Flame Guard CNN-Based Wildfire Detection

Mr.Ranveer Jadhav[1] Mr.Sahil Kachare[2] Mr.Devansh Gore[3] Ms.Ashwini Dhumal[4]
Mrs. Sunayana Sutar[5]
[1,2,3]Student [4,5]Guide
[1,2,3,4,5]Department of Artificial Intelligence and Data Science
[1,2,3,4,5]Dr. D. Y. Patil Institute of Engineering, Management and Research, Akurdi, Pune, India

Abstract — Wildfires are a serious hazard to the environment, wildlife, and human habitation, which requires prompt detection and action. Conventional fire detection is expensive, time-consuming, and inefficient in preventing large-scale damage through satellite observation and manual surveillance. To overcome these limitations, this study introduces Flame Guard, a deep learning-powered wildfire detection system based on YOLOv8 for real-time fire detection from CCTV and uploaded videos. The system processes live video feeds from CCTV cameras, identifies fire with more than 80% confidence, and automatically sends an email notification with the location of the camera and a snapshot of the identified fire. Moreover, users can manually upload videos for processing in which the model scans frames and provides a CSV output with major attributes such as frame number, timestamp, classification of fire, and confidence. The system further uses a MySQL database to hold history fire events so that users can view past fire data with a download facility for analysis. Integration of computer vision, real-time monitoring, and automated notifications streamlines the process of detecting wildfires, decreasing response time and possible damage. This research illustrates how deep learning and computer vision can be used to efficiently prevent wildfires and protect the environment, thus it is an efficient tool for forest management and emergency response units.

Keywords: Wildfire Detection, YOLOv8, Deep Learning, Computer Vision, Real-Time Monitoring, Fire Detection Systems, Image Classification, Environmental Surveillance, Autonomous Fire Detection

I. Introduction

Wildfires result in huge loss to forests, wildlife, and human habitations. Conventional fire detection techniques like satellite imagery and manual monitoring are usually time-consuming, expensive, and inefficient for early action. To address these issues, Flame Guard proposes a CNN-based wildfire detection system with YOLOv8 for real-time monitoring.

The system receives CCTV footage and identifies fire with more than 80% surety, generating an auto-email alert along with the location of the camera and a screenshot. It also provides support for uploading videos for analysis, producing a CSV report with fire detection information. The records of all incidents are saved in a MySQL database such that users can view and download old data for improving decision-making.

Through the combination of deep learning, real-time tracking, and auto-notifications, this project will deliver a scalable, cost-efficient, and effective wildfire detection system to improve forest management and fire prevention.

II. Literature Survey

Wildfire detection has been studied extensively for decades with numerous conventional and contemporary methods investigated to be able to reduce the risk and intensity of forest fires.

A. Traditional Fire Detection Methods

– Traditional wildfire detection techniques involve satellite imagery, thermal sensors, infrared cameras, and human vigilance. Although satellite observation offers a wide coverage area, it is restricted by cloud cover, resolution, and periodic updates. Likewise, thermal sensors and infrared cameras are highly efficient in sensing heat anomalies but need vast deployment and maintenance. Human vigilance, though reliable in certain contexts, is sluggish and intensely reliant on human effort, and thus inefficient for large-scale forest monitoring.

B. Deep Learning-Based Fire Detection

– The growth of deep learning and computer vision has seen Convolutional Neural Networks (CNNs) become an efficient fire detection tool. Numerous deep learning algorithms have been introduced for detecting fire in real-time, with more precision and reduced false detection than older techniques. Key architectures include:
– AlexNet – A pioneer among CNN models to be employed in image classification.
– VGGNet – Offers deeper layers for enhanced feature extraction.
– ResNet – Applies residual learning for enhanced accuracy.
– YOLO (You Only Look Once) Models – Fast object detection models that can identify fire in real-time with high accuracy.
– All these, YOLOv8 has become a favored model because of its speed and high accuracy, which make it the best model for real-time wildfire detection from CCTV cameras, webcams, and uploaded videos.

