Automatic Recognition of Marine Creatures using Deep Learning

Oudayrao Ittoo, Sameerchand Pudaruth
ICT Department, FoICDT, University of Mauritius, Réduit, Mauritius

***Abstract*—The identification of marine species is a challenge for people all over the world, and the situation is not different for Mauritians. It is of utmost importance to create an automated system to correctly identify marine species. In the past, researchers have used machine learning to address the issue of marine creature recognition. The manual feature extraction part of machine learning complicates model creation as features have to be extracted manually using an appropriate filter. In this work, we have used deep learning models to automate the feature extraction procedure. Currently, there is no publicly available dataset of marine creatures from the Indian Ocean. We created one of the biggest datasets used in this field, consisting of 51 different marine species collected from the Odysseo Oceanarium in Mauritius. The original dataset has a total of 5,709 images and is imbalanced. Image augmentations were performed to create an oversampled version of the dataset with 171 images per class, for a total of 8,721 images. The MobileNetV1 model trained on the oversampled dataset with a split ratio of 80% for training and 10% for validation and testing was the best performing one in terms of classification accuracy and inference time. The model had the smallest inference time of 0.10 seconds per image and attained a classification accuracy of 99.89% and an F1 score of 99.89%.**

Keywords—Marine creature identification; machine learning; deep learning; MobileNetV1; Mauritius

I. Introduction

Millions of marine creatures live in the ocean, making it the largest habitat on the planet [1]. The world's health is closely related to this marine biodiversity. These marine creatures are of the utmost importance to society as they are a source of food and a symbol of economic welfare. For example, fish are estimated to provide 20% of animal protein to about three billion people [2]. In addition, the ocean is home to a diverse range of creatures that can be utilised for the development of pharmaceutical products to treat various diseases [3]. The human race benefits from the numerous advantages that the marine ecosystem provides for its survival [4]. Therefore, the effective conservation of this biodiversity in a sustainable manner is crucial for the proper functioning of the marine ecosystem and the human race [5].

Traditionally, marine biologists identify aquatic species by visually inspecting their morphological traits [6-7]. Another popular method to correctly identify and group them is deoxyribonucleic acid (DNA) barcoding [8]. This method can be used to precisely, accurately, and quickly detect invasive alien species or marine bacteria that can cause viral outbreaks [9]. DNA barcoding has already proven its worth as a deterrent against various forms of economic fraud, such as seafood mislabelling [10]. Despite its advantages, such an identification method is labour-intensive and time-consuming. As a result, it is crucial to create an automatic marine creature recognition system to address these difficulties.

Automatic recognition of marine creatures is a topic of interest to many researchers around the world. Pudaruth et al. have developed such a system in the form of a mobile application to recognise some of the marine fish that are present in Mauritian waters [11]. However, no system has been developed to cater for other types of marine species that can be found in the Indian Ocean. Common people with no expertise in marine taxonomy have difficulties distinguishing between the different aquatic species. This poses a problem, especially when deadly ocean animals such as the stonefish, blue-ringed octopus, or the lionfish, amongst many others, are encountered [12-13]. Furthermore, some endangered species require proper protection, such as conservation laws and regulations. These are only feasible after recognising them.

The motivation for this study is to create an image recognition model capable of distinguishing between different marine creatures. It is worth mentioning that 80% of the ocean is still undiscovered. According to an interview given by Dr Gene Carl Feldman to Oceana (an organisation focused on protecting the ocean), space exploration is far simpler than ocean exploration [14]. As there is an abundance of marine life in the ocean and it is difficult to cater for all of them, the scope of this study focuses only on some marine creatures that are available at Odysseo Oceanarium in Mauritius. The recognition model has been integrated into a web application. The importance of this application is diverse. First, it will help in creating awareness about the creatures, especially dangerous species. Furthermore, it will also help in raising knowledge by providing some basic information about the creature after the recognition phase. Information such as its scientific name, common name, short description, and whether the animal is deadly has been provided. The information provided can then be used to better understand the animal. The proposed system employs computer vision and deep learning techniques to properly and accurately identify the marine creatures.

This paper is divided into different sections. In Section II, a background study and reviews of related work in this field are provided. Section III delves deeper into the methodology and Section IV assesses the model's performance and discusses the results obtained. The final section concludes this paper. Table XII in Appendix I lists all the marine creatures from the Odysseo Oceanarium which were used in this study.

II. Literature Review

Several studies have been carried out in the past to develop systems for the automated identification of marine life. This section contains summaries of several relevant works in chronological order.

A. *Fish Recognition*

Strachan et al. conducted one of the earliest studies in this field, attempting to evaluate three different image analysis methods to differentiate between images of fish from different species [15]. Methods such as invariant moments, mismatch optimisation, and geometric shape descriptors were used. Their strategy takes into account the fact that fish can be identified by their body shape. The dataset used in their experiment consisted of 60 different fish images. Their research found that the geometric shape descriptors method outperformed the other two approaches, yielding a 90% accuracy rate. However, their experiment was limited to a restricted number of species: only seven different species (two of which were identical gurnard fish, shot from different perspectives).

Fish image recognition has found its usefulness in systems such as automated fish counting. Fish counting is a challenging but critical task for the maintenance of a sustainable fishing level and the prevention of overfishing [16]. Luo et al. proposed a method for such a system by using video footage captured during fishing operations [17]. Their method involved the use of Statistical Shape Models (SSM) and Artificial Neural Network (ANN). To overcome the occlusion problem caused by people walking around the deck of the ship, the video footage was pre-processed. The colour of the images was used as a feature for the recognition process, and an Error Back Propagating ANN classifier was used to recognise the fish from the background. The next step was to use SSM to identify the fish. Lastly, a rule-based counting method was used to count the number of fish. Their method achieved an accuracy of 89.6% for a one-hour video.

