REFACTORY UGANDA LIMITED
GROUP 18 CAPSTONE PROJECT REPORT

Otim Brian, David Wagalinemera, Ssengendo Collin Linus,
David Ofoyrwoth, Raymond Ayebazibwe, Oguru Mathew Amyas

June 17, 2024

Abstract

The primary objective of this study is to create a machine learning model for sea turtle face detection. This model should be accurate and efficient so as to leverage machine learning techniques to detect the bounding box around a sea turtle's face to aid in sea turtle conservation efforts and preserve marine life ecosystems and bio diversity.

1 Introduction

Artificial Intelligence and Machine Learning are taking the world by storm. It has revolutionized many aspects of our lives as well as various domains, such as computer vision. A key application of these technologies includes the detection and recognition of images and videos, not to mention objects within them. Application areas for these technologies range from surveillance and security to augmented reality and human interaction with computers.

In this report, we present an AI/ML model used specifically for the detection of bounding boxes around the faces of turtles. Despite the fact that such algorithms already exist for humans and some animals, turtles present a unique set of distinct challenges. These range from different species to individual variations, varying lighting backgrounds causing the need for a specialized approach to produce reliable results for accurate detection.

Our model leverages deep learning architectures and specially selected datasets of turtle images, through the iterative process of testing and training to achieve an algorithm capable of quickly and accurately identifying turtle faces in different contexts that their images may have been taken. The resulting bounding boxes of the turtle faces can also serve as a means of age estimation, species recognition, among others.

2 Methodology

The graphic below shows the steps involved in each major category to be considered when developing and training the Machine Learning model for sea turtle face detection. It also served as a guide for the procedure to be followed when creating the model for each member of the team that contributed to the project.

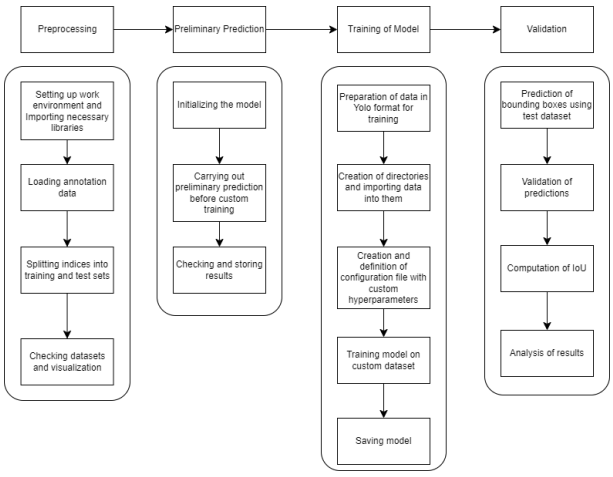

Figure 1: Steps involved when developing AI/ML model

Any computer vision project starts with effective data collection and annotation strategies. Data quality impacts model performance so it is important to understand and employ the the best practices for this purpose. Every consideration for the data should align with the project goals lest it shifts the focus or effectiveness and vice versa.
Ethical considerations and permissions are a crucial aspect of data handling and some key points considered were transparency, data minimization, purpose limitation, data security, Anonymization and De-identification. The methodology involved several structured steps to ensure a systematic approach to developing an accurate model.

2.1 Data Collection and Processing

First, data cleaning was performed to ensure the dataset was free from errors and inconsistencies. This involved loading all images of sea turtles, removing duplicates, checking annotations for accuracy, and handling any missing image data viz a viz the indices in the target feature data.

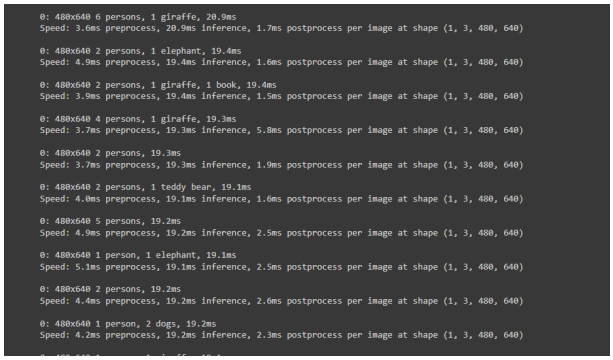

Figure 2: Inference occuring with new data

2.2 Data Split; Test and Train

Next, the dataset was split into training and test sets to evaluate the model's performance effectively. The data initially consisted of images of two sizes: 512 by 384 pixels and 1028 by 768 pixels. A split ratio of 80 to 20 for the training and test datasets was defined, and the images were randomly shuffled and split, maintaining the same distribution of classes to provide a balanced evaluation.

