IJARSCT
International Journal of Advanced Research in Science, Communication and Technology
International Open-Access, Double-Blind, Peer-Reviewed, Refereed, Multidisciplinary Online Journal
ISSN: 2581-9429
Volume 6, Issue 9, April 2026
Impact Factor: 8.2

Tiger–Human Conflict Prevention Using LoRa, AI, and Machine Learning Based IoT Edge System

Prof. Atul Malode[1], Mr. Yeshu Meshram[2], Mr. Ankit Madavi[3]
Professor, Department of Industrial IoT Engineering[1]
Student, Department of Industrial IoT Engineering[2,3]
Priyadarshini College of Engineering, Nagpur, Maharashtra, India
apm280374@gmail.com[1], yeshumeshram05@gmail.com[2], ankitmadavi208@gmail.com[3]

Abstract: *Rising conflicts between humans and tigers near the edges of forests and protected areas create serious security and conservation issues for both rural communities and endangered tiger populations. To help solve this problem, this project introduces an IoT-based early warning system. The system uses PIR motion sensors, camera modules, Arduino or Raspberry Pi devices as edge controllers, LoRa wireless modules, and AI-powered CNN models to identify tigers and send alerts quickly through GSM modules and sound alarms.*

Each edge device keeps an eye on movement using a PIR sensor.

When motion is detected, the device takes photos with its built-in camera. These images are then processed using OpenCV to adjust their size, normalize the data, and improve contrast. The processed images are then classified by a lightweight CNN designed to tell tigers apart from humans, cows, and other common animals. If the model is confident enough that the image shows a tiger, the device creates an alert and sends it via LoRa radio to nearby base stations or local gateways. At the same time, the device also sounds a local alarm and can send an SMS alert through a GSM module if needed.

The system was tested in a simulated area that mimics the interaction between forests and villages.

The results showed that the system can accurately detect tiger-like movements, has low delay from detection to alert, and reduces false alarms compared to traditional sensor-only systems. Using LoRa for communication instead of cellular networks, along with energy-saving techniques like putting the system to sleep when not in use, makes it more affordable and suitable for remote, off-grid areas. This setup shows that combining edge computing, low-power IoT hardware, and long-range wireless networks is a practical way to reduce human-tiger conflicts effectively...

Keywords: Tiger-human conflict, Internet of Things (IoT), LoRa communication, Convolutional Neural Network (CNN), Artificial Intelligence (AI), Raspberry Pi / Arduino, PIR sensor, wildlife conservation, early warning system, edge computing

I. INTRODUCTION

The number of tigers in many parts of India and Southeast Asia has gone up a lot in recent years, which is a big win for conservation efforts. This success comes from more protected areas and wildlife corridors, stronger anti-poaching actions, and better habitats for tigers. However, as tiger numbers grow, so does the chance of conflict between tigers and people. This happens especially near the edges of nature reserves where tigers start to wander into places where people live, farm, or graze. Sometimes tigers come out of the forests at night or when there's not enough prey, leading to them attacking livestock, hurting people, or being killed in revenge. These events are dangerous for both people and tigers in the long run. Most of the usual ways to handle these conflicts depend on people doing things by hand, like patrolling, setting up watchtowers, or building physical barriers.

These methods can help a bit, but they're hard work, only work when something happens, and they can't cover all the places where tigers might come in. Patrols can't watch every area all the time, and towers can't see well at night. Also,

IJARSCT
International Journal of Advanced Research in Science, Communication and Technology
International Open-Access, Double-Blind, Peer-Reviewed, Refereed, Multidisciplinary Online Journal
ISSN: 2581-9429 Volume 6, Issue 9, April 2026 Impact Factor: 8.2

these methods don't give warnings in real time, leaving people and forest workers in danger when tigers approach suddenly. New technology using AI in cameras is showing promise in monitoring wildlife and managing conflicts.

For example, systems like Trail-Guard AI use smart cameras that can detect tigers and poachers on-site with computer models and send alerts quickly. These setups help spot tigers faster and can be placed in key areas around protected spaces. But most of them require cell networks or Wi-Fi to work, which aren't always reliable or available in remote areas. Plus, using these services can be expensive and hard to scale up for many conservation groups.