The study conducted by Rathi et al. proposed a solution based on Convolutional Neural Network (CNN), deep learning, and image processing [18]. Their method involved pre-processing of the captured images with the aim of removing noise and then using CNN to classify the different fishes. The Otsu's thresholding method was adopted to obtain a histogram representation of the input grayscale image. The next step was to perform morphological operations, such as dilation and erosion, to prepare the resulting image to be processed by the CNN algorithm. To put their method to the test, they used 27,142 images from the Fish4Knowledge dataset, representing 21 species which resulted in an accuracy of 96.29%. However, due to background noise and a lack of image enhancement techniques to compensate for lost features during the pre-processing phase, some of the classifications were incorrect.

Faster R-CNN was used by Mandal et al. to create a system for the automatic detection and identification of fish species [19]. Their dataset consisted of 50 different fish species. Using a random sample technique, their dataset was divided into training (70%), validation (10%), and testing (20%) sets. They were able to achieve a mean average precision (mAP) of 82.4%.

Deep and Dash employed CNN for feature extraction, followed by Support Vector Machine (SVM) and k-Nearest Neighbour (kNN) for classification [20]. They used the Fish4Knowledge dataset which was divided into training (90%) and testing (10%) sets. The training set was further divided such that 10% of the training images were used for validation. Their research proved that using kNN for classification yields the best accuracy of 98.79%.

Rico-Díaz et al. proposed a non-invasive method for addressing the fish recognition problem by combining artificial vision techniques and ANN [21]. Their work relied on the fact that fish from different species can be distinguished based on their eye's sclera and pupil. The first step in the identification process was to employ image filtering techniques to reduce noise in the captured image. After that, background subtraction was done to segment the fish from the background. The next step was to identify the fish's eye, and for this the Hough algorithm was used. Additionally, a feed-forward ANN, being more costly, was also employed if the first method (the Hough algorithm) failed. Using their approach, they were able to achieve an overall accuracy of 74% for eye detection. Also, two underwater cameras were used to estimate the size and weight of the fish while they were swimming. Their solution, however, is dependent on the image's quality and a good background subtraction. Furthermore, performance degrades when the ANN is used if the Hough algorithm fails to detect the fish.

Liang et al. combined CNN and migration learning to distinguish between three different kinds of Chinese ornamental fish [22]. They used TensorFlow, which is an open-source library for machine learning and artificial intelligence, to train their network model. A total of 14, 000 (4*3,500) images were gathered, which were divided into 3,000 and 500 images for training and testing, respectively. Their dataset, which consisted of 3,500 images of three different fish and one set for other types of fish, was gathered from the Internet by using the web crawler technology. Following that, pre-processing was an important step in enhancing the recognition rate as real-time videos of fish were shot outside of the aquarium. The dark channel prior and gamma correction methods were used for this purpose. The latter was a significant step towards the removal of brightness from the pictures. To reduce processing power, all images were scaled to 250 * 250 pixels. Their experiment showed that an accuracy of 98.1% is achievable.

Cai et al. took a different approach to realising a system for detecting fish and counting [23]. They proposed to use the You Only Look Once Version 3 (YOLOv3) model with MobileNet as the backbone for feature extraction. Their proposed system was trained using different strategies. They found out that their system performs better than when using YOLOv3 alone. The average precision obtained was 79.61%.

Pudaruth et al. experimented with multiple machine learning classifiers to discover the most effective one for developing a smartphone application to recognise different fish species existing in the exclusive economic zone (EEZ) of Mauritius [11]. Their model was tested on 38 different fish species with a dataset consisting of 1,520 images. Using the kNN classifier, they were able to attain an accuracy of 96%. Also, their study has shown that the use of deep neural networks (DNN) with the TensorFlow framework can attain an impressive accuracy of 98%. However, pictures of the fish were taken in a controlled environment and not in their natural habitat. The fish were placed on a white background to enable easier segmentation.

Conrady et al. used the Mask Region-Based Convolutional Neural Network (R-CNN) object detection framework to perform classification of the Roman seabream fish, which is endemic in Southern Africa [24]. Their dataset consisted of 2,015 images of the fish. They were able to get a mAP of 81.45% on their test data.

B. Marine Creature Recognition

While most researchers have focused on fish recognition, Chen and Yu proposed a high-definition camera system capable of recognising marine creatures [25]. Their approach to the identification process was broken down into multiple steps. The first step was concerned with the extraction of the image frames from the original video. The following step involved pre-processing of the retrieved images. In addition, for the detection phase, the original image had to be transformed to its grayscale and binary representations. To classify seven creatures, two separate methods were used: The Back Propagation Neural Network and the SVM methods. Their study showed that the SVM approach had an accuracy of 92% in classifying the creatures, which was higher compared to the other methods. However, their proposed method does not work well for creatures with similar shapes.

Pelletier et al. developed a system capable of classifying marine animals into eight categories [26]. Their imbalanced dataset contained 3,777 images. They used two models to conduct their tests, namely AlexNet and GoogLeNet. The best performing model was GoogLeNet. The models were tested with uncropped and cropped images. They found out that by using the cropped images and GoogLeNet, they got the best accuracy, which was 96.54%. Additional tests were done by also considering the top two results during the classification process. This further increased the accuracy of the GoogLeNet model with the cropped dataset to 98.94%. Aside from image recognition, several other approaches for automatic identification of marine creatures have been used in the recent past. Demertzis et al. suggested a novel technique: the use of a Machine Hearing Framework (MHF) for the identification of marine animals through their underwater sounds [27]. They were able to recognise fish and marine animals with recognition accuracy of 96.08% and 92.18%, respectively.

