



Wild Animal Tracking for Effective Wildlife Conservation using YOLOv8 and Machine Learning Technologies

Taeho Kim¹

10.18805/IJAR.BF-2054

ABSTRACT

Background: Wild animal tracking is crucial for conservation, especially for threatened species like tigers. The Amur tiger is critically endangered, with fewer than 600 left in the wild. The integration of machine learning and drone technology has revolutionized tiger tracking. Drones, or UAVs, play a vital role in wildlife conservation, while machine learning algorithms analyze complex data, make predictions and automate decision-making. This combination enables efficient processing of drone-generated data, including images, sounds and sensor readings.

Methods: This study utilized a dataset comprising 4,413 images sourced from Kaggle, originally derived from the ATRW (Annotated Tigers in the Wild) dataset. To ensure compatibility with the YOLOv8 model, the images were reformatted from PASCAL VOC annotation format to YOLO Darknet format. A series of preprocessing techniques were implemented, including image resizing, data augmentation (such as rotation, horizontal flipping and brightness adjustment) and pixel value normalization to enhance model generalization. The dataset was systematically partitioned into training, validation and test subsets to facilitate robust model evaluation and performance assessment.

Result: The YOLOv8 model exhibited high efficacy in detecting tigers across diverse environmental conditions. It achieved a final Mean Average Precision (mAP) of 0.944, indicating robust detection accuracy. The model demonstrated strong generalization capabilities, effectively identifying tigers under varying illumination, occlusion and background complexities. These findings highlight the potential of YOLOv8 as a reliable tool for automated wildlife monitoring and conservation efforts.

Key words: Biodiversity preservation, Computer vision, Object detection, Wild animal tracking, Wildlife conservation, YOLOv8.

INTRODUCTION

Wildlife conservation is a complex, multidisciplinary challenge involving ecological, social and technological factors. Research integrating ecology with computer vision highlights the critical role of accurate wildlife recognition in maintaining ecosystem balance and stability. However, several technological challenges must be addressed to achieve reliable monitoring and effective conservation outcomes (Bhattacharya *et al.*, 2022). Ecological research often requires extensive data collection in remote regions, where inadequate infrastructure significantly hinders efficiency and data quality. Overcoming these limitations is essential for strengthening conservation strategies and ensuring sustainable ecosystem management (Pendharkar *et al.*, 2024).

Traditional wildlife recognition techniques, such as tagging and capturing animals, are invasive and can disrupt natural behaviors, induce stress and alter habitats. In contrast, non-invasive approaches offer safer and more ethical alternatives. Technologies such as camera traps, drones and footprint tracking enable researchers to collect valuable data without disturbing wildlife, thereby enhancing conservation efforts while prioritizing animal welfare (Ravor *et al.*, 2021). The responsible integration of such technologies promotes improved coexistence between humans and wildlife, which is vital for preserving biodiversity and protecting natural resources (AlZubi and Alkhanifer, 2024; Min and Kim, 2024).

¹Department of Marine Production Management and Smart Aquaculture Research Center, Chonnam National University, 50, Daehak-ro, Yeosu-si, Jeollanam-do, 59626, Republic of Korea.

Corresponding Author: Taeho Kim, Department of Marine Production Management and Smart Aquaculture Research Center, Chonnam National University, 50, Daehak-ro, Yeosu-si, Jeollanam-do, 59626, Republic of Korea. Email: kimth@jnu.ac.kr
ORCID: <https://orcid.org/0000-0001-5043-5828>

How to cite this article: Kim, T. (2026). Wild Animal Tracking for Effective Wildlife Conservation using YOLOv8 and Machine Learning Technologies. *Indian Journal of Animal Research*. **60(6)**: 1080-1087. doi: 10.18805/IJAR.BF-2054.

Submitted: 04-08-2025 **Accepted:** 28-12-2025 **Online:** 31-12-2025

A modified YOLO model, Recurrent You Only Look Once (ROLO), was implemented for wildlife surveillance and demonstrated promising performance in identifying diverse wild animals and birds. Comparative analysis with other deep learning models revealed higher precision rates, underlining its effectiveness for automated wildlife monitoring (Haldorai *et al.*, 2024; Altobel and Sah, 2021; Cho, 2024; Kim and AlZubi, 2024). Bakana *et al.* (2024) introduced WildARe-YOLO, a lightweight variant of YOLOv5s designed for wild animal recognition, incorporating Mobile Bottleneck Blocks and an enhanced StemBlock to reduce computational complexity. The model utilized Focal-EIoU for improved bounding box accuracy and a BiFPN-based

neck, achieving a 17.65% increase in FPS, a 28.55% reduction in parameters and a 50.92% reduction in FLOPs, thereby enabling efficient real-time recognition on low-end hardware.

