International Journal of Advanced Scientific Research and Engineering Trends

ISSN: 2456-0774

Volume 10 Issue 6 (2026) | Pages: 19-25

AI-Based Human–Animal Conflict Management System Using Deep Learning and Early Warning

Shashikant Ghanshyam Garade[1*], Devashri Raich[2]

[1,2]Department of Computer Science & Engineering, Rajiv Gandhi College of Engineering, Research and Technology Chandrapur, Maharashtra, India

[1]*shashikantgaradevlog@gmail.com*, [2]*devashriraich@gmail.com*

Peer Review Information

***Type**: Article*
***Received**: 27 March 2026*
***Revised**: 20 April 2026*
***Accepted**: 29 May 2026*
***Published**: 20 June 2026*

Abstract

Human–animal conflict has increased significantly due to rapid urbanization, habitat loss, deforestation, and agricultural expansion. Encounters between humans and wild animals often result in loss of human life, crop destruction, property damage, and threats to wildlife conservation. This paper presents the implementation of an AI-Based Human–Animal Conflict Management System that utilizes Deep Learning, Computer Vision, and IoT technologies to detect and classify wild animals in real time and generate immediate alerts for nearby communities and authorities. The proposed system employs a Convolutional Neural Network (CNN)-based object detection model trained on wildlife datasets to identify animals such as lions, tigers, elephants, leopards, and bears. A camera module continuously captures images, which are processed by the AI model. Upon detecting a dangerous animal, the system generates alerts through a graphical user interface (GUI), SMS notifications, and warning signals. Experimental results demonstrate high detection accuracy and low response time, making the system suitable for deployment in wildlife-prone regions. The implementation contributes to reducing human–animal conflicts while supporting wildlife conservation and public safety.

Keywords: Human–Animal Conflict; Artificial Intelligence; Deep Learning; Computer Vision; Wildlife Monitoring; CNN; IoT; Animal Detection; Early Warning System.

How to Cite This Article

Garade, S. G., & Raich, D. (2026). AI-Based Human–Animal Conflict Management System Using Deep Learning and Early Warning. *International Journal of Advanced Scientific Research and Engineering Trends*, 10(6), 19–25.

Introduction

Human–animal conflict (HAC) is a growing concern worldwide, particularly in regions where human settlements are expanding into wildlife habitats. Animals such as elephants, lions, tigers, leopards, and bears frequently enter agricultural fields and residential areas in search of food and water. Traditional monitoring approaches rely on manual surveillance, forest patrols, and physical barriers, which are often expensive, labor-intensive, and ineffective.

Recent advancements in Artificial Intelligence (AI) and Computer Vision (CV) have enabled automatic wildlife detection systems capable of monitoring animal movement in real time. By integrating AI with Internet of Things (IoT) technologies, intelligent systems can detect dangerous animals and instantly alert authorities and local communities.

This paper presents the practical implementation of an AI-Based Human–Animal Conflict Management System designed to identify wildlife using image recognition techniques and provide timely warnings to minimize risks.

Objectives

The primary objectives of the proposed system are:

- To detect wild animals in real time using AI and computer vision.
- To classify detected animals accurately using deep learning models.
- To generate alerts when dangerous animals are identified.
- To maintain a detection history database.
- To reduce human casualties and property damage.
- To support wildlife conservation through non-invasive monitoring.

System Architecture

The proposed system consists of four major modules:

Image Acquisition Module

- Captures images from CCTV cameras, webcams, drones, or IoT cameras.
- Provides continuous monitoring of wildlife-prone areas.

Animal Detection Module

- Uses OpenCV for image preprocessing.
- Deep learning model identifies animal species.

