*applied sciences*

MDPI

*Article*

# UAV-Based Forest Fire Early Warning and Intervention Simulation System with High-Accuracy Hybrid AI Model

**Muhammet Sinan Başarslan [1,*] and Hikmet Canlı [2]**

[1] Department of Computer Engineering, Istanbul Medeniyet University, İstanbul 34700, Turkey
[2] Department of Software Engineering, Istanbul Gedik University, İstanbul 34876, Turkey; hikmet.canli@gedik.edu.tr
* Correspondence: muhammet.basarslan@medeniyet.edu.tr

### Featured Application

The proposed system can be directly applied to early forest fire detection and rapid response planning using UAV-based surveillance infrastructures. By combining a high-accuracy hybrid deep learning model with a balanced drone task assignment algorithm, the system enables real-time identification of fire events and efficient allocation of available UAV resources. This approach is particularly suitable for large forest areas, national parks, and wildfire-prone regions, where fast intervention and optimal resource utilization are critical. The system can support decision-makers by reducing false alarms, minimizing response time, and improving energy-efficient deployment of firefighting drones.

### Abstract

In this study, a hybrid deep learning model that combines the VGG16 and ResNet101V2 architectures is proposed for image-based fire detection. In addition, a balanced drone guidance algorithm is developed to efficiently assign tasks to available UAVs. In the fire detection phase, the hybrid model created by combining the VGG16 and ResNet101V2 architectures has been optimized with Global Average Pooling and layer merging techniques to increase classification success. The DeepFire dataset was used throughout the training process, achieving an extremely high accuracy rate of 99.72% and 100% precision. After fire detection, a task assignment algorithm was developed to assign existing drones to fire points at minimum cost and with balanced load distribution. This algorithm performs task assignments using the Hungarian (Kuhn–Munkres) method and cost optimization, and is adapted to direct approximately equal numbers of drones to each fire when the number of fires is less than the number of drones. The developed system was tested in a Python-based simulation environment and evaluated using performance metrics such as total intervention time, energy consumption, and task balance. The results demonstrate that the proposed hybrid model provides highly accurate fire detection and that the task assignment system creates balanced and efficient intervention scenarios.

**Keywords:** forest fire; hybrid learning; real-time task assignment; VGG16; ResNet101V2



Academic Editor: Cheol-Hong Hwang

Received: 28 December 2025
Revised: 12 January 2026
Accepted: 21 January 2026
Published: 23 January 2026



## 1. Introduction

Forest fires are among the environmental disasters that are increasing in frequency and severity worldwide due to climate change, rising temperatures, low humidity, and human negligence [1,2]. Particularly in regions such as the Mediterranean basin, South America, Australia, and California, the growing areas of wildfires each year are both destroying ecosystems and causing economic and human losses [3,4]. Traditional forest fire detection methods rely on tools such as observation towers, thermal cameras, satellite imagery, and human surveillance. However, these methods often face issues such as delayed alerts, high false positive rates, or high costs [5,6]. Satellite-based systems may be insufficient for fires requiring rapid intervention due to their low imaging frequency. Therefore, the need for real-time, automated fire detection systems with low latency is growing [6,7]. In recent years, advancements in computer vision and deep learning, particularly convolutional neural network (CNN)-based approaches, have demonstrated superior performance in tasks such as image classification [8–10]. Thanks to transfer learning methods, models pre-trained on large datasets (e.g., ImageNet) can be reused for specific tasks such as forest fire detection, achieving high accuracy rates [11,12]. Such approaches provide strong generalization capabilities even on specialized datasets with a limited number of examples [12,13].

In this study, a hybrid deep learning model combining the VGG16 and ResNet101V2 architectures for image-based fire detection is proposed; then, a balanced drone guidance algorithm capable of assigning tasks to enable real-time intervention for detected fires is developed. The task assignment process is based on the classical Hungarian algorithm [14–16]; however, it has been restructured to direct approximately an equal number of drones to each fire when the number of fires is less than the number of drones, with the aim of achieving a balanced task distribution.

In the experiments conducted, VGG and ResNet derivatives were tested separately on the DeepFire dataset, and the VGG16 and ResNET101 models were determined to be the best performers with an accuracy rate of 98.16%. Based on this observation, a hybrid architecture combining the advantages of both models was proposed; integration was achieved using Global Average Pooling and layer concatenation methods. The proposed hybrid model outperformed the best performance reported in the literature with an accuracy rate of 99.72% and a precision value of 100% on the same dataset, thereby establishing a new benchmark level of performance in image-based fire detection.

The task assignment algorithm is structured not only according to the minimum cost criterion but also according to the load balancing principle. The developed system was tested in a Python version 3.14.0-based simulation environment; it was observed that the number of drones assigned to each fire was evenly distributed, total energy consumption was minimized, and the average distance was optimized. The system provides advantages in terms of both operational efficiency and time, as it can perform flexible and real-time task planning based on the instantaneous fire location.

This study contributes to the literature in the following ways:

- A new classifier has been proposed that surpasses the accuracy rates in the literature by hybridizing VGG16 and ResNet101V2 for image-based fire detection.
- An adaptive task assignment system focused on both cost and load balancing has been designed for the post-fire intervention phase.
- The entire system has been tested end-to-end in a simulation environment, demonstrating its applicability. In this regard, the study offers a holistic approach that combines image-based disaster detection with autonomous intervention mechanisms.

In Section 2 of this study, studies conducted in the literature on forest fires are examined in detail. In Section 3, the material and methods are discussed, and the data set used, the deep learning methods used, and the proposed model are explained. In Section 4, the experimental results obtained are discussed and conclusions are drawn. In Section 5, the success of the proposed method and its contribution to the literature are explained, and recommendations for future studies are made.

https://doi.org/10.3390/app16031201

## 2. Literature Review

In recent years, deep learning and transfer learning-based methods have been extensively used in the field of forest fire detection. Some recent studies in the literature are summarized below:

Khan and his colleagues made a significant contribution to the literature by creating a new and original forest fire dataset called DeepFire. The researchers used the VGG19 deep learning model, which works with a transfer learning approach, on this dataset. The model achieved 95.7% accuracy and 94.2% recall, demonstrating high success in fire detection. They supported real-time monitoring with UAV-based system recommendations [5]. In their study conducted on the FLAME and DeepFire datasets in [17], they used the EfficientNetB7 architecture and the Attention-Connected Network (ACNet) model they developed specifically for this purpose. With these two deep learning-based models, they achieved approximately 98% accuracy and F1 score. The ACNet model uses an attention mechanism to better extract visual features and enable the model to make more accurate decisions.

This study demonstrates how effective attention-based model configurations can be, particularly in complex forest fire images. Supriya and Gadekallu present an innovative approach that combines federated learning with particle swarm optimization (PSO) methods for the early detection of forest fires. The model enables the construction of systems that can learn while preserving data privacy in decentralized environments. The achieved accuracy rate of 94.47% demonstrates that such distributed systems can operate securely and effectively. This approach offers significant advantages, particularly in scenarios where collecting data on a central server is difficult or undesirable [18]. In [19], they proposed a lightweight and fast fire detection model called FFireNet, built on the MobMobileNetV2 architecture. This model is designed for use on mobile devices or embedded systems due to its low hardware requirements. The FFireNet model successfully classified fire images with 98.42% accuracy and 99.47% recall rate. Ghali and Akhloufi's proposed CT-Fire model is a hybrid architecture that combines classical convolutional neural networks (CNN) with transformer-based structures. When tested on ground images, the model achieved 99.62% accuracy. Thanks to its transformer structure, the model can learn long-range relationships more effectively and identify complex fire patterns with greater accuracy. This hybrid approach demonstrates the practical benefits of attention mechanisms, which have gained popularity in deep learning in recent years [3]. In their study, Gupta and Nihalani developed a hybrid model by combining powerful convolutional neural network architectures such as ResNet152V2 and InceptionV3. This model demonstrated very high performance in fire detection applications with a precision rate of 99.47%. This system, optimized for real-time fire detection, stands out not only for its high accuracy but also for its fast response times [6]. Saputra and Adhinata, a deep learning model developed using the DenseNet201 architecture, has demonstrated high performance with a 99% accuracy rate. The DenseNet structure enables the reuse of features by establishing dense connections between layers, thereby yielding more compact and efficient models. Thanks to this feature, effective fire classification has been achieved even in situations where the amount of data is limited [20].