Mamidi *et al.* (2024) proposed an IoT-based animal detection system aimed at improving campus safety, integrating a surveillance robot equipped with ultrasonic sensors and ESP32 cameras and employing the R-CNN machine learning technique for detection and classification. The system achieved an accuracy of approximately 97.6% in animal detection and provided real-time alerts to security personnel through push notifications.

Ferrante *et al.* (2024) assessed multiple YOLO architectures for detecting road-killed endangered Brazilian animals, including YOLOv4, Scaled-YOLOv4, YOLOv5, YOLO-R, YOLO-X and YOLOv7, using the BRA-Dataset for training. Their findings indicated that Scaled-YOLOv4 minimized false negatives most effectively, while the nano version of YOLOv5 achieved the highest FPS detection scores. Bhagabati *et al.* (2024) developed an AI-based system to mitigate human-animal conflicts around Kaziranga National Park, integrating YOLOv5 with a SENet attention layer for real-time detection from live video feeds. The model achieved 96% accuracy in day-and-night conditions and improved reliability by 1-13% compared to previous approaches, enabling timely warnings and enhancing safety for both humans and wildlife.

(Guarnido-Lopez *et al.*, 2024; Ma *et al.*, 2024; Estevez *et al.*, 2023) explored the application of the YOLO algorithm for monitoring feeding behaviors in cattle by analyzing video data of six Charolais bulls and classifying behaviors such as biting, chewing and visiting using Roboflow. Performance comparison between YOLOv8 and YOLOv10 revealed similar overall accuracy levels of approximately 90%; however, YOLOv10 demonstrated superior precision, recall, mAP50 and mAP50-95 metrics. Chappidi and Sundaram (2024) introduced a cascaded YOLOv8-based detection framework incorporating adaptive histogram equalization, super-pixel-based Fast Fuzzy C-Means (FCM) segmentation and feature extraction using ResNet50, DarkNet19 and Local Binary Pattern. The MATLAB-based system achieved 97% accuracy and exhibited robust performance across metrics including kappa, precision, sensitivity, specificity and F-measures.

Despite notable advancements in YOLO-based wildlife detection systems, persistent challenges remain in

reducing false positives and false negatives, particularly under varying environmental and ecological conditions. Therefore, further evaluation across diverse contexts and the integration of robust hybrid methodologies are necessary to enhance detection reliability.

In this study, YOLOv8 is employed to track Amur tigers with a focus on improving detection accuracy across heterogeneous environments. A dataset comprising 4,413 images is utilized, supported by preprocessing techniques such as data augmentation and normalization. By addressing classification errors and minimizing false detections, the proposed approach aims to enhance the effectiveness of wildlife monitoring systems and contribute to improved conservation practices.

MATERIALS AND METHODS

Dataset description

In this work, the dataset is sourced from Kaggle and derived from the ATRW dataset (Pendharkar, 2024). The training and validation data were initially in PASCAL VOC format, which is incompatible with the YOLO algorithm. To ensure compatibility, the datasets were converted to YOLO Darknet format, as required by the Ultralytics implementation.

The dataset contains a total of 4,413 images of tigers. It focuses on the Amur tiger, also known as the Siberian tiger. This critically endangered population is found in the Russian Far East and Northeast China, with fewer than 600 individuals remaining in the wild. Capturing sufficient image data of these free-roaming tigers is challenging due to their elusive nature and low population density. This underscores the need for advanced technologies in monitoring and conservation efforts. A sample of annotated images is presented in Fig 1.

Data preprocessing

All images are resized to a consistent dimension, 640x640 pixels, to meet YOLOv8 requirements. To enhance the diversity of the dataset, data augmentation techniques are applied. It includes rotation, flipping and brightness adjustments. This helps improve the robustness of the model. Normalization of pixel values to a range between 0 and 1 is also performed for faster training convergence.

The dataset is divided into training, validation and test sets, in an 80-10-10 split. It is important to ensure that each set has a representative distribution of classes.

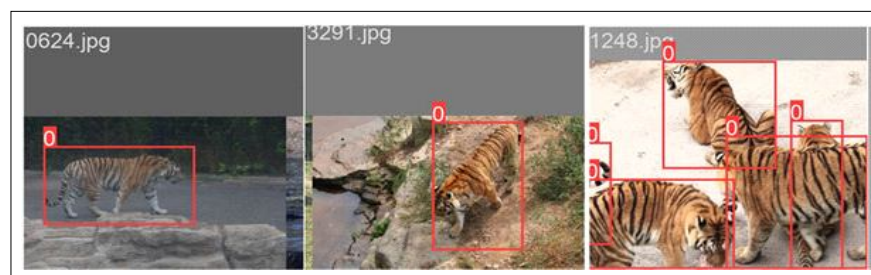


Fig 1: A sample of annotated images from the dataset.

