D. D. Pukale, 2026, 14:2
ISSN (Online): 2348-4098
ISSN (Print): 2395-4752

International Journal of Science, Engineering and Technology

An Open Access Journal

Yolo Based Real Time Animal Detection And Counting System

D. D. Pukale , Vinaya Kulkarni , Riddhi Taharabadkar , Apeksha Varangane , Pranali Barge , Sakshi Sonawane

Department of Computer Engineering, Bharati Vidyapeeth's College of Engineering for Women, Pune, India

Abstract- Wildlife monitoring is an important tool for improving biodiversity conservation efforts, conducting ecological research, and managing human/wildlife interactions. Traditional methods of monitoring wildlife such as manual observations of live animals or using captured camera images usually take considerable time and resources, and therefore they are prone to human error. In this paper, we describe an array of proposed automated systems designed to enable real-time detection and identification of wild animals through captured images and video streams, utilizing deep learning techniques. For this work, we will use the YOLO (You Only Look Once) object detection algorithm in conjunction with specific computer vision techniques in order to detect and classify different animal species based on captured images and video streams. The model was trained using both annotated datasets and defined datasets containing multiple types of species (e.g., deer, elephants, tigers, leopards). The detection process includes extracting frames from a video stream and performing image preprocessing on the individual frames, applying the trained YOLO object detection model to localize (bounding box) and identify (classification) the species of the detected animal(s) from the individual frames based on a confidence score. In addition to species detection, we have also developed a multi- object tracking system that allows for consistent identification of an animal across different frames and prevents double counting of an animal. The experimental trials conducted using this system have demonstrated high detection rates and real- time performance in various environmental conditions, including motion blur, partially obscured individuals, and shadows and reflections of light. This detection system can be used to assist with the management of forest-related resources. Recommendations made by the authors include: 1) the ability to more effectively manage forest resources through increased knowledge of wildlife behavior; 2) improved training and collaborative efforts with other researchers may result in added benefits; and 3) forest managers can derive benefits from the implementation of this new technology.

Keywords—Wild Animal Detection, Animal Face Recognition, YOLO Algorithm, Deep Learning, Computer Vision, Real-Time Object Detection, Wildlife Monitoring, Multi-Object Tracking.

I. INTRODUCTION

Wildlife monitoring is essential for conserving biodiversity, managing ecosystems, and providing for sustainable environments. Monitoring animal populations enables researchers and conservation authorities to understand the distribution, migration patterns, and habitat use of various species. The traditional methods of monitoring wildlife have predominantly consisted of using manual observation as well as analyzing camera trap images and conducting field surveys. While these methods have been an important source of ecological data, they are often labor-intensive, time- consuming, and susceptible to human error in processing large amounts of photographic and/or video data obtained from remote environments .

As advancements in artificial intelligence technologies have advanced rapidly, automated wildlife monitoring systems have attracted great interest in recent times. The implementation of deep learning techniques, particularly CNNs, has proven highly successful for the tasks of image classification, object detection, and visual recognition. These models can develop complex visual patterns when trained on large datasets, enabling them to detect objects from photographic images with great accuracy. Therefore, many researchers are now using deep learning techniques in the automated analysis of wildlife imagery and video for reducing the human resources required for conducting wildlife surveys and increasing efficiency .

The YOLO (You Only Look Once) approach is one of the leading techniques in real-time object detection among many object detection algorithms. YOLO detects both the class of the object and where it is located at the same time using one neural network; this allows for high detection speed and competitive accuracy. Both YOLO-based systems and regular two-stage detectors such as Faster R-CNN have different inference latencies and are both viable solutions for real-time monitoring systems, but YOLO-based systems are especially beneficial for applications including surveillance and monitoring wildlife .

Several studies have used deep learning architectures, specifically convolutional neural networks (CNNs), to detect and classify wildlife through camera traps and video footage. These studies have demonstrated that CNN-based solutions using the YOLO framework are capable of identifying many species of animals in complex environments as well as identifying animals even when there are changes in light, obstructions to view, or environmental clutter. By automatically recognizing animals in photos taken by a camera trap or by a video camera, these systems save vast amounts of human labor and allow scientists to conduct large scale ecological research through the monitoring of wildlife)

.