Song et al. proposed a method for the identification of marine creatures from seafloor videos [28]. Their proposed methods are twofold: extraction of valid video clips followed by recognition of the creature. During the first phase, an image segmentation method was used to determine and extract all frames from the video containing the marine creature. The next phase was concerned with identifying and labelling the creatures in the valid video clips. This was accomplished with the help of public participation. Lastly was the recognition process, which was accomplished with the help of the information submitted by the public and the membership function. Their method had an accuracy of above 80% in extracting the valid video frames, and all the creatures were successfully recognised.

Liu et al. implemented an embedded system to classify marine animals into seven categories [29]. Their dataset includes 8,455 photos of marine animals, with 80% of the images used for training and 20% used for validation. The training images were augmented by applying some transformations (rotation, translation, and flipping), which increased the number of training images to 27,056 (6,764*4). The models that were deployed on the embedded device were tested with 350 new images. Three models were used: MobileNetV1, MobileNetV2, and InceptionV3. MobileNetV2 had the highest testing accuracy of 95.0% and validation accuracy of 92.89%. Their model took an average of 0.0578 seconds to classify one image.

C. Knowledge Gap

Even though multiple studies have been conducted, very few have tested the effect of using deep learning (DL) on a big dataset. According to the review, the largest dataset had 50 species and consisted of 4,909 images [19]. Furthermore, there is no web application available in Mauritius that can perform recognition of marine animals. Adding to that, no dataset consisting of more than 50 marine creature species is currently available. This work aims to provide some answers to these research gaps as well as to contribute a dataset and a web application to perform marine creature classification.

III. Methodology

A. Data Collection at Odysseo Oceanarium

As no relevant existing dataset of marine creatures from the Indian Ocean was found at the time of this study, a custom dataset was created. Data collection can be done from multiple sources. However, due to time constraints, this is not feasible. Nonetheless, the source should be trustworthy and provide the desired information. In this regard, the Odysseo Oceanarium was chosen as the primary source of data gathering for this study. In recent years, underwater video surveillance has grown increasingly common in marine environments to acquire data on marine creatures in their natural habitat. This is a non-invasive method and provides sufficient data for research. Chen and Yu adopted this approach by using an underwater submerged video system [25]. However, for this study, due to limited resources, videos of marine creatures were taken outside of the aquariums found at Odysseo using a smartphone. All the videos were taken with a Huawei Y9 Prime 2019 smartphone, which has a resolution of 16 megapixels.

B. Data Processing

Numerous videos of marine creatures were obtained at Odysseo. Each video was converted into frames. Pelletier et al. have already shown that cropped images result in better model performance [26]. Taking this into consideration, each extracted frame was carefully cropped. Not all images were included in the final dataset. The following conditions had to be met to use the image, or else it was discarded: the image containing the creature should not be occluded by another creature or object; the creature should be recognisable and it should not be too far from the image. Fig. 1 shows an example of a good image. It follows all the criteria described above.

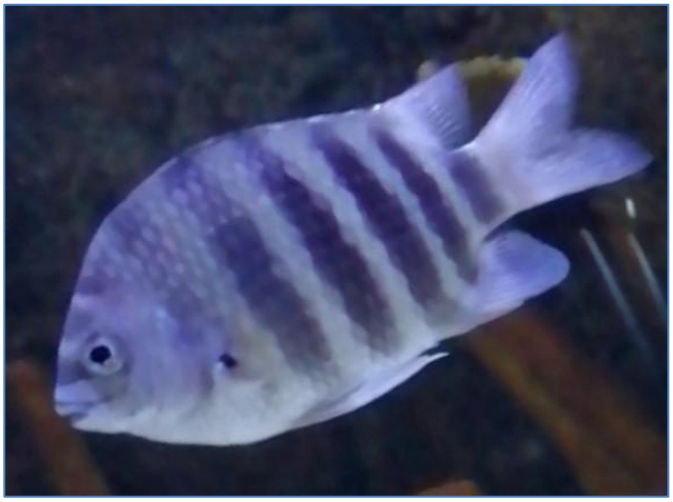


Fig. 1. Example of a good image.

C. Custom Dataset

The custom-built dataset consists of 51 classes of marine creatures as shown in Fig. 2. It has 5,709 images in total and is imbalanced. The class having the highest number of images is Dascyllus aruanus with 171 pictures, and the one with the lowest number of images is Chaetodon kleinii with 74 pictures.

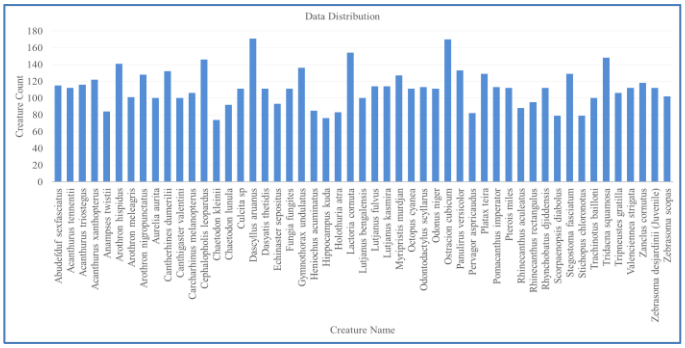


Fig. 2. Unbalanced dataset distribution.

Marine animals are challenging to photograph since they conceal their presence in aquarium by hiding beneath rocks, among plants and tank accessories. This makes it difficult to collect the same number of images for all of the creatures and is the primary reason why the dataset is initially imbalanced.

D. Oversampling

To achieve an equal distribution of images per class, the dataset had to be balanced and for this oversampling was done as shown in Fig. 3. The following transformations were used for augmentation: flipping, change in brightness, shearing and rotation. Each image is subject to three possible modifications.

