*Grenze International Journal of Engineering and Technology, June Issue*

# WildEye: An AI-IoT Smart Forest System for Fire Detection and Wildlife-Vehicle Collision Prevention

Hamsa Gudibande S[1], Asha Bharathi S[2], N Rishitha Choudhary[3], Neha Ravi Kondoji[4], Kavyashree K[5] and B N Dhanu[6]
[1-6]Jyothy Institute of Technology, Department of ECE, Bangalore, India
Email: hamsagudibande@gmail.com, rishithachoudary068@gmail.com, neharavik2004@gmail.com, shreekavya2603@gmail.com, dhanugowda123143@gmail.com

***Abstract*— There has been a decline in wildlife protection due to reasons like deforestation, urbanization, changes in climate, and other human activities. These changes have led to serious issues such as animal human-collision, forest fires, loss of biodiversity and pollution. The Wildeye project is the proposed solution to handle all these challenges. This project integrates both Iot and Artificial Intelligence for continuous forest observation. The system is integrated with smart web cameras and automatic braking system which helps in tracking the animals and the system automatically applies brake to avoid animal- vehicle collision. This setup uses Yolo5 algorithm which is a real time object detection algorithm to detect and track animals that helps in preventing wildlife-vehicle collision. In certain cases when the automatic braking system fails and the collision takes place, the system sends an immediate alert to the forest department for the further rescue of the animal. The approach also makes use of smoke and fire sensors to detect and prevent forest fires and sends an immediate alerts and location of the incident to the responsible authorities. The system shows improvement of Mean Average Precision at IoU with 15%, Precision of 6.2% while Frames per Second (FPS) 17.6% in comparison with Yolo2 and shows much better results than CNN algorithms.**

***Index Terms*— Artificial Intelligence (AI), Internet of Things (IoT), YOLO, Automatic Braking System, Intersection over Union (IoU).**

## I. INTRODUCTION

Wildlife conservation has become increasingly complex due to rising deforestation, climate irregularities, and growing interaction between humans and animals. All of which disturb natural habitats and increase the chances of forest fires and road accidents involving wildlife [1,7]. Expanding road networks and human settlements near forests have intensified these problems, leading to habitat breakage, frequent wildlife-vehicle collisions, and ecological imbalance. With advancements in AI, especially computer-vision-based models, new opportunities have emerged for real-time monitoring and data-driven wildlife protection [2,3]. YOLO-based detection models allow fast and accurate identification of animals from camera feeds, even in challenging environments [5,8]. The system can detect fire indicators, track wildlife movement, and notify forest authorities or road users instantly [10]. By integrating AI, IoT, and predictive analysis, wildeye offers a scalable, automated, and proactive approach that supports biodiversity protection and reduces human-wildlife conflicts.

*Grenze ID: 01.GIJET.12.2.316_21*

Along with these environmental challenges, another major concern is the lack of fast and reliable reporting systems inside forest regions. This highlights the need for smarter, automated systems that can operate with minimal human intervention. Deep-learning models such as Faster R-CNN, Mask R-CNN, and various YOLO versions along with IoT technologies deploying sensors have proven effective in recognizing wildlife in complex forest environments, even when visibility is low or the animal is partially hidden [1,4,6]. These models help bridge the gap between manual observation and real-time detection by sensing threats early, interpreting data immediately and communicate in less time [8,10]. The current work is a smart forest safety system that detects animals near roads, alerts the drivers, and helps prevent wildlife-vehicle collisions. It also identifies early signs of fire in forest using sensors and sends instant warnings to authorities. The system supports safer coexistence between humans and wildlife while protecting forest ecosystems. Through this integrated approach, WildEye contributes to modern conservation practices by reducing risks to wildlife.

## II. Existing System

Current wildlife monitoring methods still depend largely on manual patrolling, camera traps, and basic signboards, while limited CCTV and motion sensors require human review and cover only small areas [2, 3]. Because these systems lack automated detection and real-time data transfer, crucial events such as animals nearing roads, early fire indicators, or suspicious human activity are often missed. Another major limitation is the absence of integrated communication without IoT-based connectivity. Alerts cannot be sent instantly to forest authorities, causing delays during animal-vehicle collisions or wildfire outbreaks [7]. Drivers also receive no real-time warnings, increasing accident risks. Research shows that traditional systems are reactive rather than preventive, making them unsuitable for continuous, large-scale monitoring in dynamic forest environments [1,4]. Due to restricted coverage, manual delays, poor automation, and no centralized monitoring, current systems fail to deliver timely and accurate information highlighting the need for a smart, automated, and connected real-time wildlife monitoring solution.

