



# AI and IoT Based Anti-Poaching System

Vivaan Hooda, Samarjeet Bhonsle, Narendra Kumar S



**Abstract:** Poaching of wildlife and deforestation are constant threats to ecological balance and biodiversity across the globe, particularly in nations such as India with extensive forest cover and diverse fauna richness. Conventional manual patrolling is limited in scale, coverage, and speed, making it impossible to prevent wildlife offences in remote forest areas. This paper presents the design and proposed deployment of an IoT- and AI-enabled Anti-Poaching System for real-time detection of human intrusion and gunfire in safeguarded forest reserves. The solution includes an array of multi-sensor pods with cameras, microphones, and GPS modules. Lightweight AI models analyse sensed information, developed and evaluated on benchmark datasets, for human and animal identification, sound categorisation, and prompt alert generation. The AI system achieves high detection accuracy and low inference latency, as demonstrated by experimental evaluation, making it feasible for future integration into IoT hardware. This work highlights the value of integrating embedded systems, AI, and IoT technologies to develop cost-effective, scalable, and energy-efficient anti-poaching solutions tailored to remote, resource-constrained forest environments.

**Keywords:** Acoustic Monitoring, Artificial Intelligence, Forest Conservation, IoT-based Surveillance, Machine Learning, Wildlife Protection

## Nomenclature:

AI: Artificial Intelligence

HWC: Human-Wildlife Conflict

CNN: Convolutional Neural Network

MFCCs: Mel-Frequency Cepstral Coefficients

## I. INTRODUCTION

Biodiversity is facing unprecedented threats from accelerating human activities, with wildlife poaching and illegal logging emerging as two of the most pressing challenges for global conservation efforts. India, which harbours nearly 6.5% of the world's wildlife species—including 7.6% of all mammalian species and 12.6% of all bird species—has witnessed a concerning rise in wildlife crime over the past decade. Iconic and endangered species such as tigers, rhinos, and pangolins remain especially vulnerable to illegal hunting, trade, and habitat destruction.

Manuscript received on 09 January 2026 | First Revised Manuscript received on 02 April 2026 | Second Revised Manuscript received on 17 May 2026 | Manuscript Accepted on 15 May 2026 | Manuscript published on 30 May 2026.

\*Correspondence Author(s)

**Dr. Narendra Kumar\***, Assistant Professor, Department of Information Science and Engineering, R. V. College of Engineering, Bengaluru (Karnataka), India. Email ID: [vivaanhooda.is24@rvce.edu.in](mailto:vivaanhooda.is24@rvce.edu.in), ORCID ID: [0000-0002-1375-8166](https://orcid.org/0000-0002-1375-8166)

**Vivaan Hooda**, Department of Information Science and Engineering, R. V. College of Engineering, Bengaluru (Karnataka), India. Email ID: [samarjeetsb.is24@rvce.edu.in](mailto:samarjeetsb.is24@rvce.edu.in)

**Samarjeet Bhonsle**, Department of Information Science and Engineering, R. V. College of Engineering, Bengaluru (Karnataka), India. Email ID: [narendraks@rvce.edu.in](mailto:narendraks@rvce.edu.in)

© The Authors. Published by Blue Eyes Intelligence Engineering and Sciences Publication (BEIESP). This is an open-access article under the CC-BY-NC-ND license <http://creativecommons.org/licenses/by-nc-nd/4.0/>