In recent years, some researchers have used object tracking and detection algorithms together to improve reliability of wildlife monitoring systems. An example of this would be SORT, DeepSORT, and ByteTrack multi-object tracking algorithms that allow you to maintain the identity of each individual animal throughout the different frames of a video sequence. By maintaining this identity, duplicate counts are minimized, and accurate estimates of animal populations and movement patterns can be generated over time .

While progress has been made by combining both tracking and detection technologies for wildlife, there are still multiple obstacles that these automated systems must overcome in order to work effectively in a variety of environmental conditions. For example, dense vegetation, low light, motion blur, and visual similarities among species can all impact accuracy of detection algorithms used to identify animals. Animals that are either captured from too far away or are partially obscured by other environmental objects may also be incorrectly identified as well or even missed completely which creates a need for robust detection systems capable of performing properly in these types of real-world environments .

In the last few years, scientists have introduced a variety of improved deep learning models to improve how well we can detect wildlife. The main areas of improvement for these models are feature extraction, accuracy of detection, and speed of computer processing through optimized neural networks and new techniques for augmenting data. Furthermore, the combination of smart monitoring dashboards and automatic reporting has created an environment for real-time analysis of wildlife behaviour to assist with conservation and ecological research .

To further this line of research, this article presents a deep learning-based automated system to detect and classify wild animals through the use of still images and video footage of animal faces. This proposed system uses the YOLO object detection method for the detection and classification of animals using still photo-takes from a camera feed to produce a real-time detection of animal species. Also, a multi-object tracking system is used to enable accurate counting and tracking of animals at different times across multiple frames.

This proposed system is intended to improve the performance, accuracy and scalability of wildlife monitoring systems, while assisting with conservation efforts and intelligent monitoring of forests .

II. LITERATURE SURVEY

The initial studies examining the effectiveness of automated systems for monitoring wildlife showed how important deep learning models will be in evaluating camera trap images acquired from natural environments. Schenider et al. demonstrated how both YOLO and Faster-RCNN models could automatically detect and classify animal species in ecological based datasets from natural environments. This significantly decreased the amount of effort that was previously spent on manual annotation and improved the efficiency of conducting wildlife population studies. They further demonstrated how an automated detection system can increase the effectiveness and efficiency of large scale ecological monitoring. [1]

NLourezadeh et al. presented a method that uses deep convolutional neural network-based models to automatically identify, count and describe wildlife images found in the Snapshot Serengeti study. The researchers found that the models they used produced very high classification accuracies, and demonstrated that deep learning based models (such as CNNs) could be used to successfully replace the need for manual analysis of all wildlife images in order to conduct biodiversity assessments of wildlife. [2] Dave et al. developed a method for real-time automated detection of wildlife, utilising the YOLOv8 architecture, that was able to detect multiple types of wildlife species using their webcam video stream or from a live webcam feed. They demonstrated that their model and detection algorithms were able to increase the rate and accuracy at which they could detect animals in real time. Thus, it would be feasible for continuous surveillance of wildlife using their method in realistic situations. [3]

Nagaraj et al. created an automated detection and tracking framework using YOLOv8, which is designed to deliver real- time monitoring of wildlife in natural habitats. Their wildlife detection and tracking system integrated a dual detection and tracking system to allow continuous monitoring of wildlife, and was successfully implemented in an intelligent forest surveillance system as an efficient solution. [4]. In forest settings that can be very intricate, the paper by Ma and others set forth a simple, real-time wildlife detection algorithm called WL-YOLO. The authors examined how occluded objects, varying amounts of light, and having many things in the background can affect the performance of a detection algorithm, and showed that tropical locations achieve better detection results than temperate areas. [5]