In the study conducted in [21], the performance of three different deep learning models, ResNet50, Xception, and MobileViT, was compared in the task of forest fire detection. According to the results obtained, the modified Xception model yielded the best result with 99.22% accuracy. Xception reduces the number of parameters while increasing the model's learning capacity thanks to depth-separated convolutions. This study presents a comparative analysis of different architectures, demonstrating the effects of design choices on detection accuracy. Alkhammash, on the other hand, combined YOLOv11-based object detection models with the MobileNetV2 architecture to classify forest fires by type. The system achieved a 97% accuracy rate, successfully determining the type and spread of fires. Real-time detection and object localization became possible thanks to the YOLO architecture.

This system can not only detect the presence of fire but also classify it in detail [7]. Idroes and colleagues developed a new model called TeutongNet, which is based on a modified ResNet50V2 architecture, and achieved an accuracy rate of 98.68%. It has been noted that the model stands out for its particularly low false positive (FP) and false negative (FN) rates. In this regard, TeutongNet offers a suitable option for applications requiring high sensitivity. Minimizing misclassifications in the detection of critical events such as fires is of great importance [22]. In [23], forest fire classification was performed using large language models (LLMs). It has been demonstrated that multimodal models capable of image-supported analysis, such as GPT-4o, released on 13 May 2024, can play a complementary role in fire detection systems. Large language models can enhance the accuracy of fire-related decision support systems because they have the capacity to analyze both visual data and textual descriptions. In his study, Diaconu examined machine learning-based forest fire detection systems, classified existing methods, and analyzed their strengths and weaknesses. The author identified gaps in the existing literature and provided guiding recommendations for future research [4].

Table 1 presents a comparison of studies conducted on forest fires in the literature. These studies show that image-based systems for forest fire detection have become popular in recent years and that transfer learning-based approaches provide high accuracy in fire detection tasks.

**Table 1.** Comparison of Studies Conducted for Forest Fire Detection.

| No | Study | Year | Model/Architecture | Data Set | Performance Metrics | Contribution |
|---|---|---|---|---|---|---|
| 1 | Khan et al. [5] | 2022 | VGG19 + Transfer Learning | DeepFire | 95.7% Acc, 94.2% Recall | Real-time surveillance with UAV-based systems |
| 2 | Sokolova et al. [17] | 2009 | EfficientNetB7, ACNet | FLAME, DeepFire | ~98% Acc and F1 | Model optimized with attention mechanism |
| 3 | Supriya & Gadekallu [18] | 2023 | Federated Learning + PSO | - | 94.47% Acc | Early detection with distributed systems |
| 4 | Khan & Khan [19] | 2022 | FFireNet (Based on MobileNetV2) | - | 98.42% Acc, 99.47% recall | Lightweight, fast mobile-friendly model |
| 5 | Ghali & Akhloufi [3] | 2023 | CT-Fire (CNN + Transformer) | Ground Images | 99.62% Acc | Hybrid architecture, high precision |
| 6 | Gupta & Nihalani [6] | 2024 | ResNet152V2 + InceptionV3 | - | 99.47% Precision | Hybrid architecture optimized for real-time detection |
| 7 | Saputra [20] | 2023 | Based on DenseNet201 | - | 99% Acc | Use of deep and effective convolution network |
| 8 | Davis & Shekaramiz [21] | 2023 | ResNet50, Xception, MobileViT | - | 99.22% Acc | Model comparative analysis, Xception advantage |
| 9 | Alkhammash (1) [7] | 2025 | YOLOv11 + MobileNetV2 | - | 97% Acc | Classification specific to fire types |

**Table 1.** *Cont.*

| No | Study | Year | Model/Architecture | Data Set | Performance Metrics | Contribution |
|---|---|---|---|---|---|---|
| 10 | Idroes et al. [22] | 2023 | TeutongNet (modified ResNet50V2) | - | 98.68% Acc | Low FP/FN rates, emphasis on reliability |
| 11 | Alkhammash (2) [23] | 2025 | LLM + Image Analysis (ChatGPT) | - | - | Image-assisted analysis with ChatGPT |
| 12 | Diaconu [4] | 2023 | Review study | Various | - | Systematically analyzed gaps in the literature |

## 3. Materials and Methods

In this section, the data set used in the study, the proposed model, and the transfer learning methods used to derive this model will be discussed in detail, along with the experimental setup for the balanced drone assignment simulation.

### *3.1. Deep Fire Data Set*

The dataset used in this study is an open-source image collection titled "Dataset for Forest Fire Detection" published by Ali Khan and Bilal Hassan on the Mendeley Data platform [5]. The dataset consists of a total of 1900 images in RGB format, each measuring 250 × 250 pixels. Of these images, 950 belong to the fire-containing class, while the remaining 950 belong to the fire-free class.

Figure 1 shows examples of images with and without fires taken from the deep fire dataset. Here, factors such as smoke from fires, sunsets, sun glare, and red leaves in autumn stand out as elements that make it difficult for models to detect fires.

**Figure 1.** Examples of images with and without fire.

The DeepFire dataset was divided into training, validation, and test sets using a stratified split to preserve class balance. Images belonging to the same scene were not shared across different subsets, preventing data leakage. The test set was kept completely unseen during training and model selection.

### *3.2. Challenges in Forest Fire Image Classification*

Forest fire image classification presents unique challenges that limit the effectiveness of single deep learning architectures. Fire-related visual patterns are highly heterogeneous and often ambiguous due to environmental factors. Early-stage fires may appear as small flame regions with low contrast against complex natural backgrounds. Smoke can be visually similar to clouds or fog, while sunlight reflections, sunset illumination, and seasonal color variations (e.g., red and brown foliage in autumn) may lead to false positives.

These challenges require a model capable of simultaneously capturing fine-grained texture details (e.g., flame edges, smoke dispersion patterns) and high-level contextual semantics (e.g., spatial relationships between fire regions and surrounding vegetation). Empirical observations during preliminary experiments revealed that single CNN architectures tend to specialize in either low-level or high-level representations, resulting in misclassification under visually complex conditions. This motivates the need for a hybrid learning strategy that integrates complementary feature representations.

### *3.3. Deep Learning Models*

#### 3.3.1. ResNet

ResNet (Residual Network) is a structure developed to solve the vanishing gradient problem that arises during learning in deep neural networks [24,25]. In a traditional CNN architecture, the outputs learned in each layer are directly transferred to the next layer, while the ResNet architecture adds identity connections to this transition [26].

This structure can be expressed mathematically as in Equation (1):

$$y = F(x, \{W_i\}) + x \tag{1}$$

Here:
$x$: Input vector
$F$: Residual function to be learned
$y$: Layer output
This allows the network to focus on learning only the residual instead of learning. This ensures that learning remains efficient despite the increase in depth.

In this study, ResNet50, ResNet101, ResNet152, and their v2 versions were used. The v2 versions contain architectural differences in terms of batch normalization and ReLU activation order.

#### 3.3.2. VGGNet

VGGNet is a simple yet deep CNN architecture developed by the Visual Geometry Group (VGG) at Oxford University [27,28]. The VGG16 and VGG19 variants consist of 16 and 19 layers, respectively. All convolutional layers have a $3 \times 3$ filter size, and the same structure is repeated throughout the network. In this architecture, the number of parameters is high, but its structure gives it strong generalization power [12].

The VGG architecture typically works as follows:

- In each block:

$$\text{Conv } (3 \times 3) \rightarrow \text{ReLU} \rightarrow \text{MaxPool } (2 \times 2)$$

- Then:

$$\text{FC} \rightarrow \text{ReLU} \rightarrow \text{Dropout} \rightarrow \text{FC} \rightarrow \text{Softmax}$$

VGG networks are widely preferred, especially in classification problems such as medical imaging, object detection, and fire detection [29,30]. In this study, the VGG16 and VGG19 models were trained separately and with hybrid learning.

#### 3.3.3. Hybrid Learning (VGG16 + ResNet101V2)

In recent years, hybrid approaches obtained by combining different deep neural network architectures have provided effective solutions for improving classification performance, particularly in complex visual recognition tasks. In this study, a hybrid model was designed by integrating the VGG16 and ResNet101V2 architectures for image-based forest fire detection.

Although VGG16 has a relatively shallow architecture, it is highly effective in extracting low-level visual features such as edges, textures, and color transitions. These characteristics are especially important for detecting flame boundaries and smoke patterns. On the other hand, ResNet101V2, with its deeper structure and residual learning mechanism, is capable of learning more abstract and high-level semantic features, allowing it to better represent contextual information related to fire scenes. Preliminary experiments with single-backbone models indicated that low-level texture cues or high-level contextual representations alone were insufficient to robustly discriminate visually ambiguous fire and non-fire scenes. By combining these two architectures, the proposed hybrid model aims to simultaneously learn both local texture-based features and global contextual representations, which are critical for distinguishing fire and non-fire images under challenging environmental conditions [12,31].

