UNIVERSIDAD SAN FRANCISCO DE QUITO USFQ

Colegio de Ciencias e Ingenierías

Hammerhead Shark Detection Using Regions with Convolutional Neural Networks (Faster R-CNN)

.

Gabriela Salomé Ulloa Sotomayor
Vicente Antonio Vásconez Pérez

Carrera

Trabajo de fin de carrera presentado como requisito
para la obtención del título de
Ingeniero Electrónico

Quito, 12 de Mayo de 2020

UNIVERSIDAD SAN FRANCISCO DE QUITO USFQ

Colegio de Ciencias e Ingenierías

HOJA DE CALIFICACIÓN
DE TRABAJO DE FIN DE CARRERA

Hammerhead Shark Detection Using Regions with Convolutional Neural Networks (Faster R-CNN)

Gabriela Salomé Ulloa Sotomayor

Vicente Antonio Vásconez Pérez

Nombre del profesor, Título académico **Diego Benítez, Ph.D.**

Quito, 12 de Mayo de 2020

DERECHOS DE AUTOR

Por medio del presente documento certifico que he leído todas las Políticas y Manuales de la Universidad San Francisco de Quito USFQ, incluyendo la Política de Propiedad Intelectual USFQ, y estoy de acuerdo con su contenido, por lo que los derechos de propiedad intelectual del presente trabajo quedan sujetos a lo dispuesto en esas Políticas.

Asimismo, autorizo a la USFQ para que realice la digitalización y publicación de este trabajo en el repositorio virtual, de conformidad a lo dispuesto en el Art. 144 de la Ley Orgánica de Educación Superior.

Nombres y apellidos: Gabriela Salomé Ulloa Sotomayor

Código: 00130422

Cédula de identidad: 1104636053

Lugar y fecha: Quito, Mayo de 2020

Nombres y apellidos: Vicente Antonio Vásconez Péres

Código: 00123814

Cédula de identidad: 1718640517

Lugar y fecha: Quito, Mayo de 2020

ACLARACIÓN PARA PUBLICACIÓN

Nota: El presente trabajo, en su totalidad o cualquiera de sus partes, no debe ser considerado como una publicación, incluso a pesar de estar disponible sin restricciones a través de un repositorio institucional. Esta declaración se alinea con las prácticas y recomendaciones presentadas por el Committee on Publication Ethics COPE descritas por Barbour et al. (2017) Discussion document on best practice for issues around theses publishing, disponible en http://bit.ly/COPETheses.

UNPUBLISHED DOCUMENT

Note: The following capstone project is available through Universidad San Francisco de Quito USFQ institutional repository. Nonetheless, this project – in whole or in part – should not be considered a publication. This statement follows the recommendations presented by the Committee on Publication Ethics COPE described by Barbour et al. (2017) Discussion document on best practice for issues around theses publishing available on http://bit.ly/COPETheses.

RESUMEN

A lo largo de los años, la captura ilegal de tiburones en el Océano Pacífico ha aumentado drásticamente, hasta el punto en que el tiburón martillo se ha convertido en una especie en peligro de extinción. El monitoreo para esta especie es un procedimiento bastante complicado debido a que la mayoría de los métodos utilizados para este proceso son invasivos. Dadas estas circunstancias, los biólogos marinos optaron como solución usar cámaras subacuáticas para hacer este análisis directamente de los videos, pero este sigue siendo un proceso lento y costoso. Una herramienta importante e innovadora para resolver este problema es mediante el uso de métodos automatizados. En este artículo, se aplicó un detector de objetos basado en Regiones más rápidas con redes neuronales convolucionales (R-CNN más rápido) para detectar tiburones martillo obtenidos de videos y bases de datos de imágenes. La capacitación utilizó como extractor de funciones el ResNet50, que es una red neuronal convolucional con 50 capas de profundidad, para obtener un detector exitoso y aplicarlo en un rastreador en tiempo real para observar el comportamiento y el movimiento de las comunidades de tiburones martillo. El trabajo consistió en crear una base de datos lo suficientemente grande con imágenes etiquetadas, preprocesar estas imágenes, lo que significa cambiar su tamaño con la restricción de red de características y crear un conjunto de datos aumentado para mejorar las condiciones de entrenamiento. Después de eso, el detector fue entrenado usando un R-CNN más rápido el cual crea un sistema de seguimiento de objetos en tiempo real para observar tiburones martillo bajo el agua, con imágenes obtenidas con el mínimo efecto en el ecosistema. Se obtuvo una precisión promedio del 90% con todas las imágenes de prueba utilizadas en la experimentación.