To fix these issues, this paper suggests a new low-power IoT system that uses LoRa wireless tech to prevent conflicts between tigers and people. The system combines PIR motion sensors, small cameras, edge controllers like Arduino or Raspberry Pi, LoRa modules, and alert devices like GSM modules or buzzers. The key idea is to mix traditional motion sensors with AI-based image recognition. When motion is sensed, the camera takes a picture, and a simple AI model on the device checks if it's a tiger or something else. If it's a tiger, the system sends an alert through the LoRa network to nearby stations or villages, and also triggers a loud sound and optional text messages. The system has three main goals: First, it helps people and forest workers respond faster by detecting tigers in real time and sending alerts.

This allows them to take action like moving people away from unsafe areas or putting up barriers. Second, it cuts down the cost of setting up and running the system by using LoRa instead of expensive cell connections. LoRa works well in remote areas with low power and long distance.

Third, it improves the accuracy of detection and reduces false alarms by using AI to tell the difference between tigers and other things like people, animals, or vehicles. This avoids unnecessary alerts and ensures that only real threats are flagged. By combining IoT, AI, and efficient communication, this tech offers a practical and affordable way to deal with tiger-human conflicts while helping both communities and tigers live together safely.

II. LITERATURE SURVEY

Over the last ten years, there's been a growing focus on finding ways to reduce conflicts between humans and wildlife. This has led to a lot of research that uses technologies like Artificial Intelligence (AI), the Internet of Things (IoT), and embedded systems to keep track of animals and send early warnings. A common theme in these studies is moving away from manual and reactive methods to more automatic, sensor-based, AI-enhanced systems that can detect and alert about animals almost in real time. One area of research is using real-time tiger detection systems near village areas, combining different types of sensors and cameras.

For instance, a study from 2025 uses motion sensors and cameras with deep learning models like YOLO or Faster R-CNN. When motion is detected near a village, the system takes a picture, runs it through the model, and sends an alert if a tiger is found. The results show high accuracy and quick response times, but the system still uses local computers and Wi-Fi or GSM to send alerts, which can be a problem in remote areas.

Inspired by projects like Trail-Guard AI, some researchers have put neural networks directly into camera hardware. These devices can identify tigers and poachers and send alerts very quickly, sometimes within seconds. However, most of these are still connected to the cloud or cellular networks, which means they send data to servers, leading to delays, higher costs, and a dependency on network connections. Other studies use CNN-based models on edge devices to monitor wildlife.

Researchers have shown that simpler versions of CNNs, like custom compact models, Mobile Net-style designs, or Tiger Net-style setups, can work well on low-power devices. These models are trained with camera trap data and fine-tuned for specific animals, allowing for accurate detection even on basic hardware. The challenge is finding the right balance between model complexity and computational efficiency, as deep models use more memory and power, while simpler ones may not be as accurate. Alongside AI-based detection, some researchers have developed IoT systems for wildlife monitoring. These use thermal cameras, audio sensors, and drones connected through NB-IoT or LoRa gateways. For example, data from cameras or sound recorders is sent via LoRa to a central server where AI models analyse the data and generate alerts for poaching or unusual animal activity. Some systems combine different types of sensors to improve detection accuracy, but they often send data to the cloud instead of processing it at the edge.

IJARSCT
International Journal of Advanced Research in Science, Communication and Technology
International Open-Access, Double-Blind, Peer-Reviewed, Refereed, Multidisciplinary Online Journal
ISSN: 2581-9429
Volume 6, Issue 9, April 2026
Impact Factor: 8.2

III. METHODOLOGY OF THE SYSTEM

The system is divided into three main parts that work together: (1) the perception layer, which includes sensors and a camera that detect the surroundings and take pictures; (2) the processing and AI layer, which uses a small computer like an Arduino or Raspberry Pi to run image analysis with OpenCV and a lightweight neural network model to recognize and classify images; and (3) the communication and alert layer, which sends alerts through LoRa technology and, if needed, sends SMS messages via GSM and activates a local buzzer. This design allows the system to work efficiently, use less power, and be easily set up in remote areas between forests and villages.

