Animal Species Detection Using Deep Learning

Abstract

The alarming number of animals killed by vehicle collisions highlights the need for effective measures to mitigate roadkill and protect vulnerable animal populations. This paper focuses on the use of deep learning techniques, specifically the state-of-the-art YOLOv8 algorithm, for animal species detection. By developing an accurate and efficient model for detecting and classifying different animal species, this project aims to contribute to wildlife conservation efforts. This paper presents the methodology, including model training and evaluation, and provides insights into the performance of the YOLOv8 model. The dataset used for training, validating and testing the model consists of images of various animal species. The overall performance of the model was good, with a mean average precision (mAP50) score of 0.965. The results demonstrate the potentials of this approach in monitoring and studying animal populations, promoting the safety of both wildlife and humans, and informing wildlife conservation and management strategies.

1. Introduction

Roadkill[1], which refers to animals that are struck and killed by vehicles, is a significant concern for wildlife conservation globally. The alarming statistics reveal that over 1 million vertebrate animals in the United States [1] alone lose their lives due to vehicle collisions each day, with the global figure reaching around 5.5 million daily and exceeding 2 billion annually. Even in Pilanesberg National Park [2], car accidents claim the lives of numerous zebras and leopards annually. This issue has prompted a recent study [3] to identify animal populations at a heightened risk of extinction due to roadkill, including leopards (with an 83% increased risk), Brazilian wolves (with a 34% increased risk), Brazilian cats (with a risk ranging from 0 to 75%), and Southern African hyenas (with a risk ranging from 0 to 75%).

These wildlife vehicle collisions (WVCs) not only endanger animals but also pose risks to vehicle occupants and result in damage. To address this pressing issue, implementing measures such as highway fencing, wildlife crossings, escape routes, utilizing high beam headlights at night, and integrating animal sensors can significantly contribute to the safety of both wildlife and humans. This underscores the importance of projects like animal species detection using deep learning, which can aid in mitigating the impact of roadkill on vulnerable animal populations [4].

The paper presents a project focused on animal species detection using deep learning, specifically the YOLOv8 algorithm. The objective of the project is to develop an accurate and efficient model for detecting and classifying different animal species. The authors aim to contribute to wildlife conservation efforts by providing a reliable tool for monitoring and studying animal populations.

2. Literature Review

The detection and recognition of animal species is a crucial task in wildlife conservation and management. Deep learning techniques have been used to achieve automated object recognition and filtering of images. Ibraheam *et al.* used Convolutional Neural Network (CNN) architectures to train a system capable of filtering images from the British Columbia Ministry of Transportation and Infrastructure's (BCMOTI) wildlife program, and the Snapshot Wisconsin dataset. The system achieved 99.8% accuracy in indicating an object being animal or human, and 97.6% accuracy in identifying animal species.[5].

Other researchers have also used deep learning techniques for animal species detection. Villa et al. used a very deep CNN for species identification on

Snapshot Serengeti (SSe) dataset, achieving an accuracy of 88.9% in the evaluation set [6]. Chen et al. proposed a novel deep convolutional neural network-based species recognition algorithm for animal classification on standard camera trap dataset of 20 common species in North America. The method gave them 38.31% accuracy [7].

Other deep learning techniques were used for animal species detection. However, these techniques had some limitations. For instance, the Region-based Convolutional Neural Networks (R-CNN) and its variants such as Fast R-CNN and Faster R-CNN were computationally expensive and slow. Another technique, Single Shot Detector (SSD), was faster than R-CNN but had lower accuracy [8].

The YOLO algorithm was introduced to address these limitations. It is a real-time object detection algorithm that processes images very fast and has high accuracy [9]. The YOLO algorithm uses a single neural network to predict bounding boxes and class probabilities directly from full images in one evaluation [10]. This makes it faster and more accurate than other deep learning techniques for object detection.