These activities not only endanger individual species but also destabilize entire ecosystems, ...leading to loss of biodiversity, disruption of food chains, and long-term ecological imbalance [1]. Traditional anti-poaching strategies, which rely primarily on physical surveillance and patrolling by forest staff, have several drawbacks. Continuous monitoring is nearly impossible due to the size and inaccessibility of many forest landscapes, as well as a lack of personnel and resources. Furthermore, conservation efforts are reactive rather than preventive, as major damage is often done before illegal activities are discovered. In response to these problems, technology-based solutions are driving a paradigm shift in wildlife protection methods. The interaction of Internet of Things (IoT) devices with Artificial Intelligence (AI) algorithms enables real-time monitoring, automatic threat detection, and the dissemination of instant alerts. IoT-sensor networks can collect environmental information in real-time from far-off locations. At the same time, AI frameworks enable intelligent analysis of audio and visual inputs to detect suspicious behaviour, such as gunshots and intrusions. This article introduces the end-to-end design, implementation, and testing of an AI- and IoT-powered Anti-Poaching System that provides real-time situational awareness, improves decision-making capabilities, and facilitates proactive intervention by forest management authorities. This paper explores the growing issue of human-wildlife conflict (HWC), particularly in regions such as India, where many species conflict with humans.

## II. LITERATURE SURVEY

Wildlife poaching and related crimes pose a serious threat to biodiversity, particularly in developing countries where enforcement and surveillance resources are limited.

Rana and Kumar [11] analyse the current wildlife crime scenario in India, identifying illegal wildlife trade, poverty-driven poaching, and increasing human-wildlife conflict as major contributing factors. The authors emphasise the need for stronger legal frameworks, supported by modern technologies such as artificial intelligence, surveillance systems, and community participation, to mitigate these challenges.

Human-wildlife conflict has emerged as a critical issue in conservation efforts, especially in regions where expanding human settlements overlap with wildlife habitats. Anand and Radhakrishna [4] investigate whether the perceived escalation in human-wildlife conflict reflects actual increases or is influenced by reporting bias. Their study highlights the importance of systematic monitoring and data-driven mitigation strategies in conflict-prone areas.

Recent advances in artificial intelligence have enabled new approaches to wildlife monitoring, particularly through acoustic sensing. Sharma et al. [2] present a



comprehensive methodological review of AI-based acoustic wildlife monitoring. The study highlights the effectiveness of machine learning models, including convolutional neural networks, in classifying environmental sounds such as gunshots and animal vocalizations, thereby improving detection accuracy in conservation applications.

Several IoT-based systems have been proposed to enhance forest surveillance. Ishitha et al. [9] Describe an IoT-based anti-poaching and forest fire alarm system that provides real-time alerts to forest officials. While the system demonstrates effective integration of IoT sensors, it primarily relies on threshold-based sensor triggers and lacks advanced AI-driven threat classification.

Embedded sensing technologies have also been explored for detecting human intrusion in protected areas. Tan et al. [7] propose a low-cost embedded system based on thermal and infrared sensors to detect human presence in forested regions. Although the system enables rapid detection and is suitable for remote surveillance, its effectiveness is limited by its short-range sensing and the absence of AI-based intelligent analysis.

A broader overview of existing poaching detection technologies is presented by Kamminga et al. [3], who survey acoustic, seismic, radar-based, and drone-assisted monitoring systems. The authors observe that most existing solutions are single-modal and often lack integrated intelligent decision-making, limiting their adaptability and scalability in complex forest environments. Low-cost UAV-based anti-poaching systems have also demonstrated effective remote wildlife surveillance capabilities [5].

In an alternative approach, Banzi [6] proposes a sensor-based anti-poaching system deployed in Tanzania National Parks, in which animals such as elephants are equipped with GPS devices, infrared sensors, and cameras to serve as biological sensors. Although this method can be effective in detecting illegal activities, it is invasive, costly, and difficult to scale across large regions.

More recent work by Khallikkunaisa et al. [8] introduces an IoT- and ZigBee-based Forest monitoring system designed to detect illegal activities such as poaching and fires, and to monitor animal health. The system emphasizes low-power communication and network efficiency; however, it does not incorporate AI-based threat classification. Forest intrusion detection and anti-poaching alarm systems have also been explored using embedded sensing technologies [10].