Acoustic data was used in the YOLO11-APS study by Du et al. to improve upon existing object detection solutions with an additional emphasis on protecting wildlife at camera traps when being monitored. Some of the improvements were improved feature extraction and reduced processing time for inference on embedded hardware. [6] In conjunction with YOLOv8n, Yang et al. have developed a new algorithm called YOLO-SAG that aims to be more accurate while maintaining the same speed as current algorithms to find wildlife. The authors discussed the difficulties of finding wildlife that are of different sizes as well as within multiple environments (e.g. urban, rural). [7]

Amidst the global increase in population and habitation practices, the need for surveillance techniques to monitor how wildlife are responding to human development has gained importance and urgency. In this regard, Rahman et al. have developed an edge-based architecture for tracking wildlife using hybrid tracking methods and YOLOv8 detection. Their proposed architecture is capable of real-time tracking of animals from embedded devices even in outdoor/remote monitoring applications. [8]

Tan et al. researched deep learning methodologies to predict the automatic detection and classification of animals in video taken from camera traps. By using an automated vision system to process large data sets (wildlife images) researchers were able to increase the efficiency of their ecological studies. [9] Haldorai et al. used a Recurrent YOLO (ROLO) framework to monitor and track wildlife. By synchronizing temporal data the accuracy of animal recognition was substantially improved across the duration of the video sequences, allowing for reliable monitoring of fauna and flora. [10] Kadam et al. developed a YOLOv8 based animal detection methodology by using some initial pre-processing combined with feature extraction techniques. The deep learning models used in their study were able to support accurate species identification from images and video for use in conservation studies. [11]

Mustafić et al. studied the use of computer vision techniques to automate the identification and classification of species from video images sent from camera traps and suggested that the success of AI-based monitoring could ultimately improve biodiversity assessment and conservation planning. [12] Researchers such as Pishdast et al. have demonstrated how AI- enabled smart camera traps that utilize the YOLO algorithm for real-time detection of wildlife can be created. They show low-cost deployment capabilities with respect to animal detection capabilities in field environments. [13] Raza et al.'s research found that both CNN-YOLO and transformer-based approaches to animal detection in dense environments are valid options for monitoring animals. Additionally, their research concluded that YOLO and other one-stage detectors would provide better outcomes (i.e., quicker results) than two- stage detectors for real-time surveillance of wildlife. [14]

The research of Rajagukguk et al. looked at deep learning techniques used in visual monitoring of animals and found that accuracy of counting and behavioral analysis of animals in the monitored environment was enhanced by the implementation of YOLO detection combined with other tracking algorithms such as DeepSORT and ByteTrack. [15] Researchers Du and colleagues created a lightweight model based on YOLO for the real-time monitoring of wildlife. This research focused on increasing efficiency in detecting wildlife while reducing the complexity of computation in order that the system could run on an edge device that are limited in available resources. The results of their proposed approach were effective in detecting wildlife species in successful applications for wildlife research and monitoring. [16]

The second research by Tan and colleagues developed an automated system which detects wildlife species from camera trap images utilizing deep learning techniques; this automated system enables the ability to classify wildlife species automatically as part of the analysis of large amounts of camera trapping data which has dramatically reduced the amount of time needed to manually analyse images collected in ecological research studies. [17]

A real-time detection system to detect wildlife using video streams was presented by Nagaraj et al. They used YOLOv8 to scan the environment and track animals in real-time. This method significantly increased the accuracy of the detection, and speed of the processor making it suitable for continuous monitoring of wildlife in the forests. [18]

A real time tracking system called DeepSORT was proposed by Wojke et al. This is achieved by combining appearance descriptors with motion information. The major improvement from this method is to increase tracking accuracy and maintain the identity of object in the frames used for multi-object tracking applications. [19]

The development of the Simple Online and RealTime Tracking algorithm by Bewley et al. was based on using Kalman filtering and data association to enable effective tracking of an object with a fast and reliable performance in real time detection systems. [20] ByteTrack was developed by Zhang et al. for the purpose of improving multi-object tracking accuracy by associating the detection boxes between frames. This technique improved tracking accuracy and robustness in a number of scenarios where multiple objects are on the move. [21] The Faster R-CNN framework for improved object detection performance was introduced by Ren et al. Their approach achieved higher object detection accuracies through the combination of convolutional neural networks with region proposal networks in a single model. [22]