Animal species detection and classification have become important tasks in the field of computer vision, with applications ranging from wildlife conservation to ecological research. Deep learning algorithms have shown remarkable success in addressing these challenges. Among them, the YOLO (You Only Look Once) algorithm has gained significant popularity due to its real-time object detection capabilities.

The YOLOv8 model used in this project is an advanced version of the YOLO algorithm, known for its high accuracy and efficiency. It leverages a deep neural network architecture to detect and classify animal species in images. The model's ability to process images in real-time makes it suitable for applications where quick and accurate analysis is required.

3. Methodology

3.1 Data Collection

Datasets play a crucial role in training deep learning models for animal species detection. For the implementation of YOLOv8 in animal species detection, three datasets were used for data collection.

The first dataset, sourced from Kaggle, contains 4 classes of African wildlife and all images are annotated in YOLO format. The images in this dataset have high image resolution, which is beneficial for accurate object detection.

The second dataset, also sourced from Kaggle, consists of approximately 10 classes of animals at risk of extinction. However, none of the images in this dataset are annotated. Most of the images have a low resolution of around 250 pixels, which is not ideal for object detection as it makes it difficult for the model to extract features. To address this, only the images with high resolution were selected and annotated using the Make Sense tool.

The third dataset, again sourced from Kaggle, contains around 80 classes of animals with high image resolution. All images in this dataset are annotated in Pascal format. From this dataset, only the classes available in the second dataset were chosen, and an additional class for tigers was included.

The final dataset used for the YOLOv8 implementation consists of 10 classes: buffalo, cheetah, elephant, fox, jaguar, lion, panda, rhino, tiger, and zebra. Four animals (buffalo, elephant, rhino, and zebra) come from the first dataset, while the remaining six animals (cheetah, fox, jaguar, lion, panda, and tiger) come from the second and third datasets.

By combining these datasets and selecting specific classes, a diverse dataset was created for training the YOLOv8 model in animal species detection.

3.2 Data Cleaning/preprocessing

The dataset underwent preprocessing to clean and standardize it. The steps included

- removing identical images that were mistakenly placed in the wrong folders and dropping the leopard class.
- Images not belonging to cheetahs or jaguars were also removed, and additional high-resolution images were added to compensate for the low number of images in the cheetah and jaguar classes.
- Low-quality images were filtered out, ensuring the dataset only contained high-quality images.
- Furthermore, annotations in Pascal format were converted to YOLO format for consistency across all classes.

These preprocessing steps improved the accuracy and reliability of the model during training and serving.

3.3 Exploratory Data Analysis and Visualization

Our dataset consists of a diverse range of animal species, including zebras, rhinos, pandas, lions, elephants, buffalos, foxes, tigers, cheetahs, and jaguars. It comprises a total of 2609 images, each accompanied by an annotation file. These annotations provide information about the bounding boxes surrounding the animals in the images, allowing us to train our deep learning model. The dataset is well-balanced, with each animal species having a substantial representation. The zebras and rhinos have the highest count with 376 images each, while the panda class has the lowest count with 91 images. It is worth noting that the majority of images contain only one or two objects, while a few images exhibit a higher number of objects.

	Class Label	Image Count
0	zebra	376
1	rhino	376
2	panda	91
3	lion	208
4	elephant	375
5	buffalo	375
6	fox	150
7	tiger	291
8	cheetah	194
9	jaguar	173

