International Journal of Engineering Technology and Management Sciences
Website: ijetms.in Issue:2 Volume No.10 March - April – 2026
DOI:10.46647/ijetms.2026.v10i02.045 ISSN: 2581-4621

Wildlife Monitoring & Fire Robot

S Ravi Kumar, CH Jaideep Bharadwaj, A Nagi Reddy
Electronics and Communication Engineering, MVSR Engineering College

Abstract— Wildlife monitoring and fire hazard detection are critical challenges in remote and forest environments, where real-time human supervision is often impractical. This paper presents an integrated intelligent robotic system designed for automated wildlife monitoring, fire detection, and response. The proposed system combines embedded artificial intelligence, Internet of Things (IoT) communication, and autonomous robotic navigation to achieve efficient and real-time operation.
An embedded vision-based AI module performs on-device inference to detect entities such as fire, humans, and animals using a trained machine learning model. Upon detection, a camera module is triggered to capture and transmit images to a remote user through a cloud-based messaging platform. The system also incorporates environmental monitoring through periodic temperature sensing.
In parallel, an autonomous robotic platform equipped with obstacle avoidance and fire suppression capabilities actively responds to hazardous situations. The robot utilizes multiple sensors to detect fire and automatically activates a water pump mechanism to extinguish it. The integration of real-time detection, wireless communication, and autonomous response makes the system highly effective for applications such as wildlife surveillance, forest fire prevention, and industrial safety monitoring.

Keywords— Embedded AI, Wildlife Monitoring, Fire Detection, Autonomous Robot, ESP32, IoT, Object Detection, Smart Surveillance, Edge Computing, Fire Extinguishing Robot

I. INTRODUCTION

In recent years, the need for intelligent monitoring systems has increased significantly due to rising concerns in areas such as wildlife conservation, forest fire prevention, and industrial safety. Traditional monitoring approaches rely heavily on manual supervision, which is often inefficient, time-consuming, and impractical in remote or hazardous environments. These limitations highlight the importance of developing automated systems capable of real-time detection, decision-making, and response.

Advancements in embedded systems and artificial intelligence have enabled the deployment of machine learning models directly on low-power devices. Edge computing allows data to be processed locally, reducing latency and enabling faster response times without relying on continuous cloud connectivity. This capability is particularly useful in critical scenarios such as fire detection and wildlife monitoring, where immediate action is required.

The integration of IoT technologies further enhances system capabilities by enabling seamless communication between devices and remote users. Real-time alerts, image transmission, and environmental data monitoring can be achieved through wireless communication platforms, improving situational awareness and decision-making.

This paper proposes a Wildlife Monitoring & Fire Robot System, which combines embedded AI-based vision detection, wireless communication, and an autonomous robotic platform. The system is capable of identifying fire, human presence, and animal activity using a trained machine learning model deployed on an embedded processor. Upon detection, visual data is captured and transmitted to a remote user, enabling real-time monitoring.

In addition to monitoring, the system includes an autonomous robotic unit equipped with obstacle avoidance and fire suppression mechanisms. The robot actively navigates its environment and responds to fire hazards by activating a water pump system, thereby reducing potential damage and risk.

The proposed system is designed to be scalable, cost-effective, and efficient, making it suitable for applications such as wildlife conservation, smart surveillance, forest protection, and industrial

International Journal of Engineering Technology and Management Sciences
Website: ijetms.in Issue:2 Volume No.10 March - April – 2026
DOI:10.46647/ijetms.2026.v10i02.045 ISSN: 2581-4621

hazard management. By integrating AI-driven perception, IT-based communication, and autonomous robotic control, the system provides a comprehensive solution for intelligent environmental monitoring and response.

II. LITERATURE REVIEW

A. Traditional Detection Methods

Early monitoring and detection systems relied on basic sensor-based techniques and manual observation. Fire detection was commonly performed using temperature sensors, smoke detectors, or infrared flame sensors, which operate based on predefined threshold values.

While these methods are simple and cost-effective, they lack the ability to understand context and often result in false alarms due to environmental variations such as sunlight, heat, or reflections.