After oversampling, all classes got an equal distribution of images. Each marine creature now has 171 images. Fig. 4 shows the data distribution of the dataset after oversampling.

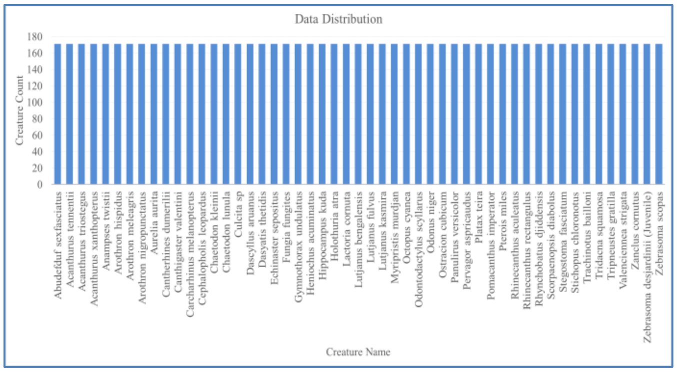


Fig. 3. Distribution after oversampling.

E. Splitting the Dataset

The oversampled dataset (171 images per class) were split using splitting ratios of 8:1:1, 7:2:1 and 6:3:1 for training, validation and testing sets. As a result of multiple manipulations, different dataset versions were created. A proper naming convention was devised to properly organise the work being done. Table I lists the various datasets that were used throughout this paper.

TABLE I. SUMMARY OF DATASET VERSIONS

#	Dataset Name	Description
1	DS_oversample_8_1_1	This is the oversampled dataset, which contains 171 images for each class. Three different splits are done.
2	DS_oversample_7_2_1	
3	DS_oversample_6_3_1	

F. Feature Extraction and Classification

For this research, pre-trained CNN models were used both for feature extraction and classification. The pre-trained models that were employed are: MobileNetV1, InceptionV3 and VGG16. Different image sizes were utilised depending on which model was imported. The image sizes used are shown in Table II [30-31].

TABLE II. IMAGE SIZES FOR DIFFERENT MODELS

Model Name	Image Size
MobileNetV1	224*224
InceptionV3	299*299
VGG16	224*224

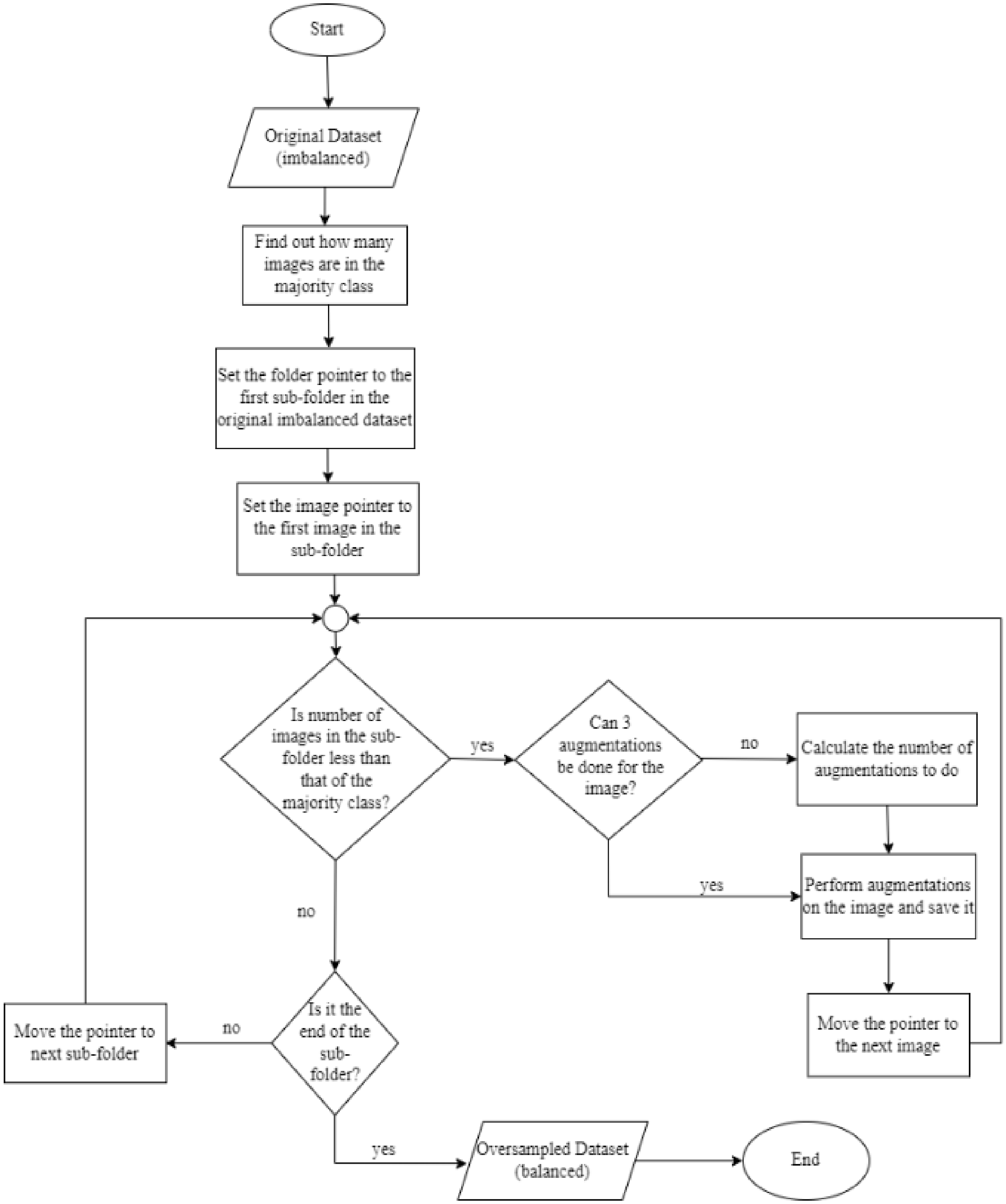
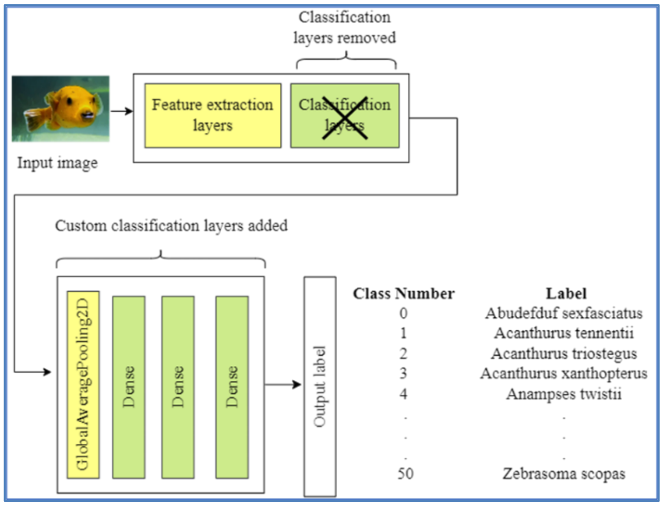