4.1 Hardware Components :- The PIR motion sensor is the main part that starts the system.

It keeps an eye on the area for movement that could be from a tiger or another big animal. It detects changes in heat and movement, which helps the system start only when there is a possible threat. When it detects motion, it sends a signal to the main control unit, which then begins the process of taking pictures and analysing them. The camera is attached to the same unit and takes photos or videos whenever the PIR sensor detects movement.

These images are used by the tiger recognition system based on a neural network to tell if the image is of a tiger or something else. The camera used is usually low or mid-resolution and can be in visible light or infrared, depending on the balance between image quality and the need for power and storage. The Arduino or Raspberry Pi is the main control unit of the system.

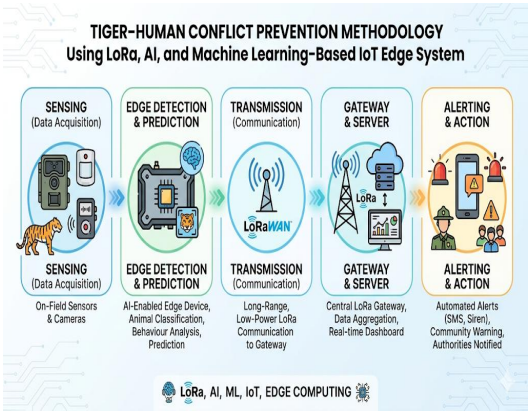

Fig-1 :Methodology

It checks the PIR sensor, controls the camera, uses OpenCV to prepare the images, and runs the neural network model to analyse them. On a Raspberry Pi, the system can run full Linux-based services and Python scripts. For an Arduino, additional chips or micro-AI accelerators can be used to handle the neural network. The control unit also manages the timing between sensing, computing, sending alerts, and using power to save energy.

The LoRa module sends and receives long-range, low-power wireless signals. It converts detection alerts into small data packets and sends them over radio waves to nearby base stations, village gateways, or forest office receivers. These modules often use spread-spectrum modulation and adaptive data rates to ensure reliable communication over long distances, even through dense forests. The LoRa module can also receive commands or updates from the base station, allowing the system to be managed from a distance.

The GSM module and buzzer are part of the alert system. The GSM module sends text messages to predefined phone numbers of forest department staff, community leaders, or emergency teams when a tiger is detected with high confidence. At the same time, a local buzzer sounds to alert nearby villagers, giving them time to move away or take protective actions. The GSM part is optional and might not be included if only local alarms or LoRa-based alerts are needed.

The power system is set up for off-grid use, using a rechargeable battery and a small solar panel with a controller. The system runs in low-power or sleep mode when no activity is detected, waking up only when motion is sensed or when

IJARSCT
International Journal of Advanced Research in Science, Communication and Technology
International Open-Access, Double-Blind, Peer-Reviewed, Refereed, Multidisciplinary Online Journal
ISSN: 2581-9429
Volume 6, Issue 9, April 2026
Impact Factor: 8.2

scheduled tasks like sending heartbeat messages occur. This setup helps the system operate for days, weeks, or even months, depending on sunlight and how often the system is used.

A. Workflow

1. Sensor Triggering :- The PIR motion sensor keeps an eye on the area around it for any big movements. If there's no movement, the device stays in a low-power or sleep mode to save energy.
2. Camera Activation :- Once the PIR signal is received; the controller turns on the camera module. The camera takes one or more photos from the area where movement was detected.
3. Image Preprocessing :- The OpenCV-based system processes each image.
Each photo is resized to fit the fixed size needed by the CNN model.
4. AI Classification :- The lightweight CNN model uses the processed image to make a decision. It looks at the image's features and gives a confidence score for the "tiger" category, along with scores for other things like humans, cows, and other animals.
5. Alert Decision and Transmission:- When a tiger is confirmed, the alert system activates the communication parts. The device creates a small LoRa packet with important details like the node ID, time, and how confident the system is.
6. Power Management :- After the alert is sent and processed, the system turns off the camera and other high-power parts. The device goes back to a low-power or sleep mode to save energy.