In wildlife monitoring, traditional approaches included static surveillance cameras and human patrol systems. These systems required continuous human supervision to interpret visual data, making them inefficient for large-scale or remote environments. Additionally, such methods do not provide automated decision-making or real-time response capabilities, limiting their effectiveness in critical situations.

Although traditional methods are reliable for basic detection, they are limited in scalability, intelligence, and adaptability to dynamic environment.

B. Deep Learning Approaches

The introduction of deep learning has significantly improved detection and monitoring systems. Convolutional Neural Networks (CNNs) and embedded AI models enable systems to automatically learn features from visual data and perform accurate object detection. These approaches allow identification of complex entities such as humans, animals, and fire directly from images or video streams.

Recent advancements in edge computing have made it possible to deploy AI models on embedded devices, enabling real-time inference without dependence on cloud processing. This reduces latency and improves system responsiveness, which is critical for applications like fire detection and security monitoring.

Deep learning-based systems also offer higher accuracy and robustness compared to traditional methods, as they can adapt to variations in lighting, background, and object appearance. However, many existing implementations focus primarily on detection and lack integration with autonomous response mechanisms or real-time communication systems.

C. Drawbacks of Existing System

Despite advancements in monitoring technologies, existing systems still face several limitations.

Traditional sensor-based methods rely on fixed thresholds, making them prone to false alarms and unable to interpret complex environmental conditions. They also lack the ability to differentiate between multiple entities such as fire, humans, or animals.

Many surveillance systems depend on continuous human monitoring, which is inefficient and leads to delayed responses, especially in remote areas. Although deep learning approaches improve detection accuracy, several systems require high computational resources and depend on cloud processing, resulting in increased latency.

Furthermore, most existing solutions focus only on detection and do not include automated response mechanisms or real-time communication. The lack of an integrated approach combining monitoring, communication, and action reduces their effectiveness in real-world applications.

D. How the System Works

The proposed system operates through a coordinated sequence of detection, communication, and response. Initially, the embedded AI module continuously captures image frames and performs real-time inference using a model developed on the Edge Impulse platform. The model analyzes the input and detects the presence of fire, humans, or animals based on learned features.

When a target entity is detected, a trigger signal is generated and transmitted to the camera module. Upon receiving this signal, the camera captures an image and sends it to a remote user through a

International Journal of Engineering Technology and Management Sciences
Website: ijetms.in Issue:2 Volume No.10 March - April – 2026
DOI:10.46647/ijetms.2026.v10i02.045 ISSN: 2581-4621

cloud-based messaging platform, enabling real-time alerts and situational awareness. In addition to visual monitoring, the system periodically measures environmental temperature and transmits updates to the user.

Simultaneously, the autonomous robotic platform operates independently, navigating its environment using obstacle detection sensors. The robot continuously monitors for fire using flame sensors, and upon detection, it activates a water pump mechanism to extinguish the fire. The system maintains this response until the hazard is cleared, after which normal operation resumes.

Through the integration of AI-based detection, wireless communication, and autonomous control, the system ensures efficient monitoring and timely response in dynamic environments.