Start
Original Dataset (imbalanced)
Find out how many images are in the majority class
Set the folder pointer to the first sub-folder in the original imbalanced dataset
Set the image pointer to the first image in the sub-folder
Is number of images in the sub-folder less than that of the majority class?
yes
Can 3 augmentations be done for the image?
no
Calculate the number of augmentations to do
yes
Perform augmentations on the image and save it
Move the pointer to the next image
no
Is it the end of the sub-folder?
no
Move the pointer to next sub-folder
yes
Oversampled Dataset (balanced)
End

Fig. 4. Oversampling flowchart.

After the features have been extracted, the next step is to use these features and a classifier to make a prediction. The architecture of the pre-trained models used is shown in Fig. 5. The input image given to the VGG16 and MobileNetV1 models is an image of size 224 * 224 compared to the InceptionV3 model, which is of size 299 * 299.

The classification layers of the pre-trained model were replaced with one global average pooling layer and three dense layers. The softmax activation function was used in the model's final dense layer for classification. Adding to that, for the two other dense layers, the rectified linear activation unit (ReLU) activation function was used. The custom model predicts the input image as one of the 51 classes of marine creatures from the dataset.


Classification layers removed
Input image
Feature extraction layers
Classification layers
Custom classification layers added
GlobalAveragePooling2D
Dense
Dense
Dense
Output label

Class Number	Label
0	Abudefduf sexfasciatus
1	Acanthurus tennentii
2	Acanthurus triostegus
3	Acanthurus xanthopterus
4	Anampses twistii
.	.
.	.
.	.
50	Zebrasoma scopas

Fig. 5. High level architecture of pre-trained model.

G. Use of Callback Functions

Different callback functions, such as ModelCheckpoint and EarlyStopping, were employed during model training. The EarlyStopping callback function is viewed as a technique to combat model overfitting. For this work, the number of epochs was fixed at 100, and then the EarlyStopping function was used to halt training if the model became overfit [32]. The ModelCheckpoint callback function, on the other hand, was used to save the model during the training phase. Failure may occur occasionally, causing the training to be disrupted. It is preferable to resume training from the last saved epoch rather than starting it from scratch [33].

H. Use of Optimiser

The Stochastic Gradient Descent (SGD) optimiser was employed during model training to adjust the weight and learning rate properties of the DL model to reduce losses during backpropagation. Additionally, to help the optimiser converge in the right direction and prevent overshooting, Nesterov momentum was employed. Due to its look ahead property, the Nesterov method takes the appropriate precautions by making smaller updates to reach the minima [34-35]. Experiments were performed using the MobileNetV1 pre-trained model to find the optimal parameters to pass to the SGD optimiser function. We discovered that setting the learning rate to 0.001, decay to 1e-6, and momentum to 0.8 produces the best results. Thus, these parameter values were used throughout this work for model training.

I. Training using Transfer Learning

Transfer learning is a concept whereby a previously trained (pre-trained) model is reused to tackle a new but comparable task. This is a popular deep learning technique as the neural network does not have to be trained from scratch with a huge volume of data. The weight that the network has already learnt is simply transferred to the new task in transfer learning. This technique aids in reducing training time and may possibly increase the performance of the neural network [36]. In this research, transfer learning is used to train the CNN models. Fig. 6 illustrates how transfer learning was applied for model training using the custom-built dataset.

