Innovation and Integrative Research Center Journal
ISSN: 2584-1491 | www.iircj.org
Volume-4 | Issue-4 | April-2026 | Page 324-333

AI-BASED EARLY FOREST FIRE DETECTION SYSTEM (FORESTGUARD): CONFIDENCE-DRIVEN ALERT MECHANISM USING CNN

[1]Manshi Singh, [2]Pawan Kumar
[1]Student, [2]Assistant Professor
[1,2]Amity University Chhattisgarh, Raipur, C.G - 493225, India,
[1]manshisinghmouhari05@gmail.com, [2]pkumar@rpr.amity.edu

Abstract

Wildfire poses a significant threat to ecosystems, wildlife, and human life, making early detection crucial for minimizing damage. This project presents an AI-based forest fire detection system using Convolutional Neural Networks (CNNs) to automatically classify images as Fire or No Fire. The system is designed with a user-friendly interface built using Streamlit, which allows users to upload aerial or ground-level images for analysis.

The backend is implemented using Flask, which connects the frontend to a trained CNN model developed using TensorFlow/Keras. The model processes input images and provides predictions along with a confidence score, enabling users to assess the reliability of the results. To enhance real-time responsiveness, an alert sound mechanism is triggered when fire is detected with a confidence level greater than 60%, ensuring immediate attention in critical situations.

The system demonstrates fast and efficient performance, making it suitable for early-stage wildfire detection. While the current implementation works on static images, future enhancements include integration with real-time drone (UAV) data, region-based environmental analysis, and explainable AI techniques such as Grad-CAM for visual interpretation of predictions. This approach highlights the potential of AI in improving disaster management and supporting proactive forest conservation efforts.

Keywords: Convolutional Neural Network, Deep Learning, UAV Aerial Imagery, Grad-CAM, Transfer Learning, Binary Classification.

Introduction

Forests are some of the most valuable natural resources on our planet. They play a crucial role in maintaining ecological balance, supporting biodiversity, regulating the climate, and sustaining millions of human lives. Covering nearly 4 billion hectares globally, forests are essential to life on Earth. However, these vital ecosystems are increasingly under threat and one of the most serious dangers they face today is the growing frequency and intensity of forest fires. Forest fires or wildfires are among the most devastating natural disasters. Their impact goes far beyond the areas where they occur, affecting ecosystems, economies, and communities on a massive scale. Between 2002 and 2016, an average of more than 420 million hectares of forest burned every year worldwide. This level of destruction is staggering; entire ecosystems are wiped out, wildlife loses its habitat, and communities that depend on forests for survival are left vulnerable.

The environmental damage caused by these fires is profound. When forests burn, they release large amounts of stored carbon into the atmosphere, contributing directly to global warming. At the same time, the loss of forest cover reduces the Earth's ability to absorb carbon, making climate change even worse. Fires also destroy habitats, pollute air and water, cause soil erosion and push many species closer to extinction. In many cases, the damage cannot be fully reversed within a human lifetime.

The economic consequences are equally serious. Governments spend enormous amounts of money each year on firefighting efforts, emergency response, evacuations and recovery operations. Industries such as agriculture, forestry, and tourism suffer heavy losses and infrastructure in rural areas is often damaged or destroyed. In developing regions where communities rely heavily on forest resources, a single major fire can lead to long-term economic hardship.

What makes the situation even more concerning is that the risk of forest fires is expected to increase. Climate change creates conditions that make fires more likely, such as longer dry seasons, higher temperatures, and increased lightning activity. As a result fires are not only becoming more frequent but also more difficult to control. This makes early detection more important than ever.

Detecting fires at an early stage can significantly reduce damage and save lives. The sooner a fire is identified, the faster response teams can act, limiting its spread. Even a short delay can result in significant losses. An effective system that can detect fires quickly possibly even before they are visible to the human eye can make a huge difference in managing this problem.

Unfortunately, traditional fire detection methods have several limitations. Satellite-based systems often lack the resolution needed to detect small fires and may not provide real-time data. Ground-based sensors, while more accurate locally, are expensive to install and maintain over large areas. Human monitoring from watchtowers is still used in some regions, but it is slow, inconsistent, and not practical for continuous large-scale surveillance.

Recent advances in deep learning, especially Convolutional Neural Networks (CNNs) offer a promising solution. These models are highly effective at analyzing visual data and can help us learn to identify patterns such as smoke and fire in images. As a result, they are well suited for fire detection tasks.