For complete flow chart, refer to fig.1

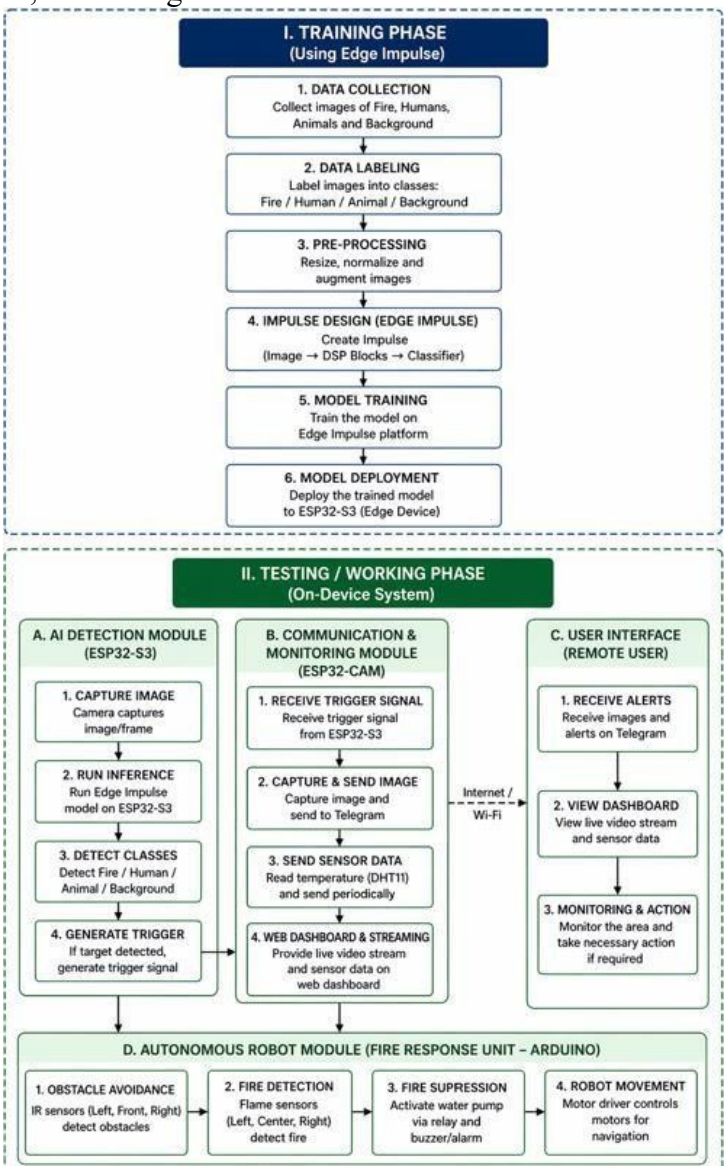

Fig. 1. Overall System Block Diagram

E. Training Phase

1. Dataset

A dataset containing images of humans, animals, fire, and background scenes is collected for training the AI model. The images can be taken from public datasets, online sources, or manually captured using cameras. Each image is labelled according to the object category so that the model can learn the differences between fire, humans, animals, and normal background scenes.

2. Pre-processing & Data Augmentation

International Journal of Engineering Technology and Management Sciences
Website: ijetms.in Issue:2 Volume No.10 March - April – 2026
DOI:10.46647/ijetms.2026.v10i02.045 ISSN: 2581-4621

Before training, all collected images are resized to a fixed size such as 96 x 96 pixels. The images are normalized to improve consistency during model training. Data augmentation techniques such as rotation, flipping, brightness adjustment, zooming, and cropping are applied to increase dataset diversity.

These techniques help the model perform better in different lighting conditions and viewing angles.

3. Learning Patterns (Training)

After pre-processing, the dataset is used to train the object detection model. The ESP32-S3 uses a lightweight AI model created through Edge Impulse. During training, the model learns patterns related to the shape, colour, and structure of humans, animals, and fire. The training process is repeated over multiple epochs until the model can accurately classify the different objects.

4. Validation Accuracy

A separate validation dataset is used to evaluate the model performance during training. Validation helps measure how accurately the model can detect different objects on unseen images. If the validation accuracy is low, additional training or parameter tuning is performed. This step is important to reduce overfitting and improve the overall performance of the AI model.

5. Saved Trained Model

Once the model reaches satisfactory accuracy, the best-performing model is saved. The saved model is then deployed to the ESP32-S3 for real-time object detection. This allows the robot to perform intelligent monitoring directly on the hardware without requiring cloud processing.