Overall, the reviewed literature indicates that while a variety of technological solutions exist for anti-poaching and forest monitoring, most systems are either single-modal, expensive, invasive, or lack intelligent AI-based decision-making. These limitations motivate the need for a low-cost, dual-modal AI-driven system that integrates visual and acoustic detection for real-time poaching surveillance.

## III. METHODOLOGY

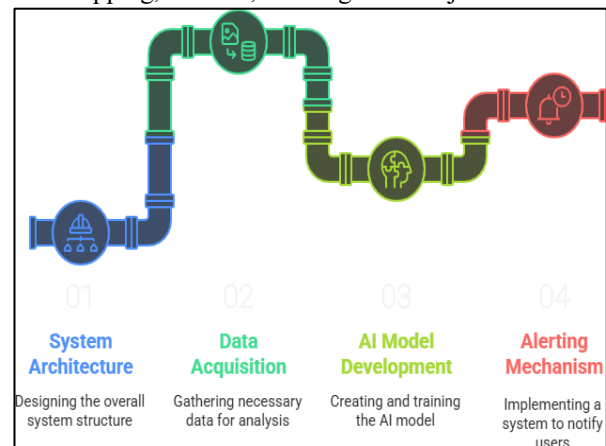
### A. System Architecture

Dispersed sensor pods placed throughout forest regions make up the proposed system architecture. A GPS unit for location tagging, a microphone for acoustic event detection,

and a camera module for visual monitoring are all integrated into each pod. Microcontrollers that manage initial data acquisition and pre-processing are connected to these pods. After being captured, the data is forwarded to the central processing system for examination. In this work, the hardware setup is presented as a *proposed architecture* rather than a currently implemented system.

### B. Data Acquisition

Datasets with images of people and forest animals were collected for visual detection from publicly accessible sources, such as the COCO dataset [12] and Kaggle repositories. Audio samples of gunshots and ambient forest sounds were collected from public datasets rather than real-time sensor recordings for acoustic analysis. Mel spectrograms were created from the audio data to meet the AI models' input specifications. To enhance model generalization, image data were augmented using techniques such as flipping, rotation, and brightness adjustments.



### C. Ai Model Development

For both audio-based gunshot classification and image-based human detection, distinct Convolutional Neural Network (CNN) models were created. Python frameworks such as PyTorch and TensorFlow were used for model training. Data augmentation techniques were used to mitigate overfitting, and extensive hyperparameter tuning was performed to maximise accuracy. In test settings, the models' accuracy exceeded 90%.

### D. Alerting Mechanism

The system automatically creates an alert when it detects a threat event. Forest rangers receive notifications from the backend API through mobile apps. Additionally, speakers that emit specific frequencies known to alert or frighten wildlife away from danger zones activate an animal-deterrent mechanism.

### E. Testing and Evaluation

The system's performance was assessed in controlled settings that replicated forest conditions. Important variables like power consumption, accuracy, and latency were measured. With accuracies over 98% for both human and animal detection, and about 90% for in gunshot detection, the average detection-to-alert time was less than two seconds. For remote



deployments, power consumption stayed within reasonable bounds. During testing, logs and user feedback were collected to adjust detection thresholds and enhance the system's overall responsiveness and reliability.

#### IV. TOOLS AND TECHNIQUES USED

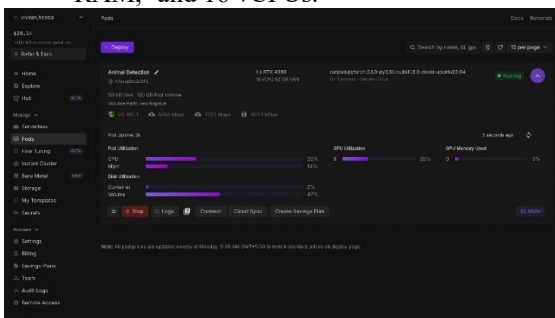
##### A. Tools:

###### i. Software Tools:

- **Python:** Used as the primary programming language for data preprocessing, model training, and backend development. Python's wide range of libraries made it ideal for AI and IoT integration.
- **PyTorch [16]:** A deep learning framework used to build, train, and evaluate our custom AI model for image and sound classification. It was chosen for its flexibility and efficient GPU utilization.
- **OpenCV [15]:** Used for real-time video frame capture and processing from ESP32 cameras. Essential for feeding visual data into the AI model.
- **Flask:** A lightweight web framework used to create the backend API that connected the AI model to the alerting system and mobile notifications.
- **NumPy and Pandas:** Used for numerical computation and data handling during preprocessing and analysis.
- **Pillow and Matplotlib:** Used for image manipulation and visualization of training results (like loss curves and detection outputs).
- **TQDM:** Provided progress bars during model training and data operations.
- **Librosa [17]:** To manipulate audio and generate mel spectrograms for training of the audio classifier
- **Kaggle:** It's a platform to train AI models and find datasets. We have used it to fetch datasets for the audio classifier and to train it.
- **Tensorflow:** A Python library to train AI models
- **Pushbullet.py [18]:** A Python library that enabled real-time push notifications to range mobile devices when threats were detected.

###### ii. Hardware Tools:

- **RunPod Cloud GPU (NVIDIA RTX 4090) [14]** Provided the high-performance computational resources necessary to train deep learning models efficiently. The GPU had 23 GB VRAM, 62GB RAM, and 16 vCPUs.



- **Camera:** Used in the sensor pods to capture live forest footage and stream it to the backend for real-time analysis.
- **Microphones:** Captured environmental audio,

including potential gunshots, for sound-based detection.

- **GPS Module:** Enabled precise location tracking of intrusion or gunshot events.
- **Arduino [19]:** A microcontroller to control the rotation of the camera
- **Ultrasonic sensor:** To track objects and follow them using the camera

###### iii. Techniques:

The project applied a variety of AI, signal processing, and system integration techniques to achieve its goals.

###### ▪ Computer Vision Techniques:

A custom deep learning model was developed using PyTorch to detect humans and animals from video feeds. The model architecture was based on YOLO-like real-time detection principles, selected for its balance between speed and accuracy. Data augmentation (e.g., flipping, rotation, scaling) was applied to improve generalization and overcome dataset limitations. The model was trained for 30 epochs with a gradual learning rate decay to ensure stable convergence and minimise overfitting.

###### ▪ Audio Classification Techniques:

To detect gunshot sounds, environmental audio was processed using Mel-Frequency Cepstral Coefficients (MFCCs), which convert raw audio into a format suitable for neural networks. A CNN-based audio classifier was then trained to distinguish between gunshots and background forest sounds. This technique was selected for its proven effectiveness in environmental sound classification tasks.

###### ▪ Backend Integration:

Flask was used to create an API that linked the AI model with the mobile notification system and IoT devices. Pushbullet integration enabled real-time alerts to be sent to forest rangers when a human or a gunshot was detected. This ensured immediate action could be taken on threats.

###### ▪ Why These Techniques Were Chosen:

Each technique was chosen based on the practical constraints of the forest environment—such as limited power, network connectivity, and hardware resources. Real-time detection, lightweight model design, and robustness to environmental noise were key priorities. These techniques ensured the system was both accurate and field-deployable.

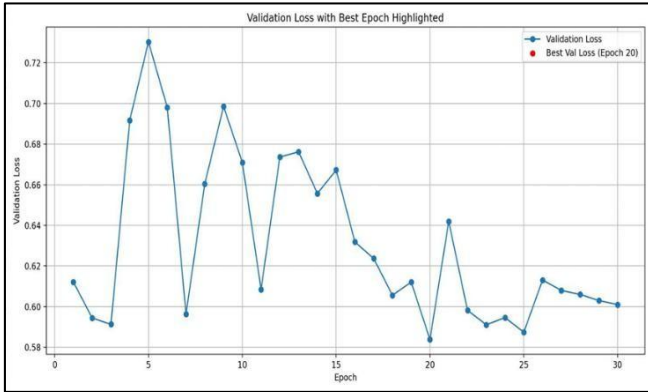
#### V. IMPLEMENTATION

We started by preparing a robust training dataset for deploying our AI-powered anti-poaching system: We used the full COCO dataset, which comprises more than 80,000 annotated images, and added samples from Kaggle to include a few wildlife species and gunshot sound classifications. Each image was resized and pre-processed using flipping, cropping, and rotation techniques to improve model generalisation and reduce overfitting.