Palabras claves: R-CNN más rápido, aprendizaje profundo, seguimiento de objetos, detección de objetos, detección y seguimiento de tiburones martillo, detector en tiempo real, ResNet50.

ABSTRACT

Over the years the illegal catch of sharks in the Pacific Ocean has drastically increased, to the point where the Scalloped Hammerhead Shark has become an endangered species. The monitoring of these for this endangered species is a very difficult procedure due to the fact that most of the methods used for this process are invasive. Given this circumstances marine biologists have chosen as solution the use of underwater cameras to make this analysis directly from videos, but this is still a slow and expensive process. An important and innovative tool for solving this problem is by using automated methods. In this paper, an object detector based on faster Regions with convolutional neural networks (faster R-CNN) was applied to detect hammerhead sharks obtained from videos and image datasets. The training used as a feature extractor the ResNet50, which is a Convolutional Neural Network with 50 layers deep, to obtain a successfully detector and to apply this in a real time tracker to observe the behavior and movement of hammerhead shark communities. The work consisted on creating a large enough database with labeled images of hammerhead sharks, pre-process these images, which means to resize them with the feature network restriction, and create an augmented dataset as well to improve training conditions. After that, the detector was trained by using a Faster R-CNN and this create an object tracking system in real time to observe hammerhead sharks underwater, with footage obtained with the minimum effect on the ecosystem. An average accuracy of 90% was obtained with all the testing images used in the experimentation.

Key Words: Faster R-CNN, Deep Learning, Object Tracking, Object Detection, hammerhead shark detection and tracking, real-time detector, ResNet50.

TABLA DE CONTENIDO

ÍNDICE DE TABLAS

ÍNDICE DE FIGURAS

INTRODUCTION

Marine species, such as sharks, contribute significantly to the preservation of healthy marine ecosystems (Sharma, Scully-Power & Blumenstein, 2018). The Galapagos Marine Reserve (RMG) protects some of the last shark aggregations that remains in the world (Chiriboga, 2018). Among the fish types that are key to the RMG, both at an ecological and tourist level, a large number of shark species can be found, including hammerhead sharks (*Sphyrna lewini*) (Vilema, 2015).

Data related to the quantity and spreading of these species are important when it comes to monitoring their status and condition. Although there are manual methods used to estimate the size of these shark populations and their taxonomy, they may involve invasive and time-consuming measures, some examples of this are physical catch and release fishing sampling and underwater visual census made by divers. Video monitoring, on the other hand, has gained popularity over the years for being a method that prevents invading and destroying these ecosystems. However, this analysis is a slow and expensive process. A solution to solve this problem can be system for recognizing these species automatically, this will improve the analysis efficiency. (Siddiqui et al., 2017).

Research on marine species recognition is not a very commun area on computer vision, but with recent developments of deep learning, the interest on these topics has increase. (Xu, Bennamoun, An, Sohel & Boussaid, 2019). Deep Learning is considered a machine learning technique, that works as a neural network extension, where a computer model acquires the knowledge to perform classification tasks directly from images, texts, or sound (MathWorks). Where a computer model learns representations of data with multiple levels of abstraction (Kim, 2017). These methods fed the machine with raw data and to discover important representations for detection or classification automatically (LeCun, Bengio & Hinton, 2015).

There are diferent types of deep learning architectures, some of these are: recurrent neural networks (RNN), long short term memory networks (LSTM), convolutional neural networks (CNN), deep belief networks (DBN), deep sparse-coded Networks (DSN) among others. These architectures are applied in a wide range of scenarios. A CNN, can be used for image recognition and video analysis (Jones, 2017). One deep learning approach is the so called regions with convolutional neural networks (R-CNN), which trains end-to-end CNNs to categorize rectangular region proposals into object or background (Ren, He, Girshick & Sun, 2015). There are three variants of a R-CNN (R-CNN, Fast R-CNN, Faster R-CNN). Each one attempts to optimize or speed-up the results of the processes (MathWorks). For this work, we will use the Faster R-CNN for detection and tracking of scalloped hammerhead sharks.