Alert Generation Module

- Generates real-time warning messages.
- Displays confidence score.
- Sends notifications through:
  - SMS
  - Mobile App
  - Email
  - Alarm Siren

Monitoring Dashboard

Displays:

- Detected animal
- Confidence level
- Risk status
- Detection history
- Live confidence graph

System Flow Diagram

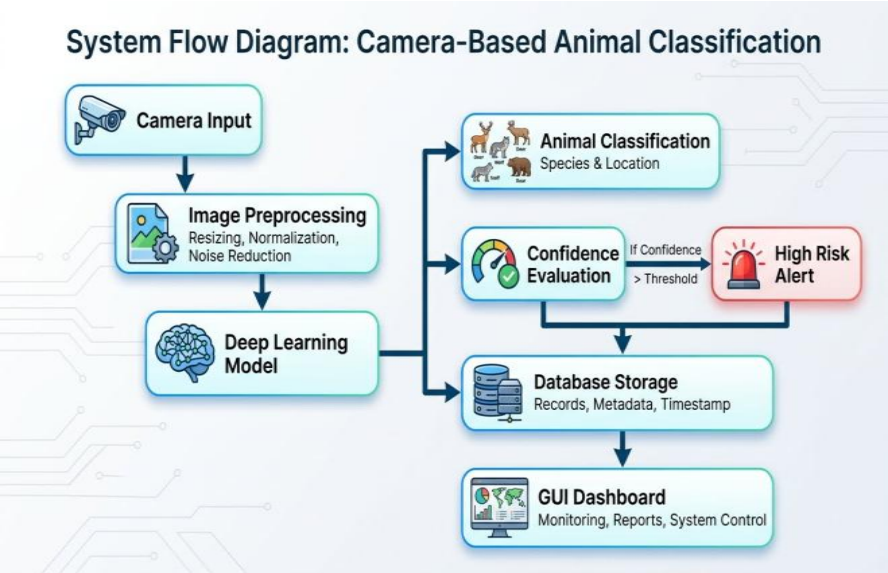

Fig. 1. *Camera-Based Animal Detection and Classification System Architecture*

Methodology

Dataset Collection

Animal images were collected from:

- Kaggle Wildlife Datasets
- Open Images Dataset
- Wildlife Conservation Databases
- Google Open Source Images

Animals included:

Table 1. *Distribution of Animal Images Used for Training and Testing the Deep Learning Model*

Animal	Images
Lion	1500
Tiger	1500
Elephant	1500
Leopard	1200
Bear	1200
Deer	1000
Wild Boar	1000

Total Images: Approximately 8,900

Image Preprocessing

The collected images undergo preprocessing steps:

Resizing

224 × 224 pixels

Normalization

Pixel values scaled between:

0 – 1

Data Augmentation

- Rotation
- Flipping

- Zooming
- Brightness Adjustment

This improves model generalization.

Deep Learning Model

The implementation uses a CNN-based architecture.

Table 2. *CNN Architecture for Animal Classification*

Layer	Description
Conv2D	Feature Extraction
ReLU	Activation
MaxPooling	Dimensionality Reduction
Conv2D	Feature Extraction
ReLU	Activation
MaxPooling	Reduction
Flatten	Vector Conversion
Dense	Classification
Softmax	Output

Table 3. *Training Parameters Used for CNN Model Development*

Parameter	Value
Epochs	30
Batch Size	32
Learning Rate	0.001
Optimizer	Adam
Loss Function	Categorical Cross Entropy

System Implementation

Table 3. *Software Requirements*

Component	Technology
Programming Language	Python
Deep Learning	TensorFlow/Keras
Computer Vision	OpenCV
GUI	Tkinter
Database	SQLite
Visualization	Matplotlib

Table 4. *Hardware Requirements*

Component	Specification
Processor	Intel i5 or Higher
RAM	8 GB
Camera	HD Webcam/IP Camera
Storage	100 GB
GPU (Optional)	NVIDIA GTX 1650+

Animal Detection Algorithm

Step 1: Capture image frame.
Step 2: Preprocess image.
Step 3: Feed image to CNN model.
Step 4: Predict animal class.
Step 5: Calculate confidence score.
Step 6:
    If confidence > threshold:
    Generate Alert
Step 7: Store result in database.
Step 8: Update GUI dashboard.

