

Real-Time Multi-Object Detection and Tracking Using Deep Learning

1st Dr. Nilesh T. Gole

Dept. Of Computer Science & Engineering
Vidarbha Institute of Technology, Nagpur
Email: n.gole@vitnagpur.edu.in

2nd Ms. Shital Raut

Master of Technology
Dept. Of Computer Science & Engineering
Vidarbha Institute of Technology, Nagpur

Abstract Human-wildlife encounters have become increasingly frequent in forest-fringe communities, agricultural regions, and rural settlements, creating significant risks for both humans and animals. This paper presents an AI-driven wildlife surveillance and threat detection system designed for the real-time identification of potentially dangerous animals, particularly tigers, leopards, and lions. The proposed system employs the YOLOv10 deep learning architecture to perform rapid object detection from surveillance video streams. In addition to animal identification, the system incorporates object tracking, automatic snapshot generation, timestamp recording, SQLite-based event logging, and email-based alert notification within a unified monitoring framework. A customized dataset obtained from the Roboflow Universe, containing approximately 2,000 tiger images, 700 lion images, and more than 500 leopard images, was utilized for model development. The YOLOv10s model was trained using the prepared dataset and subsequently integrated into a Python-based surveillance application. The resulting system provides automated wildlife detection and event monitoring, thereby reducing dependence on continuous manual observation. The proposed approach demonstrates the potential of deep-learning-based computer vision for real-time wildlife surveillance in forest-border areas, agricultural zones, and other remote locations.

Keywords— YOLOv10, Wildlife Monitoring, Deep Learning, Object Detection, Computer Vision, Animal Detection, Real-Time Surveillance

I. INTRODUCTION

The increasing encroachment of human activities into natural habitats has contributed to a rise in wildlife intrusion into human-populated regions. Deforestation, habitat fragmentation, agricultural expansion, and changes in land utilization have reduced the separation between wildlife habitats and residential or agricultural areas. As a result, encounters involving potentially dangerous animals such as tigers, leopards, and lions can pose considerable challenges to local communities.

Conventional wildlife monitoring practices frequently depend on CCTV cameras, forest personnel, or continuous human observation. Although these approaches can provide useful surveillance information, they require substantial human involvement and may not provide an immediate response when an animal enters a monitored area. Continuous manual monitoring can also become difficult when surveillance needs to operate for extended periods.

Recent developments in Artificial Intelligence (AI), Deep Learning, and Computer Vision have provided new opportunities for automated wildlife monitoring. Object-detection algorithms can analyze video frames and identify specific animal categories without requiring an operator to inspect every frame manually.

This research proposes an AI-based wildlife surveillance system utilizing YOLOv10 for real-time detection of tiger, leopard, and lion classes. Beyond basic object detection, the proposed platform combines detection with object tracking, event snapshots, timestamp management, database storage, and automated email notifications. The objective is to provide a comprehensive surveillance mechanism capable of identifying animal presence and recording relevant detection events automatically.

LITERATURE REVIEW

Early computer-vision-based wildlife recognition systems commonly relied on conventional image-processing and machine-learning approaches. Techniques such as Histogram of Oriented Gradients (HOG) combined with Support Vector Machines (SVM) were used to extract visual features and classify objects. However, such approaches generally depend on manually designed features and may encounter difficulties under variations in illumination, animal pose, background complexity, and camera conditions.

The development of deep convolutional neural networks significantly improved the capabilities of automated object recognition. Frameworks such as Faster R-CNN introduced region-based object detection, while Single Shot Detector (SSD) provided a more efficient single-stage detection

mechanism. The YOLO family further emphasized high-speed object detection by performing localization and classification within a unified architecture.

Despite improvements in detection accuracy and processing speed, wildlife surveillance applications frequently require functionality beyond simple object classification. A practical monitoring platform may need to record detection events, retain visual evidence, track movement, and immediately inform responsible personnel.

The proposed work therefore combines a YOLOv10-based detection model with supporting surveillance functionalities. The system is intended not merely to recognize an animal in an image but to provide an integrated mechanism for detection, tracking, evidence generation, event storage, and alert communication..