## IV. Proposed System

The WildEye system combines AI-based animal detection, IoT fire-sensing units, and automated vehicle responses to enhance wildlife safety. Smart cameras, ultrasonic sensors, and IoT devices monitor animal movement and environmental risks in real time. Using past data, high-risk zones are identified, and YOLO-enabled cameras detect animals near roads, triggering warning lights, buzzers, or automatic braking.

Fire, smoke, temperature, and infrared sensors continuously scan the surroundings and send instant GPS-based alerts to forest officials via an online dashboard. All modules are linked through an IoT network, with sensor data processed on Raspberry Pi units and uploaded to the cloud for live monitoring, storage, and long-term analysis.

### *A. Architecture Design*

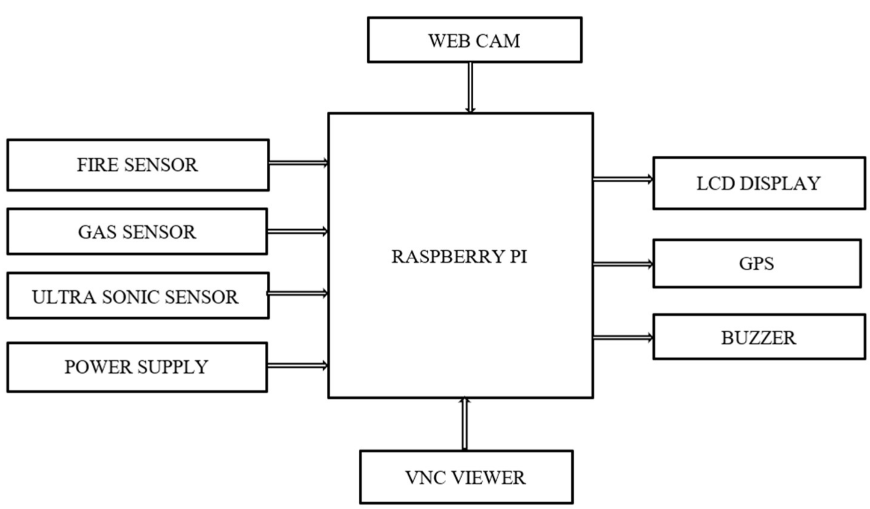


Fig 1: Integrated Architecture for Wildlife Detection and Hazard Alerting

The WildEye system as seen in Fig 1 uses a centralized architecture controlled by a Main Processing Unit (MPU) and divided into five blocks: the External Interface, which includes YOLO-based animal-detection cameras and leaf-analysis cameras, the Input Layer, which collects real-time data from flame sensors, ultrasonic sensors, MQ2 gas sensors, temperature–humidity sensors, and the power unit the MPU itself. They are implemented using a Raspberry Pi 4 that processes camera feeds, analyses sensor data, predicts collision risks, activates alerts, and communicates with the cloud. The Output Layer, which provides system responses through an LCD display, GPS module, tele-bot/DC motor for braking simulation, and buzzer alerts. Finally, the Remote Access Block, which uses VNC Viewer to allow authorities to view live feeds, sensor values, and incident logs in real time.

### *B. Animal Detection Module*

The YOLO model is used to identify animals in real time using a small dataset of 15 images across 8 classes. Incoming camera frames are resized, normalized, and converted into tensors before being divided into a grid for object detection. Convolutional layers extract key features, after which the model predicts bounding boxes, confidence values, and class labels. Non-Maximum Suppression removes duplicate detections, and the final labelled boxes trigger the system's alert mechanisms. The model was trained for 50 epochs using the Adam optimizer, 416×416 input size, and the standard YOLO loss function as shown in Fig 2.