After verifying the accuracy of the annotations, a YOLOv8 configuration file is created, detailing class names and image paths. Finally, a data-loading pipeline is established to handle image loading, preprocessing and batching during training. This thorough preprocessing ensures the model learns to accurately detect and track tigers in various environments. Fig 2 illustrates the step-by-step workflow of the proposed methodology, starting from dataset collection and preprocessing, followed by image annotation and YOLOv8 model training. The trained model is then validated and evaluated to ensure accurate and reliable wildlife detection performance.

Model architecture

The YOLOv8 model consists of three primary components: the backbone, neck and head.

- The backbone is responsible for extracting essential features from the input images. It utilizes a series of convolutional layers to process the images and create a rich set of features that will be used in the detection process.
- The neck Acts as a bridge between the backbone and head, it refines the feature representations. It enhances the resolution of features while simultaneously reducing the dimensions of the feature maps. This helps to optimize the information passed to the head for final object detection.

- The head of the YOLOv8 model is composed of three detection networks, each tailored to detect objects of different sizes: small, medium and large. This multi-scale detection capability allows the model to be versatile and adaptive, ensuring it can accurately identify tigers regardless of their size in the image. The block diagram of YOLOv8 is given in Fig 3.

Parameters and metrics

Learning rates (lr/pg0, lr/pg1, lr/pg2)

Learning rates are important hyperparameters that dictate the model's parameters adjusted during training. Setting the learning rate appropriately is vital for achieving effective convergence. If the learning rate is too high, the model may oscillate or diverge, while a rate that is too low can result in slow training and potentially getting stuck in local minima. Therefore, fine-tuning these rates is essential to maximize the performance of the tiger tracking model.

Features of the model

• Model/GFLOPs

The computational complexity of the YOLOv8 model is reflected in its GFLOPs (Giga Floating Point Operations per Second). This value indicates the number of floating-point operations the model can perform in a second.

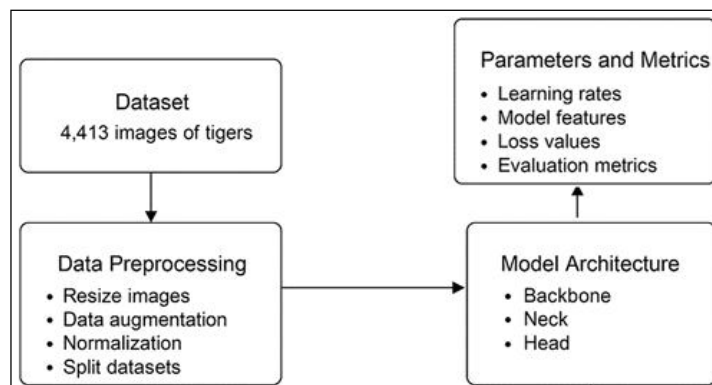


Fig 2: Flowchart of the YOLOv8-based wildlife detection methodology.

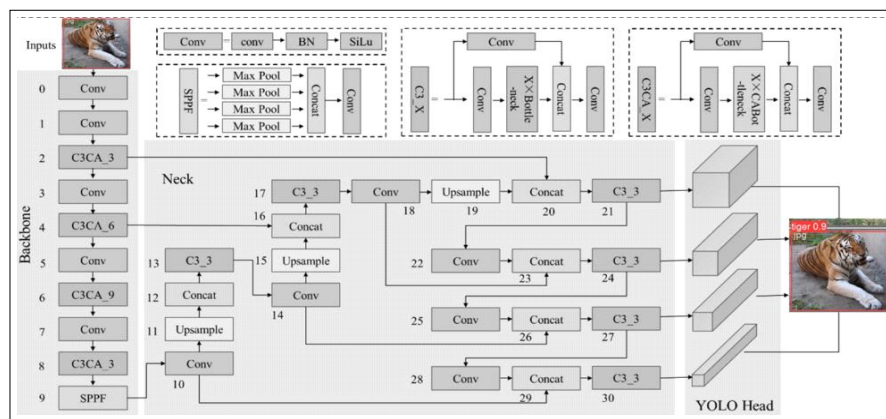


Fig 3: YOLOv8 block diagram for tiger detection (Reference: <https://docs.ultralytics.com/models/yolov8/>).

