WILD ANIMAL DETECTION USING YOLOV8

For

Smart Engineering Project (G2M)

by

SAMEER PRASAD SAHOO (220301120309)

BEDAPR AKASH SWAIN (220301120319)

SUBHRANSU SEKHAR JENA (220301120320)

VISWARANJAN (220301120321)

Under the Supervision of

Mrs. Swarupa Pattanaik

SCHOOL OF ENGINEERING AND TECHNOLOGY
BHUBANESWAR CAMPUS
CENTURION UNIVERSITY OF TECHNOLOGY AND MANAGEMENT
ODISHA

APRIL,2025

CERTIFICATE

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

SCHOOL OF ENGINEERING AND TECHNOLOGY
BHUBANESWAR CAMPUS

BONAFIDE CERTIFICATE

Certified that this project report "WILD ANIMAL DETECTION USING YOLOV8" is the bonafide work of "SAMEER PRASAD SAHOO (220301120309), BEDPRAKASH SWAIN (220301120319), SUBHRANSU SEKHAR JENA (220301120320), VISWARANJAN (220301120321)" who carried out the project work under the supervision of Mrs. Swarupa Pattanaik. This is to certify that this project has not been carried out earlier in this institute and the university to the best of my knowledge.

SIGNATURE
(Mrs. Swarupa Pattanaik)
Assoc. Professor, Dept. of CSE

Certified that the project mentioned above has been duly carried out as per the college's norms and the university's statutes.

SIGNATURE

(Prof. Raj Kumar Mohanta)

HEAD OF THE DEPARTMENT

Professor of Computer Science and Engineering

DECLARATION

We hereby declare that the project entitled "**WILD ANIMAL DE- TECTION USING YOLOV8"** submitted for the "Smart Engineering Project G2M" of 6^{th} semester B. Tech in Computer Science and Engineering our original work and the project has not formed the basis for the award of any Degree or any other similar titles in any other University / Institute.

SAMEER PRASAD SAHOO (220301120309)
BEDAPRAKASH SWAIN (220301120319)
SUBHRANSU SEKHAR JENA (220301120320)
VISWARANJAN (220301120321)

Place: BHUBANESWAR

Date:

ACKNOWLEDGEMENTS

We wish to express our profound and sincere gratitude to **Mrs. Swarupa Pattanaik**, Department of Computer Science Engineering, SoET, Bhubaneswar Campus, who guided me into the intricacies of this project nonchalantly with matchless magnanimity.

We thank **Prof. Raj Kumar Mohanta**, Head of the Department of Computer Science and Engineering, SoET, Bhubaneswar Cam- pus, and **Prof. Sujata Chakrabarty**, Dean, School of Engineering and Technology, Bhubaneswar Campus for extending their support during the course of this investigation.

We would be failing in my duty if we did not acknowledge the cooperation rendered during various stages of image interpretation by **Mrs. Swarupa Pattanaik.**

We are highly grateful to **Mrs. Swarupa Pattnaik**, who evinced keen interest and invaluable support in the progress and successful completion of our project work.

We are indebted to my friends for their constant encouragement, cooperation, and help. Words of gratitude are not enough to describe the accommodation and fortitude they have shown throughout our endeavor.

Sameer Prasad sahoo (22030120309)

Subhransu sekhar Jena (220301120320)

Beda Prakash Swain (220301120319)

Viswaranjan (220301120321)

Place:

Date:

Abstract

Wild animal detection plays a crucial role in biodiversity conservation and wildlife monitoring. This project focuses on developing an automated system for detecting wild animals, including Indian waterbird species, using advanced deep learning techniques. Leveraging the YOLOv8 (You Only Look Once, Version 8) object detection model, the system is designed to identify and localize various species in real-time, ensuring high accuracy and efficiency.

The project workflow involves the collection of annotated datasets featuring diverse wild animal species, preprocessing the images to enhance model robustness, and training YOLOv8 to recognize specific species. Key performance metrics, such as mean Average Precision (mAP), precision, and recall, are used to evaluate the model's effectiveness. To address real- world challenges, the model incorporates optimization techniques for im- proved speed and reduced computational overhead, enabling deployment in resource-constrained environments.

This project demonstrates the potential of deep learning in supporting wildlife conservation efforts by enabling non-invasive and efficient monitoring of wild animal populations. By promoting biodiversity awareness, the system contributes to ongoing efforts to preserve India's rich ecological heritage and mitigate human-wildlife conflicts.