Innovation and Integrative Research Center Journal
ISSN: 2584-1491 | www.iircj.org
Volume-4 | Issue-4 | April-2026 | Page 324-333

At the same time, Unmanned Aerial Vehicles (UAVs) or drones have become powerful tools for monitoring and managing natural disasters. They can easily access remote and dangerous areas, capturing high-resolution images for analysis. When combined with CNN-based models UAVs can provide a fast, flexible and reliable system for detecting fires in real time, without the limitations of human observation.

This work introduces ForestGuard AI, a complete aerial wildfire detection system. It integrates a CNN-based binary classifier with a Flask REST API backend and a Streamlit-based user interface. The system can analyze drone-captured images and classify them as either "Fire" or "No Fire" in less than a second. It also provides confidence-based alerts. The goal is not just accurate detection but also transparency and usability in real-world scenarios.

Objective of the Study:

- Develop an AI-powered early wildfire detection system using CNN.
- Classify aerial drone frames as Fire / No Fire in real time.
- Deploy as a web-accessible system with Flask + Streamlit.

Scope of the Work:

The proposed system focuses on binary image classification rather than more complex tasks like segmentation or object detection. It takes RGB aerial images as input, which are resized to a standard resolution of 224 by 224 pixels for consistent processing. The CNN model is trained offline and then deployed through a REST API, allowing efficient communication between different components of the system. In addition, a demo-ready frontend interface is provided for monitoring predictions and generating alerts, making the system practical and easy to use.

Literature Survey

A. Traditional Methods of Forest Fire Detection

Forest fire detection has traditionally relied on satellite imagery, ground sensors, and human observation. While satellites cover large areas, they often lack real-time responsiveness. Ground sensors provide localized data but are affected by environmental conditions, and human monitoring can be slow and inconsistent. Overall, these methods are limited in providing quick and accurate detection.

B. Role of Deep Learning in Forest Fire Detection

Deep learning, especially Convolutional Neural Networks, has improved fire detection by enabling systems to analyze images and identify fire and smoke patterns more accurately. CNNs can adapt to variations in lighting and background, but some models still face challenges in achieving real-time performance.

C. CNN Based Approaches for Fire Detection

Various CNN architectures such as VGGNet and ResNet have been used for fire detection with promising results. These models can achieve high accuracy, but their complexity can increase computational requirements, making deployment on lightweight systems more difficult.

D. Importance of Diverse Datasets

The effectiveness of CNN models depends heavily on the quality and diversity of training data. Including different fire conditions and environments helps improve accuracy and reduces false detections, though collecting such data is time-consuming.

E. Integration of Multi Modal Data

Combining visual data with other inputs like infrared imagery and sensor data can improve detection accuracy, especially in challenging conditions such as low light or dense smoke.

F. Current Challenges and Research Directions

Despite progress, challenges remain in detecting fires under difficult conditions and ensuring real-time performance on limited hardware. Future work focuses on improving efficiency, accuracy, and practical deployment of these systems.

Problem Statement

1. Detection Delay
   - Traditional methods like satellite monitoring and watchtowers often take too long to detect fires. Since wildfire spread quickly, delayed detection leads to greater damage and reduced chances of control.
2. Accessibility
   - Many AI-based detection systems require specialized hardware and are not easy to deploy. There is a need for a simple and accessible solution that can be used through web-based platforms.
3. Explainability Gap
   - Most existing models act like black boxes and do not explain their predictions. This reduces user trust and makes decision-making more difficult. Challenges also exist in detecting early fire signs due to limited data and feature complexity.
4. Need for an Improved System

Innovation and Integrative Research Center Journal
ISSN: 2584-1491 | www.iircj.org
Volume-4 | Issue-4 | April-2026 | Page 324-333

- An effective system should detect fires early, work in real time, and provide clear and reliable results. This project addresses these needs using a CNN-based approach for fire detection.

Proposed Methodology

System Architecture

- The system is built using a simple modular design with three main parts working together.
- The user interface, developed using Streamlit, allows users to easily upload images and view the results. It provides a clean and interactive way to monitor predictions without needing technical knowledge.
- The backend server, created using Flask, acts as a bridge between the interface and the model. It receives the uploaded image, processes it, and sends it to the model for prediction.
- At the core of the system is the CNN model, which analyzes the image and classifies it as either Fire or No Fire. It also generates a confidence score to indicate how certain the prediction is.

