Università degli Studi di Padova

Dipartimento di Ingegneria dell'Informazione

Artificial Intelligence Report

Sea Turtle Face Detection for Ocean Conservation

Author: Alberto Formaggio

1 Objectives

The identification of individuals is a crucial task for animal conservation. In the case of sea turtles, tracking where and when they are spotted can help reveal patterns of movement and residency, and allow more accurate estimates of the population. Nowadays, this is done thanks to artificial tags that are attached to the animals: a technique that, unfortunately, is both time demanding and difficult due to the high tag loss rate that requires multiple interventions on the turtles [1].

Photo-identification (photo-ID) for turtle identification is becoming increasingly used thanks to the development of technology: it is less invasive, cheaper and more reliable over time. This identification relies on the naturally-occurring scales on turtles' heads, which are unique for each turtle and invariant over time [2] as it can be observed in Figure 1.

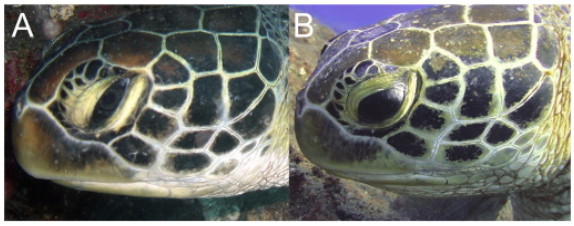

Figure 1: Evolution of the scales on the head of a turtle over a span of five years. The only difference is due to the pigmentation, while the scales did not change.[2]

As a first step towards such a system, there is the need to develop a tool able to crop an image to show only the turtles' face, thus reducing the possibility of having an incorrect registration. This task, usually referred to as Object Localisation, is what we are going to focus on in this project.

2 Impact on people's life

The issue explored in this project belongs to category 14 of the Sustainable Development Goals defined by the United Nations: Life Below Water. The inspiration for the project has been taken from Zindi website [3], which is also providing the dataset for the task.

A solution for the problem can be of help to Local Ocean Conservation (LOC): a Kenyan NFPO whose goal is the preservation of the local marine environment.

Commercial fishing and climate change are among the human-related causes that are leading sea turtles not only towards physical extinction but also toward ecological extinction, i.e. when the number of individuals in a species is so low that they are unable to perform their ecological role. Although the sea turtle population has shrunk over the years, it is still playing an important role in ocean ecosystems by, for example, grazing on seagrass, feasting on jellyfish, transporting nutrients, and supporting other marine life [4].

For this reason, the preservation of this species is crucial: by protecting and enhancing the environmental elements that these animals need to thrive, humanity and other species will flourish as well [5].

3 Setup description

Convolutional Neural Networks (CNNs) play an important role in Computer Vision tasks and, therefore, also in Object Localization. CNNs, thanks to their deep structure, are able to extract very complex features when compared to former approaches.

The framework of object detection models can be divided into two groups: one-stage and two-stage detectors [6].

- **Two-stage detectors**: two-stage detectors like the pioneer R-CNN consist, as the name suggests, of two networks where the first one is responsible for generating a *sparse* set of proposals, while the second CNN will process these regions classifying them as foreground objects or background [7]. These networks lead to high-precision results albeit slowly since two processings are needed for each image. Due to their slowness during the inference phase, two-stage detectors have not been considered for the task.
- **One-stage detectors**: one-stage detectors process the image once and are able to produce immediately the coordinates of the bounding boxes. This results in high-speed networks that lead researchers to study thor-

oughly this kind of detectors over the years. In this project, the main focus was on RetinaNet and, in smaller amounts, on YOLOv8.

RetinaNet: RetinaNet [8] is a single-stage detector that consists in the usage of Feature Pyramid Networks (FPNs) and Focal Loss (FL) as an improvement over other object detection models. FPNs are used to detect objects on different scales by extracting features on different levels and combining them with a top-down path after a bottom-up one. RetinaNet incorporates FPNs and adds a classification and regression head to create an object detection model.
For what concerns Focal Loss, instead, it can be seen as an enhancement over Cross-Entropy loss to tackle the class-imbalance problem. This is useful in single-stage detectors as the anchors containing background occur in greater amounts than the ones associated with foreground objects.

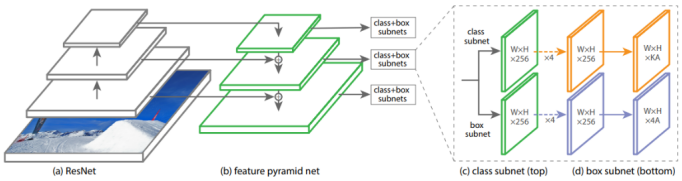

FIGURE 2: The RetinaNet architecture.