Table of Contents

List of Figures

List of Tables

CHAPTER 1
Introduction

Wild animals play a critical role in maintaining ecological balance, yet tracking and identifying them in their natural habitats remains a complex and demanding task. Monitoring wildlife populations is essential for conservation efforts, scientific research, and safety management. However, manually identifying wild animals from images is both time consuming and reliant on specialized knowledge. The similarities in appearance between different species, combined with environmental challenges such as variable lighting and complex backgrounds, make accurate identification even more difficult. To address these challenges, artificial intelligence provides an effective solution by enabling fast and accurate animal identification. Deep learning, particularly through models like YOLOv8 (You Only Look Once, version 8), has emerged as a powerful approach for this purpose. YOLOv8, optimized for object detection tasks, can analyze images of wild animals and identify them by recognizing distinct patterns in their features and surroundings.

Studies have demonstrated that deep learning models like YOLO often achieve high accuracy in animal detection, frequently exceeding 80 per- cent. This project aims to improve the accuracy of species identification in complex environments. By utilizing YOLOv8, an efficient animal detection system is being developed to classify and locate various wild animals across diverse habitats. The model will be integrated into a user-friendly web application, making it accessible to users—including researchers and conservationists—who may lack extensive field expertise. This tool is designed to facilitate wildlife monitoring, improve safety measures, and contribute to biodiversity conservation efforts.

CHAPTER -2

Literature Survey

The classification of wild animals holds particular importance in efforts to monitor biodiversity and protect ecosystems. In rural or forested areas, people, including local conservationists, often have practical knowledge of native wildlife. However, in urban areas and among field researchers who rely on data-driven insights, this knowledge may be limited, creating a gap in wildlife identification capabilities. This gap becomes significant as hu- man activities increasingly encroach on natural habitats, making it crucial to have efficient methods to track and identify wildlife. Recent advances in deep learning, particularly models like YOLO v8, offer promising solu- tions for accurate, real-time detection of wild animals. This study focuses on using YOLO v8 to classify four specific classes of wild animals, with 740 images for each class to ensure a balanced data set. YOLO v8, a re- fined version of the YOLO model family, excels at object detection tasks by processing images quickly and precisely - qualities essential for monitoring wildlife in dynamic or resource-constrained environments.

According to recent studies[1], YOLOv8 has shown promising advances in wildlife detection due to its rapid and accurate object recognition capa- bilities. Research has found that cascaded YOLOv8 models, paired with adaptive preprocessing techniques such as histogram equalization and fuzzy C-Means clustering, can achieve high accuracy in complex environments with dense foliage or low visibility. This approach enables more effective segmentation, overcoming challenges such as lighting variations and back- ground noise. By integrating additional feature extraction models, such as ResNet50 and DarkNet19, YOLOv8 outperformed previous detection techniques, achieving an impressive accuracy of 97 percent for wild animal detection tasks, underscoring its potential for reliable wildlife monitoring and conservation efforts.

According to recent studies [2], in wildlife detection through object de- tection algorithms. Two-stage methods like Cascade R-CNN have been effective but computationally intensive, limiting real-time application. One- stage algorithms, particularly various YOLO (You Only Look Once) ver- sions, are popular for their speed and accuracy balance, though recent stud- ies focus on further improving detection in complex environments with dense vegetation, low light, and occlusions.

The authors propose the YOLO-SAG model based on YOLOv8, incor- porating features like the Softplus activation function, an intra-scale fea- ture interaction mechanism (AIFI), and lightweight convolutions (GSConv and VoV-GSCSP) to enhance accuracy, speed, and robustness against envi- ronmental interference. These innovations address limitations in real-time wildlife detection and monitoring across diverse habitats and environmental conditions.

According to the findings [3], The YOLOv8 model surpasses other archi- tectures in wild animal detection tasks, particularly for endangered species monitoring. The study emphasizes YOLOv8's architecture improvements, such as CSPNet and PANet, which enhance feature extraction and pro- cessing efficiency, making it well-suited for diverse conditions in wildlife en- vironments. Previous research demonstrated that YOLO models achieved high accuracy for species classification, with models like YOLOv4 reaching up to 92.85 percent in animal classification tasks. YOLOv8 continues this trend, showing significant potential in fast, accurate detection for real-time applications.

According to the findings [4], The detection of wild animals, focusing on the classes of lions, tigers, bears, and leopards, using YOLOv8 models, including YOLOv8m, YOLOv8l, and YOLOv8x. This study leverages the concept of deep learning to extract features from images of wild animals. Additionally, data augmentation techniques are used to improve the training dataset, leading to robust model performance. With around 1619 images in the dataset, this work aims to enhance the YOLOv8 models for tracking and monitoring wild animals. Various detection strategies, including CNN-based classifiers and transfer learning techniques, have also improved detection accuracy. Prior approaches by researchers, such as the use of Faster R- CNN, SSD, and GANs, demonstrate diverse methods for animal detection under different conditions. Building on this, the paper applies YOLOv8 for detecting specific wildlife species, leveraging its robust performance and efficient training capabilities.