The input layer of the proposed model accepts fixed-size images of $224 \times 224 \times 3$. The VGG16 and ResNet101V2 networks are executed in parallel using pre-trained ImageNet weights. The fully connected layers of both backbone networks are removed and replaced with GlobalAveragePooling2D layers to reduce the number of trainable parameters and mitigate overfitting. As a result, VGG16 and ResNet101V2 produce 512- and 2048-dimensional feature vectors, respectively. These feature vectors are concatenated along the channel dimension to form a unified representation, which is then passed to the classification layers without explicit feature weighting, allowing the network to learn the relative importance of each feature set during training. Using transfer learning, only the final layers of the networks are fine-tuned during training, which significantly reduces training time while preserving the generalization capability of the pre-trained models.

Experimental results show that the proposed hybrid model achieved an accuracy rate of 99.72%, outperforming the individual VGG16 and ResNet101V2 models, which achieved accuracy rates of 98.16%. This improvement demonstrates the complementary nature of the fused architectures and indicates that the performance gain is not merely due to increased model depth, but rather to the effective integration of heterogeneous feature representations. The effectiveness of this fusion strategy is further supported by ablation experiments, which confirm that combining features from both backbones yields superior performance compared to single-architecture configurations. Although similar hybrid CNN-based approaches have been reported in the literature [32,33], the use of such a hybrid architecture specifically for forest fire detection remains limited. Therefore, the proposed model provides a novel and effective contribution both in terms of architectural design and practical application.

#### 3.3.4. Simplifying Assumptions

(a) Fire locations are treated as static during a single assignment cycle.
(b) UAV flight speed and energy consumption are assumed constant.
(c) Weather and wind effects are not explicitly modeled.
(d) Communication delays are neglected.
(e) All drones are assumed to be operational and available.

### 3.4. Fire Detection Models

After detecting a fire, the existing drone fleet must be appropriately assigned to the fire locations so that the system can respond quickly and efficiently. This process can be formulated as a multi-task assignment problem and solved within the framework of the classic assignment problem. The objective is to assign the most suitable drone to each fire location in a way that minimizes the time and energy required to respond to all fire zones. However, the classical assignment problem is based on the assumption that each task is assigned to only one drone. In some applications, this is insufficient. Especially in situations like fires that can spread over large areas, sending multiple drones to each fire location is more realistic and effective. Additionally, actively assigning all drones increases the system's efficiency. Therefore, in this study, we propose a new task assignment strategy that ensures a balanced load distribution. We also calculate the distance and energy costs for each drone and evaluate their performance.

The recommended task assignment problem is expressed with the following parameters and variables:

Parameters:

- $D = \{1, 2, \ldots, m\}$: Drone set (total of m drones)
- $F = \{1, 2, \ldots, n\}$: Fire point set (total of n fires)
- $dist_{ij}$: Euclidean distance between drone $i$ and fire $j$
- $\alpha$: Energy weighting coefficient (positive constant)
- $\beta$: Delay time weighting coefficient
- $q$: Base (average) number of drones assigned to each fire
- $m_{drones}$: Total number of drones
- $n_{fires}$: Number of detected fires

Decision Variable: This binary decision variable indicates whether drone $i$ is assigned to fire point $j$ (Equation (2)).

$$x_{ij} = \begin{cases} 1, & if\ i.\ drone\ j.\ is\ assigned\ to\ the\ fire\ point \\ 0, & otherwise \end{cases} \tag{2}$$

Objective Function (Equation (3)):

The total assignment cost, including distance and energy costs for each drone, is minimized:

$$\min_{xij} Z = \sum_{i=1}^{m} \sum_{j=1}^{n} x_{ij} \cdot (dist_{ij} + \alpha.\ energy_{ij}) \tag{3}$$

Energy is defined as follows, depending on distance:

$$energy_{ij} = e.\ dist_{ij}$$

Here, e is the energy consumption coefficient per unit distance. In this case, the objective function simplifies to (Equation (4)):

$$Z = \sum_{i=1}^{m} \sum_{j=1}^{n} x_{ij} \cdot\ dist_{ij} \cdot (1 + \alpha e) \tag{4}$$

Since response time is critical in fires, the formula has been expanded to include a delay coefficient as follows (Equation (5)):

$$Z = \sum_{i=1}^{m} \sum_{j=1}^{n} x_{ij} \cdot\ (\beta.time_{ij} + \alpha \cdot energy_{ij}) \tag{5}$$

When the total number of drones exceeds the number of fires, assignments must be balanced. Here, $q$ is computed as the integer average number of drones per fire to ensure balanced assignment. In this case, the average number of drones assigned to each fire is calculated as follows (Equation (6)):

$$q = \frac{m_{drones}}{n_{fires}} \tag{6}$$

The remaining $r = m_{drones} \bmod n_{fires}$ drone idrones are assigned not randomly but based on the distance-based minimum cost principle, so that they are closest to the fires. Thus, the number of drones assigned to each fire point is (Equation (7)):

$$m_{drones}^{(j)} \in \{q, q+1\}, \qquad \forall\{1, \ldots, n_{fires}\}) \tag{7}$$

This ensures that the number of drones assigned to each fire differs by at most one, providing a balanced distribution. This approach reduces the total cost and balances the intervention burden between fires.

Limitations:

At least one drone should be assigned to each fire location (Equation (8)):

$$\textstyle\sum_{i=1}^{m} x_{ij} \geq 1 \qquad \forall j \in F \tag{8}$$

A drone can only be assigned to one mission (Equation (9)):

$$\textstyle\sum_{j=1}^{n} x_{ij} \leq 1 \qquad \forall i \in D \tag{9}$$

Decision variables are binary (they can take values of 0 or 1) (Equation (10)):

$$x_{ij} \in \{0, 1\} \; \forall i, j \tag{10}$$

Application:

- All drone-fire costs (distance + energy) are calculated.
- An assignment list is created, sorted from lowest to highest according to the cost matrix.
- A maximum of $q$ drones are assigned to each fire.
- The remaining drones are distributed to fires that have not reached the $q$ + 1 level, again based on cost.

The model was solved using the Hungarian Algorithm (Kuhn–Munkres), which can provide a complete solution as long as the solution space is small [14]. This algorithm finds the solution that provides the minimum total assignment cost on a two-dimensional cost matrix in polynomial time [34]. The proposed task assignment strategy is based on the Hungarian algorithm, which has a computational complexity of O($n^3$), where n represents the maximum of the number of drones and fire locations. Given the moderate fleet sizes typically used in UAV-based fire monitoring scenarios, the algorithm can generate assignment plans within negligible computation time. In dynamic wildfire scenarios, the assignment process can be executed iteratively at fixed time intervals, enabling adaptive reallocation of drones as fire conditions evolve. While the current study focuses on static snapshots for simulation purposes, the proposed framework is inherently compatible with real-time updates.

### 3.5. Proposed Approach

In this study, the fire detection and response process is structured as an end-to-end integrated system. The system consists of two main components: (1) image-based fire detection system, (2) real-time task assignment and drone guidance module.

In the first stage, RGB images obtained from fixed or mobile cameras are analyzed by passing them through a trained hybrid model. The model determines with high accuracy whether the image contains a fire. The detected fire images are then subjected to geolocation, and the location information is transmitted to the task assignment system.

In the second stage, all unmanned aerial vehicles (drones) defined in the system are fed into the task assignment algorithm along with their current locations. This module, designed based on the Hungarian algorithm, generates an assignment plan that ensures minimum distance and energy consumption for each drone. This algorithm has been reconfigured to direct approximately the same number of drones to each fire in scenarios where the number of fires is less than the number of drones. This balances the task load and increases intervention efficiency.

Figure 2 shows the general workflow of the system. The process begins with image collection, followed by detection of fires using a hybrid CNN model. Location information is extracted for each detected fire and sent to the task assignment algorithm. The task assignment module assigns the appropriate number of drones to the appropriate fire locations and transmits this information to the command center. In the final stage, task information is transmitted to the drones to initiate intervention. Decisions made at each step are optimized according to predefined threshold values and cost functions. Figure 2 visually summarizes the data flow and processing sequence, revealing both the decision structure and processing sequence of the system. The drone task assignment block illustrates a conceptual representation of the proposed balanced assignment strategy rather than a direct visualization of the Hungarian algorithm steps.