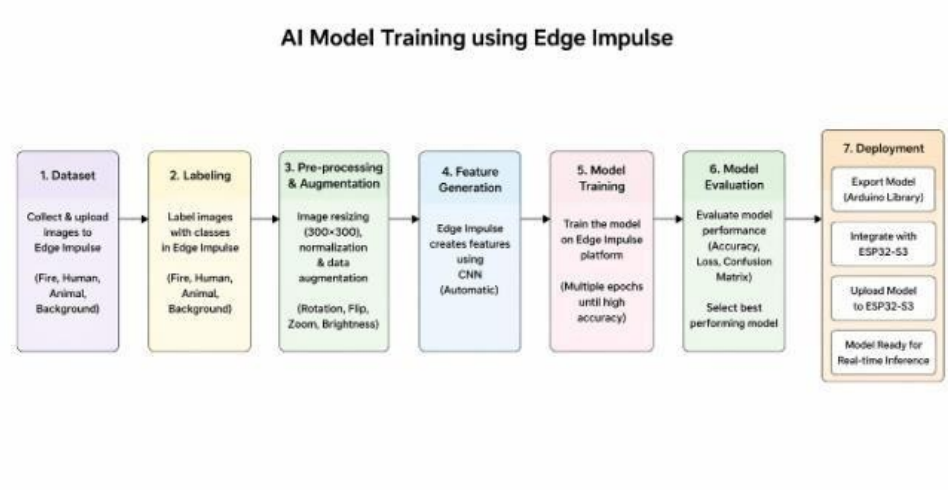

Fig. 2. Training Phase

F. AI Model Training using Edge Impulse

The AI model used in this project was developed using the Edge Impulse platform. Edge Impulse provides an easy way to build, train, and deploy machine learning models for embedded systems such as the ESP32-S3.

A dataset containing images of humans, animals, fire, and normal background scenes was collected for training. These images were uploaded to Edge Impulse and labeled according to their respective classes. The platform then performed image preprocessing, including resizing and normalization, to ensure consistency across the dataset.

To improve model performance, data augmentation techniques such as image flipping, rotation, zooming, brightness adjustment, and cropping were applied. These techniques helped the model learn different object positions, lighting conditions, and viewing angles.

After preprocessing, the training process was started within Edge Impulse. The platform automatically created an object detection model and trained it using the uploaded dataset. During training, the model learned important visual features related to fire, humans, and animals. The training process was repeated for multiple epochs until satisfactory accuracy was achieved.

Once the training was completed, the best-performing model was selected and exported as an Arduino library. The exported model was then integrated into the ESP32-S3 code and uploaded to

International Journal of Engineering Technology and Management Sciences
Website: ijetms.in Issue:2 Volume No.10 March - April – 2026
DOI:10.46647/ijetms.2026.v10i02.045 ISSN: 2581-4621

the microcontroller. This allowed the ESP32-S3 to perform real-time object detection directly on the hardware without requiring cloud processing.

G. Testing Phase (fig 4)

1. Input Image (96 x 96)

During the testing phase, the camera connected to the ESP32-S3 captures an image of the surrounding environment. The captured image may contain fire, humans, animals, or normal background scenes. The image is resized to 96 x 96 pixels so that it matches the input size required by the trained AI model.

2. Pre-processing

Before the image is passed to the trained model, it undergoes preprocessing. In this step, the image is resized, normalized, and converted into the required format for inference. Preprocessing ensures that the image data is consistent and suitable for accurate object detection.

3. Trained Model

The pre processed image is then passed to the trained Edge Impulse model running on the ESP32-S3. The model has already been trained to recognize different classes such as fire, humans, animals, and background scenes. During inference, the model analyses the input image and predicts the most suitable object class.

4. Object Detection

The trained model examines the image and identifies whether any target object is present. If a fire, human, or animal is detected, the system marks it as a valid object of interest. If no important object is detected, the image is classified as a normal background scene.

5. Confidence Score Generation

For every detected object, the model generates a confidence score. This score indicates how strongly the model believes that the detected object belongs to a particular class. Higher confidence values indicate more reliable predictions. The system only responds when the confidence score crosses a predefined threshold.

6. UART Trigger Signal to ESP32-CAM

When the confidence score is sufficiently high, the ESP32-S3 sends a UART signal "1" to the ESP32-CAM. This signal acts as a trigger to inform the ESP32-CAM that an important object has been detected.

