IJARCCE

ISSN (O) 2278-1021, ISSN (P) 2319-5940

International Journal of Advanced Research in Computer and Communication Engineering

Impact Factor 8.471 ⁙ Peer-reviewed & Refereed journal ⁙ Vol. 15, Issue 1, January 2026

DOI: 10.17148/IJARCCE.2026.15127

WildGuard: A Smart Guardian for Wildlife Using YOLOv8 and Audio Classification

Mahalakshmi C V[1], Catherine Ananya M[2], Supriya L J[3], Supritha Jogin[4], Sushmitha S[5]

Department of Computer Science & Engineering, Bangalore Institute of Technology, Bengaluru, India[1-5]

Abstract: WildGuard is an intelligent, real-time surveillance system designed to protect wildlife and forest ecosystems. It integrates state-of-the-art computer vision using YOLOv8 and audio classification based on MFCC features and machine learning classifiers to detect humans, vehicles, animals, and gunshot sounds from live camera and microphone feeds. The system automates threat detection, provides instant email alerts, and stores event evidence in a centralized database with a web-based dashboard for monitoring and analysis. This paper presents the system objectives, architecture, modules, algorithms, implementation details, and experimental outcomes.

Keywords: Wildlife Monitoring, YOLOv8, MFCC, Audio Classification, Real-Time Detection, Flask

I. INTRODUCTION

Protecting wildlife habitats requires continuous monitoring to detect illegal activities such as poaching, unauthorized entry, and misuse of forest resources. Conventional monitoring practices rely heavily on manual patrols and offline inspection of camera-trap data, which often results in delayed responses and limited situational awareness. These constraints reduce the effectiveness of conservation efforts, especially in large and remote forest regions.

Advances in computer vision and audio signal processing have enabled automated systems to assist environmental monitoring. Deep learning–based object detection models can identify humans, vehicles, and animals from video streams under varying outdoor conditions. In parallel, audio-based event detection methods using spectral features have shown promise in recognizing abnormal sounds such as gunshots, which are commonly associated with illegal hunting activities.

This work introduces WildGuard, a multimodal surveillance system that combines visual object detection and audio classification to support real-time wildlife monitoring. The visual component employs YOLOv8 to analyze live camera feeds and detect relevant objects, while the audio component processes environmental sound samples using MFCC features and supervised machine learning classifiers to identify gunshot events. The system records detection events, generates automated email alerts, and provides a centralized dashboard for monitoring and analysis.

II. OBJECTIVES

The objectives of the proposed WildGuard system are:

- To design an automated wildlife monitoring system using visual and audio data.
- To detect humans, vehicles, and animals in forest environments using YOLOv8.
- To identify gunshot sounds using MFCC features and machine learning classifiers.
- To generate real-time alerts and maintain centralized event logs for monitoring.

III. LITERATURE SURVEY

Vision-based models have been successfully applied to detect animals and human activities from camera-trap images and live video feeds. YOLO-based architectures, in particular, offer real-time performance and have been adapted for wildlife environments to improve detection accuracy under varying lighting and background conditions.

Audio-based monitoring has also been explored as a complementary approach for identifying illegal activities such as gunshots and chainsaw noises. Techniques using Mel Frequency Cepstral Coefficients (MFCC) with machine learning classifiers have demonstrated reliable performance in acoustic event detection, especially in low-visibility scenarios. However, most existing systems rely on either visual or audio data independently and lack real-time alerting and centralized monitoring. This motivates the need for an integrated multimodal surveillance system for effective wildlife protection.

IJARCCE ISSN (O) 2278-1021, ISSN (P) 2319-5940
International Journal of Advanced Research in Computer and Communication Engineering
Impact Factor 8.471 ⁙ Peer-reviewed & Refereed journal ⁙ Vol. 15, Issue 1, January 2026
DOI: 10.17148/IJARCCE.2026.15127

IV. EXISTING SYSTEM & PROBLEM STATEMENT

Existing wildlife monitoring often relies on manual review of camera-trap footage and occasional patrols. These approaches suffer from:

- Delayed detection and response to illegal activities.
- Lack of real-time alerting and centralized logging.
- No integrated audio detection for gunshots or suspicious sounds.