For real-time applications such as tiger tracking, a lower GFLOP value is advantageous, as it allows for faster processing and quicker decision-making.

• Model parameters

The number of parameters in the model indicates its complexity. While a higher number of parameters can enhance the model's ability to capture intricate details of the tigers, it also raises the risk of overfitting, where the model learns noise in the training data rather than generalizable features. Thus, finding the right balance is important.

• PyTorch model/speed (ms)

The processing speed of the model, measured in milliseconds, indicates its efficiency. For real-time tiger tracking applications, faster processing speeds are preferable as they enable the model to react quickly to changes in the environment, enhancing overall responsiveness.

• Loss values

Various loss values, including validation losses (val/box_loss, val/cls_loss, val/dfi_loss) and training losses (train/box_loss, train/cls_loss, train/dfi_loss), provide insight into the learning efficiency of model. Lower loss values during training and validation signify successful training and suggest that the model is effectively learning to generalize from the tiger tracking dataset. This efficiency is critical for ensuring the model performs well when applied to new, unseen data.

$$\begin{aligned}\text{Box loss} &= \frac{1}{N} \sum_{i=1}^N \text{IoU}(y_i, \hat{y}_i) \\ \text{Class loss} &= -\frac{1}{N} \sum_{i=1}^N y_i \cdot \log(\hat{y}_i) \\ \text{DFL} &= \frac{1}{N} \sum_{i=1}^N L_{\text{dfi}}(y_i, \hat{y}_i)\end{aligned}$$

Where,

IoU= The Intersection over Union between the true box y_i and the predicted box \hat{y}_i .

L_{dfi} = The loss function.

Evaluation metrics

To evaluate the performance of the YOLOv8 model in tiger detection, several key metrics are evaluated.

Mean average precision (mAP)

Specifically, we focus on mAP50-95(B) and mAP50(B). Mean Average Precision is a crucial metric in object detection that measures the accuracy of the model in localizing objects. Higher mAP values, especially within the 50-95% confidence interval, indicate improved tracking precision. This metric allows us to assess how well the model performs across varying confidence thresholds.

$$\text{mAP} = \frac{1}{C} \sum_{c=1}^C \text{AP}_c$$

Precision and recall

These metrics provide insights into the predictive performance of the model. Precision (B) measures the accuracy of the model's tiger predictions, indicating how many of the predicted tigers were correctly identified. Recall (B), on the other hand, assesses the model's ability to detect all actual tigers in the dataset. Achieving a balance between high precision and high recall is essential; high precision minimizes false positives, while high recall reduces the chances of missing actual detections.

$$\text{AP} = \sum_n (r_n - r_{n-1}) P_n$$

The parameters used for evaluation are summarized in Table 1.

RESULTS AND DISCUSSION

The results of the training process are presented in Fig 4, which provides insight into the model's performance over the epochs. Initially, both training and validation losses for box, class and distribution focal loss (DFL) show a downward trend, indicating that the model is learning effectively. For instance, the training box loss decreased from 1.5152 in the first epoch to 1.2224 by the twentieth epoch, while class loss reduced from 1.0523 to 0.61601. This decline signifies improved model accuracy in localizing and classifying objects.

Metrics such as precision, recall and mean Average Precision (mAP) also reveal significant progress [Fig 5 (a, b)]. Starting with a low precision of 0.07882 and recall of 0.13051 in the first epoch, these values improved substantially, reaching 0.9438 and 0.90257, respectively, by the twentieth epoch. The mAP metrics, including mAP @ 50 and mAP @ 50-95, further highlight the model's capability to detect objects across various thresholds. The mAP@50 increased from 0.03891 to 0.95817, illustrating a robust enhancement in detection performance.

The confusion matrix provides valuable insights into the model's performance [Fig 5(c)]. For the tiger class, the model identified 544 true positives. This indicates a strong ability to detect tigers in the dataset. However, there were also 38 false negatives, meaning the model missed 38 tiger instances. In the background category, the model recorded 18 false positives. This shows that it incorrectly classified 18 images as background when they were not. These results highlight the model's strengths and weaknesses. The high number of true positives reflects its effectiveness in detecting tigers. However, the false negatives suggest that the model could improve its sensitivity. Additionally, the false positives in the background category indicate a need for further refinement. Overall, the confusion matrix complements the earlier metrics.

This dataset included a diverse range of conditions, such as varying lighting and backgrounds. Fig 6 showcases a selection of these detections, highlighting the model's effectiveness in identifying tigers in different environments.