7. Image Capture

After receiving the UART trigger signal, the ESP32-CAM captures a high-quality image of the detected scene. This image is stored temporarily and prepared for transmission to the user.

8. Telegram Alert

The ESP32-CAM sends the captured image to the user through Telegram using a bot API. Along with the image, the system may also send a text message indicating the type of detected object, such as fire, human, or animal. This provides real-time alerts to the user even when they are far from the robot.

9. Live Streaming and Monitoring

The ESP32-CAM also hosts a web server that provides live video streaming. Users can access the robot through its IP address in a browser and monitor the surroundings in real time. The web interface also allows users to control robot movement and enable or disable auto mode.

10. Autonomous Robot Module

The Autonomous Robot Module operates simultaneously during the testing phase. The robot navigates using IR sensors for obstacle avoidance and continuously monitors fire using flame sensors. When fire is detected, the system immediately activates the water pump through a relay and triggers a buzzer for alert. The robot continues the fire suppression process until the fire is completely extinguished, after which it resumes normal navigation.

11. Final Output Display

The final output of the system is presented to the user through multiple interfaces, including image alerts, temperature updates, and live monitoring via the web dashboard. This ensures effective real-

International Journal of Engineering Technology and Management Sciences
Website: ijetms.in Issue:2 Volume No.10 March - April – 2026
DOI:10.46647/ijetms.2026.v10i02.045 ISSN: 2581-4621

time awareness and response, validating the system's capability for intelligent monitoring and autonomous action.

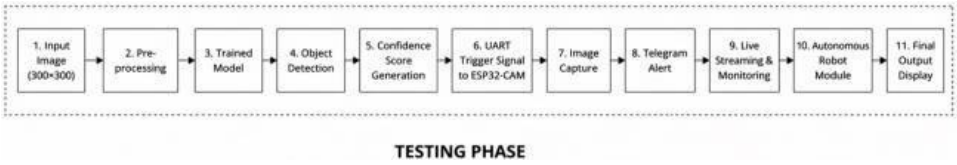

Fig. Testing phase Block Diagram

III. DATASET AND EXPERIMENTAL SETUP

A. Where the Data Came From

The dataset used in this work was constructed by combining multiple publicly available image datasets from Kaggle, ensuring diversity and robustness for training the model using the Edge Impulse platform. For fire detection, the Wildfire Detection Image Data dataset was utilized, which contains images categorized into two primary classes: fire and non-fire. This dataset includes real-world wildfire scenarios captured under different environmental conditions, enabling the model to learn visual patterns associated with fire presence.

To incorporate animal detection capabilities, an animals image dataset was used, consisting of various animal classes captured in different environments, poses, and lighting conditions. This diversity helps the model generalize better when identifying animals in real-world wildlife monitoring scenarios. Similarly, for human detection, two separate datasets were utilized to improve accuracy and variability. These datasets contain images of people in different backgrounds, orientations, and scales, allowing the model to learn robust human features across diverse conditions.

By combining these datasets, a multi-class dataset was created containing four categories: fire, human, animal, and background. The integration of multiple sources ensures variability in image quality, lighting, and environmental conditions, which improves the generalization capability of the trained model. This combined dataset was then used within the Edge Impulse platform for preprocessing, labeling, and training, enabling efficient deployment on embedded hardware for real-time inference.

TABLE I. DATASET CLASS SOURCES

Class	Main Source Datasets	How Samples Were Made
Fire	Wildfire Detection Image Dataset (Kaggle)	Real wildfire and non-fire images collected from natural environments
Human	People Detection Dataset + Human Detection Dataset (Kaggle)	Images of humans in different poses, backgrounds, and lighting conditions
Animal	Animals Detection Images Dataset (Kaggle)	Images of various animals captured in different environments and viewpoints
Background	Combined from all datasets	Images without fire, humans, or animals used as negative/background class

B. Training Settings

The experimental setup and training configuration of the proposed system are summarized in detail in Table II. This includes parameters such as image input size, processing, training cycles, and data splitting methods used in the Edge Impulse platform. All relevant implementation details related to dataset preparation, feature generation, and model evaluation are also incorporated.