PROPOSED METHODOLOGY

The proposed wildlife surveillance system consists of several interconnected stages, beginning with dataset preparation and ending with automated alert generation.

A. Dataset Collection

A customized wildlife dataset was prepared using images obtained from the Roboflow Universe. The dataset represents three animal categories:

Tiger – approximately 2,000 images

Lion – approximately 700 images

Leopard – more than 500 images

The collected images were prepared for training the object-detection model.

B. Data Pre-processing

The collected images were organized according to their corresponding animal classes and converted into a suitable object-detection dataset format. Appropriate pre-processing and annotation preparation were performed before model training.

C. YOLOv10 Model Training

The YOLOv10s architecture was selected for the proposed system because the surveillance application requires a balance between detection capability and real-time processing. The prepared dataset was used to train the model using the specified training configuration.

D. Real-Time Video Acquisition

After training, the resulting model was incorporated into a Python-based surveillance application. The application obtains frames from the surveillance source and processes them continuously.

E. Animal Detection

Each incoming frame is analyzed by the trained YOLOv10 model. When an animal belonging to one of the trained classes is identified, the system generates a corresponding bounding box and class label.

F. Object Tracking

Following detection, the system tracks the identified animal across successive frames. Tracking enables the surveillance application to maintain information about the movement of detected objects rather than treating every frame as an independent detection.

G. Snapshot Generation

When a relevant animal detection occurs, the application can capture and preserve a snapshot of the event. This provides visual evidence that can subsequently be reviewed by surveillance personnel.

H. Database Logging

Detection information is recorded using an SQLite database. Event records include timestamp-related information and other detection details generated by the surveillance application.

I. Automated Notification

The system incorporates an email notification mechanism. When a predefined wildlife detection event occurs, an alert can be transmitted to the concerned personnel, reducing the need for continuous manual observation.

Flow Chart

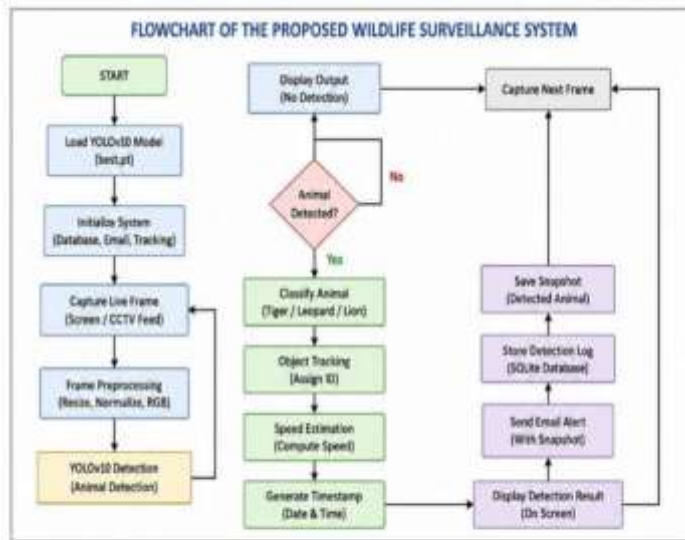


Fig. 1. Flowchart of the proposed wildlife surveillance system.

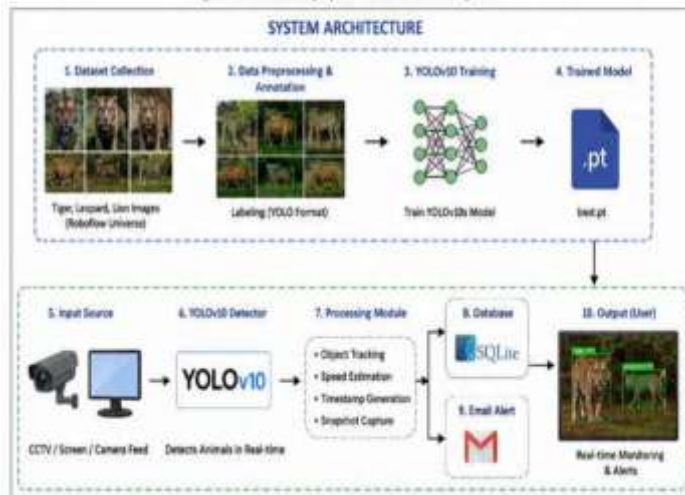


Fig. 2. System architecture of the proposed wildlife surveillance system.