Problem: Build an automated system that detects intrusions (humans, vehicles), classifies animals, identifies gunshots from audio feeds, triggers instant alerts, and logs events centrally.

V. PROPOSED SYSTEM

WildGuard uses:

- Visual pipeline: YOLOv8 for real-time detection (humans, vehicles, animals). Detected animals are optionally passed to a custom YOLO model trained on wildlife images for species classification.
- Audio pipeline: Record short audio segments, extract MFCC features, and classify using Random Forest / XGBoost / KNN models to detect gunshots.
- Alerting and storage: When thresholds are exceeded, the system saves evidence (image/audio), logs the event to a MySQL database, and sends email alerts via Flask-Mail.
- Dashboard: Flask-based web interface for live camera view, audio monitoring, log browsing, and reports.

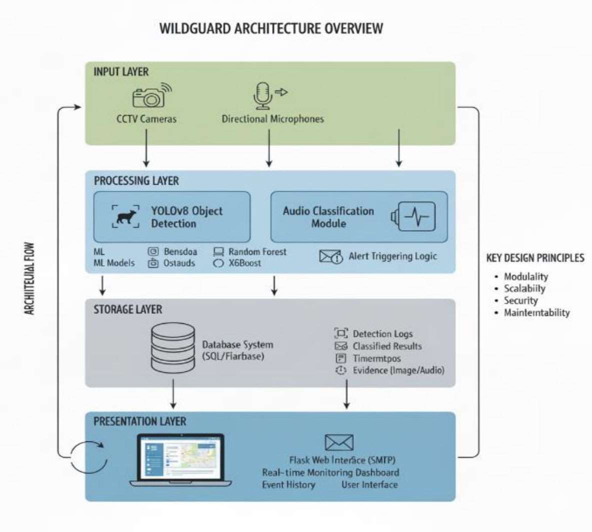

Fig. 1 System Architecture

VI. REQUIREMENT ENGINEERING

A. Software

Python, Flask, PyTorch (YOLOv8), OpenCV, Librosa, Scikit-learn, MySQL, HTML/CSS/JavaScript for frontend.
Development tools: VS Code, Anaconda, Google Colab/Kaggle for training.

A. Hardware

GPU-enabled machine for model training/inference, HD cameras, directional microphones, stable internet for remote alerts.

B. User Requirements

Secure admin login, simple responsive UI, ability to switch between camera/audio monitoring, view logs, download evidence, and receive email alerts.

VII. SYSTEM ARCHITECTURE

The architecture is modular: acquisition (camera/microphone) → preprocessing → detection/classification → alerting → storage/dashboard.

DOI: 10.17148/IJARCCE.2026.15127

VIII. DETAILED MODULES

A. User Authentication Module
Manages admin login and access control. Provides secure sessions to access the monitoring dashboard.

B. Camera Monitoring and Object Detection Module
Uses YOLOv8 for real-time frame-by-frame detection. If object class is person or vehicle, it triggers an alert; if animal, the crop is optionally classified by the custom animal classifier.

C. Animal Classification Module
Custom-trained YOLO model (trained on Kaggle wildlife dataset) for species-level identification (elephant, deer, etc). Improves monitoring precision and enables species-specific alerts.

D. Audio Monitoring and Classification Module
Continuously records short audio chunks (e.g., 3 seconds), applies preprocessing (mono conversion, normalization, noise attenuation), extracts MFCC features, and classifies using a trained model (Random Forest/XGBoost/KNN). If gunshot is detected above confidence threshold, an alert is generated.

E. Database Management Module
MySQL stores detection events, user data, images/audio evidence, and alert logs. XAMPP or a similar stack can be used for local deployments.

F. Alert and Notification Module
Flask-Mail with SMTP sends email alerts containing event type, timestamp, confidence, and evidence attachment or link.

G. Web Application Interface Module
Flask-based dashboard for live video with bounding boxes, audio waveform classification, logs, and downloadable reports.