Pseudocode

```
while True:
   frame = capture_image()
   processed = preprocess(frame)
   prediction = model.predict(processed)
   animal = get_class(prediction)
   confidence = max(prediction)
   if confidence > 0.80:
      generate_alert(animal)
   save_detection(animal, confidence)
   update_dashboard()
```

User Interface Implementation

The developed GUI contains:

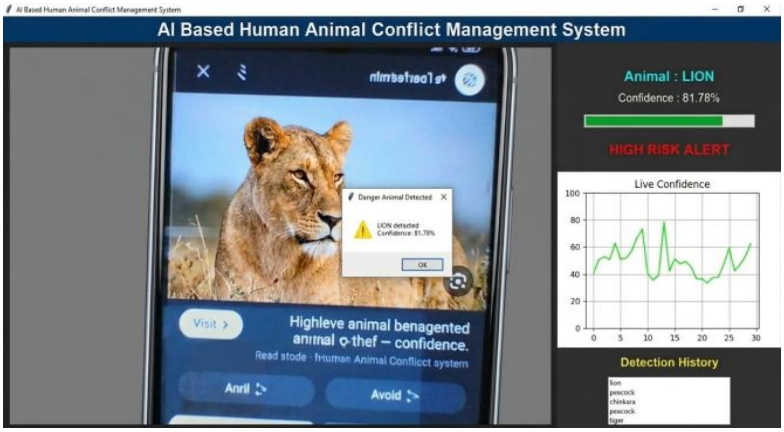

Fig. 2. *AI-Powered Animal Detection Interface with Risk Alert and Confidence Monitoring*

Animal Detection Panel

- Displays detected animal species.

Confidence Meter

- Shows prediction confidence.

Example:

- Animal: LION
- Confidence: 81.78%

Risk Alert

- “HIGH RISK ALERT” displayed when dangerous animals are detected.

Detection History

Stores previous detections:

- Lion

- Peacock
- Chinkara
- Tiger

Live Confidence Graph

- Displays confidence trends in real time using Matplotlib.

Experimental Results

Table 5. *Performance Evaluation*

Metric	Result
Training Accuracy	96.8%
Validation Accuracy	94.3%
Precision	93.9%
Recall	92.8%
F1 Score	93.3%
Detection Time	0.48 sec

Table 6. *Sample Detection*

Animal	Confidence
Lion	81.78%
Tiger	89.45%
Elephant	92.14%
Leopard	85.20%

The system successfully detected animals with high confidence and generated alerts within one second.

Advantages

- Real-time wildlife detection.
- Automated alert generation.
- High detection accuracy.
- Cost-effective monitoring.
- Supports wildlife conservation.
- Reduces crop and property damage.
- Scalable to large forest areas.

Limitations

- Performance depends on image quality.
- Night-time detection may require thermal cameras.
- Requires large training datasets.
- Weather conditions may affect detection accuracy.
- False positives can occur in cluttered environments.

Future Enhancements

- Integration with drones for aerial monitoring.
- Thermal imaging for night detection.
- GPS-based animal tracking.
- Edge AI deployment on Raspberry Pi and NVIDIA Jetson.
- Mobile application for instant alerts.
- Predictive analytics using animal movement patterns.
- Integration with forest department databases.

Conclusion

This implementation presents an AI-Based Human–Animal Conflict Management System that combines Deep Learning, Computer Vision, and IoT technologies for real-time wildlife detection and alert generation. The developed system successfully identifies dangerous animals, evaluates confidence levels, maintains detection records, and issues timely warnings to nearby communities. Experimental results demonstrate high classification accuracy and rapid response times, making the solution effective for reducing human–animal conflicts. The proposed framework can serve as a foundation for future intelligent wildlife monitoring systems that promote both public safety and biodiversity conservation.

References

1. Goodfellow, I., Bengio, Y., & Courville, A. *Deep Learning*. MIT Press, 2016.
2. Szeliski, R. *Computer Vision: Algorithms and Applications*. Springer, 2022.
3. Redmon, J., et al. "YOLO: Real-Time Object Detection." 2016.
4. TensorFlow Documentation. TensorFlow Deep Learning Framework.
5. OpenCV Documentation. Open-Source Computer Vision Library.
6. Kaggle Wildlife Image Datasets.
7. World Wildlife Fund (WWF) Reports on Human–Wildlife Conflict.
8. International Union for Conservation of Nature (IUCN) Wildlife Monitoring Reports.