The present work shows good results compared to the existing literature. Dave *et al.* (2023) presented a deep-learning model designed to track wild animals in real time using camera footage. The study focused on detecting four animal categories: Lions, Tigers, Leopards and Bears. The authors created a dataset of 1,619 annotated images sourced from documentaries and YouTube videos. They trained three YOLOv8 models: medium, large and

extra-large. The extra-large model achieved a mean Average Precision (mAP) of 94.3% and could detect animals in real-time at 20 frames per second (FPS). Wu *et al.* (2024) proposed a method for identifying individual Amur tigers using an improved InceptionResNetV2 model. Initially, YOLOv5 detected and segmented facial and stripe areas from 107 tiger images, achieving 97.3% accuracy. Enhancements such as a dropout layer and dual-attention mechanism improved feature capture and reduced overfitting. The model reached an average recognition accuracy of 95.36%, with left stripes at 99.37%. This research provided a practical solution for identifying rare animals and supported conservation efforts. Ranëić *et al.* (2023) aimed to detect and count deer populations in northwestern Serbia using UAV images and deep neural networks. They compared several architectures, including three YOLO versions and a Single Shot Multibox Detector (SSD), trained on a manually annotated dataset. The best results showed a mean average precision of up to 70.45% and confidence scores of 99%. YOLOv4 achieved the highest precision (86%) and recall (75%), while its compressed version, YOLOv4-tiny, had a counting error of 7.1%. Prabhu *et al.* (2022) introduced RescueNet, a YOLO-based deep learning model designed for detecting and counting flood survivors in disaster-stricken areas. The model demonstrated high effectiveness, achieving a precision of 0.98, recall of 0.97, F1-score of 0.94 and mean average precision (mAP) of 98%. Senbagam and Bharathi (2024) aimed to develop a highly accurate object detection system using the YOLO algorithm for wildlife conservation.

Table 1: Parameters used to perform YOLO programming.

Parameter	Value
Model	YOLOv8m.pt
Python version	3.11.3
PyTorch version	2.1.1
CPU	11 th Gen Intel Core i5-11320H 3.20GHz
Task	Detection
Epochs	20
Batch size	16
Image size	640
Workers	8
Optimizer	AdamW
Learning rate (lr0)	0.002
Momentum	0.9
Weight decay	0.0005
Model parameters	2,58,59,899
GFLOPs	76.1
Final mAP50	0.944
Final mAP50-95	0.613

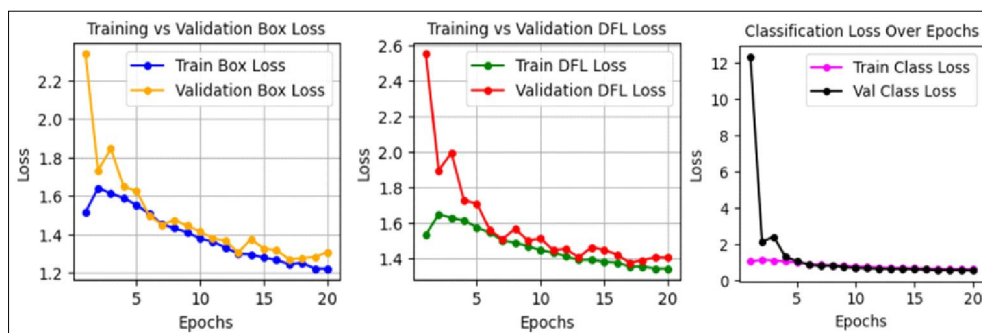


Fig 4: Loss measurement for training and validation.

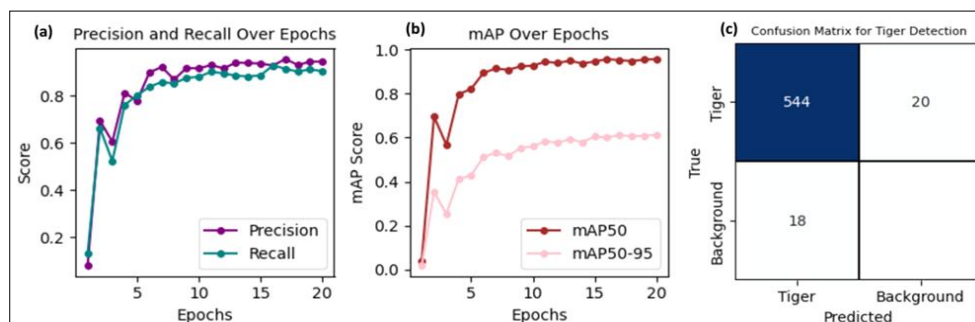


Fig 5: (a) Precision and recall measurements (b) mAP vs epochs (c) Confusion matrix.