Object detection is considered a challenging problems of computer vision. (Liu et al., 2018). The purpose of this method is to determine the location of objects in a given image (Zhao, Zheng, tao Xu & Wu, 2019). On the other hand, object tracking is an automatic estimation of the trajectory of an object around the video footage (Wang & Yeung, 2013). Despite the progress that has been made in this area in recent years, this methos are still a challeng, due to the object appearance variation caused by illumination variations, obstructions, change possess, cluttered scenes, backgrounds and others (Chen et al., 2015). In this project we want to detect the location of hammerhead sharks on video images which were taken at the Galapagos Islands.

MATERIALS AND METHODS

Faster R-CNN detection network

A Faster R-CNN is composed by a feature extraction network which is a pretrained CNN, on this case a Residual Neural Network (ResNet50), followed by two subnetworks, a RPN and a second proposal network to predict the class of each object (MathWorks).

The RPN is designed to predict region proposals. It takes an image as input and gives a set of rectangular proposals as outputs, each one with a probability score (Ren et al., 2015). For this process, it is important to use anchor boxes, the number of anchor is obtained by calculating the mean intersection-over-union (IoU), which is the area of overlap between the prediction and the training data divided by the area of union between the prediction and the ground truth, of the training data. It is important to use a number of boxes that give as result a mean IoU greater than 0.5, this will ensure that the anchor boxes overlap with the boxes in the training data (MathWorks).

The training of the RPN consist on setting the anchor boxes and the training data boxes. The anchors with the highest IoU that overlap with a training data box will be labeled as foreground, and they will pass to the next level (ROI pooling) as proposals (Ren et al., 2015).

The region of interest (ROI) pooling layer accepts the convolutional features generated by the CNN and the predicted bounding boxes given by the RPN (Ren et al., 2015), to produce a set of matting on the feature maps according to the proposal boxes, for which it scales them to a pre-defined size (Yan, Chen, Chen, Kendrick & Wu, 2018). The process done up to the ROI pooling corresponds to feature extraction. The results of this process pass then to the object classification, where the classification layers take the output produced by the ROI pooling layer and passes them through a series of convolutional layers which produce a classification and a bounding box refinement layer. This process is observed on Figure 1.

The Faster R-CNN requires to specify several inputs:

Network input size.

The image input size required by this network is [224x224x3]. This means that it accepts RGB images with a maximum size of 224x224. In consequence, images and bounding boxes resizing have to be done in a previous step.

Anchor boxes.

Anchors are a key component for Faster R-CNN. They consist on boxes with different ratios (Ren et al., 2015). This method is the most efficient, given than all predictions can be evaluated at once, and it only labels the images, instead of cropping and resizing them, as it is done with the R-CNN method. For calculating the number of anchor boxes the graphic number of anchors versus mean IoU has to be done, this can be observed on Figure 2. As it was mention before, it is essential to have a mean IoU higher than 0.5. On the graphic, it can be seen that by using only one anchor, the mean IoU is equal to 0,44 and with more than seven anchors, it yields only a marginal improvement in mean value. While using a large number of anchor boxes in the object detector can lead to an overfitting (MathWorks), therefore for this study, it was opted to use six anchor boxes.

Feature extraction networks.

The detector works with ResNet50. This is a pre-trained 50 layers deep CNN used mainly as a feature detector. The ResNet50 was trained with more than a million images from ImageNet database, where it can recognize over 1000 different categories (Espinosa, 2019). ResNet50 block diagram can be observed on Figure 3.

Shark Database

Two video footage sources that were filmed during diving sessions at the Galapagos Islands [0° 39' 59.99" N -90° 32' 59.99" W], where the diver encounters a great number of scalloped hammerhead sharks among some other marine species, were used for this study. Furthermore, a dataset of hammerhead shark images taken from the internet, where the sharks characteristics could be better observed and analyzed, was also used.

The videos were processed using MATLAB with the ``video labeler" application from the Image Processing and Computer Vision toolbox. This application allows to label every frame from a video using a point tracker algorithm, which tracks various ROIs using the Kanade-Lucas-Tomasi (KLT) algorithm (MathWorks). Although the apps provided the algorithms help with this process, it was necessary to manually resize the ROIs rectangle for each frame of the video to ensure proper proposed regions. For the image data, a similar process was used, it was necessary to manually label every scalloped hammerhead shark within each image using the Image Labeler application.

The purpose of using these applications was to create a ground truth file for each video and images data store which contains the information of the data source, label definition and label data (MathWorks). With this information it was possible to create a database that contains the all the images obtained from the videos plus the ones retrieved online with the information previously described. The labeled images were mixed in one data-base. The ones retrieved online where only used for training, given their high clarity respect the shape characteristic of the hammerhead sharks, while the ones obtained from the videos were separated on two groups, one for training the other one for validation, given a total of two databases.