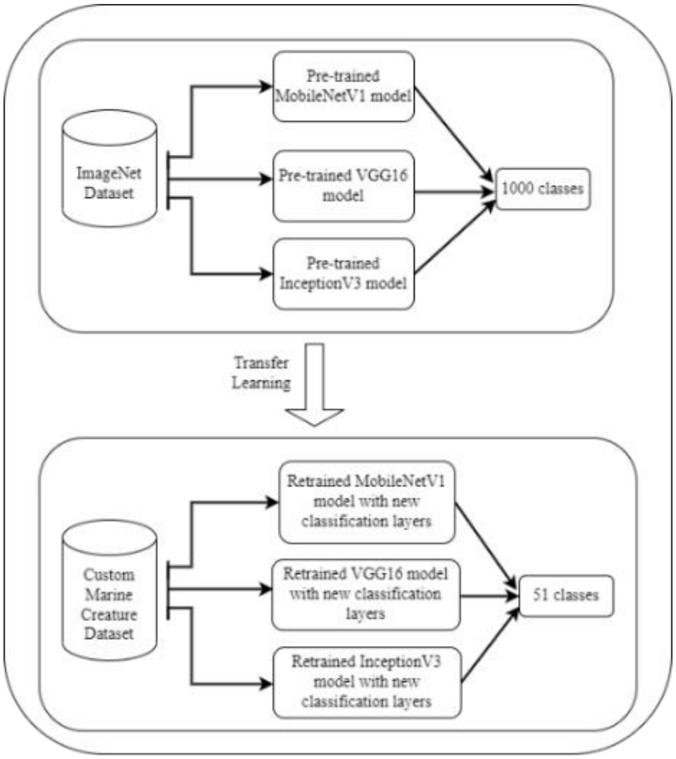

ImageNet Dataset
Pre-trained MobileNetV1 model
Pre-trained VGG16 model
Pre-trained InceptionV3 model
1000 classes
Transfer Learning
Custom Marine Creature Dataset
Retrained MobileNetV1 model with new classification layers
Retrained VGG16 model with new classification layers
Retrained InceptionV3 model with new classification layers
51 classes

Fig. 6. Block diagram of the proposed training strategy.

There are different types of fine-tuning that can be done on pre-trained CNN models, such as training the entire model, training some layers and leaving others frozen, and freezing the convolutional base by not training the feature extraction layers [37]. For this work, each test will be done by training the entire model and freezing the convolutional base.

J. The Web Application

A good and simple user interface is crucial for the user to efficiently use the application. Taking this into consideration, the user interface of the web application is divided into 4 areas: image upload area, prediction area, creatures in dataset area and modal displaying creature information.

1) Image upload area: Fig. 7 shows the image upload area when the application is accessed through a desktop and a mobile phone.

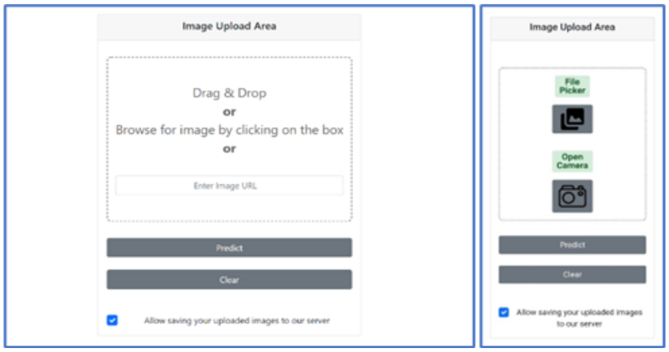
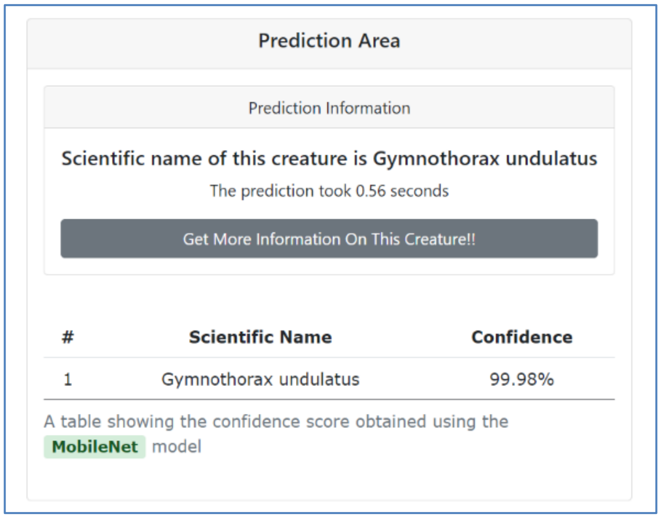


Image Upload Area
Drag & Drop
or
Browse for image by clicking on the box
or
Enter Image URL
Predict
Clear
Allow saving your uploaded images to our server
Image Upload Area
File Picker
Open Camera
Predict
Clear
Allow saving your uploaded images to our server

Fig. 7. Image upload area for desktop view (left) and mobile view (right).

2) Prediction area: The prediction area is illustrated in Fig. 8. It shows the predicted creature, along with a table containing creatures with confidence scores greater than the threshold value.



Prediction Area

Prediction Information

Scientific name of this creature is Gymnothorax undulatus

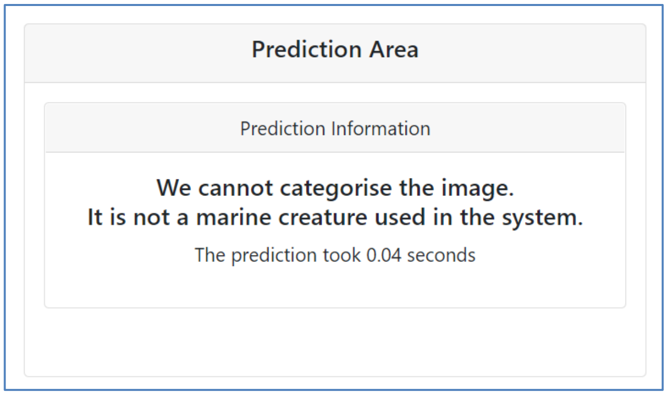
The prediction took 0.56 seconds

#	Scientific Name	Confidence
1	Gymnothorax undulatus	99.98%

A table showing the confidence score obtained using the **MobileNet** model

Fig. 8. Prediction area (system can categorize the image).

If the confidence scores computed for the input image are lower than the threshold, an appropriate message is displayed to the user as shown in Fig. 9. In this context, a threshold value of 0.5 is employed.