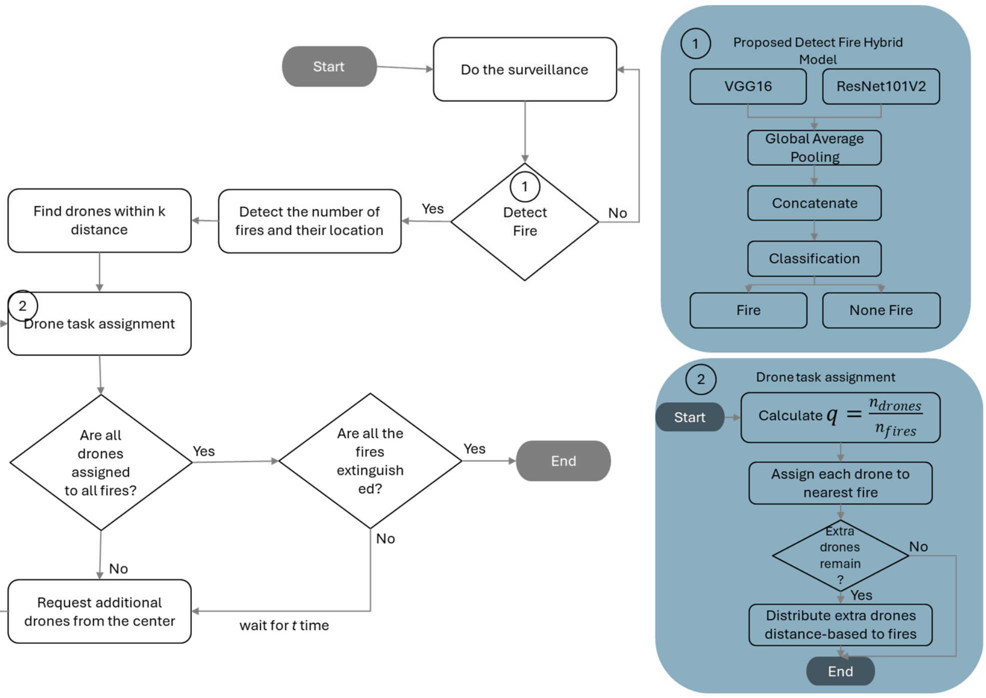


**Figure 2.** Proposed fire detection and real-time intervention architecture.

### *3.6. Experimental Setup*

The hybrid deep learning model proposed in this study was developed using the Python-based Keras API (with TensorFlow infrastructure). The training and testing processes were carried out in a CUDA-supported GPU environment, and the system was equipped with an Intel i7 processor, 32 GB RAM, and NVIDIA RTX 3080 GPU hardware.

The model is configured to accept RGB images with dimensions of 224 × 224 × 3 as input, and all images are normalized prior to training. The two architectures that form the basis of the hybrid structure, VGG16 and ResNet101V2, are preloaded with weights trained on ImageNet and used as feature extractors with the top classification layers removed. This approach has been widely demonstrated in the literature to increase the effectiveness of transfer learning, especially in limited datasets [35]. To prevent model naming conflicts, all layer names were redefined, and only the final layers (the last 15 layers in VGG16 and the last 30 layers in ResNet101V2) were left open for training in both architectures. This strategy preserves pre-learned generic visual features while enabling task-specific adaptation, and has been shown to reduce overfitting in small-scale datasets [36].

The features obtained from both architectures were transferred to Global Average Pooling layers and then combined. The resulting feature vector was passed through three Dense layers with 1024, 256, and 64 neurons, respectively, with 50% Dropout applied after each layer. The Dropout technique is a widely used regularization method that prevents overfitting and improves the model's generalization ability [37]. The model's output layer was configured with a single-neuron sigmoid activation function, and the binary cross-entropy loss function was used for the classification process. The optimization process was performed using the Adam optimization algorithm [38], configured with a small learning rate. The model was trained for up to 50 epochs with a batch size of 32.

Two callback functions were used during training to monitor learning behavior and further mitigate overfitting. EarlyStopping terminated training when the validation loss did not improve for six consecutive epochs, while ReduceLROnPlateau dynamically reduced the learning rate when the validation loss reached a plateau. These mechanisms help prevent excessive parameter updates and stabilize convergence. In addition, class weight parameters were applied during training to compensate for class imbalance and to prevent biased learning toward the dominant class.

To ensure robustness and reproducibility, the dataset was divided into training, validation, and test subsets using a stratified split, preserving class balance across all subsets. The test set was kept completely unseen during training and model selection, thereby preventing data leakage.

All CNN-based models evaluated in this study were trained under identical experimental conditions to ensure a fair comparison. The same data splits, optimization settings, learning rate, batch size, and fine-tuning strategy were applied to all architectures. Early stopping and learning rate scheduling were used uniformly. As a result, the reported performance differences reflect architectural characteristics rather than variations in training configurations.

The computational characteristics of the proposed hybrid model were analyzed to assess its practical feasibility in UAV-assisted wildfire monitoring systems. The hybrid architecture integrates VGG16 and ResNet101V2 backbones, resulting in a model with a relatively high parameter count and memory footprint when deployed in full precision. As summarized in Table 2, while VGG16 remains suitable for lightweight edge execution under constrained settings, ResNet101V2 and the proposed hybrid configuration are more appropriate for deployment on resource-rich edge servers or ground control stations.

Rather than executing the hybrid model directly onboard lightweight UAV platforms, the proposed system follows a hierarchical processing paradigm commonly adopted in UAV-based surveillance. In this paradigm, aerial platforms are responsible for image acquisition, while computationally intensive inference is performed at a centralized ground station or edge server. This architecture enables the system to leverage high-capacity deep learning models for accurate fire detection without exceeding the computational and energy limitations of individual UAVs.

**Table 2.** Model Complexity, Memory Footprint, and UAV/Edge Deployment Suitability.

| Model | Input Size | Backbone Output | Total Parameters | Trainable Parameters | Model Size (FP32) | UAV/Edge Suitability |
|---|---|---|---|---|---|---|
| VGG16 | $64 \times 64 \times 3$ | $2 \times 2 \times 512$ | 14,747,650 | 32,962 | 56.26 MB | Suitable (low-resolution input, frozen backbone, lightweight classifier) |
| ResNet101V2 | $64 \times 64 \times 3$ | $2 \times 2 \times 2048$ | 42,626,560 | Limited (upper layers only) | 163 MB | Conditionally suitable (with quantization/pruning) |

To ensure reproducibility and transparency, the complete training configuration and hyperparameter settings used in this study are summarized in Table 3. The table details the backbone configuration, optimization strategy, learning rate selection, regularization mechanisms, and overfitting mitigation techniques applied during training.

**Table 3.** Model Training Parameters and Optimization Settings.

| Parameter | Configuration Used in This Study |
|---|---|
| Backbone model | ResNet101V2 (weights = ImageNet, include_top = False, input_shape = $64 \times 64 \times 3$) |
| Transfer learning/freezing | All backbone layers frozen (layer.trainable = False) |
| Classification head | GlobalAveragePooling2D $\rightarrow$ Dense (ReLU, tuned units) $\rightarrow$ Dense (2, Sigmoid) |
| Dense layer units | Tuned range: 32–256 (step size = 32) |
| Output layer | Dense (units = 2, activation = sigmoid) |
| Loss function | Binary cross-entropy |
| Evaluation metrics | Precision, Recall, BinaryAccuracy, Accuracy |
| Optimizer selection | Automatically selected via tuner (Adam, Nadam, RMSprop, SGD) |
| Initial learning rate | Tuned within range $1 \times 10^{-4}$ to $1 \times 10^{-2}$ (logarithmic sampling) |
| Learning rate scheduling | ReduceLROnPlateau (monitor = val_loss, patience = 3, factor = 0.3, min_lr = $1\times 10^{-6}$) |
| Optimizer hyperparameters | Default Keras settings (optimizer-dependent) |
| Hyperparameter search method | RandomSearch (objective = val_loss, max_trials = 5) |
| Batch size | Default Keras setting (typically 32) |
| Early stopping | Enabled (monitor = val_loss, patience = 5, restore_best_weights = True) |
| Overfitting mitigation | Early stopping and adaptive learning rate scheduling |
| Training data format | One-hot encoded labels |
| Test evaluation | best_model.evaluate(test_data, test_labels) |

### 3.7. Performance Metrics

Performance evaluation methods such as accuracy, precision, recall, and F-score are used to evaluate models created for classification problems such as image processing [17,39,40]. These methods are obtained from the confusion matrix. The confusion matrix is given in Table 4.