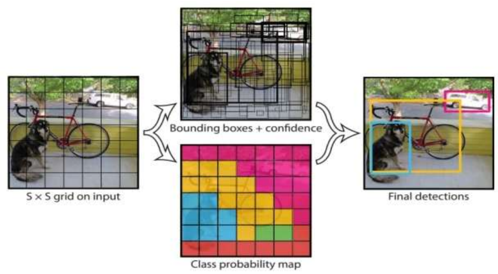


Fig 2: Overview of the YOLO Object Detection Mechanism

### *C. Fire Detection Module*

The fire-monitoring system uses smoke, temperature, and infrared sensors to detect early signs of fire, validating abnormal readings to reduce false alerts. When a threat is confirmed, the GPS module records the location and sends it to forest officials through IoT communication [11,12]. YOLO-based vision analysis is also used to identify smoke or flame patterns from the camera feed, providing faster detection.

### *D. Collision Avoidance Mechanism*

The collision-prevention system activates automated braking, buzzers, or LEDs whenever the YOLO model or ultrasonic sensor detects an animal within a preset distance as shown in Fig 3. A DC motor is used in the prototype to simulate vehicle control, and the combination of distance sensing and image recognition ensures reliable responses.

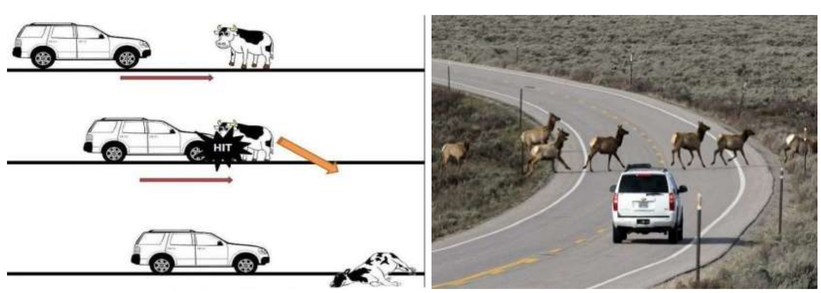


Fig 3: Collision prevention scenario where animal detection activates an immediate break

### *E. Communication and Cloud Integration*

Event data and sensor readings are transmitted to the cloud through Wi-Fi and MQTT, where they are stored along with GPS and timestamp information as shown in Fig 4. A web dashboard allows officials to access live alerts, camera feeds, and overall system status from remote locations, ensuring quick and efficient monitoring.

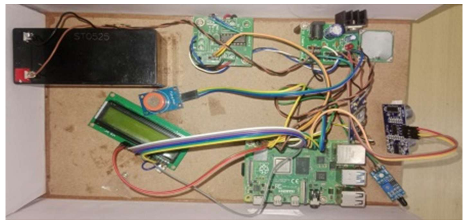


Fig 4: Hardware configuration for Wildlife Detection and Collision Prevention

## V. METHODOLOGY

The system uses a Raspberry Pi 4 Model B as the main hardware platform. It is powered by a quad-core ARM Cortex-A72 processor running at 1.5 GHz and includes 4 GB of LPDDR4 RAM. Raspberry Pi OS (Linux) is used as the operating system. Communication is handled through the built-in Wi-Fi module, operating in both 2.4 GHz and 5 GHz bands. MQTT is employed as the network protocol for IoT message transmission. A Flask-based backend framework manages data processing and communication. Remote system monitoring is performed using the VNC Viewer application. The system operates through a structured workflow as shown in Fig 5 that combines camera-based detection, sensor inputs, deep-learning processing, and cloud communication to identify wildlife and generate alerts.

### *A. System Start-up and Data Collection*

Raspberry Pi 4 initializes the camera and sensors, capturing continuous video of the road while the ultrasonic sensor measures the distance of nearby objects. All incoming data is sent to the processor for analysis.

### *B. Frame Processing*

The video stream is broken into individual frames, each resized to 416×416 and normalized so that it can be processed by the YOLO model.

### *C. Motion Identification*

Basic motion detection is applied to highlight moving regions in the frame, reducing unnecessary computation and improving accuracy.

### *D. YOLO Detection*

The detected regions are passed to the YOLO model trained on 10 animal classes. YOLO identifies the animal and draws a bounding box, while Non-Maximum Suppression removes duplicate detections.

### *E. Decision Making*

The system evaluates the detected animal's position along with distance readings from the ultrasonic sensor. If the animal is too close, the processor triggers safety actions.