International Journal of Engineering Technology and Management Sciences
Website: ijetms.in Issue:2 Volume No.10 March - April – 2026
DOI:10.46647/ijetms.2026.v10i02.045 ISSN: 2581-4621

TABLE II. MODEL TRAINING CONFIGURATION

Setting	Value
Training Platform	Edge Impulse
Model Type	Image Classification (No Bounding Boxes)
Dataset Classes	Fire, Human, Animal
Data Split	Training: 80 % / Testing: 20 %
Image Input Size	96 × 96 pixels
Learning Rate	Default (Edge Impulse)
Training Cycles (Epochs)	50
Deployment Target	ESP32-S3 (On-device inference)
Output	3-class Softmax

IV. RESULTS AND DISCUSSION

A. Overall Results

The final model achieved an accuracy of 84.5%, weighted average precision of 0.84, weighted average recall of 0.84, weighted average F1-score of 0.84, and ROC-AUC of 0.94 with a loss of 0.36. These results show that the model performs well for detecting animals, fire, and humans.

TABLE III. FINAL MODEL PERFORMANCE

Metric	Validation	Test
Accuracy (%)	84.50	84.50
Weighted Average Precision	0.8400	0.8400
Weighted Average Recall	0.8400	0.8400
Weighted Average F1-Score	0.8400	0.8400
Area Under ROC Curve	0.9400	0.9400
Loss	0.36	0.36

B. Per-Class Results

The per-class F1-scores for animals, fire, and humans are shown in Table IV. Fire achieved the highest F1-score of 0.97, followed by animals with 0.85. Humans achieved the lowest F1-score of 0.69 due to confusion with animal images.

TABLE IV. PER-CLASS TEST F1-SCORES

Class	F1-Score	Notes
Animals	0.8500	Good performance; some confusion with humans
Fire	0.9700	Best-performing class; highly distinguishable features
Humans	0.6900	Lowest-performing class; often confused with animals

International Journal of Engineering Technology and Management Sciences
Website: ijetms.in Issue:2 Volume No.10 March - April – 2026
DOI:10.46647/ijetms.2026.v10i02.045 ISSN: 2581-4621

Macro Average	0.8367	Balanced performance across all three classes

C. User Interface and Output Demonstration

This section presents the user interface and output results of the proposed autonomous wildlife observation robot. It includes the web dashboard, live video streaming, AI detection outputs, Telegram alerts, and the robot's autonomous actions during real-time operation..

1. Web Dashboard Interface

This figure shows the main web dashboard used to control the robot. The dashboard includes options for starting or stopping the live video stream, selecting the operating mode, and monitoring sensor data.

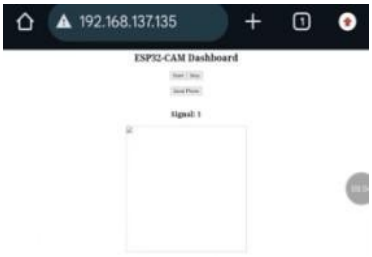

Fig. 6. Web Dashboard Interface

2. Live Video Streaming Output

This figure shows the live video stream from the camera mounted on the robot. The user can monitor the surrounding environment in real time through the web interface.

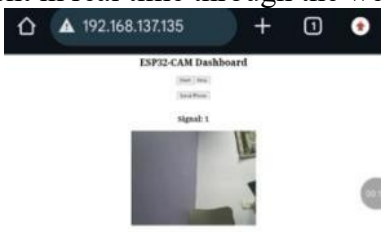

Fig. 7. Live Video Streaming Output

3. Animal Detection Output

This figure shows the output when the AI model detects an animal. The detected class label and confidence score are displayed on the screen.

Fig. 8 Animal Detection Output

International Journal of Engineering Technology and Management Sciences
Website: ijetms.in Issue:2 Volume No.10 March - April – 2026
DOI:10.46647/ijetms.2026.v10i02.045 ISSN: 2581-4621

4. Human Detection Output:

This figure shows the output when the AI model detects a human. The system identifies the person and displays the corresponding detection result.

