Automated biometric system for individual identification of Mugger crocodile (Crocodylus palustris): A Deep learning approach

Authors - Parth Joshi, Shubham Apat, Shubham Shah

***Abstract*-** In the field of animal behaviour and ecology, individual identification plays a critical role. Traditionally, invasive methods have been employed for this purpose. This study explores adapting a deep learning approach for non-invasive individual identification of free-ranging mugger crocodiles (Crocodylus palustris) utilising Unmanned Aerial Vehicle (UAV) imagery and the YOLOv8 deep learning model. A key aspect of this adaptation involves transitioning to the latest YOLOv8 model and an expansion of the dataset.

***Keywords* -** Convolutional neural network, Deep learning Free-ranging, Individual identification, Mugger crocodile, Unmanned aerial vehicle, Deep learning, YOLOv8, UAV imagery, Individual identification, Mugger crocodile

I. Introduction

Individual identification of animals plays a vital role in behavioural and ecological studies. Research by Desai et al. [1] used deep learning to identify muggers based on dorsal scute patterns instead of the traditional methods involving capturing the animals and tagging them. This approach achieved a True Positive Rate, TPR (re-identification of trained muggers) and a True Negative Rate, TNR (differentiating untrained muggers as 'unknown') of 88.8% and 89.6%, respectively. The trained model showed 100% TNR for the non-mugger species it was tested on. The images were captured using UAVs. In this study, we use a newer model - YOLOv8 and an expanded dataset, going from 143 to 160 individual mugger crocodiles.

II. Methodology

The original study[1] leverages existing research that employed YOLOv5l, a convolutional neural network (CNN) model, for individual mugger crocodile identification using UAV imagery. We aim to adapt this approach using the latest YOLOv8 model.

A. Dataset Preparation

The project began by merging an existing dataset of 143 individual mugger crocodiles with an expanded dataset containing 134 individuals. Each individual mugger has 125 images focused on their dorsal body region.

B. Computing Environment

Param Shavak, the supercomputer at Ahmedabad University, was utilised for training the YOLOv8 model. We configured a Python 3 environment suitable for YOLOv8 on the supercomputer.

C. Data Split

The combined dataset was divided into training, validation and test sets. The training set will be used to train the YOLOv8 model, and the validation set will be used to monitor the model's performance during training and prevent overfitting. The test set is used to measure the model's performance.

D. Yolov8 model:(YOLO-You Only Look Once)

Yolov8 is the latest SOTA model (state-of-the-art model) which can be used for object detection, image classification and instance segmentation tasks. It is built by Ultralytics.[2][3].

Yolov8 uses anchor box free prediction which means it directly predicts the centre of object instead of offset from anchor box. The YOLOv8 model has an advanced backbone and Neck architecture. The model we used is YOLOv8l for detection.

YOLOv8 employs a more powerful and efficient backbone architecture, specifically the Efficient Rep-Parameterized Convolution (ERPC) backbone. This new backbone architecture is designed to improve the model's performance while reducing computational requirements.

E. Difference between YOLOv8 and YOLOv5 model:

The YOLOv5 network, architecture comprises three essential components: (1) the Backbone, which utilises CSPDarknet for feature extraction, (2) the Neck, which employs PANet for feature fusion, and (3) the Head, represented by the YOLO Layer. The input data undergoes a sequence of processing, starting with feature extraction using CSPDarknet, followed by feature fusion through PANet. The final step involves the YOLO Layer, which outputs the detection results, including class labels, confidence scores, object locations, and sizes.

YOLOv8 represents a significant evolution from its predecessor, One of the most notable changes is the adoption of the Efficient Rep-Parameterized Convolution (ERPC) backbone, which replaces the traditional backbone architectures used in YOLOv5, such as CSPDarknet or CSPResNeXt. This new backbone architecture aims to improve the model's performance while reducing computational requirements. Moreover, YOLOv8 introduces an anchor-free approach, eliminating the need for predefined anchor boxes, which were a core component of YOLOv5's object detection methodology. This anchor-free approach allows the model to better adapt to various object sizes and shapes, potentially improving detection accuracy.

YOLOv8 offers various deployment options, including quantization and model pruning, to enable efficient inference on various hardware platforms, from high-end GPUs to mobile devices.

III. Results

The Model was trained for 150 epochs with a batch size of 16 and the image size was kept for all images as 640*640 pixels. The GPU which was used to compute the results is NVIDIA Quadro RTX 5000.

The mean average precision at IoU (Intersection over Union) threshold of 0.50 (mAP@50) was 0.964. The mAP@50-95 (mAP across IoU thresholds from 0.5 to 0.95) was at 0.755. The metrics precision value was found out to be 0.865.

The model was able to classify the images as per the given class. Below are some of the results of the classification process.

Figure 1

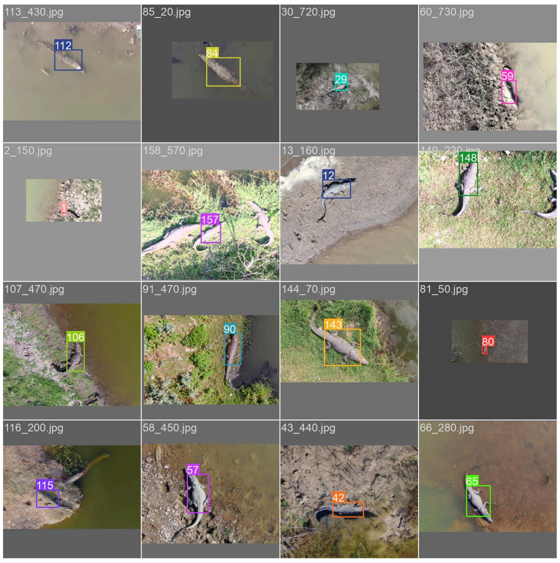
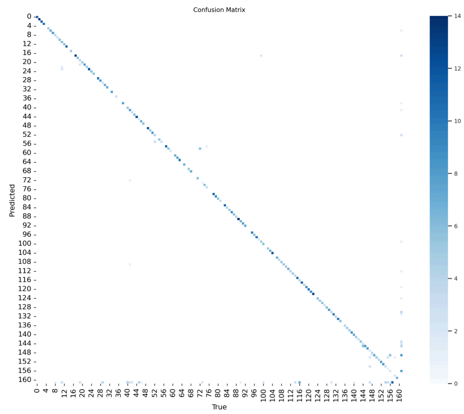