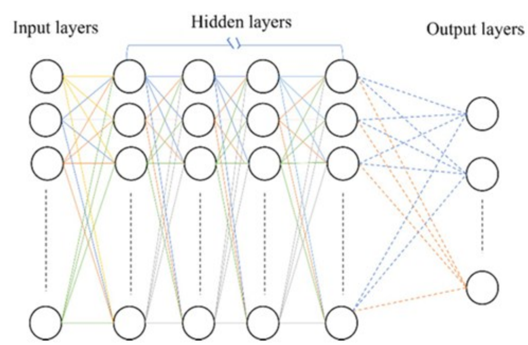

Fig 1[3]: Architecture of CNN from the Development of a Deep Learning-based surveillance system for forest fire detection and monitoring using UAV

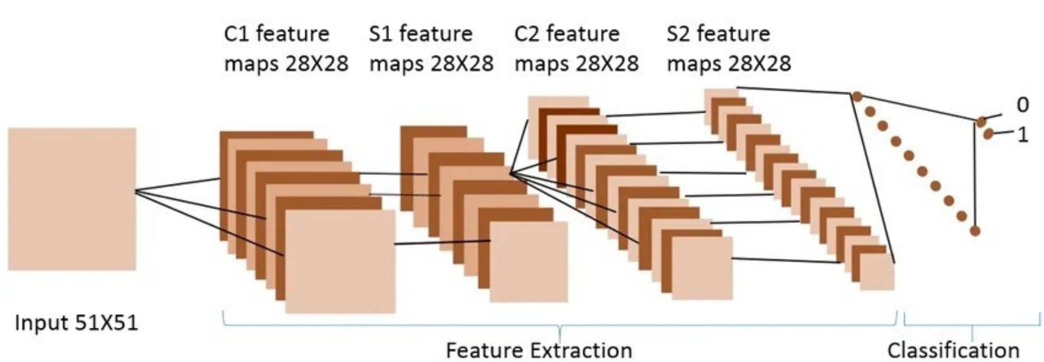

Fig 2 [2]: A CNN is composed of two basic parts of feature extraction and classification. Feature extraction includes several convolution layers followed by max-pooling and an activation function.

Innovation and Integrative Research Center Journal
ISSN: 2584-1491 | www.iircj.org
Volume-4 | Issue-4 | April-2026 | Page 324-333

The classifier usually consists of fully connected layers from Detection of Nuclei in H&E Stained Sections Using Convolutional Neural Networks

Workflow

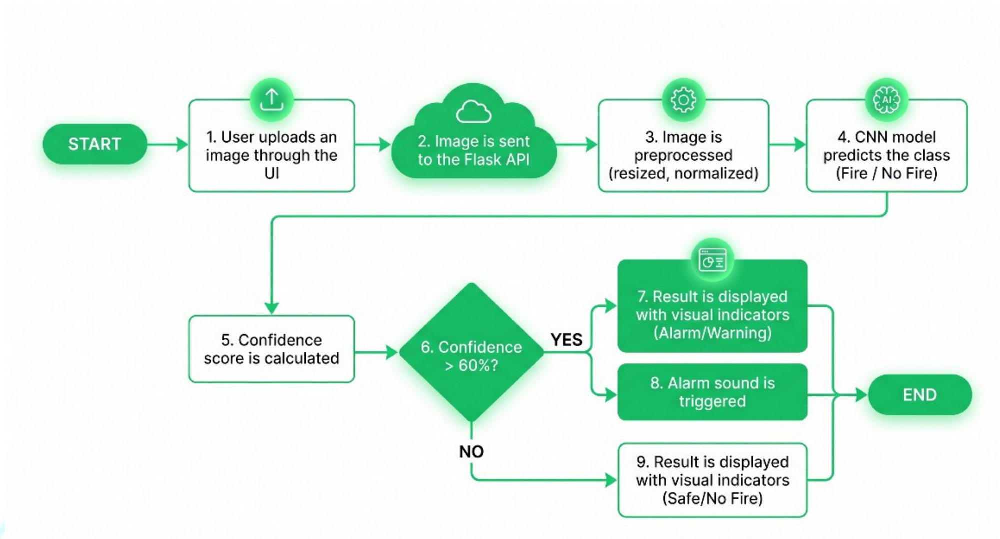

Fig 3: Workflow of Forest Fire detection system

Algorithms / Techniques Used

1. Convolutional Neural Network (CNN)
   - Used to extract important visual features like fire and smoke from images.
   - Helps with accurate image classification.
2. Binary Classification
   - Classifies images into two categories: Fire and No Fire.
   - Binary Cross-Entropy Loss: Standard for binary classification.
   - Keeps the system simple and efficient.
3. Image Preprocessing
   - Resizing images to a fixed size (e.g., 224 × 224).
   - Normalizing pixel values.
   - Converting images into the required input format.
4. Confidence-Based Decision System
   - Provides a confidence score along with prediction.
   - Helps in understanding reliability of results.
   - Confidence Thresholding: Three-tier system (>80% high, 60–80% moderate, <60% review).
   - Sigmoid Activation: Output layer gives probability 0-1.
5. Alert Mechanism
   - Triggers an audio alarm when fire confidence is greater than 60 percent.