**Table 4.** Confusion Matrix.

| | | Actual Value | |
|---|---|---|---|
| | | Positive | Negative |
| Estimate Value | Positive | True Positive (TP) | False Positive (FP) |
| | Negative | False Negative (FN) | True Negative (TN) |

The accuracy of the model is expressed as the ratio of correctly classified examples to the total number of examples. High accuracy indicates that the model has a high ability to make correct predictions, while low accuracy indicates that improvements need to be made to the model's performance [41]. The calculation of the accuracy value is given in Equation (11).

$$Accuracy = \frac{T_P + T_N}{T_P + F_P + F_N + T_N} \tag{11}$$

Precision refers to the ratio of examples that the model correctly predicts as positive to the actual positives. High precision means that there are few false positives and that most of the examples classified as positive are actually positive [42]. Precision is calculated as in Equation (12).

$$Precision \ = \frac{T_P}{T_P + F_P} \tag{12}$$

Recall is a performance metric used in classification problems. It refers to the ratio of all correct positive examples that are correctly predicted as positive. Sensitivity is an important metric for reducing the number of false negatives and minimizing missing correct positive examples [17,43]. Recall is calculated as in Equation (13).

$$Recall = \frac{T_P}{T_P + F_N} \tag{13}$$

The F-score is a measure frequently used in classification tasks such as information retrieval and machine learning. It provides a balance between precision and recall to evaluate a model's performance. Mathematically, the F-score is defined as the harmonic mean of precision and recall [17]. The F-score is given in Equation (14).

$$F - score = \frac{2 * Precision * Recall}{Precision + Recall} \tag{14}$$

## 4. Experimental Results

This section presents detailed experimental results and simulation outputs of the proposed hybrid model and task assignment algorithm on the DeepFire dataset. Image classification performance was evaluated using metrics, followed by an analysis of intervention efficiency obtained from the task assignment simulation.

Table 5 contains the classification performance metrics achieved by different single CNN-based models on the DeepFire dataset. Accuracy, F1 Score, Precision, Recall, Specificity, and AUC-PR (area under the Precision-Recall curve) were used as comparison criteria. The highest accuracy rate (98.16%) was achieved by the VGG16 and ResNet101 models. This indicates that both models are similarly effective in distinguishing between fire and non-fire images. In terms of F1 Score and Recall, ResNet101 slightly outperformed VGG16 with 98.13% F1 and 97.35% recall compared to VGG16's 98.14%/97.88%. However, these differences are too small to be considered statistically significant.

**Table 5.** Performance of a single model.

| Model | Acc | F1 | Precision | Recall | Specificity | AUC-PR |
|---|---|---|---|---|---|---|
| ResNET50 | 0.9789 | 0.9786 | 0.9892 | 0.9683 | 0.9895 | 0.9866 |
| ResNET50v2 | 0.9316 | 0.9293 | 0.9553 | 0.9048 | 0.9581 | 0.9537 |
| ResNET101 | 0.9816 | 0.9813 | 0.9892 | 0.9735 | 0.9895 | 0.988 |
| ResNET101v2 | 0.9211 | 0.9185 | 0.9441 | 0.8942 | 0.9476 | 0.9455 |
| ResNET152v2 | 0.9421 | 0.9415 | 0.9465 | 0.9365 | 0.9476 | 0.9573 |
| VGG16 | 0.9816 | 0.9814 | 0.984 | 0.9788 | 0.9843 | 0.9867 |
| VGG19 | 0.9053 | 0.9043 | 0.9091 | 0.8995 | 0.911 | 0.9293 |

Precision and specificity values indicate the success of models in reducing false positives. ResNet50 (98.92% Precision, 98.95% specificity) is noteworthy in this regard. However, its overall success rate (97.89%) is slightly lower. Deeper or alternative architectures such as VGG19 and ResNet50v2/101v2 have not stood out due to lower Acc (90–93%) and more unstable precision–recall ratios. In summary, the ResNet101 and VGG16 models are by far the two best-performing models in terms of both accuracy and other metrics. This justifies the selection of these two architectures in hybrid model design.

Table 6 shows the classification performance of the proposed hybrid model (VGG16 + ResNet101). The model has achieved a significant improvement in almost all metrics compared to single architectures: The accuracy value has reached a very high level of 99.72%. This rate is well above the maximum success rate of 98.16% achieved by single models. The Precision value is 100%; that is, the system has not mistakenly classified a non-fire image as a "fire." This indicates that false alarms have been reduced to zero. The Recall value is 98.9%, indicating that the system successfully detects nearly all fire images. Specificity is again 100%; that is, the model correctly classifies non-fire data entirely. The F1 Score is 99.44%, indicating that the classification balance is quite high for both fire and non-fire classes.

**Table 6.** Hybrid Model Performance (ResNET101 + VGG16).

| | Acc | F1 | Precision | Recall | Specificity |
|---|---|---|---|---|---|
| Hybrid (Proposed Model) | 0.9972 | 0.9944 | 1.000 | 0.989 | 1.000 |

These results show that the hybrid model significantly outperforms the single models in terms of both overall success and class balance. It also makes a significant contribution in terms of exceeding the best results in the literature (98%).

Figure 3 shows a graph comparing the performance metrics of all models. It is clear that the proposed Hybrid model performs better than other models in all performance metrics.

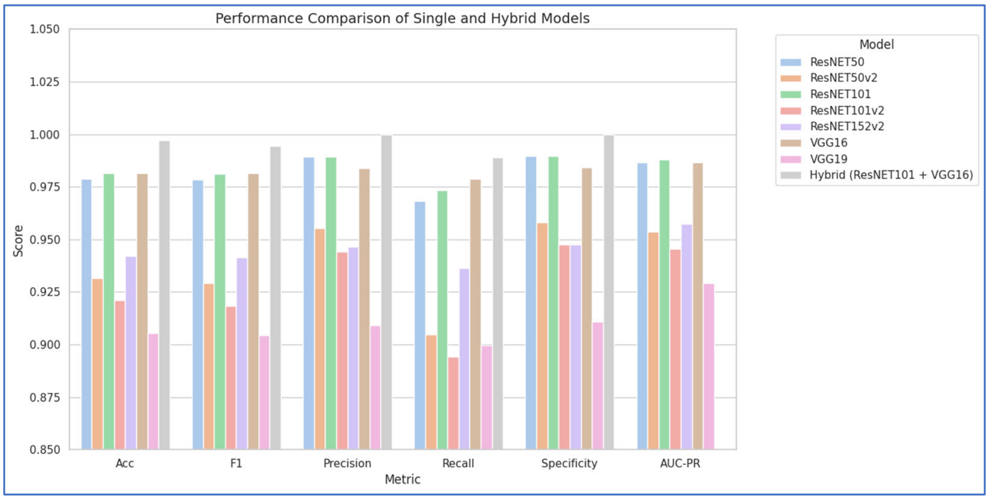


**Figure 3.** Performance Comparison of Single and Hybrid Models.

Figure 4 shows the change in accuracy and loss values of the ResNet101 model during the training process according to the number of epochs. The accuracy curve shows that the model quickly and steadily achieved high performance, while the loss curve shows that errors decreased rapidly and the model avoided overfitting. When both graphs are evaluated together, it is clear that ResNet101 underwent a balanced learning process and performed the classification task with high success.

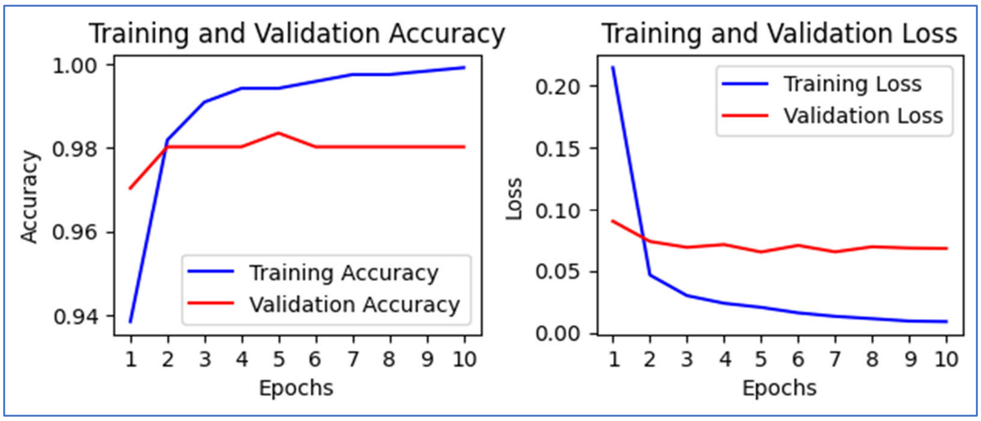


**Figure 4.** ResNET101 Acc/Epoch and Loss Epoch graph.

Figure 5 shows the epoch-based changes in accuracy and loss values during the training process of the VGG16 model. The accuracy curve shows that the model quickly reaches high accuracy levels and stabilizes in the early epochs. The loss curve, on the other hand, shows that the model consistently reduces its errors during training and reaches a minimum level without excessive fluctuations. These graphs demonstrate that VGG16 effectively learns from the dataset and offers good generalization capabilities.