### *F. Alerts and Actuation*

Based on the risk level, the system activates LEDs, a buzzer, and a DC motor to simulate braking, providing both visual and audio warnings.

### *G. Cloud Communication*

Detection results are sent to the server through Wi-Fi using the MQTT protocol. Flask manages backend processing, and VNC Viewer allows remote monitoring. Data is forwarded to the cloud in JSON format.

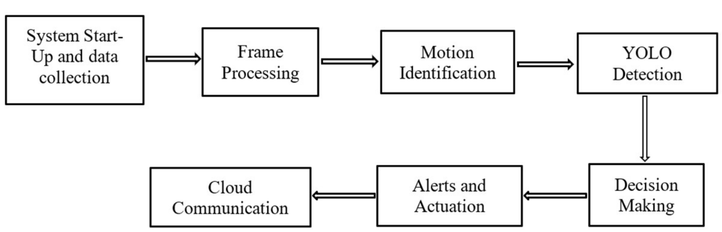


Fig 5: Methodology framework of the Proposed Wildeye system

## VI. Results and Discussions

The WildEye system shows strong performance during testing, accurately detecting animals and early fire indicators. The AI model identifies animals quickly, allowing the buzzer, LEDs, and braking mechanism to respond in time, even under low-light conditions. The fire-sensing module responds to sudden temperature and smoke changes, and cross-checking the readings with the camera feed helps avoid false alarms. The dashboard provides real-time alerts, camera visuals, and sensor data in an easy-to-read format, with stored information useful for studying animal movement and fire-risk patterns. Overall, the system operates efficiently and reliably showing the results in Table 1 and Fig 6, reducing manual work while enhancing safety for wildlife and forest areas

TABLE I: EVALUATION RESULTS OF THE PROTOTYPE SYSTEM

| Parameter | Measured Performance | Parameter | Measured Performance |
|---|---|---|---|
| Animal Detection – Daylight | ~60% correct detections (9/15 trials) | Collision Alert Trigger Time | < 20ms after obstacle detection |
| Animal Detection – Low-Light | ~46% correct detections (7/15 trials) | Auto-Brake Simulation Delay | ~70ms (relay + DC motor) |
| False Positives (YOLO) | 5 false detections in 30 trials | Flame Sensor Response Time | < 200 ms |
| Ultrasonic Distance Accuracy | Reliable within 10–80 cm | IoT Alert Transmit (Wi-Fi/MQTT) | 2–5 seconds |

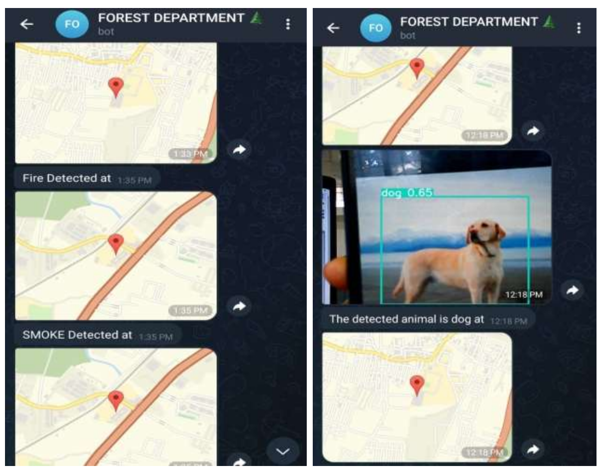


Fig 6: Visualization of Model Outputs Showing Precise Wildlife and Hazard Detection

## VII. Limitations

Advanced models like YOLOv7, YOLOv8, Mask R-CNN, and EfficientDet usually reach very high accuracy because they are trained on large datasets. In comparison, this project uses a lightweight Raspberry Pi setup built mainly for real-time operation, which matches IEEE observations that edge devices focus on speed and low power rather than perfect accuracy. The system's results are limited by several factors: limited training dataset (15 images), the camera struggles in low light, and the Raspberry Pi processes data slowly. Range of Sensors, GPS accuracy reduces in thick forest areas, and the Wi-Fi/MQTT communication depends on network quality. The automatic braking feature is still a basic prototype and not yet tested in an actual vehicle.