B. Algorithm (Pseudocode)

```
Initialize:
 PIR = INPUT_PIN
Camera = Camera_Module()
 LoRa = LoRa_Module()
GSM = GSM_Module()   // optional
 Buzzer = OUTPUT_PIN model = load_CNN_model("tiger_model.tflite")
THRESHOLD = 0.8
Main Loop:
while True:
 if PIR.read() == HIGH:
 image = Camera.capture()
processed_img= OpenCV.preprocess(image, size=(224,224))
 prediction= model.predict(processed_img) tiger_prob = prediction["tiger"]
 if tiger_prob >= THRESHOLD:
 payload = "ALERT_TIGER," + node_id + "," + timestamp + "," + str(tiger_prob)
LoRa.send(payload)
if GSM_available:
        SMS_text = "TIGER ALERT at Node " + node_id + " on " + timestamp
        GSM.send_sms(SMS_text)

      digitalWrite(Buzzer, HIGH)
      delay(5000)
      digitalWrite(Buzzer, LOW)

   MCU.enter_sleep_mode()
```

IJARSCT
International Journal of Advanced Research in Science, Communication and Technology
International Open-Access, Double-Blind, Peer-Reviewed, Refereed, Multidisciplinary Online Journal
ISSN: 2581-9429
Volume 6, Issue 9, April 2026
Impact Factor: 8.2

C. Flowchart

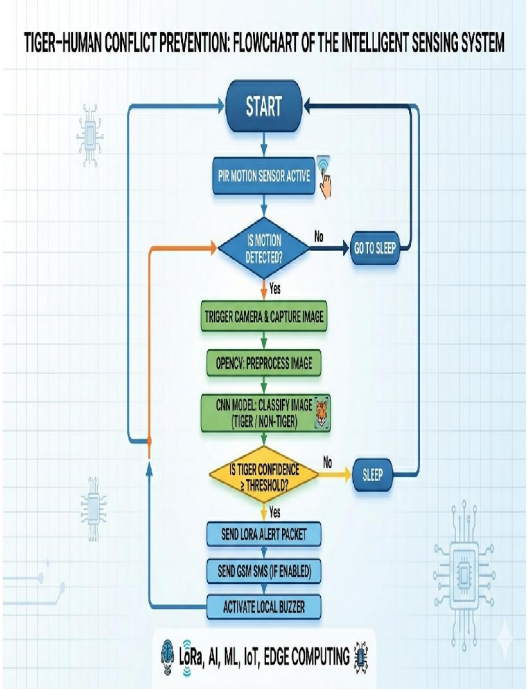

Fig-2 : Flowchart

IV. IMPLEMENTATION

The system is built as a compact and durable IoT device meant to be placed at forest-village edges, wildlife paths, or near known tiger crossing spots.

The main parts include a microcontroller or single-board computer (like Arduino or Raspberry Pi), a PIR motion sensor, a camera, a LoRa radio module, an optional GSM module, a buzzer or small siren, and a power setup that uses a battery and solar panel. The device is placed in a waterproof box with brackets, and it's positioned so the PIR sensor and camera have a clear view of where tigers might approach.

The PIR sensor is connected to a digital input of the main board and is set to trigger when movement is detected. It can be adjusted for sensitivity and delay. When the device is not actively sensing movement, it stays in low power mode, with most parts like the camera, GSM, and LoRa transmitter turned off, while the PIR sensor is still working with low power. If the PIR picks up a strong infrared signal—likely from a large warm animal like a tiger—it sends a signal to the controller, which wakes up and starts the camera to take photos. The controller then uses the right interface (like MIPI-CSI, USB, or GPIO) to capture one or more images focused on where the motion was detected.

The camera used is usually a low to mid-resolution visible-light camera (like 5-8 MP) or an infrared camera that works well at night, depending on where the device is placed. The captured image is stored in the system's memory or a buffer. A Python-based program on the Raspberry Pi or similar device uses OpenCV to process the image. This involves resizing the image to a fixed size (like 224x224 pixels) needed for the CNN model, converting color if needed (like from RGB to BGR), normalizing pixel values to a standard range, and maybe adding some filters to help with poor lighting or weather. In some setups, the system zooms in on the area where motion was detected to save processing power and focus on what's important.