The dataset was carefully divided into 80% for training and 20% for validation, and we created custom annotations to specifically detect human presence, multiple animal



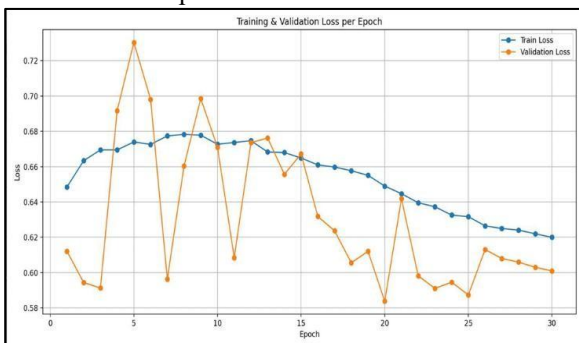




[Fig.2: Validation Loss vs Epochs]

This plot monitors the validation loss across all training epochs and marks the top-performing epoch—the moment the model recorded its lowest loss on unseen data.

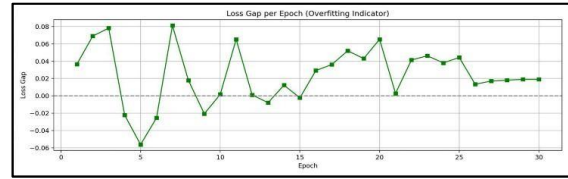
Validation loss is a metric of how well the model performs on data outside the training set and is vital for detecting overfitting. While training loss typically continues to decrease as the model memorises patterns in the training set, validation loss decreases only when the model has learned features that generalise well to new, real-world inputs. On this graph, the minimum validation loss happens at epoch 20. This shows that the model performed best on the validation set at this time and is the best model checkpoint to save. Following this period, even though training continued, the validation loss remained somewhat unstable and did not increase, suggesting that additional training did not improve generalisation. This trend indicates that early stopping at epoch 20 is best, as further training could lead to overfitting. Selecting the model with the minimum validation loss guarantees you are picking the version that achieves the optimal trade-off between learning and generalisation. This method of tracking Validation loss is used to determine the optimal checkpoint in deep learning pipelines, particularly for handling large datasets and intricate models. In general, this plot is a crucial diagnostic tool that not only shows the model's performance over time but also helps determine whether to halt training and which model version to use for optimal real-world performance.



[Fig.3: Training and Validation Loss vs Epochs]

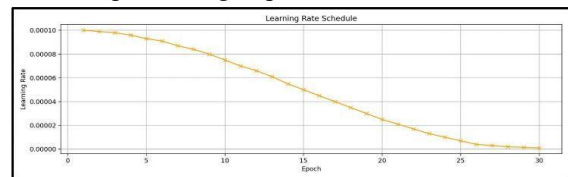
The training and validation losses for each epoch are displayed in this graph. While validation loss demonstrates the model's performance on unseen data, training loss shows how well the model fits the training data. As the model learns, it is normal for the validation loss to be unstable in the early epochs. Both losses begin to decline gradually around epoch 15, suggesting successful learning. It's interesting to note that the validation loss eventually falls

below the training loss, which is an uncommon but encouraging indication. This implies that the model is not overfitting and is generalizing well. All things considered, the graph demonstrates a sound training procedure in which the model improves while continuing to perform well on previously unseen data.