Experimental setup

Image Resizing.

Original images from the dataset are from frames of high definitions videos which are [4096x2160x3]. Therefore, it is necessary to readjust the size of these images to the required [224x224x3] size, defined by the feature extraction network, and repeat this process for each labeled box defined in every image so that the final images and the labeled boxes will match with their respective ones.

Data Augmentation.

The original training data set contains 247 images. Data augmentation methods were used to enlarge the dataset (He, Zhang, Ren, & Sun, 2016). It is also used to improve network accuracy by transforming the training data randomly, adding more variety without actually increasing the number of labeled training samples (MathWorks). For this instance, every image was rotated and scaled. The final Training Dataset consisted of 988 shark images.

Training Options.

For the training options defined for the training Process of the Faster R-CNN, a stochastic gradient descent with momentum (SGDM) is used. This method calculates the error for each training data and adjust the weights immediately. By instance, if the trainer has 100 training data points, the SGDM adjust the weights 10 times. It updates the network parameters, such as weights and biases, and ensure a minimal loss function. At each iteration, the SGDM, using a subset of the training data, updates the parameters. Mini-batch is also used at each iteration. One epoch means that all the training data passed through the training algorithm using mini-batches (MathWorks).

Validation Metrics.

The validation consisted in two sections. The first one is the use of the true ground labeled boxes, which are then later compared to the ones obtained by the detector. The second method consists in running a video footage, specifically on .avi format, to obtain as result the same video but with labeled bounding boxes of the detected hammerhead sharks and their respective matching score.

The implementation and application of the proposed shark detector was made in MATLAB (Release R2019b) while the employed Faster R-CNN is available within the Deep Learnig Tool Box (version 13.0) (MathWorks).

RESULTS AND DISCUSSION

Once all the parameters are correctly defined, the training process can take place. While the training is in progress, the MATLAB console displays the results of each epoch. Within these results, it is possible to observe the GPU processing time required by each epoch, the mini-batch loss, and the mini-batch accuracy. For the Faster R-CNN training, a total number of 8 epochs were used. The total time required by the GPU for training was 31 min and 50 s. For each epoch carried out, it was observed that the mini batch loss was less than 1, except for the first one that had a loss of 1.59, and the accuracy for most of the epochs was around 99%.

Training Validation

After accomplishing the training process, it is necessary to validate the dataset inside the detector. An image dataset with labeled images was obtained, which correspond to every hammerhead shark found in the images obtained by the detector. To ensure a successful training process, it is necessary to compare the obtained results with the Bounding Box data-stored, set at the beginning of the whole process. On the other hand, a video source that was

not considered during the training step was used to test the proposed method in real-time. The performance of the method was based on the accuracy (ACC) of hammerhead shark detection and tracking across a set of retrieved frames from the test video. The detection will give the following results:

- True positives TP: It refers to all the correct detections we obtain on the test.
- False positives FP: It refers to all the wrong detections we obtain on the test.
- True negative TN: It refers to a correct misdetection.
- False negative FN: It refers to all results we obtain with any objects detected.

The comparison process between the validation box labels and the ones given by the detector, was used to compute the precision and the recall of our trained detector. The precision was obtained by dividing the true positive results by all the detections, while the recall was obtained by dividing the true positives by the validation data.

The comparison is done by observing the superposition of the detector-obtained bounding box over the bounding box manually placed during training, as illustrated by Figure 4, where the bounding box represented in red, was taken from the dataset created manually for validation, while the blue rectangle is the bounding box obtained as a result of running the same image on the detector. A true positive result was obtained for this case since the area of the obtained bounding box completely covers the area of the validation bounding box.

This process must be done for every image tested on the detector. Figure 5 shows the Precision vs Recall curve obtained by testing the validation images. The graph shows that most of the analyzed images got a precision of 100% for detection. By calculating the average of this results we can obtain a precision of 0.9 over 1. This means that the proposed detector is capable of detecting hammerhead sharks within the video or images under test in almost all cases. For this propose the testing image dataset consisted of 163 images. Once the detector

was confirmed to be working properly, other test images or videos were processed, using the following criteria:

Object detection

The performance of the detector was evaluated by the recognition of all hammerhead sharks present in a given image. For this process, any image where a shark is present was imported, the size of this picture was irrelevant for the process. A bounding box was placed on the detected object and the score that represents the certainty that it is a shark was also displayed, for example as it is observer on Figure 6. For this study, we considered a threshold value of 0.85 for considering a true shark detection.