Figure 1: Table showing the different animal species and the image count

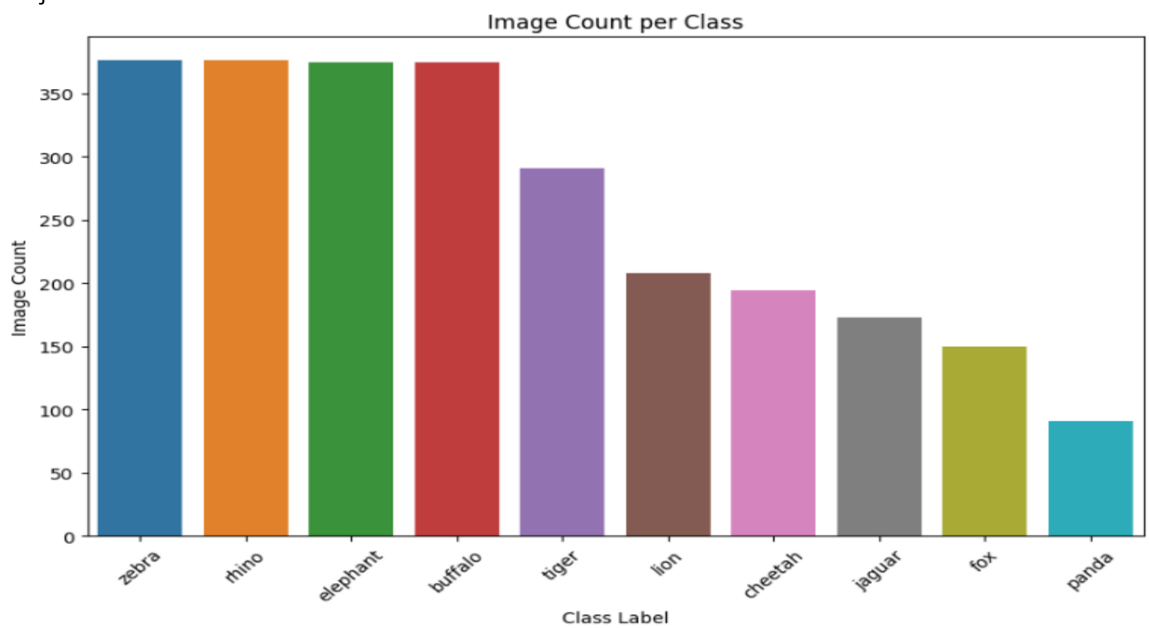


Figure 2: Bar plot of animal species according to the image count

3. 4 Data Modelling and Evaluation

The animal detection model was built using the You Only Look Once (YOLO) deep learning algorithm, specifically YOLOv8. YOLOv8 is known for its speed and accuracy, as well as its ability to train models for Object Detection, Instance Segmentation, and Image Classification within a unified framework. With YOLO, bounding boxes and class probabilities are predicted directly by a single network during a single evaluation. This simplicity enables real-time predictions.

The Dataset was split into train, validation and test sets in the ratio of 0.7, 0.15, 0.15 respectively. To build a new model, pre-trained weights were used with task argument as detect, mode as train, model as yolov8n. The model was trained for 50 epochs.

3.4.1 Model Training

The YOLOv8 model has two phases in each training cycle - training and validation. During these phases, it calculates the training and validation loss for bounding box and classification. In addition, it measures precision, recall, mean average precision *(mAP)* at 50% threshold and at different thresholds (mAP50-95).

The table above presents a summary of the performance of a YOLO v8 computer vision model that detects and classifies different animals in images.

The model has 168 layers and a total of 3,007,598 parameters, which are the values that the model learns during training to make predictions. The model was evaluated on a dataset of 243 images containing a total of 329 instances of animals.

Model summary (fused): 168 layers, 3007598 parameters, 0 gradients

Class	Images	Instances	Box(P	R	mAP50
all	226	302	0.944	0.915	0.95
Buffalo	226	32	1	0.85	0.945
Elephant	226	40	0.92	0.95	0.953
Rhino	226	32	1	0.952	0.994
Zebra	226	54	0.938	0.842	0.953
Cheetah	226	25	0.837	0.8	0.793
Fox	226	28	0.919	0.964	0.988
Jaguar	226	22	1	0.906	0.962
Tiger	226	24	1	0.956	0.958
Lion	226	29	0.908	0.931	0.956
Panda	226	16	0.916	1	0.995

Figure 3: Table showing a summary of the performance of a YOLO v8 computer vision model.

