International Journal of Scientific Research in Engineering and Management (IJSREM)
Volume: 09 Issue: 07 | July - 2025 SJIF Rating: 8.586 ISSN: 2582-3930

Real-Time Object Detection Using Open CV

Author: JAYENDRA UPADHAYAY(MCA student), M. BALA NAGA BHUSHANAMU[2] (Asst. Prof)
Department of Information Technology & Computer application, Andhra University College of Engineering, Visakhapatnam, AP.
Corresponding Author: Jayendra Upadhayay
(email-id:jayendra20551013@gmail.com)

Abstract: With the increasing demand for intelligent surveillance and environmental protection, object detection has emerged as a critical technology in computer vision. Traditionally, object detection systems relied on handcrafted features and shallow architectures, which often struggled with scalability and adaptability in complex real-world scenarios. These methods typically combined low-level image features with high-level contextual cues using ensemble strategies, but their performance plateaued when faced with dynamic and diverse inputs. The advent of deep learning has drastically changed this landscape, introducing robust models capable of learning semantic, high-level features through deep neural networks. These architectures vary significantly in terms of network design, training strategies, and optimization functions. In this paper, we present a comprehensive overview and implementation of a Python-based AI object detection system aimed at real-time animal monitoring and threat detection using the YOLOv8 framework. Designed for sensitive environments such as forests, farms, and conservation zones, our system performs real-time object classification and tracking with low latency. We review the evolution of object detection, from traditional methods to modern deep learning approaches, and evaluate the performance of YOLO-based systems in specialized tasks such as animal detection, pedestrian recognition, and environmental threat alerts. The proposed model demonstrates high accuracy and responsiveness, making it a strong candidate for practical deployment in multipurpose monitoring solutions.

Introduction: Machine Learning (ML) has emerged as a transformative technology in building intelligent systems capable of learning from large datasets and making autonomous decisions. Among its various subfields, object detection plays a fundamental role in Computer Vision by enabling machines to identify, classify, and localize objects within digital images or video streams. The effectiveness of object detection has led to widespread adoption in areas such as autonomous driving, surveillance, retail analytics, industrial automation, and environmental monitoring.

In recent years, the need for real-time animal monitoring and environmental threat detection has grown significantly, especially in regions where wildlife is vulnerable to threats such as poaching, unauthorized human intrusion, or natural habitat disruption. Traditional surveillance methods relying on manual monitoring or fixed sensors often prove to be insufficient, error-prone, and expensive to scale across large geographic areas.

To address these challenges, this research explores the application of modern ML-based object detection particularly the You Only Look Once (YOLO) family of models. The latest iteration, YOLOv8, developed by Ultralytics, offers significant improvements in accuracy, speed, and resource efficiency, making it ideal for real-time applications even on edge devices.

This paper proposes an detection framework that uses YOLOv8 to automatically identify animals and potential threats within forest areas, farmlands, and conservation zones. The system leverages Python-based tools and libraries for training, deployment, and integration with alert mechanisms. By combining cutting-edge ML techniques with practical deployment considerations, the project aims to create a cost-effective, adaptable, and scalable solution for wildlife protection as well as domestic animals and environmental security.

2: Literature Review

Over the past decade, Convolutional Neural Networks (CNNs) have significantly advanced the field of computer vision, particularly in areas such as image classification, object detection, and semantic segmentation. Among the earliest milestones in object detection, R-CNN and its variants (Fast R-CNN, Faster R-CNN) introduced region proposal mechanisms that significantly improved accuracy but came with high computational cost and latency. These models laid the foundation for more efficient architectures but were not ideal for real-time applications due to their multi-stage processing.

A major breakthrough in real-time object detection came with the introduction of the YOLO (You Only Look Once) family of models by Redmon et al., which reframed object detection as a single regression problem. YOLO directly predicts class probabilities and bounding boxes from entire images in a single forward pass of the network, enabling extremely fast inference speeds. Over the years, multiple YOLO versions have been released, with continuous improvements in accuracy, scale generalization, and computational efficiency.