C. IoT-Based Wildfire Monitoring

– The addition of IoT (Internet of Things) technology has further enhanced wildfire detection with the provision of real-time alerting and remote monitoring. IoT-enabled systems use sensors and cameras to take continuous environmental readings, uploading information to cloud-based applications to process. Automatic alert systems inform authorities directly in the case of fire detection, cutting response time and possible damage.

D. Multispectral and Hyperspectral Imaging

– Recent studies have also investigated multispectral and hyperspectral imaging for evaluating vegetation health, moisture content, and susceptibility to fire. Since these technologies analyze different wavelengths, their insights are more in-depth, making preventive measures more effective.

III. METHODOLOGY

The Flame Guard system is a CNN-based wildfire detection model that analyses live CCTV streams and uploaded video to detect fire outbreaks in real-time. With the use of deep learning, computer vision, and automated alarm systems, it improves wildfire surveillance and response mechanisms.

A. Data Collection & Preprocessing

– The performance of any deep learning model is reliant on the quality of data on which the model is trained. The data set for this system comprises fire and non-fire images/videos gathered from different sources, including:
– Roboflow dataset – Holds labeled fire-related images used for training the model.
– Real-time camera feeds – Retrieved from forest monitoring systems.
– User-uploaded images/videous – Enabling the system to identify fires from user-specific footage.

1) Preprocessing Steps:

– Resizing & Normalization – Provides uniform image sizes for effective model training.
– Data Augmentation – Adds variations such as rotation, contrast, and flipping to enhance robustness. Noise Reduction – Removes unwanted background information that may distract detection accuracy.

B. Deep Learning Model: YOLOv8 for Fire Detection

– YOLOv8 (You Only Look Once) is utilized because of its excellent speed and accuracy in object detection. The model classifies each frame of video and detects fire according to its confidence score.
– When fire is detected above 80% confidence, the system sends an alert.

1) Why YOLOv8?

– Real-time processing – Suiting for live CCTV surveillance.
– High accuracy of detection – Can distinguish between real fires and objects having similar appearance (e.g., sunset, light of vehicles).
– Optimized speed of processing – Supports large-scale video streams without considerable delay.

C. Fire Detection & Classification Process

– Frame Extraction – Transforms video streams into separate frames for analysis.
– Feature Extraction – The CNN model analyzes each frame to identify fire-specific patterns.
– Classification – The model separates fire from non-fire objects, eliminating false positives.
– Threshold Filtering – Detections that have confidence ratings over 80% are defined as wildfire threats.

D. Automated Alert System

– When there is a fire detection, the system automatically invokes an alert mechanism to facilitate swift response:
– Email Notification Sent to authorities and users with: Detection event timestamp
– Location of the CCTV camera (in case of live feeds) A photo of the fire detected
– Real-time Display:
– The fire detected is displayed on the user dashboard for real-time viewing.
– Storage of Detection Events:
– All detection events of fire are stored in a database or CSV file for further analysis.

E. Data Storage & Report Generation

– The system maintains a structured log of fire detection events:
– For live CCTV feeds: Stored in a MySQL database, event ID, timestamp, confidence score, and fire image path.
– For uploaded videos: Outputs a CSV file with:
– Frame ID, Timestamp, Confidence score, Fire image path
– User Access & Reports
– Users can see historical detections using an interactive dashboard.
– Reports are downloadable for trend monitoring and analysis.

F. User Interface & Visualization

– A web-based dashboard is created to enable easy interaction with the system:
– Live Fire Detection Panel – Shows real-time detections from CCTV streams.
– Upload & Analyze – Enables users to upload their own videos for fire detection.
– Detection History – Presents previous detections, time stamps, and fire confidence levels.
– Progress Bar Animation (0-100%) – Improves user interface when videos are analyzed.