Prediction Area

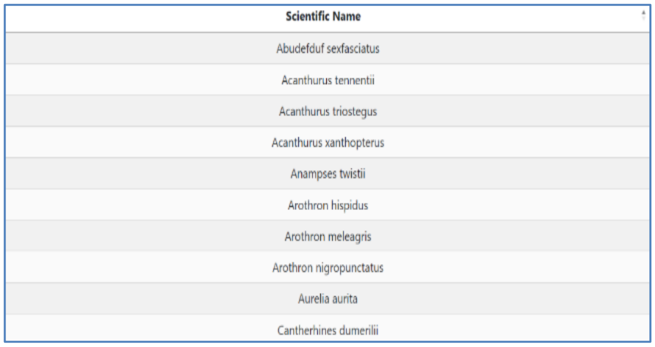
Prediction Information

We cannot categorise the image.
It is not a marine creature used in the system.

The prediction took 0.04 seconds

Fig. 9. Prediction area (system cannot categorize the image).

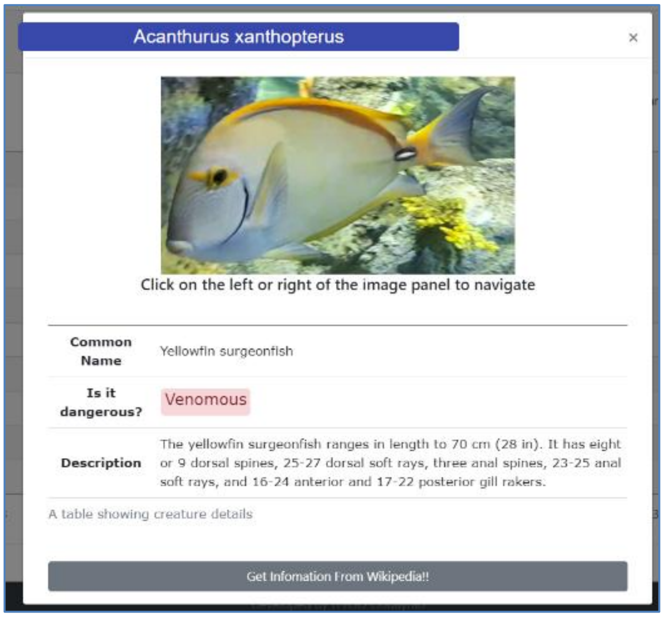
3) Creatures in dataset area: The creatures that the system can classify are shown in Fig. 10.



Scientific Name
Abudefduf sexfasciatus
Acanthurus tennentii
Acanthurus triostegus
Acanthurus xanthopterus
Anampses twistii
Arothron hispidus
Arothron meleagris
Arothron nigropunctatus
Aurelia aurita
Cantherhines dumerilii

Fig. 10. DataTable for displaying creatures from the dataset.

4) Modal displaying creature information: The modal component is used to display information about a creature as shown in Fig. 11.



Acanthurus xanthopterus

Click on the left or right of the image panel to navigate

Common Name	Yellowfin surgeonfish
Is it dangerous?	Venomous
Description	The yellowfin surgeonfish ranges in length to 70 cm (28 in). It has eight or 9 dorsal spines, 25-27 dorsal soft rays, three anal spines, 23-25 anal soft rays, and 16-24 anterior and 17-22 posterior gill rakers.

A table showing creature details

Fig. 11. Modal displaying creature information.

IV. Results

This section provides an evaluation of the different models.

A. Testing Different Split Ratios

Table III shows the three splits for the oversampled dataset.

TABLE III. OVERSAMPLED DATASET SPLIT SUMMARY

Dataset	Split Ratio	Train	Val	Test
DS_oversample_8_1_1	Train: 80% Val: 10% Test: 10%	136	17	18
DS_oversample_7_2_1	Train: 70% Val: 20% Test: 10%	119	34	18
DS_oversample_6_3_1	Train: 60% Val: 30% Test: 10%	102	51	18

The three pre-trained models employed were trained on the three versions of the oversampled dataset. Table IV, Table V and Table VI shows the results obtained.

TABLE IV. DS_OVERSAMPLE_8_1_1 RESULT

Model	Trainable Layers	Classification Accuracy (%)	F1 Score (%)
MobileNetV1	False	99.56	99.56
	True	99.89	99.89
VGG16	False	98.58	98.58
	True	98.91	98.91
InceptionV3	False	97.17	97.16
	True	99.35	99.35

TABLE V. DS_OVERSAMPLE_7_2_1 RESULT

Model	Trainable Layers	Classification Accuracy (%)	F1 Score (%)
MobileNetV1	False	99.67	99.68
	True	99.89	99.89
VGG16	False	98.47	98.48
	True	99.89	99.89
InceptionV3	False	97.60	97.58
	True	99.46	99.45

TABLE VI. DS_OVERSAMPLE_6_3_1 RESULT

Model	Trainable Layers	Classification Accuracy (%)	F1 Score (%)
MobileNetV1	False	98.58	98.58
	True	99.46	99.45
VGG16	False	97.82	97.82
	True	99.02	99.03
InceptionV3	False	96.51	96.52
	True	99.35	99.35

TABLE VII. BEST MODELS FOR OVERSAMPLED DATASET

Dataset	Split Ratio	Best Model	Classification Accuracy (%)	F1 Score (%)
DS_oversample_8_1_1	Train: 80% Val: 10% Test: 10%	MobileNetV1	99.89	99.89
DS_oversample_7_2_1	Train: 70% Val: 20% Test: 10%	MobileNetV1	99.89	99.89
		VGG16	99.89	99.89
DS_oversample_6_3_1	Train: 60% Val: 30% Test: 10%	MobileNetV1	99.46	99.45

The best accuracy obtained in all cases is when the feature extraction layers are trained. Table VII shows the best performing model for each of the different splits.