According to the findings [6], this study explores the use of deep learning models, particularly YOLOv8, Yolo-NAS, and Fast-RNN, for detecting wild animals that may pose threats to farm animals. Traditional farm monitoring methods have limitations, such as manual oversight and response delays, making it difficult to effectively protect livestock. Advanced models in object detection, like those in the YOLO series, provide high accuracy in identifying animals in diverse environments, offering a real-time, automated solution.

According to the findings [7],YOLOv8-based wildlife detection highlights the importance of object detection algorithms for wildlife conservation by addressing challenges posed by traditional monitoring methods. Traditional methods, such as GPS collars and manual surveys, are limited by their intrusive nature, resource demands, and inefficiencies. Camera traps offer a non-invasive alternative but generate a massive volume of data, often containing irrelevant images, which complicates manual processing. Object detection algorithms can automate wildlife detection with high efficiency.

Author	Methodology	Dataset	Accuracy
Dave et al. [1]	YOLOv8	1619 images from documentaries, YouTube, Kaggle datasets, re-labeled using Makesense.ai, with four classes: Lions, Tigers, Leopards, Bears.	94.3% (YOLOv8x)
Chappidi et al. [2]	Cascaded YOLOv8 with adaptive preprocessing, superpixel-based Fast Fuzzy C-Means (FCM) segmentation, and feature extraction with ResNet50, DarkNet19, Local Binary Pattern.	KAD dataset, Missouri Camera Traps (25,000 images of 20 species), WILD dataset (5,784 images of 28 species)	97% (KAD dataset), 98% (Missouri Camera Traps), 96.6% (WILD dataset)
Sharma et al. [3]	DenseNet, ResNet, VGG, YOLOv8	Custom dataset of 23 species, 1150 images sourced from iNaturalist and ZooChat; balanced by species	97.39% (YOLOv8 training), 99.13% (YOLOv8 validation)
Abraham et al. [4]	YOLOv8	2000 images across 20 wild animal categories from Kaggle	NA
Chen et al [5]	YOLO-SAG, an improved YOLOv8 model with Softplus activation, AIFI, and Slim-Neck for efficient wildlife detection.	Custom wildlife dataset with 11,348 images of 10 species from sources like Animal World documentaries and Snapshot Serengeti	mAP 0.5 Accuracy 96.8%, mAP 0.5 Accuracy 0.95: 79.9%
Korkmaz et al [6]	YOLOv8, Yolo-NAS, and Fast-RNN for detecting threats to farm animals using deep learning models.	2462 images of animal species from Kaggle, Roboflow, and Ultralytics, representing species that threaten farm animals	YOLOv8: Precision 93%, Recall 85.2%, mAP50 93.1%; Yolo-NAS: Precision 52%, Recall 98.7%, mAP50 93.1%; Fast-RNN: Precision 85.2%, Recall 91.8%, mAP50 91.2%
Jiang et al [7]	Enhanced YOLOv8 model with deformable convolutions (DCNv3), EMGA attention, ASPFC feature fusion, and Extended Kalman Filter for wildlife detection and tracking	17,169 augmented images featuring various animals in complex environments.	mAP 0.5, Accuracy 88.54%, MOTA: 40.35%
Zhong et al. [8]	YOLOv3, YOLOv5, and YOLOR (YOLO-based models) applied for real-time marine animal detection in underwater images	Custom dataset of marine animals in coral reef environments (fish and turtle classes)	YOLOv3: AP50=0.842 (fish), AP50=0.995 (turtle); YOLOv5: AP50=0.768 (fish), AP50=0.995 (turtle); YOLOR-W6: AP50=0.805 (fish), AP50=0.995 (turtle)

Table 1: Literature Survey

CHAPTER – 3

Proposed Method

This project focuses on designing an efficient model for classifying and detecting four classes of wild animals using the YOLOv8 architecture. The process begins with Dataset Preparation, where a curated dataset contain- ing images of various wild animal species is gathered, totaling 3,008 images. Four specific species Buffalo, Elephant, Zebra, Rhino are selected, each rep- resented by 752 labeled images. The images are organized into separate folders, one for each species.

By focusing on these four distinct species and ensuring each image is clearly labeled, the dataset is prepared to support the development of a YOLOv8 model that is both accurate and optimized for the detection of wild animals.

Figure 3.1: **Methodology**

The dataset for this project was sourced from Kaggle, where no pre existing working directories were provided. The YOLOv8 model dataset is struc- tured into image and label folders, each containing subfolders for testing, training, and validation. To prepare the data for use with the YOLOv8 model, working directories were created. Following this, the test images were extracted and visualized, and the dataset was divided into training and validation sets. The training set constitutes 80 percent of the data, while the validation set accounts for 20 percent, enabling the evaluation of the model's performance effectively.