Visualizing the Training output

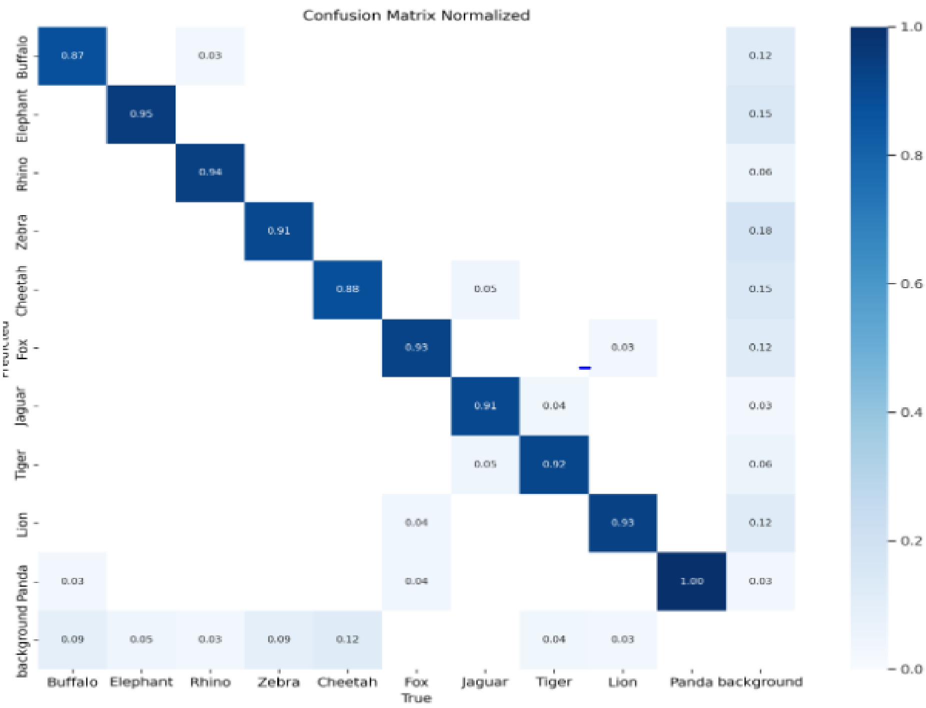


Figure 4: The normalized confusion matrix

The confusion matrix represents the performance of the deep learning model for animal species detection. It is a confusion matrix, where the diagonal elements (top left to bottom right) represent the correct predictions for each class, and the off-diagonal elements represent misclassifications.

In the context of animal species detection, the matrix shows how well the model is able to accurately identify and classify different animal species. Each row in the matrix corresponds to a true class, while each column corresponds to a predicted class. The values in the matrix indicate the number of instances that were correctly classified or misclassified for each class.

By analyzing the confusion matrix, researchers can gain insights into the strengths and weaknesses of the model.

Figure 5: Graphs showing the performance metrics

- **Metrics**

Model	Precision	Recall	F1-score	mAP@0.5	mAP@0.5:0.95
YOLOv8	0.944	0.915	0.93	0.95	0.804

The overall performance of the model was good, with a mAP50 score of 0.965. This indicates that the model was able to accurately detect and classify instances of the animals with high precision and recall. The high mean average precision (mAP) scores further demonstrate the model's effectiveness in identifying each animal in the images. The table(*Figure 5*) also reports the precision and recall scores for each animal class, which indicates how well the model was able to correctly identify instances of that class and avoid false positives