Transfer Learning: Undoubtedly, it is infeasible to train a model of such complexity from scratch on small datasets, like the one used in this project. Therefore, transfer learning is needed to achieve good results: we reuse the weights of a model pre-trained on a huge dataset used to extract features from images and then we fine-tune the network on the problem-specific data [9]. Keras CV, an extension of the Keras API, provides a version of RetinaNet pre-trained on the ImageNet dataset, available also with different backbones as feature extractors.

Naïve Regression Approach: Another idea considered for the task consists in looking at the object localization problem as a regression problem, where the goal of the network is to predict directly the top-left and bottom-right corners, which are able to fully describe the box position over the image. The implementation was done only for learning purposes since this approach has been discarded a long time ago as it was not able to achieve SotA results.

4 Experiments and Results

Backbones: We used several backbones with RetinaNet: the ResNet50 (RN) [10] from the original RetinaNet paper, Darknet (DK) [11], EfficientNetv2B0 (E0) and EfficientNetv2B2 (E2) [12].

Augmentation: Three different types of augmentations were performed:

- *NA*: No Augmentation.
- *BA*: Basic Augmentation. Similar to the one used in the original paper, consisting of a horizontal flip and resizing with jittering.
- *CA*: Complex Augmentation. Basic augmentation with also changes to the Hue and Saturation of the image (the idea was to make the model robust also under the water where images may be shot under different lighting conditions).

Optimization: We tried as optimizers SGD as described by the authors of the RetinaNet paper (momentum = 0.9, weight decay = 0.0001) [8], Adam and AdamW (with weight decay = 0.0001). The cosine decay for the learning rate has been always used since it has been proven to have good performance on many tasks [13].

Evaluation Metric: To assess and compare the performance the Intersection over Union metric (IoU) was used.

$$IoU = \frac{\text{Area of Intersection}}{\text{Area of Overlap}} = \frac{\text{Intersection Area}}{\text{GT Area + Prediction Area - Intersection Area}}$$

The evaluation was done over a validation set obtained using an 80/20 split.

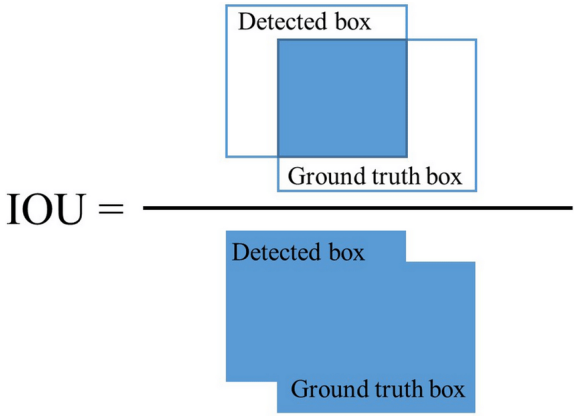

FIGURE 3: Illustration of intersection-over-union (IOU).

Model definition			
Opt	Aug	Backbone	IoU
Adam	BA	RN	0.811
		DK	**0.866**
		E0	0.837
		E2	0.858
SGD	BA	RN	*0.668*
		DK	0.559
		E2	0.525
AdamW	BA	RN	0.839
		DK	*0.848*
		E2	0.845

TABLE 1: Performance based on the optimizer

Model definition			
Opt	Aug	Backbone	IoU
Adam	BA	DK	**0.866**
		E2	0.858
	NA	DK	0.858
		E2	0.843
	CA	DK	0.845
		E2	0.845

TABLE 2: Performance based on the augmentation

Model definition				
Algorithm	Opt	Aug	Backbone	IoU
Naïve Regression	Adam	BA	E0	0.6743
YOLOv8	Adam	BA	YOLOv8	0.7938
RetinaNet	Adam	BA	DRK	**0.866**

TABLE 3: Performance based on the algorithm

Based on the results of Table 1, Adam outperforms the other two optimizers, with SGD leading to definitely worse results. Moreover, DarkNet and EfficientNetv2B2 seem to be the best backbones: the following experiments have focused on them.

Table 2 illustrates, instead, how a basic augmentation leads to the best results for generalization purposes. On the other hand, a strong augmentation that makes the images too artificial is to be avoided.

Finally, Table 3 shows the comparison between our best detection algorithm and others. Please note that YOLOv8 was not trained with the appropriate settings since the model was not able to fit in the GPU available for the experiment (Nvidia GTX 2070 SUPER). It would have been better to use a smaller model like YOLOv5 or YOLOv3, however, the pre-trained version was not available in Keras CV.