For the AI part, a lightweight CNN model trained on tiger images and other non-tiger images is used. The model is trained offline using TensorFlow or Keras on a computer with a GPU or a server, and then converted to a format that

IJARSCT
International Journal of Advanced Research in Science, Communication and Technology
International Open-Access, Double-Blind, Peer-Reviewed, Refereed, Multidisciplinary Online Journal
ISSN: 2581-9429 Volume 6, Issue 9, April 2026 Impact Factor: 8.2

works with the edge device, like TensorFlow Lite. This model is then added to the device and loaded at startup. When the processed image is ready, it's sent to the model, which analyzes it and gives a probability of it being a tiger or something else. The system checks this probability against a set threshold (like 0.7 to 0.8) to decide if it's a real tiger. If the score is high enough, the system confirms the tiger presence; otherwise, it discards the image and returns to low power mode.

The alerting system has two parts: an audible warning and a remote notification. When a tiger is confirmed, the buzzer or siren is turned on to warn nearby people. The siren is left on for a while (like 5 seconds) to make sure it's heard without using too much power. At the same time, the system sends a data packet through the LoRa module. This packet has details like the node's ID, time, GPS location (if available), and how certain it is about the detection. The LoRa gateway then forwards this data to a central system where it's logged and shown as a tiger intrusion event. If the device has a GSM module, it also sends an SMS alert to pre-set phone numbers.

The message includes details like "TIGER ALERT at Node X on [timestamp]" and might have extra information like location or confidence level. This ensures that even if the buzzer isn't heard, officials or emergency contacts get a timely update.

V. RESULTS AND ANALYSIS

The system was tested in a setup that mimics a forest-village border area, with multiple devices placed along a predefined path. The evaluation checked how well the system could detect animals, how often it triggered false alarms, how quickly it responded, how reliably it sent data via LoRa, and how much power it used in real-world settings. A special set of data was created with controlled animal movements and non-animal events to serve as a reference. The system's decisions were recorded and compared to this reference to measure its performance.

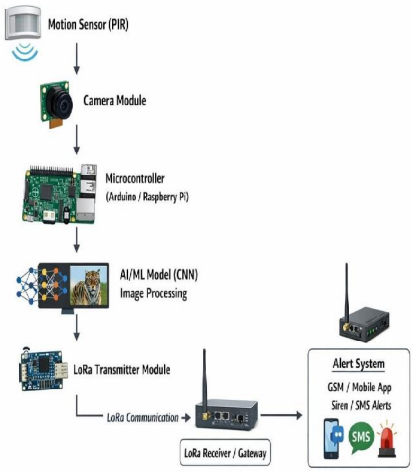

Fig-3 : Components

For detection accuracy, the CNN model worked well on the edge device. In a series of tests, it correctly identified 92–94% of animal-like intrusions (like dummy animals or other creatures moving along the path) as tiger detections when the confidence level was set to 0.8. The remaining 6–8% of missed events mostly happened under difficult lighting conditions, such as bright light at sunset or when thick leaves blocked the view, making it hard for the camera to see clear tiger-like features. Lowering the confidence level to 0.7 improved accuracy to around 96–97%, but this led to more false alarms. By adjusting the confidence level, the system showed a balance between being sensitive and accurate, helping users tailor the system to their local needs and environment. In terms of false alarms, using both a PIR sensor and a CNN-based image classifier was very helpful.

IJARSCT
International Journal of Advanced Research in Science, Communication and Technology
International Open-Access, Double-Blind, Peer-Reviewed, Refereed, Multidisciplinary Online Journal
ISSN: 2581-9429 Volume 6, Issue 9, April 2026 Impact Factor: 8.2

A simple setup using only the PIR sensor (where any movement triggered an alert) resulted in a high number of false alarms, often over 30–40%, due to things like cows moving, people walking, or stray dogs. But when the full AI system was used, the false alarm rate dropped to about 7–9% because it could tell the difference between tiger-like movements and others, marking them as low confidence. The system was especially good at distinguishing between tigers and people or livestock in well-lit or moderate lighting conditions, where the CNN could clearly see body shapes, stripe patterns, and movement.