He and his colleagues recommended deep residual learning or Residual Net (ResNet) for the purpose of enhancing instruction for training deep neural networks via adding residual linkages. ResNet provided superior results than before with respect to the accuracy of image identification tasks and thus became very popular among all deep learning models. [23] In the previous year Krizhevsky, along with any number of his colleagues, presented a new architecture for a deep convolutional neural network to classify images against the ImageNet dataset. Their success demonstrated the value that deep learning models provide in terms of visual recognition and helped push the visual recognition field forward significantly. [24]

In the same year Lin and his co-authors released the Microsoft COCO dataset that allows for large-scale image data collection to support object detection, segmentation and image captioning. Today this dataset is among the most widely used datasets in practice to evaluate computer vision models on a variety of benchmarks.[25] Girshick developed Fast R-CNN to allow for more efficient object detection by integrating region proposal generation with convolutional neuron networks into a single training pipeline. Fast R-CNN reduces the time to train the model and increases the accuracy of detection as compared to previous methods. [26]

Long et al. created fully convolutional networks (FCN) and used them to perform semantic-level segmentation of images. In their proposed system every pixel in an image will be classified in accordance with a visual concept. This capability will enable investigators to perform semantic segmentation of multiple images and will support a variety of computer vision-related applications such as object segmentation and scene classification. [27].

III. PROPOSED SYSTEM ARCHITECTURE

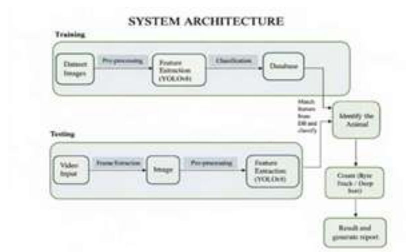

Fig. 1. System Architecture

1. Collecting Datasets

Labeled images are gathered from a range of collected datasets including animals used to train the classification models for Detecting & classifying animals.

2. Pre-Processing images

The images collected from the various datasets will undergo pre-processing to increase the quality & consistency of the images. Pre-processing will consist of resizing images, normalizing image sizes (coloration & shaping the animal), & removing any noise from the images to allow the images to be processed more effectively through the DL model.

3. Feature Extraction using YOLOv8

Using the YOLOv8 model, the DL model will be used to learn the visual features of the animals within the dataset. YOLO is able to learn how to visually differentiate between animal types & styles & how to accurately classify them based upon their feature analysis.

4. Classifying Animals

Once the features are extracted from the images, the system then stores them in a database to classify the detected objects into the various animal species.

5. Detect and Track Animals in Real-Time

When testing begins on the system, the user must provide a video input to the system. The video input will be broken down into frames and the YOLOv8 will identify each frame

with some animal(s) in them. The tracking algorithms will then track the animals through the frames where they were detected to get an accurate count.

6. Identification of Animals

The final step of the system will be to identify the animal(s) detected and report the animal's identification and their total count. The system can also generate reports based on the data for wildlife management and/or analysis.

IV. METHODOLOGY

This proposed approach to automatically detecting, classifying and counting wildlife by using deep learning and computer vision techniques is defined in six main phases: data collection, preprocessing, feature extraction, detection/tracking and results generation. Each step contributes to improving the accuracy and efficiency of the wildlife monitoring system as a whole.

A. Dataset Collection

First, the creation of a wildlife database is the first step in the system. Images and video clips of various animal types (deer, elephants, tigers and leopards) are all included in this database. The images were obtained through public wildlife databases and videos. Each picture in the database has been labeled with bounding boxes that indicate where each animal is located within the picture. The labeled databases provide the deep learning model with visual characteristics of the various animal types during the model's training stage. The model's generalization aspect will be enhanced by providing the model with sufficient diversity within the various animal type databases, through images taken in relatively different lighting, locations and perspectives.