Innovation and Integrative Research Center Journal
ISSN: 2584-1491 | www.iircj.org
Volume-4 | Issue-4 | April-2026 | Page 324-333

- Ensures quick user notification.

6. Future Enhancement
    - Integration of Grad-CAM for visual explanation.
    - Highlights important regions influencing the prediction.

Implementation

Tools and Technologies

Table 1: Tools and Technologies used for implementation

Category	Component	Description & Role in System
Software	Python 3.10	The core programming language used to develop all components of the system.
	Streamlit (Frontend)	Creates the simple, interactive frontend user interface (UI) for user interaction.
	Flask (Backend)	Serves as the backend API to bridge the frontend UI and the CNN machine learning model.
	TensorFlow / Keras	The deep learning frameworks used for building, training, and running the Convolutional Neural Network (CNN).
	OpenCV & PIL	Libraries dedicated to image processing tasks, including resizing, formatting, and preprocessing.
	SQLite3 (Database)	Can be used to store user data, prediction logs, and system records for future analysis.
	Postman	Used for testing and validating API endpoints, ensuring correct request-response handling between frontend and backend.
Hardware	Processing Unit	Runs on a standard, regular CPU-based computing system.
	GPU Requirements	None. A specialized GPU (Graphical Processing Unit) is not required for the current stage.
Future Implementation	Deployment Platform	Integration with UAVs (Unmanned Aerial Vehicles) or drone systems for real-time aerial image capture.
	Infrastructure	Migration to cloud platforms to improve system scalability and enable remote monitoring.

Results and Discussion

Output Screens

1. Uploaded Image Preview
    - Displays the input image provided by the user.
2. Classification Result
    - It shows whether the image is classified as Fire or No Fire.
3. Confidence Percentage
    - Indicates how confident the model is in its prediction.
4. Color-Coded Result Cards
    - Using simple visual cues to make results easy to understand.
5. Alert Mechanism

Innovation and Integrative Research Center Journal
ISSN: 2584-1491 | www.iircj.org
Volume-4 | Issue-4 | April-2026 | Page 324-333

- Triggers an audio alarm for high-confidence fire detection.

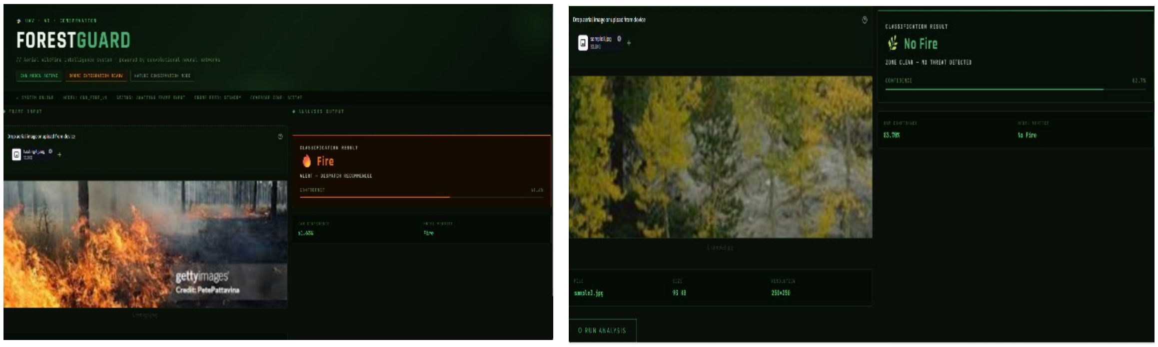

Fig 4: Model output interface with confidence scores for Fire/No Fire classification

Performance Analysis

- Prediction Speed
  - The model generates results quickly, usually within a few seconds.
  - Suitable for near real-time detection.
- Model Performance
  - Performs well on clear and well-defined fire images.
  - Accuracy depends on the quality and size of the training dataset.
- Confidence-Based Observations
  a. Above 80 percent
     - Highly reliable prediction.
  b. Between 60 and 80 percent
     - Moderately reliable and may require manual verification.
  c. Below 60 percent
     - Low reliability and not suitable for decision-making.
- Alert System Impact
  - The audio alert improves response time in critical situations.
  - Help users take immediate action when high-confidence fire is detected.