Object tracking

The object tracking works as the previous method but in this case it detects the objects as they move on a video. The footage was separated into frames and shark detection was carried out in each one, as it was done previously for the images. A box will follow the shark in the generated video. This method was tested with two video footage not previously used for training and without verification content (i.e without manually placed bounding boxes). The results can be observed on Figure 7 and Figure 8.

Different results were obtained for images and videos, where it can be seen that the detector clearly recognizes most of the sharks located in every frame. On Figure 6 we can observe that the detector recognizes the shark with a mean score of 0.91. The image used for object detection is clear and the shark was found to be swimming near to the camera, which makes it easier for the detector to recognizes the shark. There were some problems when sharks were swimming far away from the camera. As we can see on some of the frames from Figure 7 and Figure 8, when sharks are moving further from the camera, the score values decrease to the point where the detection is no longer satisfactory. The same happens when there is an

extremely high light exposure, as for example when the sharks found themselves directly under the sun, in such case the detector does not manage to find them in the image.

Specific accuracy data related to results obtained from the frames of these videos can be found inside Table 1 and Table 2, where the real number of hammerhead sharks found in each frame is compared to the number of sharks detected and tracked by the Faster R-CNN detector. To ensure the proper functionality of the proposed method, the test videos were selected to have different conditions, such as density of other fish species as well as light exposure.

CONCLUSIONS

In this study, we developed an automated shark detection method using a Faster R-CNN architecture. The obtained results are considered satisfactory, since there was a fairly good prediction level in the detection, as well as in the tracking of hammerhead sharks found in the tested video sources. However, there are still parameters that could be changed within this architecture for a better detection. For instance, when sharks are located further away from the camera location, and in situations where there is low or high light exposure, the detector was not able to find them, as well as when the sharks were located directly underneath the sun light or behind another fish species. For fixing this problem it would be necessary to manually increase or modify the layers that are used for training, so the detector would be able to find all the sharks presented in the image. Another found challenge was the casual mistake between sharks and other marine species, such as a variety of large fishes and other shark species. The detector would sometimes confuse them with a hammerhead shark, by mixing their physical features leading to a fault in the results, however in this cases the precision value was considerable low, never exceeding 75%. Furthermore, we must also take into consideration that the cameras used for recording the videos were not static and the video footage from these cameras do not have the best quality, this may cause some problems when using such images.

Achieving lower false negative or false positive responses on the results is still a challenge on this type of videos. For a future work we will seek to fix these problems in order to obtain a higher quality detector, for which we could opt for more complete architectures.

REFERENCES

Chen, Y., Yang, X., Zhong, B., Pan, S., Chen, D., Zhang, H. (2015). CNNTracker: Online discriminative object tracking via deep convolutional neural network. *Applied Soft Computing*(38), 1088-1098. doi: 10.1016/j.asoc.2015.06.048

Chiriboga, Y. (2018). Ecología Espacial y Conservación de tiburones neonatos y juveniles punta negra (Carcharhinus limbatus) en la Isla San Cristóbal – Reserva Marina de Galápagos. (Spanish) [On reserva marina de galápagos]. Retrieved from http://repositorio.usfq.edu.ec/bitstream/23000/7545/1/139489.pdf

Espinosa, J. (2019). Detection And Tracking of Motorcycles in Urban Environments by using Video Sequences with High Level of Oclussion (Doctoral dissertation, Universidad Nacional de Colombia - Sede Medellín). Retrieved from http://bdigital.unal.edu.co/72867/1/93390022.2019.pdf

He, K., Zhang, X., Ren, S., Sun, J. (2016). Deep Residual Learning for Image Recognition. *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. doi: 10.1016/j.asoc.2015.06.048

Jones, T. (2017). Deep learning architectures. Retrieved from https://developer.ibm.com/technologies/artificial-intelligence/articles/cc-machine-learning-deep-learning-architectures/

Kim, P. (2017). Matlab deep learning with machine learning, neural networks and artificial intelligence. Seoul, SouthKorea: Addison-Wesley.