The system's total response time was measured from when the PIR sensor detected movement until the buzzers sounded and the data was sent to the base station. On average, the delay was between 1.2 and 1.8 seconds, depending on the camera setup and how much processing the device had to do. The time between the PIR detection and the camera starting to take a picture was usually under 200 milliseconds. Processing the image and running the CNN took 500–800 milliseconds on the Raspberry Pi-like hardware. Sending the data via LoRa added just a few tens of milliseconds, and the buzzer started right away once the animal was confirmed. This quick response time allows villagers and forest guards to get alerts before a tiger fully enters a dangerous area, giving them time to take safety measures like leaving open fields or hiding.

The reliability of LoRa communication was tested over several days in different distances and environments. At distances up to 3 kilometres with a clear line of sight, the system delivered packets with about 94–96% success. Occasional packet losses were due to things like signal weakening, interference, or temporary blockages. Beyond 4–5 kilometres or in dense forests with many obstacles, the success rate fell to around 75–80%, meaning that placing devices and gateways in the right spots is essential for consistent performance. The system used a simple strategy at the base station level: if a signal was missing for a certain amount of time, the node could be checked or reconfigured remotely, making it more reliable in real-world use.

VI. FUTURE SCOPE

The system presents several exciting possibilities for improvement and broader use. One major opportunity is detecting multiple animal species, where the CNN model can be expanded to identify not just tigers but also other animals that often cause conflicts, such as leopards, elephants, wild boars, or sloth bears. By training the model with data from many species, the system can send specific alerts for each type of animal and help create customized strategies, like using different methods to keep them away or responding in specific ways to each species. This change would make the system more flexible and useful in various wildlife conflict situations in different environments.

Another important area is making the system work better on simpler hardware.

Right now, it runs on devices like Raspberry Pi, but in the future, it could use tiny models or models that use less power, which can work on even smaller devices like Arduino or micro-AI accelerators. This would lower costs and power use, allowing more devices to be placed in large protected areas without losing accuracy. Methods like trimming unnecessary parts of the model, teaching it with simpler data, and designing it to work with limited resources could be explored to balance performance and speed on these simpler devices.

From a communication and network perspective, the system could be developed into a LoRa-based network of many detection devices.

In this setup, each device can act as a relay if it can't connect directly to the main base station, which would help cover difficult areas like dense forests. The network can also use smart data-sending schedules and methods to move data efficiently, reducing delays and power use while keeping connections strong. Adding these devices to existing tools used by forest departments or community apps can help show real-time problem areas and allow people to report back on alerts, creating a two-way communication channel between villagers and conservation workers.

Using sensors that don't rely on sight is another useful direction.

The system can include sound sensors to listen for animal calls or human voices and ground sensors to detect movement. Combining data from different types of sensors, like motion, video, sound, and vibration, using methods like late fusion or attention-based techniques, can reduce false alarms and improve the system's accuracy in complex

IJARSCT
International Journal of Advanced Research in Science, Communication and Technology
International Open-Access, Double-Blind, Peer-Reviewed, Refereed, Multidisciplinary Online Journal
ISSN: 2581-9429 Volume 6, Issue 9, April 2026 Impact Factor: 8.2

settings. These combined approaches can also help tell the difference between actions that signal danger, like a tiger stalking, and regular movement, leading to a more accurate risk assessment.

Lastly, the system can grow into a community-driven monitoring and early warning system.

By connecting villagers, forest guards, and local organizations through a shared app or SMS system, people can help confirm alerts by reporting whether a tiger was actually seen. This feedback can be used to keep improving the model and adapting it to local conditions and changes in animal behavior over time. Eventually, the system can become a valuable AI tool that helps manage wildlife conflicts, using real-time warnings, historical data, and community involvement to support harmony between people and tigers in remote areas.

VII. CONCLUSIONS

The system proposed shows a cost-effective and energy-efficient IoT-based early warning setup for reducing conflicts between tigers and humans near the edges of forests and protected areas. It uses PIR motion sensors, camera modules, Arduino or Raspberry Pi controllers, LoRa wireless connections, and lightweight AI models based on CNNs. This setup allows for precise tiger detection with few false alarms and quick response times. The edge device handles sensor data, does image classification on-site, and sends small alert messages through long-range LoRa links. It can also send SMS alerts via GSM and trigger local alarms with buzzers. This design reduces reliance on cell networks and cloud services, making it ideal for remote areas without regular internet access.