2.3 Data Annotation

Here, data annotation is used to label the data such that it is usable for training the machine learning model. Without annotation, the model can't accurately learn the relationship between inputs and outputs.
To verify the correctness of annotations, the data was visualized with bounding boxes on the images. This involved loading a subset of images and their corresponding bounding box annotations, drawing bounding boxes using libraries like OpenCV and matplotlib, and displaying the images for manual verification.
Our model uses YOLOv8 which is designed for seamless integration into the model for object detection, segmentation, and classification. The ease of use of the Python Interface is a valuable resource that facilitates this incorporation. The pre-trained YOLOv8m model was then imported into the project as a starting point. Ensuring all necessary libraries were installed, the model was loaded using appropriate library functions.
Using the base model, predictions were made, restricted to detecting only sea turtle faces. A batch of test images were then loaded, the model configured, and the predictions made. To evaluate prediction accuracy, the Intersection over Union (IoU) was computed. This was done by extracting predicted and ground truth bounding boxes, implementing an IoU function to calculate the overlap between them, and calculating the average IoU for the test set to assess model performance.

2.4 Model Training

The train mode is used for training a model on a custom dataset. Here, the model is trained using the specified dataset and hyper parameters. The training process involves optimizing the model's parameters so it can accurately predict the classes and locations of objects in an image.
Preparing the training data in YOLOv8m format was essential for effective model training. The annotations

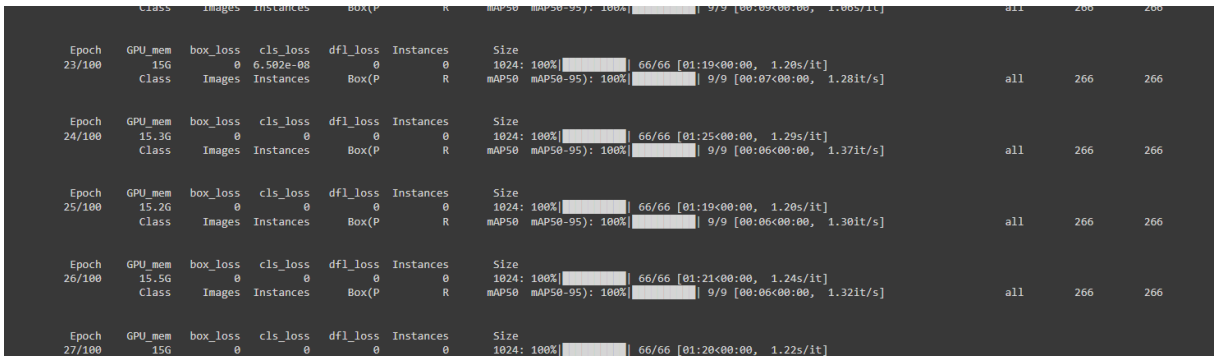

Figure 3: Model Training using epochs

were converted to the required YOLO format (x-center, y-center, width, height), images and annotation files organized in their respective directories, and a data configuration file created specifying paths to training and validation data.
The model was then trained with custom data. The training parameters such as batch size, epochs, and learning rate were set, then the training process initiated. The loss and accuracy metrics were monitored and the model hyperparameters adjusted appropriately till the model converged, and then the resulting model saved. Using the trained model, predictions of the bounding boxes were made and saved in csv format for further evaluation. In the evaluation, IoU was computed basing on predicted bounding boxes and those defined in the test set. The average IoU was also evaluated to assess the improvements.

Figure 4: Bounding Boxes around sea turtle's faces

2.5 Model Validation

Finally, to refine the model until good results were acquired, the hyperparameters were tuned further and more data used. Validation mode is used for validating a model after it has been trained. In this mode, the model is evaluated on a validation set to measure its accuracy and generalization performance. This mode can be used to tune the hyperparameters of the model so as to improve its performance.
The key hyperparameters with the biggest impact on results such as learning rate and batch size were identified,

and experiments conducted by adjusting one hyperparameter at a time and retraining the model. The resulting performances were then evaluated using IoU and other metrics, and refining continued until satisfactory results were achieved. This ensured development of a robust model for prediction of bounding boxes of sea turtle faces.