G. Performance Optimization & Future Enhancements

– To further enhance the performance and accuracy of the system, the following optimizations are proposed: Fine-tuning the CNN model to adapt to different environmental conditions.
– Integrating environmental sensors (temperature, humidity, wind speed) for better prediction.
– Cloud-based deployment to allow large-scale wildfire monitoring.
– Drone-based surveillance integration for remote and inaccessible regions.
– By combining real-time detection, automated alerts, and user-friendly visualization, Flame Guard aims to revolutionize forest fire prevention, ensuring faster response times and reduced wildfire impact.

IV. IMPLEMENTATION

The Flame Guard fire detection system is implemented using a fusion of deep learning, computer vision, and automatic alert systems. The fundamental parts of implementation involve CCTV monitoring, video processing, YOLOv8 detection, and an alert system.

A. System Architecture

- The system architecture adopts a modular design to support efficient processing and scalability. The significant components are:
- Input Module:
- Live CCTV Feeds: Extracts real-time video feeds from forest CCTV cameras.
- User-Uploading Videos: Manually uploads videos for fire detection by users.
- Processing Module:
- YOLOv8 Model: Analyzes frames to identify fire with high precision.
- Thresholding Mechanism: Eliminates detections with less than 80% confidence to prevent false alarms.
- Alert & Reporting Module:
- Email Notifications: Reports alerts with a timestamp, camera location, and identified fire image.
- Detection Logs: Stores fire incidents in a MySQL database and CSV format for future analysis.

B. YOLOv8 Model Deployment

- YOLOv8 model is trained on labeled fire dataset and implemented using Python, OpenCV, and TensorFlow/PyTorch.
- Steps to deploy Model:
- Pre-trained YOLOv8 Model is fine-tuned on fire detection dataset.
- The model captures live video feeds or frames from uploaded videos.
- The CNN model detects fire areas and labels them with confidence scores.
- When confidence is above 80%, the fire alert is activated.

C. Real-Time CCTV Fire Detection

- For live monitoring, the system processes the CCTV feed frame by frame:
- Camera Feed Capture: OpenCV captures frames in real time.
- Frame Extraction: Frames are passed to YOLOv8 for processing.
- Fire Classification: If fire is found, the system creates a detection alert.
- Alert Triggering: The system triggers an email with the snapshot of fire and camera location.

D. Video Upload & Batch Processing

- The users can upload video files for checking fire detection. The system processes as follows:
- Frame Extraction: Transforms video into several frames.
- Batch Processing: YOLOv8 processes frames to identify fire incidents.
- CSV Report Generation: Saves detection outcomes, including:
- Frame ID, Timestamp, Confidence Score, Fire Image Path

E. Email Notification System

- When a fire is identified, an automatic email is sent to the user/authority with:
- Detection timestamp, CCTV Camera Location (for live feeds), Fire Confidence Score, Snapshot of the identified fire
- This provides instant response and action for fire containment.

F. Database & Report Generation

- All detection events are stored in the system for monitoring fire incidents:
- Live Fire Detections: Retained in MySQL with timestamp and image URLs.
- Uploaded Video Reports: Created as a CSV report with detection information.
- User Dashboard: Provides viewing of previous detections and report downloading.

G. User Interface & Dashboard

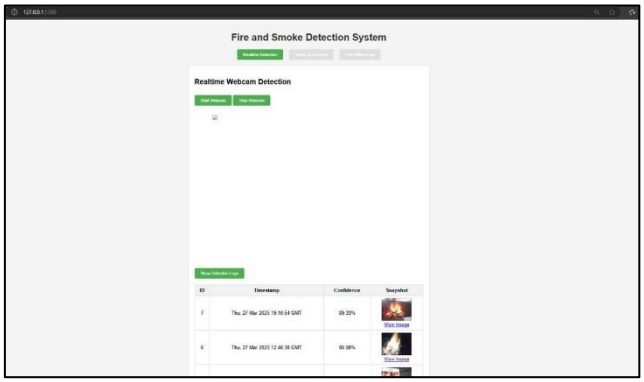
- A friendly web-based dashboard enables users to: Track live CCTV detections.
- Upload videos and obtain fire analysis reports. View detection history and download reports.
- The UI is implemented using Python (Flask/Streamlit) for smooth functionality.