IX. DATA STRUCTURE DESIGN

Key tables / structures:

- Detection: detection id, object type, confidence, timestamp, image path, location.
- Audio: audio id, mfcc features (stored as path or serialized), predicted label, timestamp, audio path.
- Users: user id, username, password hash, role.
- Alerts: alert id, event _type, description, email sent to, timestamp.

X. ALGORITHM DESIGN

A. YOLOv8 Object Detection
1) Capture frame from camera.
2) Resize and normalize to YOLO input (e.g., 640×640).
3) Run YOLOv8 inference to get bounding boxes and class scores.
4) Apply NMS; keep highest confidence detections.
5) If person or vehicle→ save event and trigger alert.
6) If animal → crop and forward to custom animal classifier.

B. Audio Classification (MFCC + ML)
1) Record short audio chunk (e.g., 3s) with sampling rate (e.g., 22050 Hz).
2) Preprocess: mono, normalize, clip, remove NaNs.
3) Extract MFCC features (13–40 coefficients).
4) Predict label using trained classifier (Random Forest / XGBoost / KNN).
5) If label = gunshot and confidence>threshold → trigger alert and log.

XI. IMPLEMENTATION

```
BEGIN
LOAD YOLOv8_model
LOAD YOLOv5_custom_model
```

International Journal of Advanced Research in Computer and Communication Engineering
Impact Factor 8.471 ⪤ Peer-reviewed & Refereed journal ⪤ Vol. 15, Issue 1, January 2026
DOI: 10.17148/IJARCCE.2026.15127

```
LOAD Audio_Classifier_Model
CONNECT to MySQL Database
FUNCTION User_Login(username, password):
   IF credentials_match_database THEN
      GRANT access
   ELSE
      DISPLAY "Invalid Login"
FUNCTION Start_Camera_Monitoring():
   OPEN camera
   WHILE monitoring_active:
      frame ← capture_frame()
      detections ← YOLOv8_model.predict(frame)
FOR each object IN detections:
         IF object == animal:
Subtype ←   YOLOv5_custom_model.predict(crop(frame))
         ELSE:
            subtype ← object
         SAVE (timestamp, location, subtype) TO database
         IF object IN [human, vehicle, dangerous_animal]:
            SEND email_alert(subtype, location)
      DISPLAY frame_with_labels()
FUNCTION Start_Audio_Monitoring():
   WHILE listening_active:
      audio ← record_3_seconds()
      mfcc ← extract_MFCC(audio)
      label ← Audio_Classifier_Model.predict(mfcc)
      IF label == "gunshot":
         SAVE (timestamp, location, label) TO database
         SEND email_alert(label, location)
UPDATE dashboard_with_latest_logs()
END
```

The proposed system was implemented using Python and web-based technologies to support real-time wildlife monitoring. Live video streams from cameras are processed using a YOLOv8 model to detect humans, animals, and vehicles. Detected animals are further analyzed using a custom-trained model for species identification.

For audio monitoring, short sound samples are continuously recorded and processed to extract Mel Frequency Cepstral Coefficients (MFCC). A machine learning classifier is used to identify gunshot sounds from the extracted features.

All detected events are stored in a centralized database with time and event details. A web-based dashboard enables monitoring of live feeds and detection logs. Whenever suspicious activity is detected, the system automatically sends email alerts to the concerned authorities.

XII. RESULTS

The WildGuard system was evaluated using wildlife video streams and gunshot audio samples, where the YOLOv8-based visual module reliably detected humans, animals, and vehicles in real time, and the MFCC-based audio module successfully identified gunshot events from short audio segments. All detected events were logged with timestamps and displayed on a web-based dashboard, and automated email alerts were generated immediately, demonstrating effective real-time monitoring and response capability.