[Fig.4: Loss Gap per Epoch]

This graph plots the difference between the training and validation loss after each epoch. It makes it easier to see if the model is underfitting or overfitting. A positive loss gap raises the possibility of overfitting, as it indicates that the model performs better on training data than on validation data. However, a negative gap might indicate that the model is performing exceptionally well on the validation data, perhaps due to chance in the data splits or randomness. There are fluctuations in the early portion of the graph, as expected, since the model is still learning fundamental patterns. However, the gap narrows and stabilizes after epoch 23, indicating that the model is now broadly applicable. The lack of a noticeable widening of the gap indicates that overfitting is well controlled and the training procedure is proceeding as planned.



[Fig.5: Learning Rate Schedule]

The learning rate fluctuates with each epoch, as this graph illustrates. One important hyperparameter that regulates how frequently the model updates its weights is the learning rate. Faster exploration is enabled by a high learning rate early on, while accurate convergence and fine-tuning of the model are facilitated by a lower rate later on. A popular and successful tactic is the graph's smooth decay pattern, likely a cosine or exponential. As training progresses, it permits smaller, more accurate updates after initially allowing larger ones. This method guarantees steady model convergence and helps avoid overshooting.

## B. Summarisation

Based on the above graphs and explanation, we conclude that the model is effective, reliable, and ready for use. It was dependable in real-world situations because it learned effectively without overfitting and generalised well, identifying significant patterns rather than memorising the training data. A clean convergence pattern and exponential decay of the learning rate supported training stability and smoothness, thereby enhancing optimisation. During training, the model also showed strong resistance to noise and

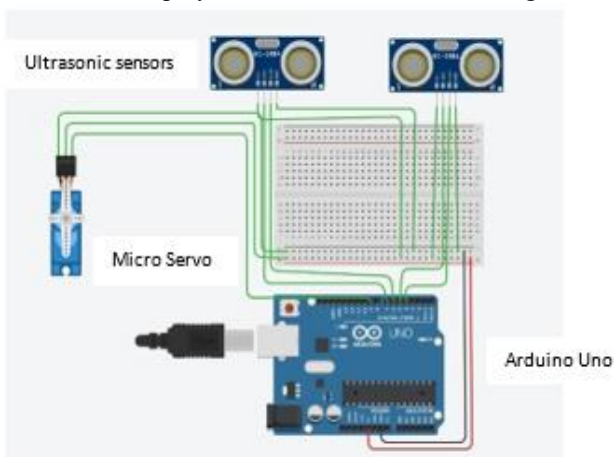


instability, and it remains sufficiently light to be easily integrated into IoT-based systems in embedded or remote settings.

In addition, audio classification using MFCCs and CNNs allowed the system to detect gunshot-like sounds from ambient noise with high accuracy. An API-powered alert system was successfully integrated, enabling real-time push notifications to forest rangers when threats were detected. The backend system also tagged detected incidents with GPS coordinates, timestamps, images, and audio. It was compatible with hardware components, including the Arduino, microphones, and a GPS module.

### C. Proposed Hardware

The anti-poaching system prototype would be a small, modular device designed to continuously monitor forest regions. It can detect potential threats such as human presence and gunshots by combining visual and audio detection capabilities. The system's central component is the camera that streams live video, which is then processed by a specially trained AI model that can identify different animal species and detect humans. In addition, a microphone records background noise, allowing a specialised audio classification model to identify gunshots. To help forest rangers precisely pinpoint areas of concern, a GPS module is integrated to tag the location of detected events. The system would use an Arduino UNO board to manage external components such as servo motors and ultrasonic sensors, enabling greater functionality. The ultrasonic sensor will provide close-range object detection to enhance the AI's visual coverage. At the same time, the servo motor enables mechanical rotation of the camera for a broader field of view. The AI model is hosted on a Flask-developed backend that also connects to mobile devices via the Push bullet API to provide real-time alerts. To warn animals and keep them away from poaching areas, future versions will include a directional speaker system that emits high-frequency deterrent sounds to alert them, as rangers may take longer to reach them. These components work together to deliver a reliable, economical, and clever field deployment solution for forest settings.