Norouzzadeh et al. made a notable contribution in wildlife monitoring by applying deep learning models to automatically analyze camera trap images. Their work demonstrated how CNNs could outperform human experts in identifying and classifying species in large datasets. However, many of these earlier efforts were constrained by the need for high-performance GPUs, extensive pre-processing, and batch-oriented offline analysis—making them unsuitable for real-time field deployment in resource-constrained environments such as forests or remote farms.

Other studies explored hybrid approaches, integrating motion detection with object classification, or using IoT sensors with vision systems to improve efficiency. While these systems showed promise, they often lacked the robustness and flexibility needed for dynamic and diverse ecosystems.

In contrast, the recently released YOLOv8 by Ultralytics introduces a cleaner architecture, anchor-free detection, and improved task modularity. It is more lightweight and flexible than its predecessors, with built-in capabilities for both detection and segmentation tasks. Its performance in terms of FPS (frames per second) and mAP (mean Average Precision) makes it suitable for real-time, on-device processing, a key requirement for autonomous wildlife monitoring systems.

Our work builds on these advancements by implementing YOLOv8 in a Python-based environment for the purpose of detecting animals and threats in forest, zoo, and agricultural settings. It emphasizes adaptability, real-time detection, and low-resource deployment, contributing to the practical implementation of AI in environmental conservation and smart surveillance applications.

3. Methodology

The development of the proposed object detection system for animal and threat monitoring follows a structured Machine Learning (ML) pipeline. The methodology involves a sequence of steps including dataset preparation, image annotation, model training, evaluation, and deployment in real-time environments. This section outlines each phase in detail.

3.1 Dataset Collection

A clear and complete dataset is needed for training a robust object detection model. For this project, a custom dataset was created by collecting thousands of images from open-source repositories such as Open Images Dataset, Kaggle, Google Images, and public camera trap datasets. These images represent various animals (e.g., tigers, elephants, sheep, deer) and potential threats such as humans (poachers or intruders), often in forest, farm, or zoo environments. The dataset was curated to ensure diversity in lighting, background, angles, and object sizes to improve generalization during inference.

3.2 Data Annotation

Once collected, the images were labeled using the LabelImg tool, a popular graphical image annotation tool for object detection. Each object in an image was assigned a bounding box and a corresponding class label (e.g., "tiger", "sheep", "human"). These annotations were saved in the YOLO format (text files containing class number and bounding box coordinates relative to image dimensions), making them compatible with the YOLOv8 training pipeline.

3.3 Model Selection and Training

For the object detection model, YOLOv8 (You Only Look Once, version 8) developed by Ultralytics was selected due to its superior performance in terms of speed and accuracy. YOLOv8 adopts an anchor-free architecture and supports multiple tasks such as detection, segmentation, and classification, making it highly versatile.

PyTorch and the Ultralytics YOLOv8 API is used for training. Key training parameters such as learning rate, batch size, image size, number of epochs, and optimizer choice were carefully tuned. The training was performed on high-performance GPUs to accelerate the process. The dataset was split into training, validation, and test sets to monitor overfitting and ensure fair evaluation.

3.4 Evaluation Metrics

The performance and reliability of the trained YOLOv8 model is assessed using some standard object detection parameters:Precision – The percentage of correctly predicted objects out of all predicted objects.Recall – Real data should be present for the better program.mAP (mean Average Precision) – A comprehensive measure that considers both precision and recall across all classes and Intersection over Union (IoU) thresholds.A high mAP score indicated that the model could detect and classify animals and threats accurately across different scenarios.

3.5 Real-Time Deployment

To deploy the model for real-time monitoring, it was integrated with OpenCV, an open-source computer vision library. The trained YOLOv8 weights (best.pt) were loaded into a Python script that captures live video feeds from a webcam, CCTV, or drone camera. Each frame is processed in real-time, and when a detected object (e.g., an animal or intruder) matches predefined criteria, an alert mechanism is triggered—either via sound, text message, or system log.

This implementation ensures the system can be used in practical field settings, such as forest outposts or automated farm monitoring stations, where real-time response is critical.

3.6 Summary

The proposed methodology is a complete end-to-end ML-based object detection pipeline—from data acquisition to deployment. Leveraging YOLOv8's speed and accuracy, this approach delivers a scalable, cost-effective, and adaptable solution for animal and environmental security, supporting multiple real-world use cases with minimal latency.