The model selection and training process employs the YOLOv8 archi- tecture, a well-regarded convolutional neural network designed for efficient object detection. The "medium" variant of the YOLOv8 model (yolov8m) was selected for this project, as it offers a balance between accuracy and computational efficiency. Pre-trained weights (yolov8m.pt) derived from large-scale datasets were used to initialize the model. These weights pro- vide a robust baseline by equipping the model with general feature recogni- tion capabilities across various object types, facilitating faster learning and improved detection performance.

The YOLOv8 model was fine-tuned for the specific application using custom training data. This process involved defining a data configuration file, setting the configuration path, and selecting key training parameters. The model was trained for 100 epochs to allow sufficient time for learn- ing specialized features in the dataset. Parameters such as an Intersection over Union (IoU) threshold of 0.5 and a confidence threshold of 0.01 were carefully chosen to optimize object localization and precision. These thresh- olds help the model prioritize accuracy in detecting objects, reducing false positives while ensuring sensitivity even in challenging conditions.

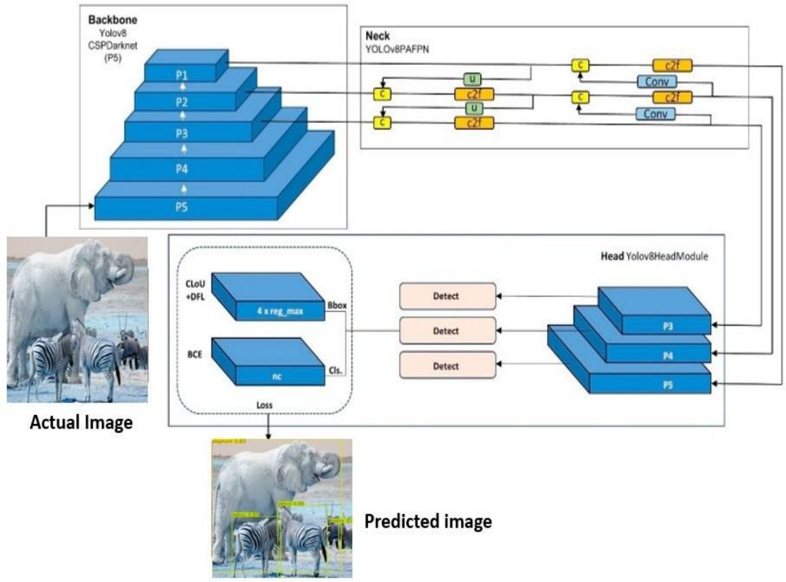


Figure 3.2: YOLOv8 Architecture

CHAPTER – 4

Result & Discussion

Using the YOLOv8 model, conducted a detection of wild animal detection across of animal dataset, focusing on four primary classes: buffalo, elephant, rhino, and zebra. The model trained over 100 epochs, resulting in a well- optimized weight file. Validation metrics demonstrated high mAP (mean Average Precision) scores, reinforcing the model's strong object detection and classification capabilities across different environments.

After 100 epochs, YOLOv8 achieved an impressive mAP50 of 0.994 on the training set, while mAP50-95 was recorded at 0.964. This high mAP re- flects the model's ability to detect and classify animals with precision across classes. Key metrics such as precision (P) and recall (R) were consistently high across all classes, with an overall precision of 0.996 and recall of 0.992. On the validation set, the model achieved a mAP50 of 0.959, indicating that it generalizes well. Class-level mAPs demonstrated that rhinos and buffaloes were detected with the highest accuracy, achieving scores above 0.97, while elephants and zebras had slightly lower scores. The model's performance on the test set was robust, achieving a mAP50 of 0.855. Pre- cision and recall values highlighted class-specific challenges, especially for buffalo and zebra, where complex backgrounds or overlapping features af- fected model accuracy. Nonetheless, the model handled difficult test images well, demonstrating resilience in identifying animal features under diverse conditions.

Certain classes, like buffalo and zebra, had some moderate error rates (with a precision of around 0.82 and a recall of around 0.74 for zebra). These errors happened mainly because of cluttered backgrounds and ob- stacles. For example, misidentifications often occurred when features like zebra stripes were partially blocked or blended into the background, mak- ing it harder for the model to correctly recognize them. Interestingly, the model performed excellently for elephants and rhinos on the test set, achiev- ing near-perfect recall and high mAP scores, suggesting that these classes had distinct features that the model learned effectively.

Observing the loss metrics during training, including box loss, class loss, and distribution focal loss (DFL), it was evident that they steadily de- creased, indicating successful optimization and minimal overfitting. This stability in loss metrics highlights that YOLOv8 efficiently captures animal- specific features without overly fitting to training images.

The YOLOv8 model demonstrated strong performance in detecting and classifying wild animals with high accuracy. Future model enhancements, like attention mechanisms and broader datasets, may help overcome current limitations and further improve detection capabilities. This project illus- trates the viability of AI-based animal detection models, underscoring their potential role in biodiversity preservation and ecological research.