B. Image Pre-processing

Prior to training the model, a series of prerendering processes are applied to the collected dataset to ensure consistent image quality and uniformity. These processes include resizing the image to a predetermined resolution, thus allowing for the same size to be inputted into the model. Pixel value normalization helps ensure stable training of the neural network. Noise filtration and image enhancement techniques are applied to assist in filtering out disturbances caused by the background. To artificially expand the size of the dataset and make the model more robust to environmental differences, a number of augmentation processes are implemented, such as rotation, reflection, scaling, and changes to brightness.

C. Feature Extraction using YOLOv8

The process of determining the meaningful visual patterns from the original images is the extract feature stage. For feature extraction and object detection, the proposed system uses the YOLOv8 deep learning model, which is a one-step detection algorithm that gives both an object location and an object classification at once by processing the input image through multiple convolutional layers in the model. Important features from the input image, including shape, texture, edges, and patterns associated with each species of animal are extracted from the image. These features are then used by the model to differentiate between species of animals contained within the dataset of images. YOLOv8 is selected because of its accuracy in object detection and quick processing speed, which makes it suitable for use in real-time animal monitoring.

D. Animal Detection and Classification

In the first stage of animal detection, YOLOv8 analyses the image features extracted using computer vision techniques (feature extraction), detecting and capturing animals in bounding boxes as they appear in the image. Each bounding box gets a confidence score by assigning a probability value that indicates how likely it is that the identified object belongs to a specific animal class. The animal's classification during this stage produces the class labels using trained feature values. YOLOv8 can simultaneously identify multiple wildlife species in an image or within one video frame. During this phase, the animal classification process compares one animal's classification results with previously saved YOLOv8 trained feature values to verify the accuracy of the identified species.

E. Real-Time Video Processing

The system receives video data for testing from the video surveillance cameras and pre-recorded video of wildlife. The video is broken down into separate frames by an external piece of equipment called a frame extractor. The individual frames from the video are processed as images; they are sent to the system for use in the YOLOv8 detection model. YOLOv8 will receive the images it previously received as individual frames and look for animals to be classified and have bounding boxes around them. When completed, using YOLOv8 detection model, this entire animal detection process will provide real-time wildlife activity monitoring capabilities.

F. multi-object tracking and counting

The system incorporates multi-object tracking algorithms such as ByteTrack or DeepSORT to identify and track animals throughout multiple video frames. These algorithms create unique ID numbers for detected animals to help track their trajectory over time, so there won't be more than one copy of an animal when it appears in several video frames. Using the tracking module, you can also analyze movement patterns and behaviors of the animals being studied in real time within the scope of the monitoring environment.

G. Result generation and analysis

The last step in the process is result generation and recording. All detected animals will be shown on the user interface with their respective class labels and confidence scores, including the number of detected animals in the video stream, total count and monitoring information about the animals within the monitoring environment. The user interface provides real- time detection output; therefore, the detection data can be stored in a database for future review, reporting and populating wildlife.

V. RESULT

Through the use of actual video footage from the field, which include various other species, the potential for animal detection and monitoring technology was assessed. The proposed system utilizes the YOLOv8 deep learning model for detecting animals as well as algorithms such as ByteTrack or DeepSORT for tracking their movements from frame to frame. Based on the trials conducted, I have concluded that the proposed system demonstrates the ability to detect, classify, and track multiple animals concurrently in real time.

The system was evaluated through the use of video feed that contained an assortment of animals such as elephants, giraffes, zebras, rhinos, and tigers. The detection portion of the model successfully created a bounding box around each identified species and generated an associated label as well as confidence score. Using an identification number to designate each animal identified as a unique animal, as well as a method of preventing multiple counts for the same animal.

The proposed system offered a real-time monitoring dashboard that displays the numerical totals of each category of identified species, the number of individual species detected, and whether the animal was still being tracked. This interface enables users to efficiently monitor animal activity and evaluate the movement of the animal within the context of time.