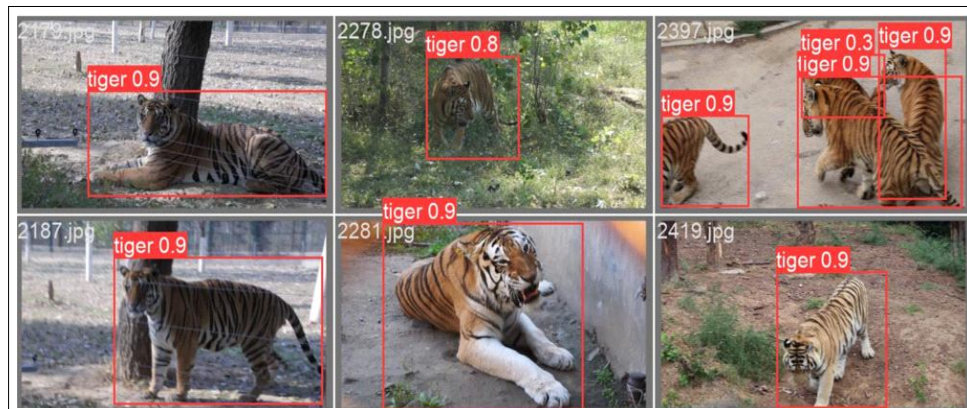


Fig 6: Output of YOLOv8-tiger detection method.

Table 2: Comparison of detection performance across different models.

Reference	Model type	Results	Detection categories
Dave <i>et al.</i> (2023)	YOLOv8m, YOLOv8l, YOLOv8x	mAP=94.3%	Lions, Tigers, Leopards, Bears
Wu <i>et al.</i> (2024)	Improved InceptionResNetV2	Average accuracy>95%	Amur tiger identification
Ranèiæ <i>et al.</i> (2023)	YOLOv4; YOLO4-tiny	YOLOv4-mAP: 70.45%, YOLOv4-tiny-mAP: 71%,	Deer
Prabhu <i>et al.</i> (2022)	RescueNet (YOLO-based)	Precision: 0.98, Recall: 0.97, F1-score: 0.94, mAP: 98%	Individuals and animals
Senbagam and Bharathi (2024)	YOLO-based object detection	mAP: 93.8%	Wildlife species
Naresh <i>et al.</i> (2023)	YOLO-based detection	Precision: 87%	Harmful snakes
This work	YOLOv8 detection method	mAP: 94.4%	Amur tiger identification

The study achieved a mean average precision (mAP) of 93.8% in detecting and identifying animal species under varying weather conditions. Naresh *et al.* (2023) developed a machine learning model using the YOLO algorithm to identify harmful snakes, aiming to help farmers recognize and avoid them. The algorithm achieved a precision of 87% in detecting snakes, facilitating real-time identification and improving safety in agricultural environments. Table 2 compares the performance of various object detection models reported in the literature with the proposed YOLOv8-based approach.

The presented work successfully used the YOLOv8 detection method to achieve an mAP of 94.4% for Amur tiger identification. The focus on this critically endangered species aids targeted conservation efforts. It also highlights the potential of advanced object detection technologies in wildlife monitoring.

Despite the strong performance demonstrated by the YOLOv8 model, certain limitations remain. The dataset size for Amur tiger detection is relatively limited, which may restrict the model's generalizability across broader geographic regions and more complex ecological scenarios. Variations in extreme lighting, heavy occlusion, dense vegetation and motion blur may also affect detection accuracy, as reflected in the observed false negatives. Future improvements will include expanding the dataset with more diverse real-world images, incorporating data augmentation techniques tailored for low-light and

occluded conditions and integrating temporal information from video sequences to enhance detection consistency. Additionally, exploring ensemble models or hybrid approaches combining YOLOv8 with attention mechanisms or transformer-based networks will further improve robustness and sensitivity, particularly for rare and cryptic wildlife species.

CONCLUSION

The YOLOv8 model successfully detected tigers in a variety of environments, demonstrating its potential for aiding wildlife monitoring and conservation efforts. With a final mAP of 0.944 and significant reductions in both training and validation losses, the model shows promise in accurately identifying tigers. However, limitations exist, including instances of false negatives and false positives that indicate areas for improvement. However, some limitations were noted in the model's performance. It had difficulty detecting tigers in low-light conditions, which led to missed detections. Moreover, the model sometimes confused other animals or objects with tigers. These challenges underscore the importance of refining the model to enhance its accuracy in varied and challenging environments. For future work, it is essential to explore advanced techniques such as ensemble learning or transfer learning to enhance model performance. Increasing the diversity of the training dataset could help the model better adapt to various conditions. Furthermore, incorporating more comprehensive

data, such as behavioral patterns of tigers, may improve detection rates. Addressing these challenges can significantly advance the effectiveness of detection technologies in supporting the conservation of the Amur tiger and similar endangered species.