4.1 Dataset & Experimental setup

In this project, the dataset was structured to train the YOLOv8 model for detecting and classifying four wild animal species—buffalo, elephant, zebra, and rhino. The dataset was initially organized into species-specific folders within a main directory. To meet YOLOv8's input requirements and ensure uniformity, all images were resized to a standard size of 640x640 pixels. Working directories were created to streamline the process, with separate paths for training, validation, and testing images, along with corresponding label directories. Each directory was further divided into subfolders for images and labels.
Files were organized and renamed based on their respective class labels to facilitate model training. For each class directory, label files were copied into the label training directory, while images in jpg or JPG format were resized using OpenCV and stored in the image training path. Files were renamed to include the species name as a prefix to allow easy identification of their class. Following preprocessing, the dataset was verified, resulting in 1504 images and 1504 label files within the training directory. This ensured a balanced distribution of data across the four species, providing a solid foundation for effective model training.

Each species folder contained 752 images, leading to equal-sized datasets for all classes. An 80:20 split was performed, allocating 80 percent of the images 1,189 for training and 20 percent 297 for validation. This structured configuration was designed to enable the model to learn diverse patterns effectively while maintaining consistency.

For the detection task, the YOLOv8 model was employed. This robust convolutional neural network architecture, optimized for object detection, was particularly suited to this application. The model was initialized with pre-trained weights from general-purpose datasets, leveraging transfer learning to reduce training time and enhance detection performance by building on prior knowledge of general image features.

Images were preprocessed by resizing them to YOLOv8's required in- put dimensions and normalizing pixel values, ensuring consistency across various image conditions and habitats. The training process involved 100 epochs with an IOU threshold of 0.5 and a confidence threshold of 0.01, parameters chosen to optimize precise localization and accurate classification while minimizing false positives.

The model's performance was evaluated using metrics such as precision, recall, and mean Average Precision (mAP). The best performing version of the model, achieving the highest validation accuracy, was saved as best.pt. This final model is ready for deployment for detecting and classifying wild animal species in diverse environmental conditions.

4.2 Performance Parameters

The performance of the YOLOv8 model in the wild animal dataset is mea- sured by key metrics used for its efficacy. Those include: Overall Precision, overall recall, and Mean Average Precision will be mainly used here. For the project at hand, a validation set accuracy of an impressive 92.7 percent was obtained from the data. The mAP of the model for all images in the train dataset is 0.994. The mAP of the model for all images in the val dataset is 0.959.

Precision: It tells us how well the model avoids false positives cases where it identifies something as an animal when it's actually not. Here, the model achieved an overall precision score of 0.95 across all animal classes, meaning it is good at correctly identifying each species without mistaking non-targets in the environment.

$$P_{all} = TP_t/(TP_t + FP_t) \quad (1)$$

True Positives (TP): Instances where the model correctly identifies a species (for example, the model detects a zebra, and a zebra is indeed present).

False Positives (FP): Instances where the model incorrectly identifies something as a particular species when it is actually not that species (for ex- ample, the model identifies an object as a zebra, but it is actually something else, like a shadow or another animal).

True Positives (TP) = 95 (correctly identified zebras). False Positives (FP) = 5 (incorrectly identified non-zebra objects). The precision would be calculated as:

$$Precision = 95/(95 + 5) = 95/100 = 0.95 \tag{2}$$

Recall:

It is a metric that quantifies how effectively a model identifies all relevant instances of a particular class (in this case, different animal species) within a set of images. It is calculated by comparing the number of correctly identified positive instances (True Positives) to the total number of actual positive instances (True Positives + False Negatives).

$$R_{all} = TP_t/(TP_t + FN_t \tag{3}$$

True Positives (TP): These are instances where the model correctly iden- tifies a specific species. For example, if the model successfully detects a zebra when a zebra is indeed present, it is counted as a True Positive.

False Negatives (FN): These are instances where the model misses identifying a specific species that is actually present. For instance, if there is a zebra in the image but the model fails to detect it, this counts as a False Negative.

True Positives (TP) = 91 (correctly identified zebras). False Negatives (FN) = 9 (zebras that were missed). So, the recall calculation would be:

$$Recall = 91/(91 + 9) = 91/100 = 0.91 \tag{4}$$

Mean Average Precision (mAP):

After calculating the AP for each class (for example, buffalo, elephant, etc.),Average Precision(AP) is calculated by taking the area under the precision-recall curve. we compute the mean of these AP values across all classes.

$$mAP = 1/n * sum(AP) \tag{5}$$

where n is the number of classes

Buffalo: AP = 0.966 Elephant: AP = 0.936 Zebra: AP = 0.949 To calculate mAP for these classes:

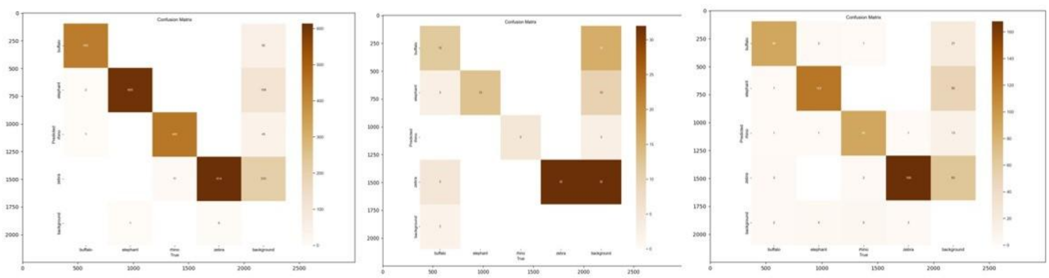
$$mAP = 1/3(0.966 + 0.936 + 0.949) = 2.851/3 = 0.950 \tag{6}$$

4.3Results

The results of the YOLOv8 model in detecting wild animal species were analyzed to assess its performance and ability to recognize four distinct species buffalo, elephant, zebra, and rhino. Several evaluation metrics were employed to gauge the model's effectiveness, including a confusion matrix, a classification report, and various graphical visualizations. These tools provided a comprehensive analysis of the model's detection capabilities, offering insights into its strengths and areas for improvement.

4.3.1 Confusion Matrix:

The confusion matrix you provided is a visualization of the performance of a machine learning model in classifying five categories: buffalo, elephant, rhino, zebra, and background.



Train Test Validation

Figure 4.1: Confusion Matrix

Training Confusion Matrix: This matrix shows how well the model performs on data it has seen during training. Darker cells along the diagonal indicate higher accuracy, suggesting the model has learned to recognize these classes. However, some misclassifications occur, seen in lighter non- diagonal cells, but overall, the diagonal dominance suggests strong class identification.

Testing Confusion Matrix: The test confusion matrix assesses the model's performance on unseen data. Here, while the model still accu- rately identifies many classes, some misclassifications are more noticeable compared to the training matrix, as expected due to unfamiliar data. Dark diagonal cells for background and specific animal classes show high accuracy, but some lighter cells in other areas indicate occasional incorrect predictions. **Validation Confusion Matrix:** This matrix reflects the model's per- formance on a separate validation dataset, typically used to fine-tune hyper- parameters or adjust the model. Similar to the test matrix, the model per- forms well overall, with diagonal dominance indicating correct predictions.

However, certain misclassifications appear, showing the model's limitations in consistently distinguishing between certain animal classes.

4.3.2 Graph and Visualization:

Additional graphical representations can provide deeper insights into the model's performance. Commonly used visualizations include loss curves, Precision-Recall curves for train, and validation which track the model's accuracy and loss over each

epoch during
learns, accuracy should increase while loss decreases, indicating effective training.

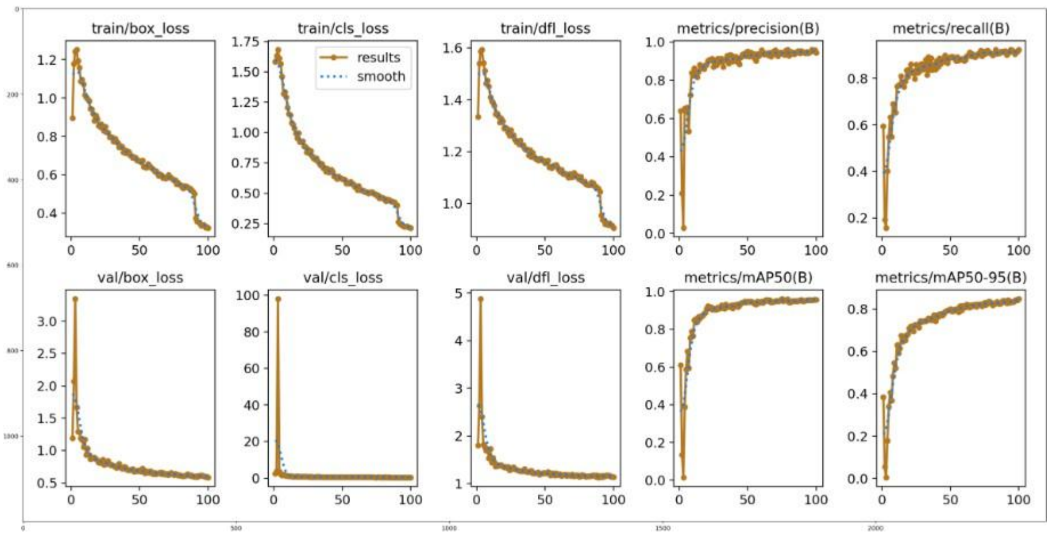


Figure 4.2: Loss Curves

This set of plots provides an overview of the model's training and valida- tion progress in wild animal detection. The first three graphs in the top row show the training losses for bounding box localization, classification, and distribution focal loss, all consistently decreasing, which indicates the model is improving in both localization and classification accuracy. Specif- ically, train/box-loss reflects better bounding box placement, train/cls-loss shows improved classification and train/dfl-loss enhances localization accu- racy. Together, these trends suggest the model effectively learns to detect and classify animals with increasing accuracy over time.
The top row also includes precision and recall metrics, which show near- perfect values as training progresses. High precision means the model ac- curately identifies true positives with fewer false positives, while high recall indicates that the model is capturing most of the actual positives, mini- mizing missed detections. Together, these metrics reflect strong detection performance during training.