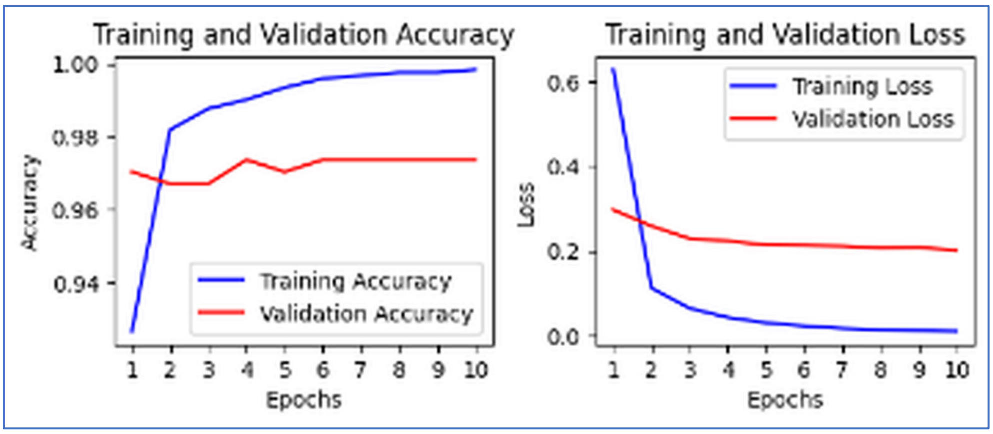


**Figure 5.** VggNET16 Acc/Epoch and Loss Epoch graph.

Figure 6 shows the accuracy curve of the proposed hybrid model. The model achieved high accuracy levels in a very short time during the training process and remained stable at over 99% from approximately the 8th to 10th epoch onwards. This demonstrates that the hybrid architecture has strong learning capacity and fast convergence capabilities. In particular, the combination of VGG16 and ResNet101 has significantly improved classification accuracy.

When comparing the ROC of the three models in Figure 7, significant differences in classification accuracy were observed. Although the ROC curve of the VGG16 model generally shows high accuracy, it is noteworthy that the false positive rates are relatively higher in some parts of the curve. In contrast, the ResNET101V2 model provides higher sensitivity and specificity thanks to its deeper architecture, and its ROC curve approaches the upper left corner more clearly in terms of slope. The best performance was achieved with the Hybrid VGG16 + ResNET101V2 model. The ROC curve of this model covers more area than the other two models, indicating that the AUC value has reached its maximum. This demonstrates that the hybrid model offers superior performance in classification tasks by learning both low-level details and high-level abstractions more effectively. In particular, the combination of VGG16 and ResNet101 has significantly improved classification accuracy.

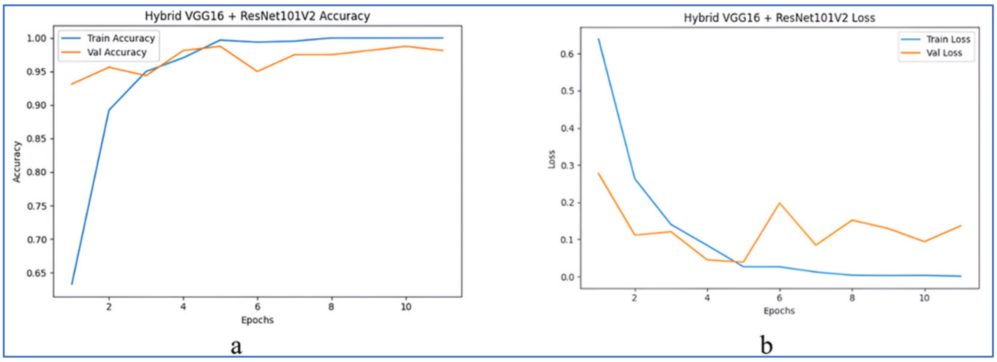


**Figure 6.** Hybrid VGG16+ ResNet101V2: (**a**) Acc/Epoch graph (**b**) Loss/Epoch graph.

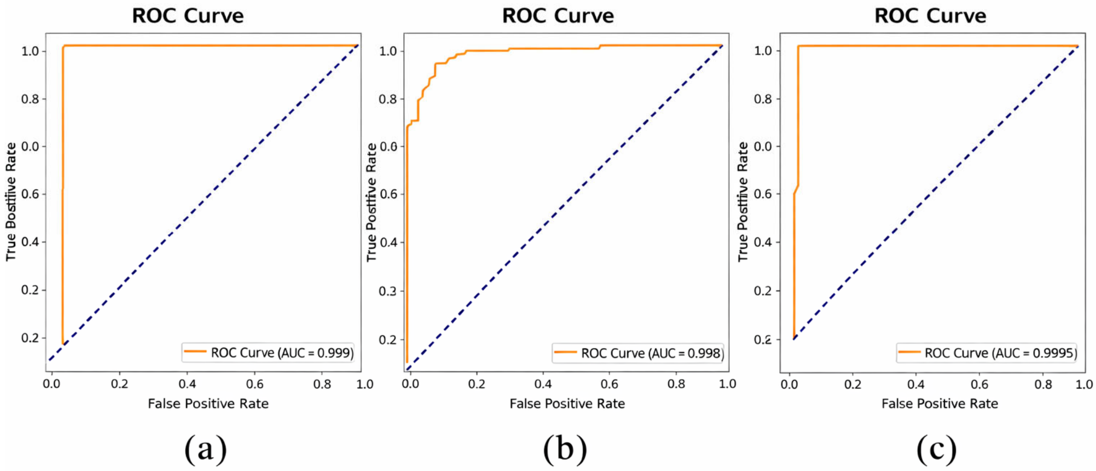


**Figure 7.** ROC Graphs: (**a**) VggNET16, (**b**) ResNET101V2, (**c**) Hybrid VGG16+ ResNET101V2.

Figure 8 visually presents the output of the proposed task assignment algorithm in the simulation environment. In this figure, fire points (Y0, Y1, . . ., etc.) and the locations of existing drones (D0, D1, . . ., etc.) are randomly placed on a two-dimensional plane. Each colored connection line represents the assigned task established between a drone and the target fire point.

In this simulation, two scenarios were considered: one where the number of fires was less than the number of drones, and another where the number of drones was less than the number of fires. In the scenario where the number of fires was less than the number of drones, multiple drones had to be directed to each fire. The algorithm took this situation into account, assigning the drones closest to the fires while also ensuring a balanced task distribution so that approximately the same number of drones were assigned to each fire.

While the traditional Hungarian algorithm only produces solutions based on minimum cost, in this study, the algorithm has been adapted to also take into account the principle of load balancing. In particular, the assignment of an excessive number of drones to some fires has been prevented, and each fire point has been adequately and evenly supported. This provides a critical advantage in terms of optimizing the firefighting process. This simulation also demonstrates the algorithm's adaptability to real-time operations. The algorithm can produce balanced solutions with different numbers of drones and fire configurations, thereby achieving effective results in terms of energy consumption, mission duration, and operational efficiency.