LeCun, Y., Bengio, Y., Hinton, G. (2015). Deep learning. *Nature*,521(7553), 436-444. doi: 10.1038/nature14539

Liu, L., Ouyang, W., Wang, X., Fieguth, P., Chen, J., Liu, X., Pietikainen, M. (2018). Deep Learning for Generic Object Detection: A Survey, *International Journal of Computer Vision*(128), 261-318. doi: https://doi.org/10.1007/s11263-019-01247-4

MathWorks. (n.d.). Deep Learning Toolbox. Retrieved from https://la.mathworks.com/help/deeplearning/

MathWorks. (n.d.). Estimate anchor boxes from training data. Retrieved from https://la.mathworks.com/help/vision/examples/estimate-anchor-boxes-from-training-data.html

MathWorks. (n.d.). Getting started with r-cnn, fast r-cnn, and faster r-cnn. Retrieved from https://la.mathworks.com/help/vision/ug/getting-started-with-r-cnn-fast-r-cnn-and-faster-r-cnn.html

MathWorks. (n.d.). groundtruth. Retrieved from https://la.mathworks.com/help/vision/ref/groundtruth.html

MathWorks. (n.d.). Introducing deep learning with matlab. Retrieved from https://la.mathworks.com/ content/dam/mathworks/ebook/gated/ 80879v00_Deep_Learning_ebook.pdf

MathWorks. (n.d.). Object detection using faster rcnn deep learning. Retrieved from https://la.mathworks.com/help/vision/examples/object-detection-using-faster-r-cnn-deep-learning.html

MathWorks. (n.d.). Train a deep learning vehicle detector. Retrieved from https://la.mathworks.com/help/driving/examples/train-a-deep-learning-vehicle-detector.html

MathWorks. (n.d.). trainingoptions. Retrieved from https://la.mathworks.com/help/deeplearning/ref/trainingoptions.html

MathWorks. (n.d.). Video labeler. Retrieved from https://la.mathworks.com/help/vision/ref/videolabeler-app.html

Ren, S., He, K., Girshick, R., Sun, J. (2015). Faster R-CNN: Towards Real-Time Object Detection with Region Proposal Networks, *IEEE Transactions on Pattern Analysis and Machine Intelligence* (39), 1137 – 1149. doi: 10.1109/TPAMI.2016.2577031

Sharma, N., Scully-Power, P., Blumenstein, M. (2018). Shark Detection from Aerial Imagery Using Region-Based CNN, a Study. *AI 2018: Advances in Artificial Intelligence,* 224–236. doi: 10.1007/978-3-030-03991-2_23

Siddiqui, S., Salman, A., Malik, M., Shafait, F., Mian, A., Shortis, M., Harvey, E. (2017). Automatic fish species classification in underwater videos: exploiting pre-trained deep neural network models to compensate for limited labelled data, *ICES Journal of Marine Science,75*(1), 474–389. doi: 10.1093/icesjms/fsx109

Vilema, D. (2015). Diseño de una campaña de comunicación y educación ambiental sobre la conservación de las agregaciones de desove del Bacalao de Galápagos (Mycteroperca olfax) y del Tiburón Ballena (Rhincodon typus) en las Islas Galápagos (Spanish). Retrieved from http://repositorio.usfq.edu.ec/bitstream/23000/4487/1/112710.pdf

Wang, N., Yeung, D.-Y. (2013). Learning a deep compact image representation for visual tracking (C. J. C. Burges, L. Bottou, M. Welling, Z. Ghahramani, K. Q. Weinberger, Eds.). Curran Associates, Inc. Retrieved from http://papers.nips.cc/ paper/5192-learning-a-deep-compact -image-representation-for-visual -tracking.pdf

Xu, L., Bennamoun, M., An, S., Sohel, F., Boussaid, F. (2019). Deep Learning for Marine Species Recognition, *Handbook of deep learning applications,* 129–145. doi: https://doi.org/10.1007/978-3-030-11479-4_7

Yan, C., Chen, W., Chen, P., Kendrick, A., Wu, X. (2018). A new two-stage object detection network without RoI-Pooling. *2018 Chinese Control And Decision Conference (CCDC),* 1680-1685. doi: 10.1109/CCDC.2018.8407398

Zhao, Z.-Q., Zheng, P., Xu, S.-T., Wu, X. (2018). Object Detection With Deep Learning: A Review, *IEEE Transactions on Neural Networks and Learning Systems(99),* 1-21. doi: DOI: 10.1109/TNNLS.2018.2876865

TABLES

Table 1 ACC results per frame obtained with the Faster R-CNN based on shark detector for footage video of figure 7