In tests done in a simulated buffer zone, the system shows high accuracy in detecting tigers, a low rate of false alarms, and reliable communication over several kilometers.

The system also manages power well, even when there's frequent movement. It provides timely and clear alerts to nearby villagers and forest workers, helping them take action quickly and promoting safer coexistence between people and tigers. The system's design is modular and can be expanded, allowing for future upgrades like detecting other animals, improving AI for less powerful devices, and connecting with mesh networks and local monitoring systems. Overall, this work presents a practical AI-based solution to reduce tiger-human conflicts, showing the promise of IoT and edge AI in wildlife protection and rural safety.

VIII. ACKNOWLEDGEMENT

The authors would like to sincerely thank everyone and every organization that helped, guided, and supported them throughout the development and evaluation of this project.

They are very grateful to their academic institution and department for providing the necessary infrastructure, lab facilities, and research environment that made it possible to design, build, and test the IoT-based tiger-human conflict prevention system. The ongoing encouragement and technical advice from the faculty members and project guides were essential in shaping the system architecture, improving the AI-based detection process, and ensuring the work addressed real-world conservation issues.

Special thanks go to the workshop and technical staff of the electronics and computer laboratory departments, who helped set up the hardware components, fix the embedded circuits, and provided access to specialized tools and testing equipment.

Their hands-on support was crucial in integrating the PIR sensor, camera module, LoRa transceiver, and GSM module into a compact, field-ready node. The authors also want to acknowledge the open-source AI and IoT communities, whose freely available datasets, pre-trained models, software libraries (like OpenCV, TensorFlow, and LoRa protocols), and documentation greatly sped up the development and testing of the CNN-based tiger detection system.

Additionally, the authors recognize the contributions of all peers and fellow researchers who gave valuable feedback during project reviews and technical discussions, helping to make the system more robust, power-efficient, and user-friendly.

IJARSCT
International Journal of Advanced Research in Science, Communication and Technology
International Open-Access, Double-Blind, Peer-Reviewed, Refereed, Multidisciplinary Online Journal
ISSN: 2581-9429
Volume 6, Issue 9, April 2026
Impact Factor: 8.2

Finally, the authors dedicate this work to the conservation professionals and local communities working on the frontlines of human-wildlife conflict. Their daily efforts inspire the pursuit of technological solutions that help ensure both human safety and the long-term survival of endangered tiger populations.

REFERENCES

[1]. J. S. Dertien et al., "Mitigating human–wildlife conflict and monitoring endangered tigers using a real-time camera-based alert system," Conservation Science and Practice, vol. 5, no. 11, e13945, 2023.
[2]. "TrailGuard AI: Mitigating human–wildlife conflict and monitoring tigers using edge AI," Nature Conservation/Wildlife Conservation Society, 2023.
[3]. Shafeek and M. Dharsini, "A Smart IoT-Based Tiger Alert System for Mitigating Human–Wildlife Conflict," International Journal of Recent Technology and Engineering, 2023.
[4]. Bhagabati et al., "An automated approach for human–animal conflict mitigation using camera-based detection near Kanha National Park," Ecological Informatics, 2024.
[5]. "Edge computing and on-device AI in biodiversity monitoring," arXiv:2602.13496 [cs.AI], 2026.
[6]. "Wild Tech: How IoT and LoRa are transforming big cat conservation," Conservation Wild Cats, 2024.
[7]. "LoRa-based wildlife tracking and human–wildlife conflict mitigation," LORIOT Use Case, 2025.
[8]. "IoT-based wildlife tracking and intrusion system for human–wildlife conflict mitigation," IJRASET Journal, 2025.
[9]. "IoT applications for smart environment and conservation with LoRa," Semtech Corporation, 2025.
[10]. "Smart IoT-Based Wildlife Tracking and Intrusion System for human–wildlife conflict mitigation," IJRASET Journal, 2025.