Testing and Validation

- Testing Approach
  - The system was tested using a mix of fire and non-fire images.
  - Aimed to evaluate overall accuracy and reliability.
- Validation Checks
  - Verified correct classification of images as Fire or No Fire.
  - Checked confidence score consistency and reliability.
  - Ensured alert sound triggers correctly for high-confidence fire cases.

   - Tested API responses and basic error handling.
- Observations
   - The system performs consistently for known and trained scenarios.
   - Works well under clear image conditions.
- Limitations
   - Performance may vary in real-world situations with complex environments.
   - Requires further testing on diverse and unseen data.
- Future Testing
   - Use of real-time image data for validation.
   - Integration of region-based environmental inputs for improved accuracy.

Conclusion

This project presents a practical and efficient AI-based solution for wildfire detection using a Convolutional Neural Network. The system is designed to analyze images quickly and accurately, classifying them as Fire or No Fire while also providing confidence scores to support better decision-making. This helps in identifying potential fire incidents at an early stage.

The integration of a user-friendly interface, a Flask-based backend, and an alert mechanism makes the system easy to use and suitable for real-world applications. Features such as confidence-based alerts and quick response time improve its usefulness in critical situations where early detection is important.

Overall, the project demonstrates how deep learning can be applied to solve real-world environmental problems. By enabling faster detection and response, the system has the potential to reduce damage caused by forest fires and support more effective fire management strategies.

Future Scope

In the future, the system can be improved in several ways to make it more robust, accurate, and suitable for large-scale real-world deployment. One key enhancement is the integration of Grad-CAM, which will allow the system to visually highlight the regions in an image that influenced the model's prediction. This will make the decision-making process more transparent and easier for users to trust and interpret.

Another important direction is the integration of real-time data from drones or UAVs. By connecting the system with aerial platforms, continuous monitoring of remote and high-risk forest areas can be achieved. This would enable faster detection and quicker response, especially in regions where human access is limited.

Innovation and Integrative Research Center Journal
ISSN: 2584-1491 | www.iircj.org
Volume-4 | Issue-4 | April-2026 | Page 324-333

The system can also be improved by incorporating region-based environmental data such as temperature, humidity, and wind speed. These factors play a major role in fire occurrence and spread and combining them with image-based analysis can lead to more accurate and reliable predictions.

For better scalability and accessibility, the system can be deployed on cloud platforms. This would allow centralized monitoring, easier data storage, and the ability to handle large volumes of incoming data from multiple sources. It would also make the system accessible from different locations without depending on a single machine.

Another area of improvement is enhancing the model's accuracy by training it on larger and more diverse datasets. Including images from different environments, lighting conditions, and fire intensities will help the model generalize better and reduce false predictions.

Finally, developing a mobile-based alert system can make the solution more practical. Instant notifications on mobile devices can help authorities or users respond quickly in emergency situations, improving overall response time and potentially reducing damage caused by forest fires.

References

1. Saleh, A., Zulkifley, M. A., Harun, H. H., Gaudreault, F., Davison, I., & Spraggon, M. (2024). Forest Fire Surveillance Systems: A Review of Deep Learning Methods. Heliyon, 10(1), e23127.
2. Khoshdeli, M., Cong, R., & Parvin, B. (Year). Detection of Nuclei in H&E Stained Sections Using Convolutional Neural Networks.
3. Shamta, I., & Demir, B. E. (2024). Development of a Deep Learning-Based Surveillance System for Forest Fire Detection and Monitoring Using UAV. PLOS ONE.
4. Karthik, K. V. G. (Year). Forest Fire Detection Using CNN. Parul University, Vadodara, Gujarat, India.
5. Abid, F. (2021). A Survey of Machine Learning Algorithms Based Forest Fires Prediction and Detection Systems. Fire Technology, 57(2), 559–590.
6. Das, D., Ghosh, M., Gehani, H., Sarwate, S., Gote, A. C., Gade, P., Rathore, S. S., Rathour, R., & Debnath, T. (2025). CNN-Based Real-Time Forest Fire Detection: A Cost-Effective Solution. Journal of Information Systems Engineering and Management, 10(22s).
7. Anjanadevi, B., Nagesh, V., Sujeeth, T., & Varma, P. N. M. (2023). An Efficient Model for Forest Fire Detection using Deep Convolutional Neural Networks. International Journal on Recent and Innovation Trends in Computing and Communication, 11(10s).
8. TensorFlow. (n.d.). TensorFlow Documentation.
9. Keras. (n.d.). Keras Documentation.
10. Kaggle. (n.d.). Fire Dataset.