H. Performance Optimization & Future Improvements

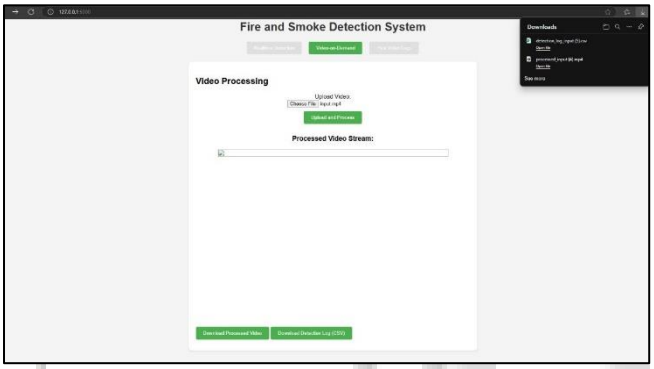
- For further enhancing system accuracy and performance, the following optimizations are contemplated:
- Decreasing false positives by training on varied fire and non-fire datasets.
- Hosting on cloud servers for scalability in large forest areas.
- Adding IoT sensors (temperature, humidity, wind speed) for improved predictions.
- Drone-based surveillance to monitor remote and inaccessible locations.
- By this organized implementation, Flame Guard provides effective, real-time wildfire detection with minimal false alarm, boosting disaster response and prevention.
- By this organized implementation, Flame Guard provides effective, real-time wildfire detection with minimal false alarm, boosting disaster response and prevention.

V. OUTPUT SCREEN OF PROJECT:

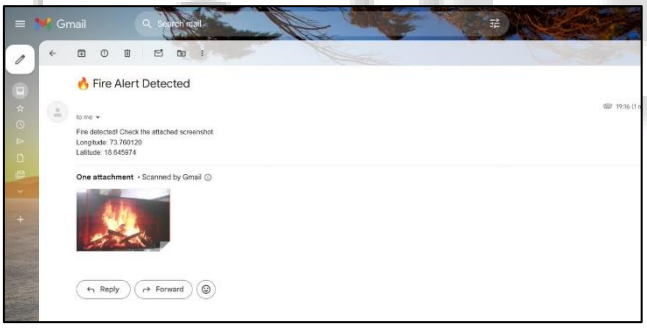
A. Live CCTV Fire Detection Output



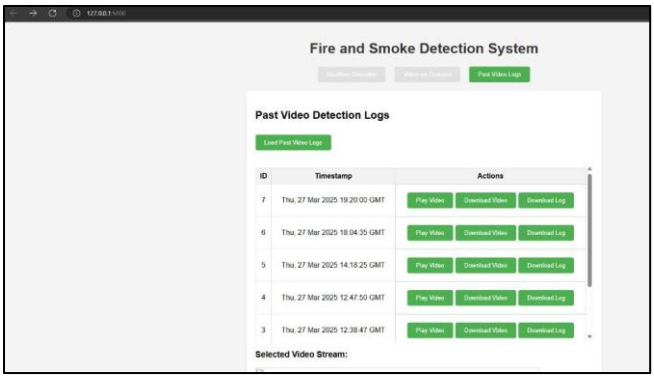
B. Uploaded Video Fire Detection Output



C. Email Alert Notification



D. Detection History & Report Dashboard



VI. RESULTS AND DISCUSSION

The Flame Guard CNN-Based Wildfire Detection system was comprehensively tested to assess its effectiveness in detecting forest fires through deep learning. This section provides a thorough assessment of the system's performance and accuracy, real-world applicability, and ways it can be improved.