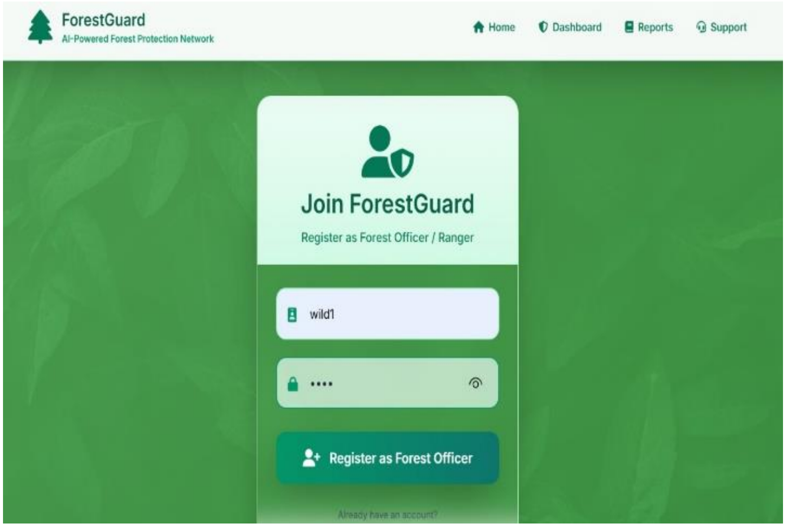

Fig. 2 Login Page

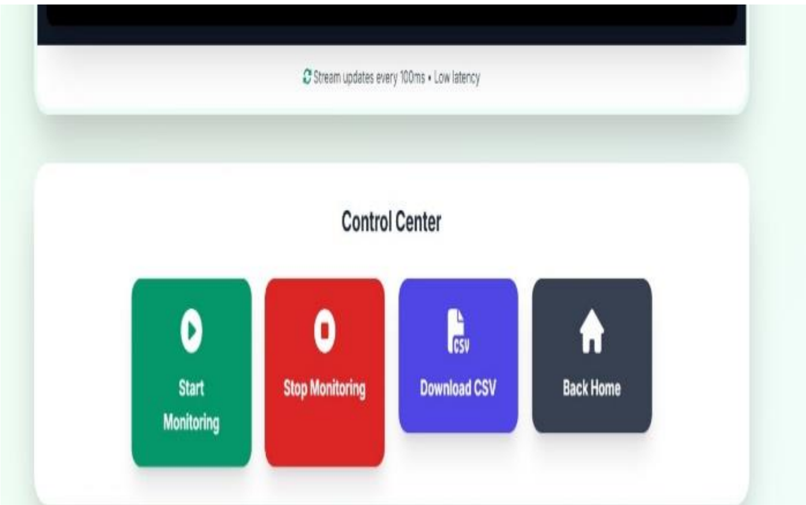

Fig. 3 Monitoring Page

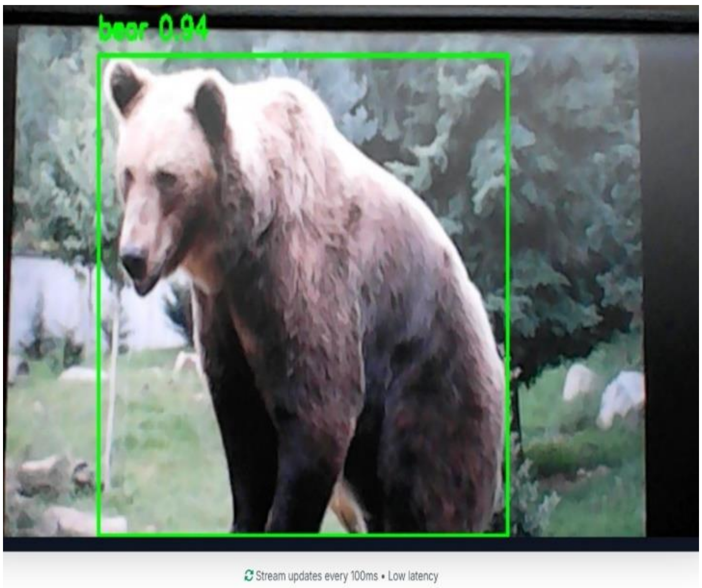

Fig. 4 Real-Time Object Detection Using YOLOv8 and Animal Classification Result

IJARCCE ISSN (O) 2278-1021, ISSN (P) 2319-5940

International Journal of Advanced Research in Computer and Communication Engineering