Figure 2

The above images show how the mugger crocodiles are detected. The bounding box is formed around the dorsal patterns and then the classification based on which class it belongs to from 1 to 160. Then the number of the class which it belongs to is itself shown on the bounding box.

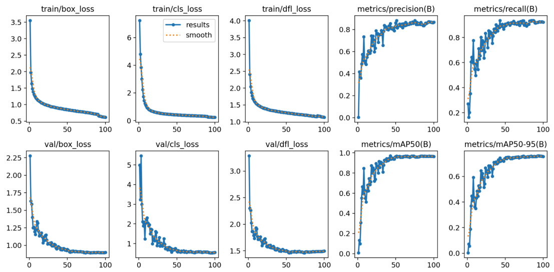


Confusion matrix

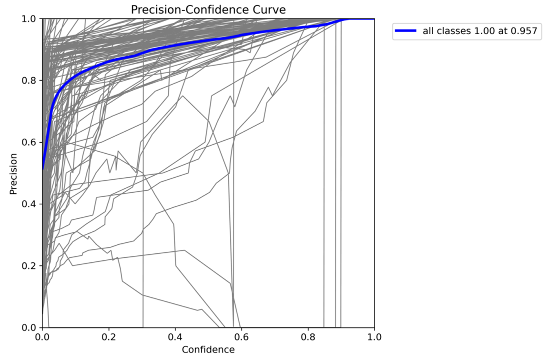
This confusion matrix displays the predicted values on the y-axis and the true values on the x-axis. The diagonal line represents perfect predictions, where the predicted value matches the true value.

The matrix shows a decreasing trend, with most of the predictions concentrated towards the lower end of the true value range. This suggests that the model tends to underestimate or underpredict the true values, particularly for higher true values.

The histogram on the right side displays the frequency of each predicted value. It shows that the most common predicted value is around 14, followed by 12 and 10.



precision(B): This metric shows some fluctuations but generally increases over time, indicating improving precision performance.
recall(B): Similar to precision, the recall metric also improves as training progresses.
mAP50(B): This metric, likely mean Average Precision with an IoU threshold of 0.5, starts low but steadily increases, suggesting better object detection/localization performance.
mAP50-95(B): This metric, probably mean Average Precision averaged over multiple IoU thresholds (0.5 to 0.95), also improves consistently during training.



The blue line represents the precision values for all classes combined. The gray lines likely correspond to individual classes or subsets of classes.

The annotation "all classes 1.00 at 0.957" highlights that for the combined set of all classes, the model achieves a precision of 1.0 (perfect precision) at a confidence threshold of 0.957. This means that if we only consider predictions with a confidence score above 0.957, the model's predictions will be 100% precise or accurate.

IV. Discussions

The process of retraining our dataset on YOLOv8, after initially training on YOLOv5, presented several challenges that required careful consideration and adaptation. To address these changes, we had to preprocess our dataset to align with the new requirements of YOLOv8. This involved adjusting the annotation format, scaling, and other preprocessing steps to ensure compatibility with the anchor-free approach.

Throughout the retraining process, we encountered several challenges related to computational resources and hardware limitations. The advanced architectures and training strategies employed in YOLOv8 often required significant computational power, which led us to explore various optimization techniques and deployment strategies.

V. Conclusion

The transition from YOLOv5 to YOLOv8 for object detection and instance segmentation tasks represents a significant leap forward in terms of architectural advancements, training strategies, and deployment optimization. Throughout this project, we successfully retrained our dataset on the YOLOv8 model, leveraging its improved backbone architecture, anchor-free approach, and advanced data augmentation techniques.

By adapting our dataset and preprocessing pipelines to align with YOLOv8's requirements, we were able to take advantage of the model's enhanced accuracy and robustness.

References

[1] - Desai, B., Patel, A., Patel, V., Shah, S., Raval, M. S., & Ghosal, R. (2022). Identification of free-ranging mugger crocodiles by applying deep learning methods on UAV imagery. *Ecological Informatics*, *72*, 101874.

[2] - Jocher, G., Chaurasia, A., & Qiu, J. (2023). Ultralytics YOLO (Version 8.0.0) [Computer software]. https://github.com/ultralytics/ultralytics

[3] - Glenn Jocher, Ayush Chaurasia, Alex Stoken, Jirka Borovec, NanoCode012, Yonghye Kwon, Kalen Michael, TaoXie, Jiacong Fang, imyhxy, Lorna, 曾逸夫 (Zeng Yifu), Colin Wong, Abhiram V, Diego Montes, Zhiqiang Wang, Cristi Fati, Jebastin Nadar, Laughing, … Mrinal Jain. (2022). ultralytics/yolov5: v7.0 - YOLOv5 SOTA Realtime Instance Segmentation (v7.0). Zenodo. https://doi.org/10.5281/zenodo.7347926

[4]- Ijraset. (n.d.). *A comparative analysis of YOLOV8 and YOLOV5 for nut thread Classification - Deep Learning approach*. IJRASET. https://www.ijraset.com/research-paper/comparative-analysis-of-yolov8-and-yolov5