SYSTEM IMPLEMENTATION

The system was implemented using Python 3.10 and the Ultralytics YOLOv10 framework. OpenCV was used for image processing and visualization, while SQLite was used for database management. The wildlife dataset collected from Roboflow Universe was trained using the YOLOv10s architecture with 150 epochs, batch size 4, and image resolution 640×640 pixels. The final trained model was saved as best.pt and integrated into the surveillance application for real-time detection.

Training Configuration

Parameter	Value
Model	YOLOv10s
Epochs	150
Batch Size	4
Image Size	640 × 640
Programming Language	Python 3.10
Framework	Ultralytics YOLO

MATHEMATICAL EXPRESSIONS

The system uses several mathematical expressions including Intersection over Union (IoU), Euclidean Distance Formula, Precision, Recall, and Speed Calculation. These expressions are used for object localization, tracking, performance evaluation, and movement estimation during real-time wildlife surveillance.

RESULTS AND DISCUSSION

The trained YOLOv10 model successfully detected tiger, leopard, and lion classes in realtime surveillance conditions. The system generated automatic snapshots, timestamps, database logs, and email alerts during animal detection events. GPU acceleration improved inference speed and enabled smooth real-time monitoring performance. The proposed system demonstrated higher automation and better monitoring efficiency compared to traditional surveillance systems.

I. COMPARISON WITH EXISTING SYSTEMS

Compared to traditional wildlife monitoring systems, the proposed system provides real-time multi-animal detection, automated alerts, object tracking, snapshot generation, and database logging. The integration of YOLOv10 and intelligent monitoring functionalities improves overall detection efficiency and response capability.

II. CONCLUSION

The proposed AI-Based Wildlife Surveillance and Animal Threat Detection System successfully integrates YOLOv10-based real-time animal detection with intelligent monitoring features. The system improves wildlife surveillance efficiency by providing automated alerts, database management, object tracking, and realtime monitoring. The developed solution can be deployed in forest-border regions, agricultural areas, and remote wildlife monitoring environments.

FUTURE SCOPE

Future improvements may include thermal camera integration, drone-based wildlife monitoring, cloud dashboard deployment, GPS tracking, mobile application support, and edge AI deployment using NVIDIA Jetson device

REFERENCES

1. You Only Look Once: Unified, Real-Time Object Detection
J. Redmon, S. Divvala, R. Girshick, and A. Farhadi, "You Only Look Once: Unified, Real-Time Object Detection," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, Las Vegas, NV, USA, 2016, pp. 779–788.
2. Simple Online and Realtime Tracking with a Deep Association Metric
N. Wojke, A. Bewley, and D. Paulus, "Simple Online and Realtime Tracking with a Deep Association Metric," in *2017 IEEE International Conference on Image Processing (ICIP)*, Beijing, China, 2017, pp. 3645–3649.
3. ByteTrack: Multi-Object Tracking by Associating Every Detection Box
Y. Zhang, P. Sun, Y. Jiang, D. Yu, F. Weng, Z. Yuan, P. Luo, W. Liu, and X. Wang, "ByteTrack: Multi-Object Tracking by Associating Every Detection Box," in *European Conference on Computer Vision (ECCV)*, 2022.
4. DeepSORT: Real Time & Multi-Object Detection and Tracking with YOLO and TensorFlow
A. Pujara and M. Bhamare, "DeepSORT: Real Time & Multi-Object Detection and Tracking with YOLO and TensorFlow," in *2022 International Conference on Augmented Intelligence and Sustainable Systems (ICAISS)*, 2022, pp. 456–460.
5. Video Object Tracking Based on YOLOv7 and DeepSORT
F. Yang, X. Zhang, and B. Liu, "Video Object Tracking Based on YOLOv7 and DeepSORT," *arXiv preprint arXiv:2207.12202*, 2022.
6. Multi-Object Tracking with Ultralytics YOLO
Ultralytics, "Multi-Object Tracking with Ultralytics YOLO," 2023