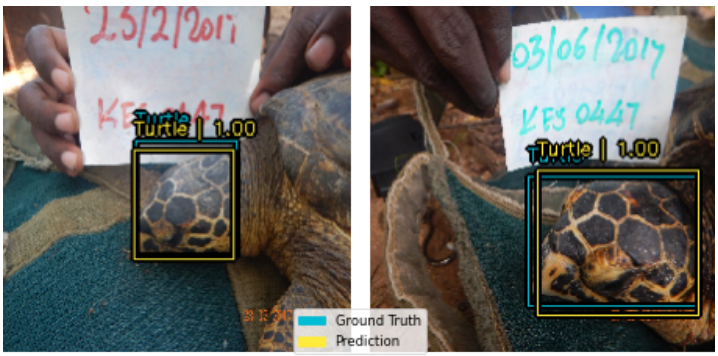

FIGURE 4: Prediction of the model over 2 images taken from the validation set.

The code used for the project can be found online at the following GitHub repository: https://github.com/AlbertoFormaggio1/turtle-head-detection.

My best solution was able to get 21st out of the 61 active participants in the competition (326 people enrolled), achieving an IoU score of 0.878 on the test set.

5 Conclusions

The research proposed a method for the detection of sea turtles' heads. The method is based on object localization with RetinaNet as detection algorithm. Our method is able to achieve a good IoU accuracy, although there is still a margin for improvement. By setting appropriately the augmentation and by changing the optimizer of the original RetinaNet paper we were able to get an improvement in the overall score of roughly 20%. It is important to point out, however, that the boundaries of a turtle's head are not well-defined even for humans, thus the score obtained can be considered already good enough to be applied on the field. In the future, other architectures can be applied: for example, YOLO, with proper hardware, can be fine-tuned on the task. Alternatively, two-stage detectors can be considered once they will be competitive in speed with one-stage ones.

References

[1] J. Reisser, M. Proietti, P. Kinas, and I. Sazima, "Photographic identification of sea turtles: method description and validation, with an estimation of tag loss," Endangered Species Research, vol. 5, no. 1, pp. 73–82, 2008.

[2] A. S. Carpentier, C. Jean, M. Barret, A. Chassagneux, and S. Ciccione, "Stability of facial scale patterns on green sea turtles chelonia mydas over time: a validation for the use of a photo-identification method," Journal of Experimental Marine Biology and Ecology, vol. 476, pp. 15–21, 2016.

[3] "Zindi competition: Local ocean conservation sea turtle face detection," last accessed 26 July 2023. [Online]. Available: https://zindi.africa/competitions/local-ocean-conservation-sea-turtle-face-detection

[4] A. D. Wilson E.G., Miller K.L. and M. M., "Why healthy oceans need sea turtles: the importance of sea turtles to marine ecosystems," Oceana, Tech. Rep., 2010. [Online]. Available: https://oceana.org/reports/why-healthy-oceans-need-sea-turtles/

[5] "Local ocean conservation," last accessed 26 July 2023. [Online]. Available: https://localocean.co/

[6] M. Straka, "Object detection." [Online]. Available: https://ufal.mff.cuni.cz/~straka/courses/npfl114/2223/slides.pdf/npfl114-2223-06.pdf

[7] R. Girshick, J. Donahue, T. Darrell, and J. Malik, "Rich feature hierarchies for accurate object detection and semantic segmentation," 2014.

[8] T. Lin, P. Goyal, R. B. Girshick, K. He, and P. Dollár, "Focal loss for dense object detection," CoRR, vol. abs/1708.02002, 2017. [Online]. Available: http://arxiv.org/abs/1708.02002

[9] M. Straka, "Convolutional neural networks," pp. 54–55. [Online]. Available: https://ufal.mff.cuni.cz/~straka/courses/npfl114/2223/slides.pdf/npfl114-2223-05.pdf

[10] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," CoRR, vol. abs/1512.03385, 2015. [Online]. Available: http://arxiv.org/abs/1512.03385

[11] J. Redmon and A. Farhadi, "Yolov3: An incremental improvement," CoRR, vol. abs/1804.02767, 2018. [Online]. Available: http://arxiv.org/abs/1804.02767

[12] M. Tan and Q. V. Le, "Efficientnet: Rethinking model scaling for convolutional neural networks," 2020.

[13] A. Lewkowycz, "How to decay your learning rate," CoRR, vol. abs/2103.12682, 2021. [Online]. Available: https://arxiv.org/abs/2103.12682