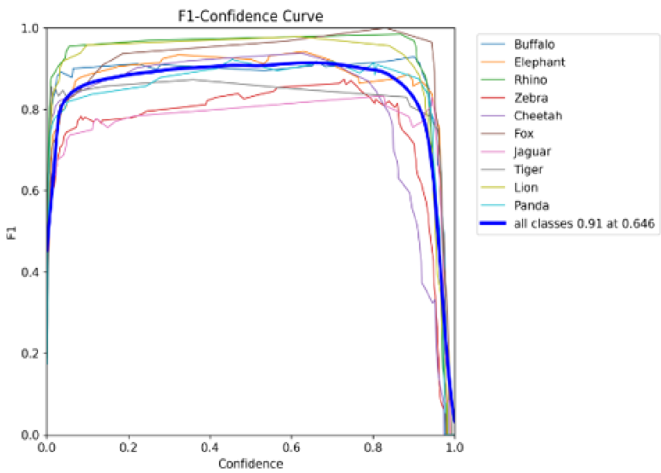


Figure 7: F1 confidence curve for train set

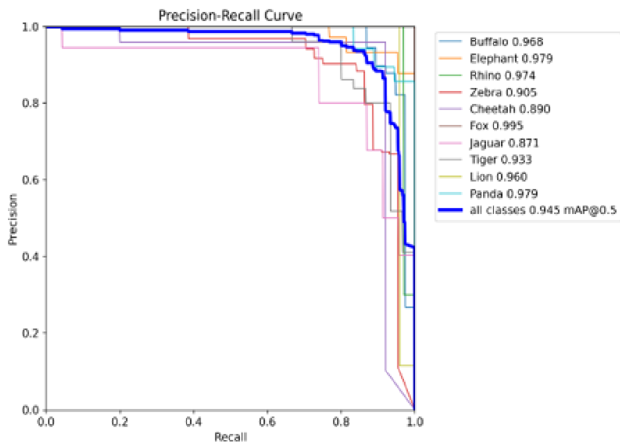


Figure 8: Precision-Recall curve for train set

3.4.2 Evaluation

The trained YOLOv8 model was evaluated using a separate test set to assess its performance in animal species detection. The evaluation metrics used included precision, recall, and the F1 score, which provide insights into the model's ability to correctly identify and classify instances of each animal class. Additionally, the mean average precision (mAP50) and mAP50-95 scores were calculated to measure the model's overall performance across different confidence thresholds.

To ensure unbiased evaluation, the test set was carefully selected to include a diverse range of animal species and challenging scenarios. The evaluation results were analyzed to identify the strengths and weaknesses of the model, including the classes that were accurately detected and those prone to misclassification.

