

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

1<sup>st</sup> Dr. Nilesh T. Gole

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

2<sup>nd</sup> Mr. Shantanu Punwatk

DEpt.Of Computer Science &Engineering  
Vidarbha Insitute of Technology, Nagpur

5<sup>th</sup> Mr. Soham Jumde

DEpt.Of Computer Science &Engineering  
Vidarbha Insitute of Technology

3<sup>rd</sup> Mr. Sanket Domki

DEpt.Of Computer Science &Engineering  
Vidarbha Insitute of Technology, Nagpur

4<sup>th</sup> Mr. Rohit Bisen

DEpt.Of Computer Science &Engineering  
Vidarbha Insitute of Technology.Nagpur

**Abstract** — Human-wildlife conflict has increased significantly in forest-border and rural regions. This research presents an AI-Based Wildlife Surveillance and Animal Threat Detection System using the YOLOv10 deep learning architecture for real-time detection of tiger, leopard, and lion. The system integrates object detection, object tracking, snapshot generation, SQLite database logging, timestamp recording, and automated email alerts into a unified surveillance platform. A custom dataset collected from Roboflow Universe consisting of approximately 2,000 tiger images, 700 lion images, and 500+ leopard images was used for model training. The trained YOLOv10s model demonstrated efficient real-time wildlife detection performance suitable for forest surveillance and remote monitoring applications

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

## I. INTRODUCTION

Wildlife intrusion into human-populated areas has become a major challenge due to deforestation and habitat destruction. Traditional monitoring systems mainly depend on manual CCTV surveillance and human observation, which are inefficient for continuous monitoring. Recent advancements in Artificial Intelligence and Computer Vision have enabled the development of intelligent surveillance systems capable of automated wildlife detection. The proposed system uses YOLOv10 for real-time detection of dangerous wild animals such as tiger, leopard, and lion. The system also provides automated alerts, database logging, snapshot generation, and realtime monitoring capabilities.

## II. LITERATURE REVIEW

Earlier wildlife detection systems used traditional image processing techniques such as Histogram of Oriented Gradients (HOG) and Support Vector Machines (SVM). These systems suffered from low accuracy and poor real-time performance. Modern deep learningbased approaches such as Faster R-CNN, SSD, and YOLO significantly improved object detection accuracy. However, many existing systems still lack automated alert systems, database logging, and efficient real-time monitoring features. The proposed system addresses these limitations using a

customtrained YOLOv10 model integrated with realtime surveillance functionalities.

## III. PROPOSED METHODOLOGY

The proposed methodology includes dataset collection, data preprocessing, YOLOv10 model training, real-time frame acquisition, animal detection, object tracking, snapshot generation, SQLite database logging, and automated email alerts. The system continuously captures frames from the surveillance source and processes them using the trained YOLOv10 model. Once an animal is detected, the system generates bounding boxes, tracks object movement, calculates movement speed, stores timestamped snapshots, and sends automated alerts.

## IV. Flow Chart

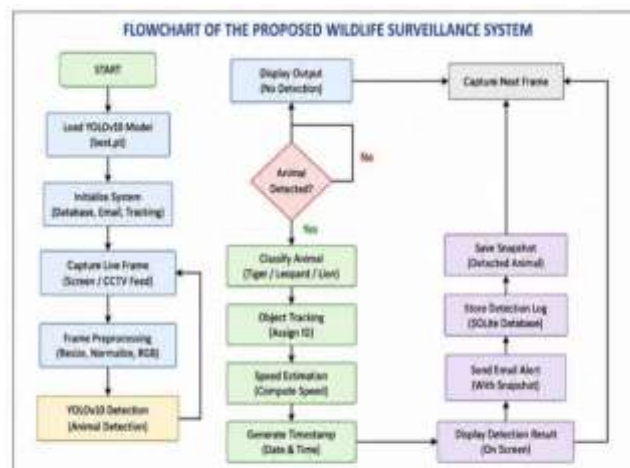


Fig. 1. Flowchart of the proposed wildlife surveillance system.

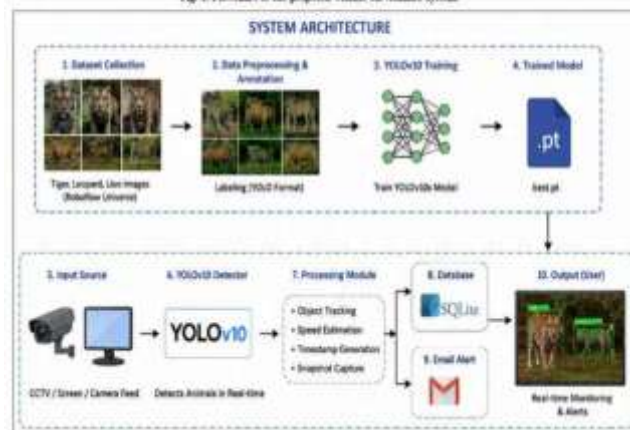


Fig. 2. System architecture of the proposed wildlife surveillance system.

V. **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.

VI. **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.

VII. **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.

VIII. **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.

IX. **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.

X. **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 devices.

#### Training Configuration

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

#### 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