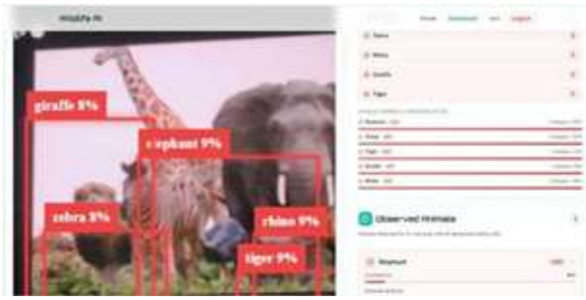

Fig. 1. Real-Time Wild Animal Detection and Identification Interface

This figure illustrates the real-time wildlife detection interface of the proposed system. The YOLOv8 model analyzes the video frame and identifies several different types of animals. A bounding box, class id and confidence level identify each detected animal.

The image depicts multiple types of animals including a giraffe, elephant, zebra, rhinoceros and tiger. The detection model assigns a confidence level to each detected animal to indicate how confident it is in proving that the animal has been correctly classified. The interface's right side panel displays the detected animal list and tracking information for each of them. The system tracks unique animals by assigning each of them a unique identifier and provides the ability to monitor wildlife events in real-time.

This interface shows that the proposed system can detect and identify multiple types of animals at the same time while performing processing operations quickly and accurately.

Fig. 2. Real-Time Animal Tracking and Monitoring Dashboard

The system produced an animal tracking and monitoring dashboard, as shown in figure 2. The tracking and monitoring dashboard provides a detail overview of the animals that were detected and the corresponding status of their tracking. In the monitoring environment, the system has detected five different animals and provides the tracking Id for each detected animal.

As part of the monitoring dashboard, the monitoring dashboard lists the five animals that have been detected which include the Elephant, Zebra, Tiger, Giraffe and Rhino. The dashboard also lists the number of different instances of unique animals detected within the monitoring dashboard by species. The tracking algorithm allows for each detected animal to have its own unique tracking ID so that an animal cannot be counted multiple times in different frames.

The monitoring dashboard has an Observed Animals section that tracks the amount of time certain animals have been detected for their corresponding observed periods, providing an accurate record of each species of animal's amount of time that the species was observed so that researchers and/or forest authorities can quickly and easily track animal activity within a monitored environment and maintain timely and accurate documentation of animal activity in the environment.

VI. CONCLUSION

This article described a novel real-time wild animal detection, classification, and tracking system based on deep learning techniques using computer vision methods. The system incorporated deep learning object detection (YOLOv8) to identify and track different species of wild animals by leveraging traditional computer vision techniques. In conjunction with the object detection system, multi-object tracking algorithms (ByteTrack and DeepSORT) were integrated into the overall system to provide accurate tracking of wild animals across multiple frames while eliminating duplicate counts. The research demonstrated that the result of the project was a successful prototype capable of detecting and classifying a range of animal species in real-time (i.e., elephant, giraffe, zebra, rhino, tiger) from static images and video streams. The prototype was also able to provide a monitoring dashboard with information about tracked animals, tracking data, and analytical statistics so researchers and forest authorities can effectively evaluate the behaviors of wild animals. The system and its capabilities offer an automated wildlife monitoring system that is scalable in nature, as well as the ability to reduce the number of human observers required to monitor wildlife while increasing the accuracy of identified species. By assisting with ecological research, monitoring local/national wildlife populations, and detecting animals in sensitive lands prior to human-caused disturbance, the proposed method can aid wildlife conservation. Thus, the combination of deep learning models with computer vision provides a unique opportunity for developing intelligent wildlife monitoring systems.

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, pp. 779–788.

[2] J. Redmon and A. Farhadi, "YOLO9000: Better, Faster, Stronger," in Proc. IEEE CVPR, 2017, pp. 7263–7271.

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

[4] C.-Y. Wang, A. Bochkovskiy, and H.-Y. M. Liao, "Scaled- YOLOv4: Scaling Cross Stage Partial Network," in Proc. IEEE CVPR Workshops, 2021.