The bottom row of plots shows validation metrics, assessing the model's performance on unseen data. The validation box loss, classification loss and distribution focal loss curves decrease steadily, indicating effective general- ization. The last two plots, metrics for mean Average Precision measure the model's precision across different IoU thresholds. High mAP values demonstrate the model's accuracy in localizing and classifying animals.

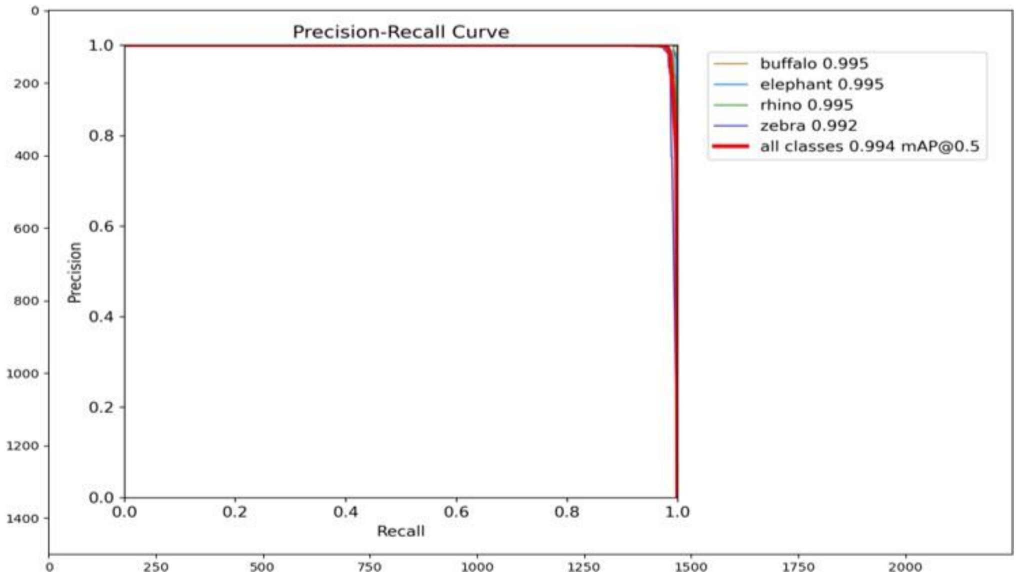


Figure 4.3 : **Precision-Recall Curve for train**

The Precision Recall (PR) curve displayed in the image represents the per- formance of a deep learning model trained to detect and classify four animal species: buffalo, elephant, rhino, and zebra. On the y-axis, the curve shows precision, which measures the accuracy of positive detections, while the x- axis displays recall, which measures the model's ability to detect all actual instances of each animal class. Each colored line represents one of the four animal classes, with high precision and recall values indicated by the tight clustering near the top-right of the graph. The legend indicates the aver- age precision (AP) for each class: buffalo (0.995), elephant (0.995), rhino (0.995), and zebra (0.992), all of which are very close to 1.0, signifying excellent accuracy. The red line represents the overall mean Average Preci- sion (mAP) across all classes. The near-ideal values in both precision and recall suggest that the model is highly effective at distinguishing between the four animal classes with minimal false positives or missed detections. This performance makes it well suited for reliable wildlife monitoring and conservation efforts.