2.6 Predict and Export

Here, predictions are made using a trained model on new images or videos. The model is loaded from a checkpoint file and the user can then provide images to perform inference. The model predicts the classes and locations of objects in the input images.
This mode is used to export the model to a format that can be used for deployment. The model is then converted to a format that can be used by other software applications or hardware devices. This is particularly useful when deploying the model to production environments.

3 Challenges Faced

- Variability in Appearance. Sea turtles exhibit all shapes, sizes and textures on their faces which makes it difficult to effectively and successfully deploy the model across the varying species and environments that we find them in.

- Environmental factors. Images of turtles taken underwater are not always of the highest quality and this can be as a result of different lightning conditions, occlusions such as algae, sand, among others. These factors can distort the appearance of the turtle's face, hindering the performance of the model.

- Scale and Perspective. Images of sea turtles can be taken from different perspectives eg. landscape, portrait. This creates a challenge of accurately detecting a turtle's face across these variations.

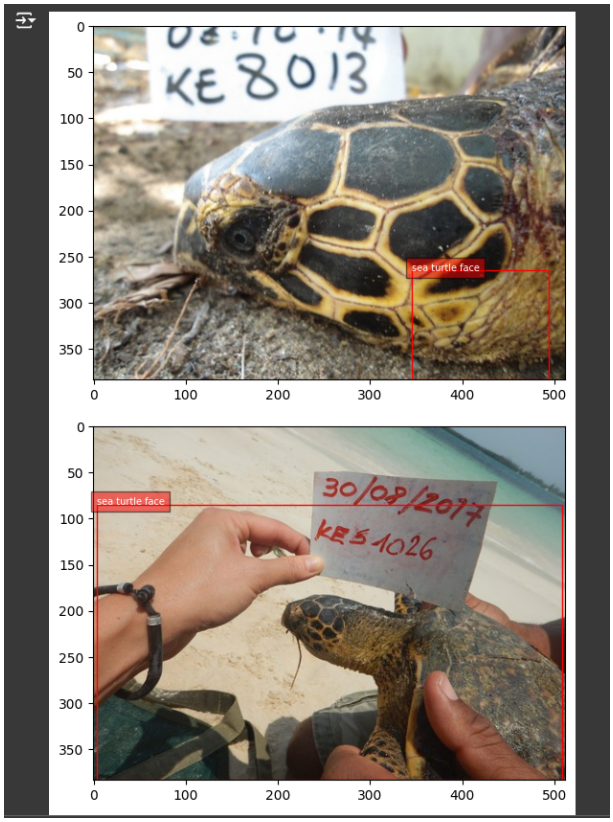

Figure 5: An example of the challenge faced with scale and perspective

- Inconsistency with Annotation. When annotating sea turtle's faces, images require consistency and accuracy to reliably train data used in deploying successful machine learning model.s This variability in annotation can impact the performance of the model.

4 Lessons Learned

- **Data Augmentation**. It is important to enhance variability by applying augmentations such as rotations, scaling to make the model more adept to different scenarios and improve its performance with unseen data.

- **Model Selection.** A key point to stress here is choosing the right architecture when developing your Machine Learning model. One can start with pre-trained models like YOLO, as seen in our case, SSD, among others specifically for object detection purposes and fine tune them depending on your dataset.

- **Training Process.** Hyperparameter Tuning is useful when experimenting with different batch sizes, learning rates to find the best training configuration for our model. Regularization techniques like data augmentation also help prevent over fitting.

- **Evaluation and Validation.** Splitting the dataset into training, validation and test sets ensures that each set represents a section/part of the overll dataset. Relevant metrics such as Intersection over Union (IoU) can then be used to get an insightful and comprehensive understanding of the model's performance.

- **Deployment Considerations.** The model should be optimized for inference in terms of speed and usage of resources especially for devices with limited computational power. Once implemented in the real world, the should be retrained periodically with new data to maintain its accuracy and relevance.