ACKNOWLEDGEMENT

Disclaimers

The views and conclusions expressed in this article are solely those of the authors and do not necessarily represent the views of their affiliated institutions. The authors are responsible for the accuracy and completeness of the information provided but do not accept any liability for any direct or indirect losses resulting from the use of this content.

Funding details

This research was supported by the Korea Institute of Marine Science and Technology Promotion (KIMST), funded by the Ministry of Oceans and Fisheries (RS-2022-KS221676).

Data availability

The data analysed/generated in the present study will be made available from corresponding authors upon reasonable request.

Use of artificial intelligence

Not applicable.

Declarations

Author declares that all works are original and this manuscript has not been published in any other journal.

Conflict of interest

Author declares that they have no conflict of interest.

REFERENCES

- Altobeli, M.Z. and Sah, M. (2021). Tiger Detection Using Faster R-CNN for Wildlife Conservation. In: 14th International Conference on Theory and Application of Fuzzy Systems and Soft Computing-ICAIFS-2020. [Aliev, R.A., Kacprzyk, J., Pedrycz, W., Jamshidi, M., Babanli, M., Sadikoglu, F.M. (eds)], ICAIFS 2020. Advances in Intelligent Systems and Computing, Springer, Cham. pp. 1306. https://doi.org/10.1007/978-3-030-64058-3_71.
- AlZubi, A.A. and Alkhanifer, A. (2024). Application of machine learning in drone technology for tracking of tigers. *Indian Journal of Animal Research*. **58(9)**: 1614-1621. doi: 10.18805/IJAR.BF-1759.
- Bakana, S.R., Zhang, Y. and Twala, B. (2024). WildARe-YOLO: A lightweight and efficient wild animal recognition model. *Ecological Informatics*. **80**: 102541. <https://doi.org/10.1016/j.ecoinf.2024.102541>.
- Bhagabati, B., Sarma, K.K. and Bora, K.C. (2024). An automated approach for human-animal conflict minimisation in Assam and protection of wildlife around the Kaziranga National Park using YOLO and SENet attention framework. *Ecological Informatics*. **79**: 102398. <https://doi.org/10.1016/j.ecoinf.2023.102398>.
- Bhattacharya, S., Sultana, M., Das, B. and Roy, B. (2022). A deep neural network framework for detection and identification of bengal tigers. *Innovations in Systems and Software Engineering*. **20(2)**: 151-159. <https://doi.org/10.1007/s11334-021-00431-5>.
- Chappidi, J. and Sundaram, D.M. (2024). Novel animal detection system: Cascaded YOLOv8 with adaptive preprocessing and feature extraction. *IEEE Access*. **99**: 1-1. <https://doi.org/10.1109/access.2024.3439230>.
- Cho, O.H. (2024). An evaluation of various machine learning approaches for detecting leaf diseases in agriculture. *Legume Research*. **47(4)**: 619-627. doi: 10.18805/LRF-787.
- Gaurav, P. (2024). Tiger Detection Dataset (Accessed: 08 October 2024) Kaggle. <https://www.kaggle.com/datasets/gauravpendharkar/tiger-detection-dataset/data>.
- Dave, B., Mori, M., Bathani, A. and Goel, P. (2023). Wild animal detection using YOLOv8. *Procedia Computer Science*. **230**: 100-111. <https://doi.org/10.1016/j.procs.2023.12.065>.
- Estevez, J.R., Manco, J.A., Garcia-Arboleda, W., Echeverry, S., Pino, I., Acevedo, A. and Rendon, M.A. (2023). Microencapsulated probiotics in feed for beef cattle are better alternative to monensin sodium. *International Journal of Probiotics and Prebiotics*. **18(1)**: 30-37. <https://www.nchpjournal.com/admin/uploads/ijpp2641-7197v18n1-30-37.pdf>.
- Ferrante, G.S., Nakamura, L.H.V., Sampaio, S., Filho, G.P.R. and Meneguette, R.I. (2024). Evaluating YOLO architectures for detecting road killed endangered Brazilian animals. *Scientific Reports*. **14(1)**. <https://doi.org/10.1038/s41598-024-52054-y>.
- Guarnido-Lopez, P., Ramirez-Agudelo, J., Denimal, E. and Benaouda, M. (2024). Programming and setting up the object detection algorithm YOLO to determine feeding activities of beef cattle: A comparison between YOLOv8m and YOLOv10m. *Animals*. **14(19)**: 2821. <https://doi.org/10.3390/ani14192821>.
- Haldorai, A., R.B.L., Murugan, S. and Balakrishnan, M. (2024). A Modified AI Model for Automatic and Precision Monitoring System of Wildlife in Forest Areas. In: Artificial Intelligence for Sustainable Development. EAI/Springer Innovations in Communication and Computing. Springer, Cham. https://doi.org/10.1007/978-3-031-53972-5_25.
- Kim, S.Y. and AlZubi, A.A. (2024). Blockchain and artificial intelligence for ensuring the authenticity of organic legume products in supply chains. *Legume Research*. **47(7)**: 1144-1150. doi: 10.18805/LRF-786.
- Mamidi, K.K., Valiveti, S.N., Vutukuri, G.C., Dhuda, A.K., Alabdelli, H., Chandrashekar, R., Lakhanpal, S. and Praveen, N. (2024). An IoT-based animal detection system using an interdisciplinary approach. *E3S Web of Conferences*. **507**: 01041. <https://doi.org/10.1051/e3sconf/202450701041>.
- Ma, L., Zhang, M., Hao, X., Meng, R., Ma, Y. and Liu, J. (2024). Nutrition education improves the dietary habits and clinical outcomes of patients undergoing orthopedic care. *Current Topics in Nutraceutical Research*. **22(3)**: 980-985. <https://doi.org/10.37290/ctnr2641-452X.22:980-985>.
- Min, P.K., Mito, K. and Kim, T.H. (2024). The evolving landscape of artificial intelligence applications in animal health. *Indian Journal of Animal Research*. **58(10)**: 1793-1798. doi: 10.18805/IJAR.BF-1742.