Figure 8: The confusion matrix for test set

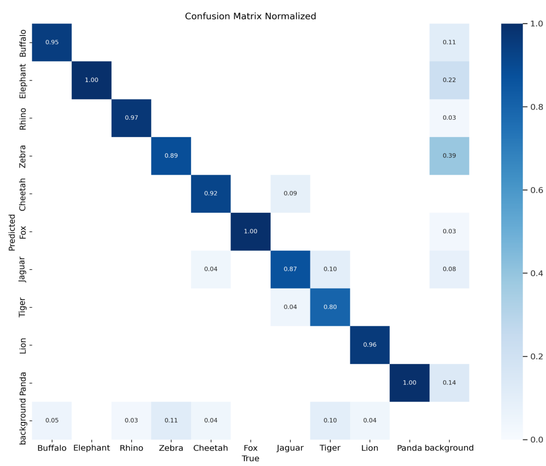


Figure 9: PR Curve for test set

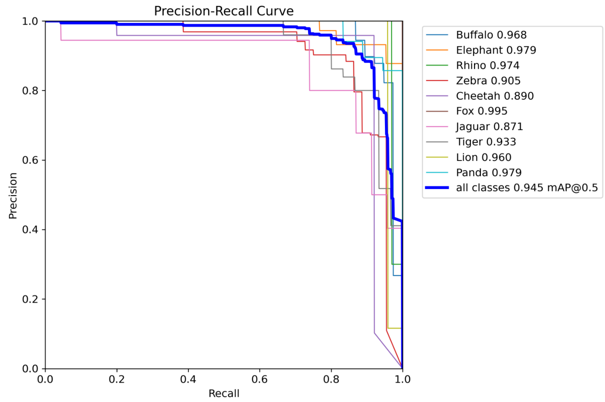
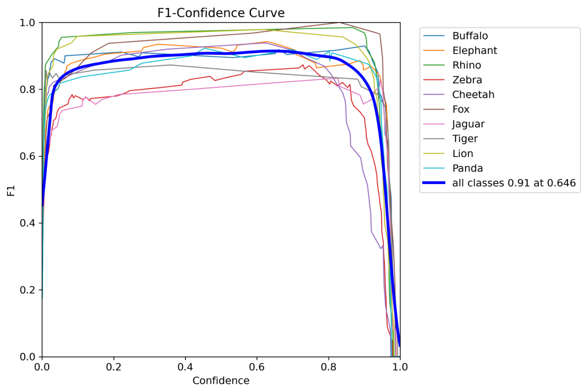


Figure 10*: F1 Curve for test set*



5. Model Deployment

Deployment of the model is done using ***Streamlit***, a user-friendly framework, and hosting it on ***Hugging Face***, a popular platform for sharing and deploying machine learning models. This allows users to easily access and utilize the trained model for their own applications. **Link**

6. Conclusion

Animal species detection using YOLOv8 provides a powerful solution for monitoring and studying animal populations, contributing to wildlife conservation efforts. By leveraging deep learning techniques, this approach enables accurate and efficient detection and classification of animal species, aiding in the mitigation of roadkill and promoting the safety of both wildlife and humans.

The result of the project on animal species detection holds significant importance in various domains, including wildlife conservation, ecological research, and environmental monitoring. Accurate and efficient

Prediction probabilities

In summary, the YOLOv8 model used for animal species detection performed well, with high precision and recall scores. These results demonstrate the effectiveness of the YOLOv8 algorithm in accurately detecting and classifying different animal species.

4. Results

Figure 11: Display of images with bounding boxes indicating class and

detection of animal species can provide valuable insights into biodiversity, population dynamics, habitat monitoring, and species distribution. It enables researchers, conservationists, and policymakers to make informed decisions and develop effective strategies for wildlife conservation and management.

7. References

[1] https://en.wikipedia.org/wiki/Roadkill

[2] https://www.natucate.com/en/blog/nature/south-africa-roadkills

[3] https://www.ucf.edu/news/new-study-shows-impact-of-roadkill-on-worlds-vulnerable-animal-populations

[4] Roadkill Alert, https://pilanesbergwildlifetrust.co.za/roadkill-alert/

[5] M. Ibraheam, F. Gebali, K.F. Li, and L. Sielecki: Animal species detection using deep learning. Advanced information networking and application (2020), pp. 523-532, doi: 10.1007/978-3-030-44041-1_47.

[6] A.G. Villa, A. Salazar, and F. Vargas: Towards automatic wild animal monitoring: Identification of animal species in camera traps using very deep convolutional neural networks. Ecological informatics, volume 41 (2017), pp.24-32, doi: 10.1016/j.ecoinf.2017.07.004.

[7] G. Chen, T. X. Han, Z. He, R. Kays and T. Forrester: Deep convolutional neural network-based species recognition for wild animal monitoring. IEEE International Conference on Image Processing (ICIP), Paris, France, (2014), pp. 858-862, doi: 10.1109/ICIP.2014.7025172.

[8] E. Zvornicanin. What is YOLO algorithm? (2023) June 13. https://www.baeldung.com/cs/yolo-algorithm

[9] Introduction to YOLO algorithm for object detection. Engineering Education (EngEd) Program. (2021) April 15 https://www.section.io/engineering-education/introduction-to-yolo-algorithm-for-object-detection/

[10] R. Kundu. YOLO: Algorithm for Object Detection Explained [+Examples]. 2023, January 17. https://www.v7labs.com/blog/yolo-object-detection