Fig. 9 Human Detection Output

5. Fire Detection Output

This figure shows the output when the AI model detects fire. The fire class label is displayed along with the confidence score.

Fig. Fire Detection Output

6. Telegram Image Output

This figure shows the captured image sent through Telegram after a detection event. The image helps the user verify the detected object remotely.

International Journal of Engineering Technology and Management Sciences
Website: ijetms.in Issue:2 Volume No.10 March - April – 2026
DOI:10.46647/ijetms.2026.v10i02.045 ISSN: 2581-4621

Fig. Telegram Image Output

7. Autonomous Navigation Output

This figure shows the robot moving autonomously while avoiding obstacles using IR sensors and motor control logic.

Fig. Fire Detection Output

8. Final System Output

This figure shows the complete working prototype of the autonomous wildlife observation robot during real-time operation.

International Journal of Engineering Technology and Management Sciences
Website: ijetms.in Issue:2 Volume No.10 March - April – 2026
DOI:10.46647/ijetms.2026.v10i02.045 ISSN: 2581-4621

V. CONCLUSION

This Robot is designed and implemented for Wildlife Monitoring & Fire Robot System, integrating embedded artificial intelligence, IoT-based communication, and autonomous robotic control. The system utilizes a machine learning model developed using the Edge Impulse platform to perform real-time detection of fire, humans, and animals on an embedded device. The use of edge computing enables fast and efficient inference without reliance on cloud processing.

Upon detection, the system provides real-time alerts by capturing and transmitting images through a wireless communication module, along with periodic environmental updates such as temperature. In addition, the autonomous robotic module enhances the system by enabling obstacle avoidance and active fire suppression using flame sensors and a water pump mechanism.

The overall system demonstrates an effective combination of intelligent monitoring and automated response, making it suitable for applications such as wildlife surveillance, forest fire prevention, and smart safety systems. The proposed solution is cost-effective, scalable, and capable of operating in real-time environments, thereby contributing to advancements in embedded AI and robotics-based monitoring systems.

VI. REFERENCES

[1] A. Shamim, "People Detection Dataset," Kaggle, 2023. [Online]. Available: https://www.kaggle.com/datasets/adilshamim8/people-detection

[2] A. Jana, "Animals Detection Images Dataset," Kaggle, 2023. [Online]. Available: https://www.kaggle.com/datasets/antoreepjana/animals-detection-images-dataset

[3] B. R. S. Dincer, "Wildfire Detection Image Data," Kaggle, 2023. [Online]. Available: https://www.kaggle.com/datasets/brsdincer/wildfire-detection-image-data

[4] DFRobot, "DFR1154 AI Camera Module Wiki," DFRobot, 2024. [Online]. Available: https://wiki.dfrobot.com/dfr1154/

[5] DFRobot, "Train the Edge Impulse Model Using DFR1154," GitHub, 2024. [Online]. Available: https://github.com/DFRobot/DFR1154_Examples

[6] Edge Impulse, "Edge AI Development Platform for Embedded Machine Learning," Edge Impulse Inc., 2024. [Online]. Available: https://www.edgeimpulse.com

[7] Espressif Systems, "ESP32-S3 Series Datasheet," Espressif Systems, 2023. [Online]. Available: https://www.espressif.com

DOI:10.46647/ijetms.2026.v10i02.045 ISSN: 2581-4621

[8] Espressif Systems, "ESP32-CAM Technical Reference Manual," Espressif Systems, 2023. [Online]. Available: https://www.espressif.com
[9] Arduino, "Arduino Uno Rev3 Documentation," Arduino, 2023. [Online]. Available: https://www.arduino.cc
[10] Motor Driver sheild, "L298P Dual H-Bridge Motor Driver Datasheet," 2023. [Online]. Available: G. Zampoglou, S. Papadopoulos, and Y. Kompatsiaris, "Detecting Image Splicing in the Wild," Proc. IEEE ICMEW, 2015