- Naresh, E., Babu, J.A., Darshan, S.L.S., Murthy, S.V.N. and Srinidhi, N.N. (2023). A novel framework for detection of harmful snakes using YOLO algorithm. *SN Computer Science*. **5(1)**: 52. <https://doi.org/10.1007/s42979-023-02366-z>.
- Pendharkar, G., Micheal, A.A., Misquitta, J., Kaippada, R. (2024). An Efficient Illumination Invariant Tiger Detection Framework for Wildlife Surveillance. In: Communication and Intelligent Systems. [Sharma, H., Shrivastava, V., Tripathi, A.K., Wang, L. (eds)], ICCIS 2023. Lecture Notes in Networks and Systems, Springer, Singapore. **5**: 968. https://doi.org/10.1007/978-981-97-2079-8_14.
- Prabhu, B.V.B., Lakshmi, R., Ankitha, R., Prateeksha, M.S. and Priya, N.C. (2022). RescueNet: YOLO-based object detection model for detection and counting of flood survivors. *Modeling Earth Systems and Environment*. **8(4)**: 4509-4516. <https://doi.org/10.1007/s40808-022-01414-6>.
- Rančić, K., Blagojević, B., Bezdan, A., Ivošević, B., Tubić, B., Vranešević, M., Pejak, B., Crnojević, V. and Marko, O. (2023). Animal detection and counting from UAV images using convolutional neural networks. *Drones*. **7(3)**: 179. <https://doi.org/10.3390/drones7030179>.
- Ravoor, P.C., Sudarshan, T.S.B. and Rangarajan, K. (2021). Digital Borders: Design of an Animal Intrusion Detection System Based on Deep Learning. In: Computer Vision and Image Processing. CVIP 2020. [Singh, S.K., Roy, P., Raman, B., Nagabhushan, P. (eds)], Communications in Computer and Information Science, Springer, Singapore. pp. 1378. https://doi.org/10.1007/978-981-16-1103-2_17.
- Senbagam, B. and Bharathi, S. (2024). Animal Detection in Wildlife Conservation Using Deep Learning. In: ICT: Cyber Security and Applications. ICTCS 2022. [Joshi, A., Mahmud, M., Ragel, R.G., Kartik, S. (eds)], Lecture Notes in Networks and Systems, Springer, Singapore. **916**. https://doi.org/10.1007/978-981-97-0744-7_18.
- Wu, L., Jinma, Y., Wang, X., Yang, F., Xu, F., Cui, X. and Sun, Q. (2024). Amur tiger individual identification based on the improved inception ResNetV2. *Animals*. **14(16)**: 2312. <https://doi.org/10.3390/ani14162312>.