4. YOLOv8 Object Detection Algorithm

YOLOv8 (You Only Look Once version 8) is the latest evolution in the YOLO family of object detection algorithms, developed by Ultralytics. Designed as a single-stage object detector, YOLOv8 excels in real-time performance while maintaining high accuracy. It improves on earlier versions by introducing an anchor-free architecture, enhanced post-processing, and advanced neural design strategies that make it well-suited for diverse applications, especially where both speed and resource efficiency are critical.

4.1 Working Mechanism

Unlike traditional multi-stage detectors (e.g., R-CNNs), YOLOv8 performs object classification and localization in one forward pass through the network. Here's how it works:

The input image is resized (e.g., to 640×640 pixels) and divided into a grid of cells.

Each cell predicts:

Bounding box coordinates (x, y, width, height)

Confidence score (how certain it is that an object exists in the box)

Class probabilities (e.g., “tiger,” “human,” “sheep”)

YOLOv8 is anchor-free, it does not rely on anything it will react as how our module is trained . Instead, it uses object center points as the basis for bounding box prediction, which simplifies training and improves speed.

Non-Maximum Suppression (NMS) is applied at the end to eliminate redundant overlapping boxes and keep the most confident detections.

4.2 Architecture Highlights

YOLOv8 introduces a more compact and efficient architecture that includes:

Backbone: A convolutional neural network (CNN) that extracts high-level features from the input image.

Neck: A path aggregation network that combines features from different stages of the backbone to capture spatial and semantic information.

Head: Predicts bounding boxes and class probabilities. Unlike YOLOv5, YOLOv8’s head is simplified for anchor-free prediction.

These components work together in a lightweight pipeline that maintains strong accuracy even on low-power devices such as Raspberry Pi, Jetson Nano, or smartphones.

4.3 Model Selection Justification

YOLOv8 was selected for this project due to its numerous advantages:

Real-time performance: With the configuration of better quality of system and hardware it will achieve good performance .

High accuracy: Competitive mAP scores on standard benchmarks (COCO, Pascal VOC).

Scalability: Supports different model sizes (e.g., YOLOv8n, YOLOv8s, YOLOv8m) depending on resource availability.

Edge-compatibility: Efficient enough to deploy in remote field locations where compute resources are limited.

Versatility: Easily integrates with Python libraries (OpenCV, PyTorch) and can be fine-tuned for specific domains like wildlife or agriculture.

4.4 YOLOv8 in Our System

In the implemented system, YOLOv8 is trained on a custom dataset of animals and intruders. The trained model (best.pt) is deployed in a Python environment to process live video feeds. Detection results are immediately analyzed, and when specific animals or threats are detected, the system triggers alerts for human intervention or logs the event.

This real-time object detection and alert mechanism is the core of the system’s functionality, enabling automatic surveillance in forests, zoos, and farms without constant human monitoring.

5. Results and Discussion

The proposed object detection system was evaluated to assess its accuracy, speed, responsiveness, and robustness in real-world scenarios. Upon training the YOLOv8 model on a custom dataset composed of annotated images of animals and human intruders, the system achieved a mean Average Precision (mAP) of 91.2% on the validation set. This indicates that the model effectively generalized across the classes it was trained on, demonstrating strong performance in distinguishing between different animals such as lions, tigers, wolves, and sheep, as well as detecting unauthorized human presence.

The real-time testing of the system was conducted using mid-range hardware including an Intel i5 processor and 8GB RAM, without GPU acceleration. Despite the modest computing resources, the model achieved a frame rate of approximately 25 to 30 frames per second (FPS) while processing live video streams. The detection-to-alert latency averaged 2.5 seconds, which is sufficiently fast for practical applications where immediate notifications are critical. Such responsiveness validates the system's capability to serve in scenarios requiring real-time monitoring and rapid decision-making.

Robustness testing was conducted under varied environmental conditions such as different lighting levels (daylight, dusk, and low light), cluttered backgrounds, and moving camera angles. The model maintained stable detection accuracy across most of these settings, though minor accuracy drops were observed in situations involving heavy motion blur or partial object occlusion. This suggests the system can operate reliably in semi-controlled and outdoor environments, though performance may benefit from the addition of infrared vision or night-mode enhancements in low-light situations.