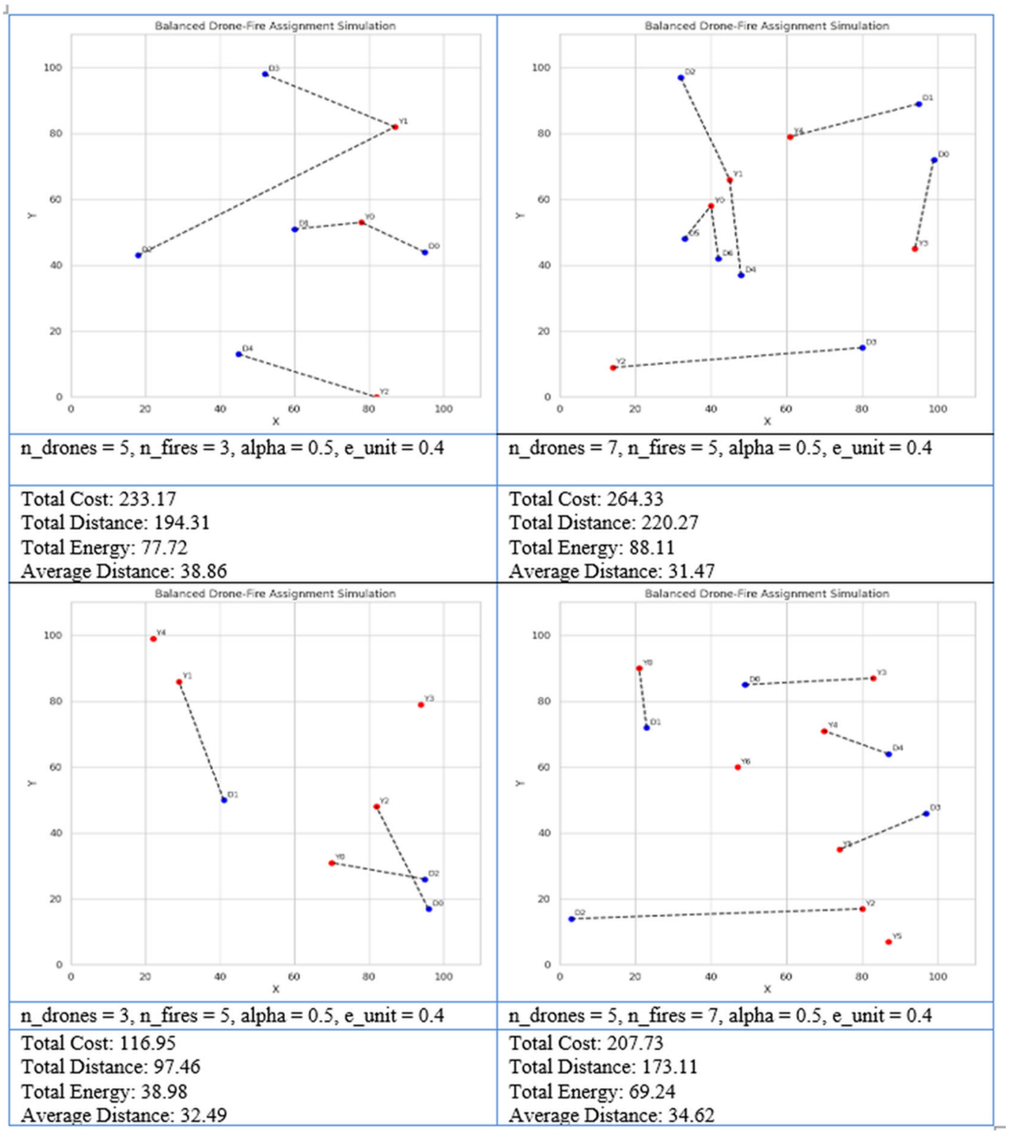


**Figure 8.** Balanced Drone-Fire Assignment Simulation.

## 5. Results and Discussion

In this study, an end-to-end system covering image-based fire detection and real-time intervention processes is proposed. As a deep learning-based classifier, a hybrid model combining the advantages of the VGG16 and ResNet101 architectures was developed; then, an optimization-based distribution algorithm was presented to perform task assignments after fire detection. In the experiments conducted, the proposed hybrid model was shown to perform with high-level metrics such as 99.72% accuracy, 100% precision, and 98.9% recall on the DeepFire dataset. This performance surpassed both single models in the literature and the current best results. Additionally, the Hungarian algorithm-based task assignment structure has reduced response times and energy consumption by distributing the load evenly and balanced, even when the number of fires is less than the number of drones.

The most significant contribution of this study to the literature is that it presents a new hybrid model based on deep learning that works with high accuracy in image-based fire detection. Developed by combining two successful architectures with different structural features, such as VGG16 and ResNet101, this model effectively learns both low-level and abstract features, demonstrating superior classification performance compared to single models. In addition, the task assignment algorithm that makes this detection system functional in the field is designed with an adaptive distribution structure that not only uses a distance-based approach but also balances the task load and minimizes the response time. Thanks to this holistic approach, the system is able to provide both high-accuracy fire detection and fast and balanced response planning at the same time. The end-to-end structure of the developed system distinguishes it from similar studies, positioning it as a significant step forward in terms of both academic research and practical applicability in the field.

Additionally, the proposed hybrid model was compared with the latest hybrid architectures in the literature. Although CNN-Transformer and multi-backbone fusion models have demonstrated strong performance, they generally rely on more complex architectures or larger datasets. Under limited data conditions, the proposed VGG16-ResNet101V2 hybrid model achieves competitive performance with a relatively simple and efficient fusion strategy.

To clarify the source of the high performance values obtained, it is assessed that the performance increase observed in the proposed system is primarily due to the hybrid design at the architectural level. While the VGG16 architecture represents strong local and textural features, the ResNet101V2 architecture can effectively learn deeper and more abstract semantic information thanks to its convolutional connections. The combination of the complementary feature representations of these two architectures at the vector level after Global Average Pooling has created a more discriminative feature space compared to individual models. The frozen backbone structure used in the training process, standard optimization settings, and commonly used training strategies do not aim to artificially inflate performance and are accepted practices in the literature. Therefore, it can be said that the improvement achieved stems from the hybrid feature fusion-based architectural approach rather than an aggressive training strategy. Furthermore, although no direct comparison has been made with more complex and data-intensive architectures in the literature, the fact that the proposed model achieves competitive results with a simple and efficient structure under limited data conditions demonstrates the practical applicability of the approach and its contribution at the system level.

The proposed framework is scalable with respect to both the number of drones and fire locations, as the assignment process can be performed iteratively in dynamic scenarios. For large-scale deployments, the system can be extended using hierarchical or regional task allocation strategies.

Future work plans include expanding the system with different spectrum data (thermal, gas, temperature, etc.) and evaluating models trained with multi-modal data. In addition, the integration of multiple fire scenarios, dynamic weather conditions, and multi-agent coordination systems will increase the system's applicability under more realistic conditions.

**Author Contributions:** Conceptualization, M.S.B. and H.C.; methodology, M.S.B. and H.C.; software, M.S.B.; validation, M.S.B. and H.C.; formal analysis, M.S.B. and H.C.; investigation, M.S.B. and H.C.; data curation, M.S.B.; writing—original draft preparation, M.S.B.; writing—review and editing, M.S.B. and H.C.; visualization, M.S.B.; supervision and project administration, H.C. All authors have read and agreed to the published version of the manuscript.

**Funding:** This research received no external funding.

**Institutional Review Board Statement:** Not applicable.

**Informed Consent Statement:** Not applicable.

**Data Availability Statement:** The data used in this study are publicly available. The DeepFire dataset analyzed during the current study can be accessed from the Mendeley Data repository at https://data.mendeley.com/datasets/gjmr63rz2r/1 (accessed on 8 September 2025).