[5] G. Jocher et al., "YOLOv5," GitHub Repository, 2020. [Online]. Available: https://github.com/ultralytics/yolov5

[6] G. Jocher, A. Chaurasia, and J. Qiu, "YOLOv8: Ultralytics YOLO," 2023. [Online]. Available: https://github.com/ultralytics/ultralytics

[7] N. Schneider et al., "Deep Learning Object Detection Methods for Ecological Camera Trap Data," Methods in Ecology and Evolution, vol. 11, no. 3, pp. 353–362, 2020.

[8] M. S. Norouzzadeh et al., "Automatically Identifying, Counting, and Describing Wild Animals in Camera-Trap Images," Proc. National Academy of Sciences, vol. 115, no. 25, pp. E5716–E5725, 2018.

[9] B. Dave et al., "Wild Animal Detection Using YOLOv8," Procedia Computer Science, vol. 218, pp. 1523–1532, 2023.

[10] Z. Ma et al., "WL-YOLO: Lightweight Wildlife Detection Model in Complex Forest Environments," Remote Sensing, vol. 16, no. 8, 2024.

[11] W. Yang et al., "Improved YOLO-Based Wildlife Detection Algorithm," Sensors, vol. 23, no. 15, 2023.

[12] L. Chen et al., "YOLO-SAG: Attention-Based Wildlife Detection Algorithm," Information Processing Letters, vol. 190, 2024.

[13] D. Velasco-Montero et al., "AI-Based Smart Camera Traps for Wildlife Monitoring," Ecological Informatics, vol. 75, 2024.

[14] R. A. Rajagukguk et al., "Deep Learning Approaches for Visual Animal Monitoring," Smart Agricultural Technology, vol. 9, 2025.

[15] A. Raza et al., "Deep Learning Models for Wildlife Monitoring Applications," Scientific Reports, vol. 15, 2025.

[16] Z. Du et al., "Lightweight YOLO-Based Wildlife Monitoring System," Sensors, vol. 25, no. 23, 2025.

[17] M. Tan et al., "Automatic Animal Detection from Camera Trap Images," Ecological Informatics, vol. 68, 2022.
[18] N. Nagaraj et al., "Real-Time Wildlife Detection Using YOLOv8," Springer Nature Computer Science, 2025.
[19] N. Wojke, A. Bewley, and D. Paulus, "Simple Online and Realtime Tracking with a Deep Association Metric," in Proc. IEEE ICIP, 2017.
[20] A. Bewley et al., "Simple Online and Realtime Tracking (SORT)," in Proc. IEEE ICIP, 2016.
[21] Y. Zhang et al., "ByteTrack: Multi-Object Tracking by Associating Every Detection Box," in Proc. ECCV, 2022.
[22] S. Ren, K. He, R. Girshick, and J. Sun, "Faster R-CNN: Towards Real-Time Object Detection," IEEE TPAMI, vol. 39, no. 6, pp. 1137–1149, 2017.
[23] K. He et al., "Deep Residual Learning for Image Recognition," in Proc. IEEE CVPR, 2016.

[24] A. Krizhevsky, I. Sutskever, and G. Hinton, "ImageNet Classification with Deep CNNs," Communications of the ACM, vol. 60, no. 6, pp. 84–90, 2017.
[25] T.-Y. Lin et al., "Microsoft COCO: Common Objects in Context," in Proc. ECCV, 2014.
[26] R. Girshick, "Fast R-CNN," in Proc. IEEE ICCV, 2015.
[27] J. Long, E. Shelhamer, and T. Darrell, "Fully Convolutional Networks for Semantic Segmentation," in Proc. IEEE CVPR, 2015.
[28] H. Ma et al., "Multi-Object Tracking for Livestock Monitoring," Expert Systems with Applications, 2025.
[29] P. Dollar et al., "Pedestrian Detection: An Evaluation," IEEE TPAMI, vol. 34, no. 4, pp. 743–761, 2012.
[30] R. Szeliski, Computer Vision: Algorithms and Applications, Springer, 2011.