Frame (ID)	Time (s)	Number of sharks per frame (u)	Correct detection Faster R-CNN	ACC based detection (%) Faster R-CNN
1	**09**	**3**	**2**	**67**
2	**10**	**4**	**3**	**75**
3	**11**	**5**	**4**	**80**
4	**12**	**4**	**3**	**75**
5	**13**	**3**	**2**	**67**
6	**14**	**4**	**3**	**75**
7	**15**	**5**	**4**	**80**
8	**16**	**1**	**1**	**100**
9	**20**	**1**	**1**	**100**

Table 2 ACC results per frame obtained with the Faster R-CNN based on shark detector for footage video of figure 8

Frame (ID)	Time (s)	Number of sharks per frame (u)	Correct detection Faster R-CNN	ACC based detection (%) Faster R-CNN
1	**09**	**2**	**1**	**50**
2	**01**	**2**	**2**	**100**
3	**02**	**2**	**2**	**100**
4	**04**	**1**	**1**	**100**
5	**31**	**2**	**1**	**50**
6	**33**	**2**	**2**	**100**
7	**37**	**2**	**2**	**100**
8	**41**	**2**	**1**	**50**
9	**46**	**3**	**1**	**33**

FIGURES

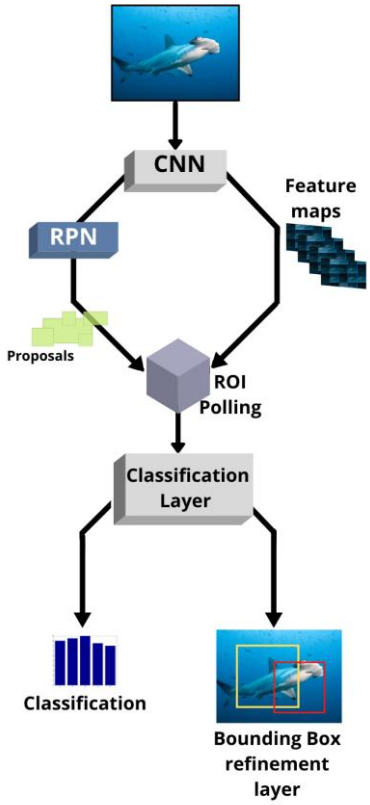

Figure 1 Block Diagram of the Faster R-CNN

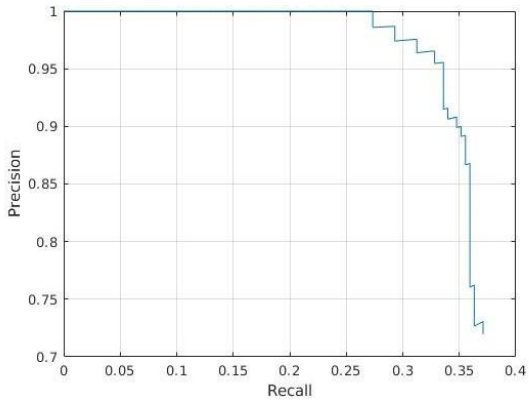

Figure 2 Behavior of the mean IoU against the number of anchor boxes

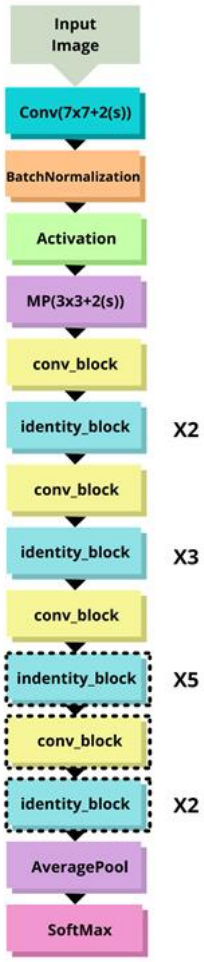

Figure 3 ResNet50 block diagram

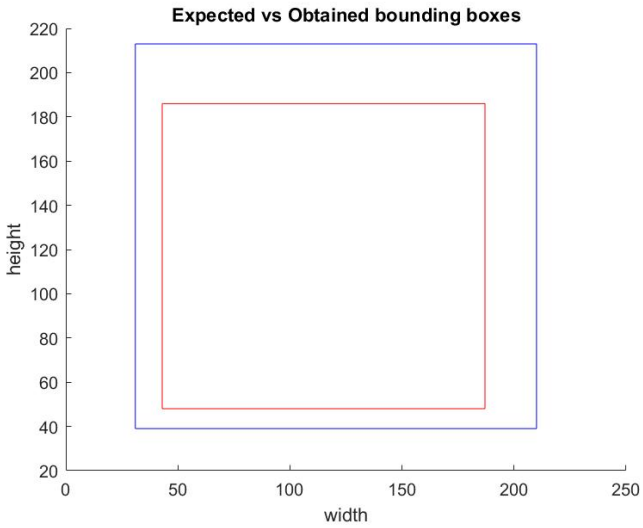

Figure 4 Validation and detection bounding box for the first image

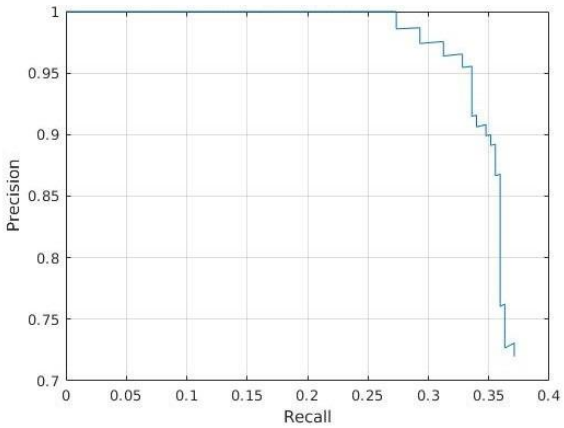

Figure 5 Precision vs Recall curve

Figure 6 Randomly selected image from the dataset with a hammerhead shark

Figure 7 Performance of the Faster R-CNN method across the frames under analysis: successfully (green box) hammer shark detection in a test video

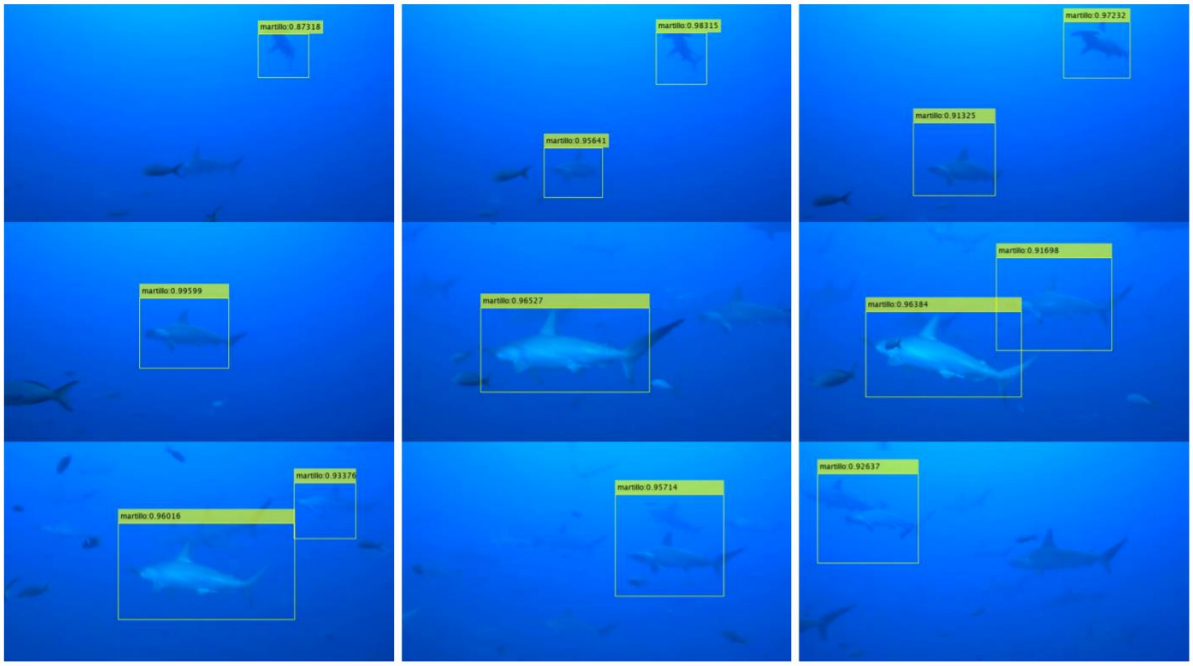

Figure 8 Performance of the proposed Faster R-CNN method across the frames under analysis: successfully (green box) hammer shark detection in another