Impact Factor 8.471 ⁒ Peer-reviewed & Refereed journal ⁒ Vol. 15, Issue 1, January 2026

DOI: 10.17148/IJARCCE.2026.15127

Fig. 5 Gunshot Detection Using MFCC-Based Audio Classification

XIII. FUTURE WORK

- Edge deployment optimizations (pruning/quantization) for on-device inference.
- Multi-sensor fusion (thermal, LiDAR) to improve detection under poor visibility.
- Mobile / SMS alert integration and role-based access control for authorities.
- Continuous model retraining with new labeled data from field deployments.

XIV. CONCLUSION

This paper presented WildGuard, an automated wildlife monitoring system that combines YOLOv8-based visual detection with MFCC-based audio classification for real-time surveillance. The system enables timely detection of intrusions and gunshot events, centralized data logging, and automated alerts through a web-based dashboard. The results demonstrate that the proposed approach can assist wildlife authorities in improving response time and enhancing conservation efforts.

ACKNOWLEDGEMENT

The authors express their sincere gratitude to **Prof. Mahalakshmi C V**, Assistant Professor, Department of Computer Science and Engineering, Bangalore Institute of Technology, for her guidance and support throughout this work.

REFERENCES

[1]. T. Wei, P. Fernandes, and S. M. Lee, “WildCLIP: Scene and Animal Attribute Retrieval from Camera Trap Images,” International Journal of Computer Vision, Springer, vol. 132, no. 2, pp. 345–362, Feb. 2024.

[2]. M. Ibraheam, K. F. Li, and F. Gebali, "An Accurate and Fast Animal Species Detection System for Embedded Devices," *IEEE Access*, vol. 11, pp. 23279–23291, 2023.

[3]. J. Chappidi and D. M. Sundaram, "Novel Animal Detection System: Cascaded YOLOv8 With Adaptive Preprocessing and Feature Extraction," *IEEE Access*, vol. 12, pp. 99387–99401, 2024.

[4]. Z. Li, Z. Yan, W. Tian, D. Zeng, Y. Liu, and W. Li, “ReDeformTR: Wildlife Re-Identification Based on Light-Weight Deformable Transformer With Multi-Image Feature Fusion,” *IEEE Access*, vol. 12, 2024.

[5]. L. Alagialoglou, I. Manakos, M. Heurich, J. Červenka, and A. Delopoulos, “A Learnable Model With Calibrated Uncertainty Quantification for Estimating Canopy Height From Spaceborne Sequential Imagery,” *IEEE Transactions on Geoscience and Remote Sensing*, vol. 60, 2022.

[6]. G. Allabadi, A. Lucic, Y.-X. Wang, and V. Adve, “Learning to Detect Novel Species with SAM in the Wild,” *International Journal of Computer Vision*, Springer, vol. 132, 2024.

[7]. W. C. O’Malley, L. M. Elbroch, K. A. Zeller, and P. Beier, “Machine Learning Allows for Large-Scale Habitat Prediction of a Wide-Ranging Carnivore Across Diverse Ecoregions,” *Springer*, 2024.

[8]. C. M. Coman, B. C. Toma, M. A. Constantin, and A. Florescu, “Ground Level LiDAR as Contributing Indicator in an Environmental Protection Application,” *IEEE Access*, vol. 11, pp. 87543–87557, 2023.

[9]. D. Stewart, A. Zare, S. Marconi, B. Weinstein, E. White, S. Graves, S. Bohlman, and A. Singh, “RandCrowns: A Quantitative Metric for Imprecisely Labeled Tree Crown Delineation,” *IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing (JSTARS)*, vol. 14, pp. 8451–8465, 2021.

[10]. M. Aubry-Kientz, A. Laybros, B. Weinstein, J. G. C. Ball, T. Jackson, D. Coomes, and G. Vincent, “Multisensor Data Fusion for Improved Segmentation of Individual Tree Crowns in Dense Tropical Forests,” *IEEE JSTARS*, vol. 14, pp. 9423–9435, 2021.