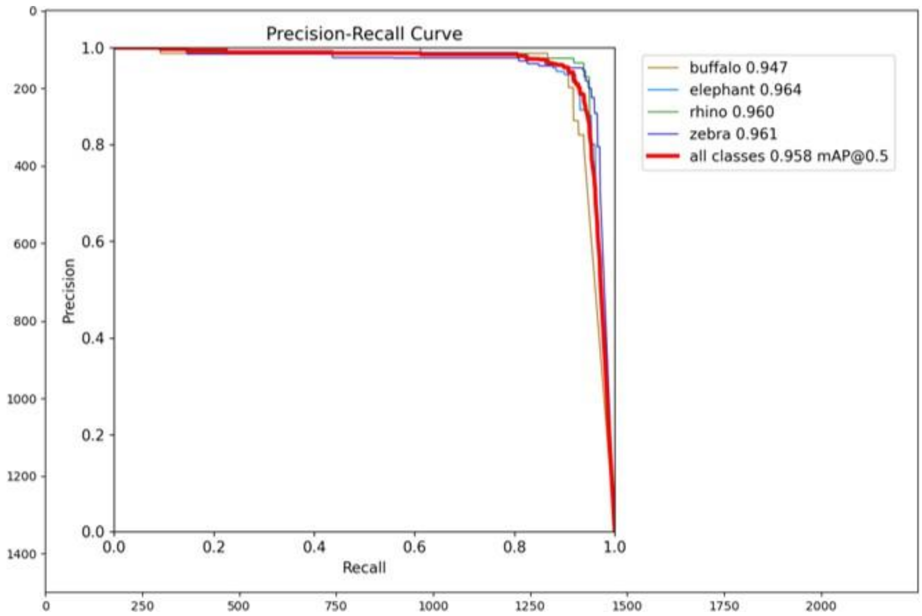


Figure 4.4: Precision-Recall Curve for Validation

The Precision-Recall (PR) curve shown here represents the validation per- formance of a deep learning model tasked with detecting and classifying four animal species: buffalo, elephant, rhino, and zebra. The y-axis shows precision, which measures how accurate the model's positive predictions are, while the x-axis shows recall, which assesses the model's ability to capture all relevant instances of each class. Each line represents one animal class, with the legend showing the average precision (AP) achieved for each: buf- falo (0.947), elephant (0.964), rhino (0.960), and zebra (0.961). These values indicate a strong performance in identifying each species, though slightly lower than perfect, suggesting some room for improvement in classification accuracy.

The red line represents the combined performance across all classes, with an overall mean Average Precision (mAP@0.5) of 0.958, meaning that, on average, the model has achieved 95.8 percent precision across all classes at a recall threshold of 0.5. This high mAP score demonstrates that the model performs well on the validation data, with high precision and recall, though with minor decreases in precision at the highest recall levels. This PR curve indicates that the model is effective in distinguishing between the four animal classes, making it suitable for wildlife monitoring applications.

CHAPTER – 5

Conclusion

In conclusion, the wild animal detection project using YOLOv8 has demon- strated impressive effectiveness and accuracy in identifying and classifying various animal species, including buffalo, elephant, rhino, and zebra. The YOLOv8 model, known for its speed and precision, proved highly suitable for real-time wildlife detection applications. With a high mean Average Precision score of around 0.958, the model showed reliable performance, achieving strong precision and recall for each target class. Analysis of the confusion matrix indicated minimal misclassifications, affirming the model's capability to differentiate between species accurately. The smooth and sta- ble loss curves further validated the model's learning efficiency and gener- alization ability without overfitting. This project highlights YOLOv8 as a robust tool for wildlife monitor- ing, capable of providing accurate and rapid animal detection results that can support conservation efforts and ecological research. Future work could focus on further enhancing the model's robustness by incorporating a more diverse dataset or employing data augmentation techniques to improve de- tection under challenging conditions such as low lighting or partial occlusion. Overall, YOLOv8's performance in this project underscores its potential for impact applications in automated wildlife detection and monitoring systems.

References

[1] Brahm Dave, Meet Mori, Anurag Bathani, and Parth Goel. Wild animal detection using yolov8. *Procedia Computer Science*, 230:100–111, 2023.

[2] Johnwesily Chappidi and Divya Meena Sundaram. Novel animal detec- tion system: Cascaded yolov8 with adaptive preprocessing and feature extraction. *IEEE Access*, 2024.

[3] Subek Sharma, Sisir Dhakal, and Mansi Bhavsar. Evaluating trans- fer learning in deep learning models for classification on a custom wildlife dataset: Can yolov8 surpass other architectures? *arXiv preprint arXiv:2408.00002*, 2024.

[4] Manna Mariam Abraham, Naveen Moncy Mathew, Richu Sakeer Hus- sain, Tima Jose Thachara, and Bibin Varghese. Wild watch sentry.

[5] Lingli Chen, Gang Li, Shunkai Zhang, Wenjie Mao, and Mei Zhang. Yolo-sag: An improved wildlife object detection algorithm based on yolov8n. *Ecological Informatics*, 83:102791, 2024.

[6] Adem Korkmaz, Mehmet Tevfik Agdas, Selahattin Kosunalp, Teodor Iliev, and Ivaylo Stoyanov. Detection of threats to farm animals us- ing deep learning models: A comparative study. *Applied Sciences*, 14(14):6098, 2024.

[7] Langkun Jiang and Li Wu. Enhanced yolov8 network with extended kalman filter for wildlife detection and tracking in complex environ- ments. *Ecological Informatics*, page 102856, 2024.

[8] Yujie Zhong, Xiao Li, Jiangjian Xie, and Junguo Zhang. A lightweight automatic wildlife recognition model design method mitigating shortcut learning. *Animals*, 13(5):838, 2023.