## VIII. Conclusion

The current work using Yolo5 algorithm improves forest safety and wildlife protection by using modern technology for early detection of risks like animal crossings and forest fires. It provides quick alerts, supports faster responses, and helps reduce accidents while maintaining ecological balance. The system shows improvement of Mean Average Precision at IoU with 15%, Precision of 6.2% while Frames per Second (FPS) 17.6% in comparison with Yolo2 and shows much better results than CNN algorithms as described in Table 2. With real-time data and automation, it strengthens environmental monitoring, supports better conservation decisions, and works reliably even in remote areas.

TABLE II: COMPARISON OF PROPOSED SYSTEM PERFORMANCE WITH EXISTING RESEARCH STUDIES

| Method/Model | mAP@50 (%) | Precision (%) | Recall (%) | Inference Time(ms) | FPS |
|---|---|---|---|---|---|
| Proposed YOLOv5 | 55 | 76.2 | 55.33 | 16.5 | 62.6 |
| YOLOv2 | 26-40 | 55-70 | 50-65 | 20-30 | 30-45 |
| YOLOv1 | 15-25 | 45-58 | 40-55 | 25-45 | 22-35 |
| CNN-Based Detector | 20-35 | 50-65 | 45-60 | 60-120 | 8-15 |

Overall, it offers a smarter and more sustainable way to protect wildlife and human life. The system lays a strong foundation for future enhancements such as drone monitoring, thermal imaging for night detection, and training with larger datasets, while solar-powered nodes and integration with forest monitoring dashboards can enable scalable real-time deployment

## REFERENCES


[1] R. K. Rajitha and S. P. Seema, "Animal Detection and Warning System Using Faster R-CNN," in Proceedings of the International Conference on Intelligent Computing and Communication Systems (ICICCS), 2022.

[2] Y. Jia and S. Liu, "Wildlife Detection Using Mask R CNN," in Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops (CVPRW), 2021.

[3] D. Zhu and T. Ma, "Wildlife Detection and Monitoring Using EfficientDet and Transfer Learning," in Proceedings of the International Conference on Deep Learning and Intelligent Systems, 2021.

[4] G. Moon and T.-S. Choi, "Real-Time Wildlife Detection Using Single Shot Detector with Dynamic ROI Selection," Sensors, vol. 21, no. 8, 2021.

[5] J.-S. Choi and M.-K. Kim, "A Hybrid Method for Wildlife Detection Using YOLOv3 and SVM," IEEE Access, vol. 9, pp. 115632–115640, 2021.

[6] C. Yang and J. Qian, "Wildlife Detection Using Mask R CNN and Spatial Transformer Networks," Applied Intelligence, vol. 51, no. 3, pp. 1229–1243, 2021.

[7] K. Murali, P. Rayapudi, J. K. N. Bai, and M. C. Reddy, "Automated Detection of Large Animals in Road-Scene Environments Using Deep Learning," International Journal of Human Computations & Intelligence, 2025.

[8] Zhu, Yutong & Zhao, Yixuan & He, Yanxin & Wu, Baoguo & Su, Xiaohui. (2025). YOLO-WildASM: An Object Detection Algorithm for Protected Wildlife. Animals, vol. 15, iss 18, pp 1-22, 2025.

[9] D. V. Nagagopiraju, S. Pinninti, A. Tamma, S. Kakkarla, "Advanced Wild Animal Detection and Alert System Using the YOLO V5 Model Powered by AI", Turkish Journal of Computer and Mathematics Education (TURCOMAT), 2024.

[10] T. Yang, et al., "Wildlife Real-Time Detection in Complex Forest Scenes Based on WL-YOLO", Remote Sensing, 2023.

[11] Hamsa, S, et.al, (2024). IoT-Driven Civil Engineering Solutions for Smart Integrated Agriculture in Controlled Environment. Recent Advances in Civil Engineering for Sustainable Communities. IACESD 2023. Lecture Notes in Civil Engineering, vol 459. Springer, Singapore.

[12] Revathi, B., et.al., (2024). Development of Artificial Intelligence of Things and Cloud Computing Environments Through Semantic Web Control Models. Emerging Technologies for Securing the Cloud and IoT (pp. 112-143). IGI Global.