**Conflicts of Interest:** The authors declare no conflicts of interest.

## References

1. Bowman, D.M.J.S.; Moreira-Muñoz, A.; Kolden, C.A.; Chávez, R.O.; Muñoz, A.A.; Salinas, F.; González-Reyes, Á.; Rocco, R.; De La Barrera, F.; Williamson, G.J.; et al. Human–environmental drivers and impacts of the globally extreme 2017 Chilean fires. *Ambio* **2020**, *49*, 521–535. [CrossRef] [PubMed]
2. Intergovernmental Panel on Climate Change (IPCC). *Climate Change 2021: The Physical Science Basis*; Cambridge University Press: Cambridge, UK, 2021. Available online: https://www.ipcc.ch/report/ar6/wg1/ (accessed on 20 January 2026).
3. Ghali, R.; Akhloufi, M.A. CT Fire: A CNN Transformer for wildfire classification on ground and aerial images. *Int. J. Remote Sens.* **2023**, *44*, 7390–7415. [CrossRef]
4. Diaconu, B.M. Recent advances and emerging directions in fire detection systems based on machine learning algorithms. *Fire* **2023**, *6*, 441. [CrossRef]
5. Khan, A.; Hassan, B.; Khan, S.; Ahmed, R.; Abuassba, A.O.M. DeepFire: A Novel Dataset and Deep Transfer Learning Benchmark for Forest Fire Detection. *Mob. Inf. Syst.* **2022**, *2022*, 5358359. [CrossRef]
6. Gupta, H.; Nihalani, N. An efficient fire detection system based on deep neural network for real time applications. *Signal Image Video Process.* **2024**, *18*, 6251–6264. [CrossRef]
7. Alkhammash, E.H. Multi-Classification Using YOLOv11 and Hybrid YOLO11n-MobileNet Models: A Fire Classes Case Study. *Fire* **2025**, *8*, 17. [CrossRef]
8. Krizhevsky, A.; Sutskever, I.; Hinton, G.E. Imagenet classification with deep convolutional neural networks. *Adv. Neural Inf. Process. Syst.* **2012**, *25*, 1097–1105. [CrossRef]
9. Rawat, W.; Wang, Z. Deep convolutional neural networks for image classification: A comprehensive review. *Neural Comput.* **2017**, *29*, 2352–2449. [CrossRef]
10. Albawi, S.; Mohammed, T.A.; Al-Zawi, S. Understanding of a convolutional neural network. In *2017 International Conference on Engineering and Technology (ICET), Antalya, Turkey, 21–23 August 2017*; IEEE: New York, NY, USA, 2017; pp. 1–6.
11. Tan, M.; Le, Q.V. EfficientNet: Rethinking model scaling for convolutional neural networks. In *Proceedings of the 36th International Conference on Machine Learning (ICML), Long Beach, CA, USA, 9–15 June 2019*; PMLR: Cambridge, MA, USA, 2019.
12. Simonyan, K.; Zisserman, A. Very deep convolutional networks for large scale image recognition. *arXiv* **2014**, arXiv:1409.1556.
13. Howard, A.G.; Zhu, M.; Chen, B.; Kalenichenko, D.; Wang, W.; Weyand, T.; Andreetto, M.; Adam, H. MobileNets: Efficient convolutional neural networks for mobile vision applications. *arXiv* **2017**, arXiv:1704.04861. [CrossRef]
14. Kuhn, H.W. The Hungarian method for the assignment problem. *Nav. Res. Logist. Q.* **1955**, *2*, 83–97. [CrossRef]
15. Munkres, J. Algorithms for the assignment and transportation problems. *J. Soc. Ind. Appl. Math.* **1957**, *5*, 32–38. [CrossRef]
16. Burkard, R.E.; Dell'Amico, M.; Martello, S. *Assignment Problems*; SIAM: Philadelphia, PA, USA, 2009. [CrossRef]
17. Sokolova, M.; Lapalme, G. A systematic analysis of performance measures for classification tasks. *Inf. Process. Manag.* **2009**, *45*, 427–437. [CrossRef]
18. Supriya, Y.; Gadekallu, T.R. Particle swarm-based federated learning approach for early detection of forest fires. *Sustainability* **2023**, *15*, 964. [CrossRef]
19. Khan, S.; Khan, A. FFireNet: Deep learning based forest fire classification and detection in smart cities. *Symmetry* **2022**, *14*, 2155. [CrossRef]
20. Saputra, M.A. Implementasi Website Deteksi Kebakaran Hutan Dan Lahan Menggunakan Densenet201. Undergraduate Thesis, Institut Teknologi Telkom Purwokerto, Kabupaten Banyumas, Indonesia, 2023.
21. Davis, M.; Shekaramiz, M. Desert/Forest fire detection using machine/deep learning techniques. *Fire* **2023**, *6*, 418. [CrossRef]
22. Idroes, G.M.; Maulana, A.; Suhendra, R.; Lala, A.; Karma, T.; Kusumo, F.; Noviandy, T.R. TeutongNet: A fine tuned deep learning model for improved forest fire detection. *Leuser J. Environ. Stud.* **2023**, *1*, 1–8. [CrossRef]
23. Alkhammash, E.H. Leveraging Large Language Models for Enhanced Classification and Analysis: Fire Incidents Case Study. *Fire* **2025**, *8*, 7. [CrossRef]
24. He, K.; Zhang, X.; Ren, S.; Sun, J. Deep residual learning for image recognition. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Las Vegas, NV, USA, 26 June–1 July 2016*; IEEE: New York, NY, USA, 2016; pp. 770–778.
25. Bello, I.; Fedus, W.; Du, X.; Cubuk, E.D.; Srinivas, A.; Lin, T.Y.; Shlens, J.; Zoph, B. Revisiting resnets: Improved training and scaling strategies. *Adv. Neural Inf. Process. Syst.* **2021**, *34*, 22614–22627.

26. He, K.; Zhang, X.; Ren, S.; Sun, J. Identity mappings in deep residual networks. In *Computer Vision–ECCV 2016: 14th European Conference, Amsterdam, The Netherlands, 11–14 October 2016, Proceedings, Part IV*; Springer International Publishing: Cham, Switzerland, 2016; pp. 630–645.
27. Zhang, X.; Zou, J.; He, K.; Sun, J. Accelerating very deep convolutional networks for classification and detection. *IEEE Trans. Pattern Anal. Mach. Intell.* **2015**, *38*, 1943–1955. [CrossRef]
28. Dhillon, A.; Verma, G.K. Convolutional neural network: A review of models, methodologies and applications to object detection. *Prog. Artif. Intell.* **2020**, *9*, 85–112. [CrossRef]
29. Shen, D.; Wu, G.; Suk, H.I. Deep learning in medical image analysis. *Annu. Rev. Biomed. Eng.* **2017**, *19*, 221–248. [CrossRef] [PubMed]
30. Ren, S.; He, K.; Girshick, R.; Sun, J. Faster R-CNN: Towards Real-Time Object Detection with Region Proposal Networks. *IEEE Trans. Pattern Anal. Mach. Intell.* **2017**, *39*, 1137–1149. [CrossRef] [PubMed]
31. Szegedy, C.; Ioffe, S.; Vanhoucke, V.; Alemi, A.A. Inception-v4, inception-ResNet and the impact of residual connections on learning. *arXiv* **2016**, arXiv:1602.07261. [CrossRef]
32. Rajpurkar, P.; Irvin, J.; Zhu, K.; Yang, B.; Mehta, H.; Duan, T.; Ding, D.; Bagul, A.; Langlotz, C.; Shpanskaya, K.; et al. CheXNet: Radiologist-level pneumonia detection on chest X-rays with deep learning. *arXiv* **2017**, arXiv:1711.05225.
33. Maji, P.; Das, S.; Roy, A. Hybrid deep CNN–SVM model for classification of Alzheimer's disease from structural MRI data. *J. Neurosci. Methods* **2020**, *339*, 108702.
34. Zhu, H.; Liu, D.; Zhang, S.; Zhu, Y.; Teng, L.; Teng, S. Solving the many to many assignment problem by improving the Kuhn–Munkres algorithm with backtracking. *Theor. Comput. Sci.* **2016**, *618*, 30–41. [CrossRef]
35. Pan, S.J.; Yang, Q. A survey on transfer learning. *IEEE Trans. Knowl. Data Eng.* **2010**, *22*, 1345–1359. [CrossRef]
36. Yosinski, J.; Clune, J.; Bengio, Y.; Lipson, H. How transferable are features in deep neural networks? In *Advances in Neural Information Processing Systems (NeurIPS)*; Neural Information Processing Systems Foundation, Inc.: San Diego, CA, USA, 2014; Volume 27, pp. 3320–3328.
37. Srivastava, N.; Hinton, G.; Krizhevsky, A.; Sutskever, I.; Salakhutdinov, R. Dropout: A simple way to prevent neural networks from overfitting. *J. Mach. Learn. Res.* **2014**, *15*, 1929–1958.
38. Kingma, D.P.; Ba, J. Adam: A Method for Stochastic Optimization. In Proceedings of the 3rd International Conference on Learning Representations (ICLR), Banff, AB, Canada, 14–16 April 2015.
39. Powers, D.M.W. Evaluation: From precision, recall and F-measure to ROC, informedness, markedness and correlation. *J. Mach. Learn. Technol.* **2011**, *2*, 37–63.
40. Han, J.; Kamber, M.; Pei, J. *Data Mining: Concepts and Techniques*, 3rd ed.; Elsevier: Amsterdam, The Netherlands, 2011.
41. Provost, F.; Fawcett, T. *Data Science for Business: What You Need to Know About Data Mining and Data-Analytic Thinking*; O'Reilly Media: Sebastopol, CA, USA, 2013.
42. Lee, J.; Seo, H.; Park, S. Evaluation metrics and statistical tests for machine learning. *Sci. Rep.* **2024**, *14*, 6086.
43. Tharwat, A. Classification assessment methods. *Appl. Comput. Inform.* **2020**, *17*, 168–192. [CrossRef]




https://doi.org/10.3390/app16031201