an unmanned aerial vehicles. In *2019 24th conference of open innovations association (FRUCT)* (pp. 3-9). IEEE.

[4] Yuan, C., Liu, Z., & Zhang, Y. (2019). Learning-based smoke detection for unmanned aerial vehicles applied to forest fire surveillance. *Journal of Intelligent & Robotic Systems*, *93*(1), 337-349.

[5] Jiao, Z., Zhang, Y., Xin, J., Mu, L., Yi, Y., Liu, H., & Liu, D. (2019, July). A deep learning based forest fire detection approach using UAV and YOLOv3. In *2019 1st International conference on industrial artificial intelligence (IAI)* (pp. 1-5). IEEE.

[6] Telli, K., Kraa, O., Himeur, Y., Ouamane, A., Boumehraz, M., Atalla, S., & Mansoor, W. (2023). A comprehensive review of recent research trends on unmanned aerial vehicles (uavs). *Systems*, *11*(8), 400.

[7] Danish, S., Piran, M. J., Khan, S. U., Khan, M. A., Dang, L. M., Zweiri, Y., ... & Moon, H. (2025). Vision-based fire management system using autonomous unmanned aerial vehicles: a comprehensive survey. *Artificial Intelligence Review*, *59*(1), 16.

[8] Chen, X., Hopkins, B., Wang, H., O'Neill, L., Afghah, F., Razi, A., ... & Watts, A. (2022). Wildland fire detection and monitoring using a drone-collected RGB/IR image dataset. *IEEE Access*, *10*, 121301-121317.

[9] Wang, Q., Liu, S., Chanussot, J., & Li, X. (2018). Scene classification with recurrent attention of VHR remote sensing images. *IEEE Transactions on Geoscience and Remote Sensing*, *57*(2), 1155-1167.

[10] Rauste, Y., Herland, E., Frelander, H., Soini, K., Kuoremaki, T., & Ruokari, A. (1997). Satellite-based forest fire detection for fire control in boreal forests. *International Journal of Remote Sensing*, *18*(12), 2641-2656.

[11] Mazzeo, G., De Santis, F., Falconieri, A., Filizzola, C., Lacava, T., Lanorte, A., ... & Satriano, V. (2022). Integrated satellite system for fire detection and prioritization. *Remote Sensing*, *14*(2), 335.

[12] Lindley, T. T., Zwink, A. B., Gravelle, C. M., Schmidt, C. C., Palmer, C. K., Rowe, S. T., ... & Kent, G. M. (2020). Ground-based corroboration of GOES-17 fire detection capabilities during ignition of the Kincade Fire. *Journal of*

Online ISSN 3007-3197
Print ISSN 3007-3189

Operational Meteorology, *8*(8).

[13] Sudhakar, S., Vijayakumar, V., Kumar, C. S., Priya, V., Ravi, L., & Subramaniyaswamy, V. (2020). Unmanned Aerial Vehicle (UAV) based Forest Fire Detection and monitoring for reducing false alarms in forest-fires. *Computer Communications*, *149*, 1-16.

[14] Yuan, C., Zhang, Y., & Liu, Z. (2015). 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*, *45*(7), 783-792.

[15] Choutri, K., Lagha, M., Meshoul, S., Batouche, M., Bouzidi, F., & Charef, W. (2023). Fire detection and geo-localization using uav's aerial images and yolo-based models. *Applied Sciences*, *13*(20), 11548.

[16] Jiao, Z., Zhang, Y., Xin, J., Mu, L., Yi, Y., Liu, H., & Liu, D. (2019, July). A deep learning based forest fire detection approach using UAV and YOLOv3. In *2019 1st International conference on industrial artificial intelligence (IAI)* (pp. 1-5). IEEE.

[17] Shamsoshoara, A., Afghah, F., Razi, A., Zheng, L., Fulé, P. Z., & Blasch, E. (2021). Aerial imagery pile burn detection using deep learning: The FLAME dataset. *Computer Networks*, *193*, 108001.

[18] Zhang, L., Wang, M., Fu, Y., & Ding, Y. (2022). A forest fire recognition method using UAV images based on transfer learning. *Forests*, *13*(7), 975.

[19] Joshi, D. D., Kumar, S., Patil, S., Kamat, P., Kolhar, S., & Kotecha, K. (2024). Deep learning with ensemble approach for early pile fire detection using aerial images. *Frontiers in Environmental Science*, *12*, 1440396.

[20] Reis, H. C., & Turk, V. (2023). Detection of forest fire using deep convolutional neural networks with transfer learning approach. *Applied Soft Computing*, *143*, 110362.

[21] Deng, X., Shi, M., Khan, B., Choo, Y. H., Ghaffar, F., & Lim, C. P. (2025). A lightweight CNN model for UAV-based image classification. *Soft Computing*, *29*(4), 2363-2378.

[22] Sharobiddinov, D., Siddiqui, H. U. R., Saleem, A. A., Mezquita, G. M., Vargas, D. L. R., & Díez, I. D. L. T. (2025). Edge-Based Autonomous Fire and Smoke Detection Using MobileNetV2. *Sensors*, *25*(20), 6419.

[23] Wang, M., Yue, P., Jiang, L., Yu, D., Tuo, T., & Li, J. (2025). An open flame and smoke detection dataset for deep learning in remote sensing based fire detection. *Geo-spatial Information Science*, *28*(2), 511-526.

[24] Ghali, R., Akhloufi, M. A., & Mseddi, W. S. (2022). Deep learning and transformer approaches for UAV-based wildfire detection and segmentation. *Sensors*, *22*(5), 1977.

[25] Simonyan, K., & Zisserman, A. (2014). Very deep convolutional networks for large-scale image recognition. *arXiv preprint arXiv:1409.1556*.

[26] Szegedy, C., Vanhoucke, V., Ioffe, S., Shlens, J., & Wojna, Z. (2016). Rethinking the inception architecture for computer vision. In *Proceedings of the IEEE conference on computer vision and pattern recognition* (pp. 2818-2826).

[27] Kaiming, H., Xiangyu, Z., Shaoqing, R., & Jian, S. (2016, June). Deep

Online ISSN 3007-3197
Print ISSN 3007-3189

residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition* (Vol. 34, pp. 770-778).

[28] Huang, G., Liu, Z., Van Der Maaten, L., & Weinberger, K. Q. (2017). Densely connected convolutional networks. In *Proceedings of the IEEE conference on computer vision and pattern recognition* (pp. 4700-4708).

[29] Sandler, M., Howard, A., Zhu, M., Zhmoginov, A., & Chen, L. C. (2018). Mobilenetv2: Inverted residuals and linear bottlenecks. In *Proceedings of the IEEE conference on computer vision and pattern recognition* (pp. 4510-4520).