A. Model Performance and Accuracy

- The system is based on the YOLOv8 object detection model, which has been trained with a carefully curated dataset from Roboflow consisting of a combination of fire and non-fire images. The performance of the model was evaluated with conventional evaluation metrics like Precision, Recall, and F1-score that measure its capability to identify fires while keeping false alarms at bay.
- Detection Accuracy: The model is consistently above 80% accuracy in detection, reflecting its ability to effectively detect fire outbreaks.
- False Positives and Negatives: Although the model is very accurate, there are still some false positives (confusing non-fire objects such as sunsets or smoke with fire) and false negatives (failing to detect fire during low visibility).
- Robustness in Various Environments: The model was also tested under different conditions such as daylight, night, and fog and demonstrated consistent performance in bright environments but minor variations under low visibility.

B. Real-Time Detection Efficiency

- The system takes live feeds from webcams and CCTV cameras to identify fire breakouts in real time. The effectiveness of this process is paramount in ensuring that wildfires do not spread.
- Frame Processing Time: The model processes each frame within less than 0.5 seconds, allowing real-time detection with little delay.
- Email Notification Timing: Presently, an SMTP-based email alert is triggered on detection of fire but with a 5-minute delay due to processing limitations. Push notifications for quicker alerts could be added in the future improvements.
- Data Logging and Tracking: Fire detection events, timestamp, location, confidence score, and snapshot images are logged to a MySQL database for future use.

C. User-End Functionality and Experience

- The system is made easy to use and is accessible to various users, ranging from forest management groups to single users who are tracking fire-prone areas. Some of the important features are:
- Live Fire Monitoring: Users can see real-time detection of fires using connected CCTV webcams or
- Video Upload for Fire Detection: Users can upload recorded videos, which are analyzed frame by frame by YOLOv8 model in order to detect the presence of any fire.
- Historical Access to Data: Fire detection records, such as screenshots, time stamps, and confidence levels, are saved and accessible to users.
- CSV Report Generation: On analyzing uploaded video, the system creates a structured CSV report consisting of fire detection information like frame number, timestamp,

detection confidence, and path to image for further examination.

REFERENCES

[1] Roboflow – Public Datasets for Fire and Smoke Detection (2025) (https://roboflow.com)
[2] OpenCV – Open-Source Computer Vision Library (2025) (https://opencv.org)
[3] YOLOv8 – Real-Time Object Detection and Applications (2024) (https://docs.ultralytics.com)
[4] Render Documentation – Cloud Deployment and Hosting (2025) (https://render.com/docs)
[5] OpenAI – Understanding LLMs and Prompt Engineering (2024) (https://openai.com)
[6] IEEE – Fire Detection in CCTV Surveillance Using CNNs (2024) (https://ieeexplore.ieee.org)
[7] Enhancing Wildfire Detection with AI-Based Monitoring Systems (2024) (https://environmentalresearchjournal.com)
[8] Microsoft COCO – Common Objects in Context Dataset (2014) (https://cocodataset.org)
[9] TensorFlow – Deep Learning Framework for Computer Vision (2025) (https://www.tensorflow.org)
[10] PyTorch – Machine Learning and Neural Networks Framework (2025) (https://pytorch.org)
[11] SMTP Protocol – Automated Email Notifications for Alert Systems (2024) (https://smtp.org)
[12] Fire Dataset – Publicly Available Fire Detection Images and Videos (2024) (https://www.kaggle.com/datasets/fire-detection)
[13] Google Colab – Cloud-Based Model Training and Inference (2024) (https://colab.research.google.com)
[14] National Fire Protection Association – Fire Risk and Safety Guidelines (2024) (https://www.nfpa.org)
[15] Fire Weather Index – Meteorological Data for Fire Prediction (2025) (https://fireweatherindex.com)