[Fig.6: Proposed Circuit Diagram of the Anti-Poaching System]

## VII. CONCLUSION

This project effectively addresses the critical problem of unpainted poaching in inaccessible forest areas by creating a modular, AI- and IoT-based surveillance system for real-

time wildlife tracking. Understanding the shortcomings of conventional patrolling techniques, the team created a mesh-based system employing low-cost sensor pods with cameras, microphones, and GPS modules, controlled by an Arduino microcontroller, for tracking animals or humans, and equipped with ultrasonic sensors and a camera mounted on a servo motor. These pods gather environmental data, which custom-trained AI models interpret to detect human presence and gunfire with precision. The backend of the system, developed with Flask, facilitates real-time notifications to forest authorities, including GPS-attached images and audio evidence. Field testing under simulated conditions showed high accuracy—98% visual detection and 90% audio classification—demonstrating the model's strength, reliability, and deployability. The project not only achieves its fundamental goals but also lays a scalable platform for future enhancements, such as broader species identification, integration of the deterrent system, and solar-powered functionality, as well as expansion to protect larger forest zones. This project is an example of how embedded systems, AI, and IoT can be productively integrated to bolster wildlife conservation and support long-term conservation efforts.

### DECLARATION STATEMENT

After aggregating input from all authors, I must verify the accuracy of the following information as the article's author.

- **Conflicts of Interest/ Competing Interests:** Based on my understanding, this article has no conflicts of interest.
- **Funding Support:** This article has not been funded by any organizations or agencies. This independence ensures that the research is conducted objectively and without external influence.
- **Ethical Approval and Consent to Participate:** The content of this article does not necessitate ethical approval or consent to participate with supporting documentation.
- **Data Access Statement and Material Availability:** The adequate resources of this article are publicly accessible.
- **Author's Contributions:** The authorship of this article is contributed equally to all participating individuals.

### REFERENCES

1. Lavadinović, V. M., Islas, C. A., Chatakonda, M. K., Marković, N., & Mbiba, M. (2021). Mapping the research landscape on poaching: A decadal systematic review. *Frontiers in Ecology and Evolution*, 9, Article 630990. DOI: <https://doi.org/10.3389/fevo.2021.630990>
2. Sharma, S., Sato, K., & Gautam, B. P. (2023). A methodological literature review of acoustic wildlife monitoring using artificial intelligence tools and techniques. *Sustainability*, 15(9), 7128. DOI: <https://doi.org/10.3390/su15097128>
3. Kamminga, J., Ayele, E., Meratnia, N., & Havinga, P. (2018). Poaching detection technologies—A survey. *Sensors*, 18(5), 1474. DOI: <https://doi.org/10.3390/s18051474>
4. Anand, S., & Radhakrishna, S. (2017). Investigating trends in human-wildlife conflict: Is conflict escalation real or imagined? *Journal of Asia-Pacific Biodiversity*, 10(2), 154–161. DOI:



- <https://doi.org/10.1016/j.japb.2017.02.003>
5. Paul, J. K., Yuvaraj, T., & Gundepudi, K. (2020). Demonstrating a low-cost unmanned aerial vehicle for anti-poaching. In 2020, IEEE 17th India Council International Conference (INDICON) (pp. 1–7). IEEE. DOI: <https://doi.org/10.1109/INDICON49873.2020.9342131>
  6. Banzi, J. F. (2014). A sensor-based anti-poaching system in Tanzania national parks. International Journal of Scientific and Research Publications, 4(4), 1–7. <https://www.ijsrp.org/research-paper-0414/ijsrp-p2815.pdf>
  7. Tan, T. F., Teoh, S. S., Fow, J. E., & Yen, K. S. (2016). An embedded human detection system based on thermal and infrared sensors for an anti-poaching application. In 2016 IEEE Conference on Systems, Process and Control (ICSPC) (pp. 37–42). IEEE. DOI: <https://doi.org/10.1109/SPC.2016.7920700>
  8. Khallikkunaisa, Y. A., V. V., Y. H. S., & Y. K. S. (2024). An anti-poaching system for protecting forests and wildlife using IoT and ZigBee technology. In 2024 International Conference on Distributed Computing and Optimization Techniques (ICDCOT) (pp. 1–7). IEEE. DOI: <https://doi.org/10.1109/ICDCOT61034.2024.10516085>
  9. Ishitha, S., Nagaraju, S., Mohan, H. A., Harshitha, M., Gowda, G. R., & J. N. (2021). IoT-based anti-poaching and fire alarm system for forests. In 2021, IEEE Mysore Sub Section International Conference (MysuruCon) (pp. 711–715). IEEE. DOI: <https://doi.org/10.1109/MysuruCon52639.2021.9641539>
  10. Neetu, D., Sreedhar, R., Sreekumar, R., T., & P. J. (2024). Forest intrusion detection and anti-poaching alarm system. In 2024 International Conference on Smart Systems for Electrical, Electronics, Communication and Computer Engineering (ICSSECC) (pp. 366–368). IEEE. DOI: <https://doi.org/10.1109/ICSSECC61126.2024.10649453>
  11. Rana, A. K., & Kumar, N. (2023). Current wildlife crime (Indian scenario): Major challenges and prevention approaches. Biodiversity and Conservation, 32(5), 1473–1491. DOI: <https://doi.org/10.1007/s10531-023-02577-z>
  12. COCO Dataset. [Online]. Available: <https://cocodataset.org/>
  13. Kaggle Gunshot Audio Dataset. [Online]. Available: <https://www.kaggle.com/datasets>
  14. RunPod Cloud GPU Service. [Online]. Available: <https://www.runpod.io>
  15. OpenCV Library. [Online]. Available: <https://opencv.org/>
  16. PyTorch: An open-source machine learning framework. [Online]. Available: <https://pytorch.org/>
  17. Librosa: Audio and Music Signal Processing in Python. [Online]. Available: <https://librosa.org/>
  18. Pushbullet Python Library. [Online]. Available: <https://github.com/randomchars/pushbullet.py>
  19. Arduino Documentation. [Online]. Available: <https://www.arduino.cc/>

## AUTHOR'S PROFILE



**Vivaan Hooda** is an undergraduate student pursuing a Bachelor of Engineering in Information Science at RV College of Engineering, Bengaluru. His academic interests include systems programming, artificial intelligence and machine learning, blockchain technologies, and web development. He is focused on

building practical, efficient, and scalable computing solutions and actively engages in technical projects and collaborative learning.



**Samarjeet Bhonsle** is an undergraduate student pursuing a Bachelor of Engineering in Information Science and Engineering at R.V. College of Engineering, Bengaluru. His interests include artificial intelligence, machine learning, web development, and intelligent systems. He has been involved in projects focused on

AI-driven applications and data-centric problem-solving, and aims to build practical and scalable software solutions with real-world relevance.



**Dr. Narendra Kumar S** is an Assistant Professor at RV College of Engineering (RVCE), Bengaluru, India. His research interests include fermentation technology and its engineering applications, with a particular focus on fermentation processes relevant to the biotechnology, pharmaceutical, and food industries. His scholarly work demonstrates expertise in the analysis and development of fermentation-based technologies. In addition to his research, he actively mentors undergraduate engineering students in bioprocess and chemical engineering.

**Disclaimer/Publisher's Note:** The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of the Blue Eyes Intelligence Engineering and Sciences Publication (BEIESP)/ journal and/or the editor(s). The Blue Eyes Intelligence Engineering and Sciences Publication (BEIESP) and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.