Further evaluation included practical deployment tests in simulated environments representing forests, farmlands, and zoo enclosures. The model successfully identified relevant objects and triggered alerts accordingly. These trials validate the system's usability as a non-intrusive monitoring solution capable of reducing the need for manual surveillance. The alert mechanism was consistent and generated notifications with minimal delay, enhancing its potential for applications in wildlife conservation and animal safety.

When compared with conventional motion-based detection systems or earlier deep learning models like YOLOv3 or SSD, the YOLOv8-based implementation showed superior efficiency in both detection speed and accuracy. YOLOv8's lightweight architecture and anchor-free design offer advantages in terms of deployment flexibility, particularly on edge devices where computational resources are limited. Unlike traditional sensor-based systems that can only detect movement, this solution provides meaningful identification of object types, allowing for more intelligent response strategies.

Nonetheless, some limitations were observed. The model had reduced performance in very dark environments where standard camera input was insufficient, and in certain cases, it struggled to distinguish between visually similar animals. Additionally, in high-traffic zones where animals frequently moved, the frequent triggering of alerts could lead to notification fatigue, which may require enhancements such as region-based filtering or object tracking to suppress repetitive alerts for known non-threatening entities.

In conclusion, the results demonstrate that the proposed machine learning-based object detection system using YOLOv8 is a promising tool for real-time wildlife monitoring and environmental threat detection. It offers an effective balance of accuracy, speed, and scalability, making it suitable for practical deployment across a range of conservation and agricultural domains. The outcomes of this study affirm that with further refinement, such systems can significantly enhance our ability to protect animals and secure sensitive environments with reduced human effort.

6. Conclusion:

This research highlights the practical and technical viability of using machine learning, particularly deep learning-based object detection, to build an intelligent and multipurpose animal monitoring and threat detection system. By utilizing the capabilities of YOLOv8, one of the most advanced object detection models, the system achieves high accuracy and responsiveness while maintaining a lightweight architecture suitable for deployment on standard computing devices without the need for expensive GPU infrastructure. The custom dataset created for this project reflects real-world scenarios involving various animals and intruders, ensuring that the system is not just theoretically sound but practically relevant. The deployment pipeline—encompassing dataset creation, image annotation, training, evaluation, and integration with real-time video processing—has been designed to deliver a complete, end-to-end solution. Real-time tests validate that the system can function efficiently in dynamic, outdoor environments such as forests, farms, and zoos, where conditions are unpredictable and threats are often time-sensitive. The built-in alert mechanism adds a critical layer of utility by facilitating immediate human intervention when necessary. Compared to traditional surveillance systems that are costly and reactive, this system offers a cost-effective, proactive alternative. Its modular nature also ensures that the underlying architecture can be adapted for related tasks like crop monitoring, livestock counting, or even perimeter security, making it a versatile solution. In essence, this project serves as a strong demonstration of how machine learning and object detection can be combined to address real-world challenges in environmental monitoring, animal safety, and security automation, paving the way for future innovations in the field.

7. References

1. J. Redmon, S. Divvala, R. Girshick, and A. Farhadi, "You Only Look Once: Unified, Real-Time Object Detection," in *Proc. IEEE Conf. Computer Vision and Pattern Recognition (CVPR)*, 2016.

2. A. Bochkovskiy, C.-Y. Wang, and H.-Y. M. Liao, "YOLOv4: Optimal Speed and Accuracy of Object Detection," *arXiv preprint arXiv:2004.10934*, 2020.

3. G. Jocher et al., "YOLOv5 and YOLOv8 Documentation and Models," *Ultralytics*, 2023. [Online]. Available: https://docs.ultralytics.com/

4. M. S. Norouzzadeh, A. Nguyen, M. Kosmala, A. Swanson, M. Palmer, C. Packer, and J. Clune, "Automatically identifying, counting, and describing wild animals in camera-trap images with deep learning," *Proceedings of the National Academy of Sciences (PNAS)*, vol. 115, no. 25, pp. E5716–E5725, 2018.

5. OpenCV: Open Source Computer Vision Library. [Online]. Available: https://opencv.org/

6. LabelImg: Image Annotation Tool. [Online]. Available: https://github.com/tzutalin/labelImg