Irrespective of the split ratios used, the best models were able to achieve very good accuracy of above 99%. The variations in the scores, as indicated in Table VII are less than 1%. This is due to the fact that randomness is used in weight initialization when training of the model starts. The weights are adjusted at every epoch. This produces different outcomes for the same model each time it is trained on the same dataset [38-39]. This means that if the same experiments were repeated, different scores would have been obtained. To conclude, the differences obtained are considered insignificant.

Judging the models solely on accuracy is not enough to give a fair evaluation. The model's prediction time must also be considered. The inference time of the models to predict all the images in their test directory was repeated five times. In Table VIII, the total time taken is displayed in seconds.

Using the values shown in Table VIII the average inference time can be calculated by dividing the time taken to predict all the images by five. The prediction time for one image can then be computed by dividing the resulting value by the number of test pictures as shown in Table IX.

TABLE VIII. PREDICTION TIME

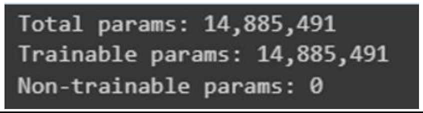
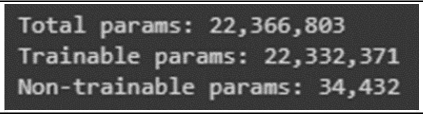
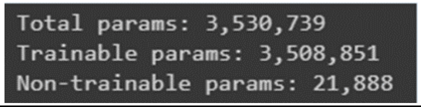
Dataset	Number of Images	MobileNetV1	VGG16	InceptionV3
DS_oversample_8_1_1	918 (18*51)	475.5	2453.5	1002.6

TABLE IX. PREDICTION TIME PER IMAGE

Dataset	MobileNetV1	VGG16	InceptionV3
DS_oversample_8_1_1	0.10	0.53	0.22

From Table IX, it can be seen that the MobileNetV1 models achieved the lowest inference time. Table X shows the number of trainable parameters and the sizes of the three models.

TABLE X. MODEL SIZES

Model	Parameters	Size
VGG16	 Total params: 14,885,491 Trainable params: 14,885,491 Non-trainable params: 0	113.7 MB
InceptionV3	 Total params: 22,366,803 Trainable params: 22,332,371 Non-trainable params: 34,432	171.8 MB
MobileNetV1	 Total params: 3,530,739 Trainable params: 3,508,851 Non-trainable params: 21,888	27.3 MB

Among the three models, MobileNetV1 has the least number of trainable parameters. Furthermore, the MobileNetV1 model is smaller in terms of size. As a result, it has the shortest inference time. For deployment, the MobileNetV1 model trained on the DS_oversample_8_1_1 dataset was chosen since it had the lowest inference time of 0.10 seconds per image.

B. Comparisons with Related Works

Even though the accuracies obtained in this study cannot be truly compared with other researchers because the same datasets were not employed, an attempt to compare our work with previous studies is made in this section.

Deep and Dash conducted several experiments using a dataset of 23 creatures [20]. They used CNN for both feature extraction and classification. Additionally, they used a hybrid strategy in which CNN was used to extract features and a classifier (kNN or SVM) was used to classify them. They got the best accuracy of 98.79% when they used their custom-made CNN for feature extraction and kNN as a classifier. However, in this study, we were able to achieve higher accuracy when the pre-trained CNN models were used for both extraction and classification.

The training methodology adopted by Liu et al. is the same as the one we have used [29]. They also used transfer learning to train their models. They used the MobileNetV1, MobileNetV2 and InceptionV3 models to perform feature extraction and classification. Table XI shows a comparison between the best performing model in their work and in ours.

TABLE XI. COMPARISION OF THE BEST PERFORMING MODEL

Work	Best Model	Classification Accuracy (%)	Inference Time (Seconds)
Ittoo and Pudaruth	MobileNetV1 trained on the DS_oversample_8_1_1 dataset	99.89	0.10
Liu et al. (2019)	MobileNetV2	95.0	0.0578

The MobileNetV2 model presented by Liu et al. is limited to predicting 7 species [29]. However, the model presented in this study can perform classification between 51 creatures. The larger the number of classes in a deep learning model, the more time the model generally takes to predict the image.

V. CONCLUSIONS

Automatic recognition of marine creatures is a topic of interest to many researchers around the world. People who are unfamiliar with marine taxonomy have trouble discriminating between different aquatic organisms. This poses a problem, especially when deadly ocean animals are encountered. Several studies have been undertaken over the last few decades, but not many have examined the effect of training deep learning pre-trained CNN models on a large dataset.

In this research, three deep learning models, namely, MobileNetV1, InceptionV3, and VGG16, were investigated and implemented for the task of marine creature classification. To achieve the objectives of this study, a customised dataset of 51 available creatures from the Indian Ocean was built and used for training and testing the effectiveness of models. Images of marine creatures were collected at Odysseo Oceanarium in Mauritius.

Several experiments with different split ratios were carried out. The splits for the training and validation sets were varied, and that for the testing set was fixed at 10%. Transfer learning was used, and the models were fine-tuned by replacing their classification layers with new ones. Adding to that, two experiments were performed for each model: training the feature extraction layers and not training them. All of these tests were carried out in order to determine the optimal split ratio, dataset, and model.

It has been concluded that the best suited model was MobileNetV1 trained with the oversampled dataset with a split ratio 80% for training, 10% for validation and 10% for testing. The model attained a classification accuracy and an F1 score of 99.89%. The model had an inference time of 0.10 seconds per image. This model was then integrated into a web application.

Our research has thus demonstrated that deep learning models offer enormous potential for automating the process of marine creature recognition. The developed web application with the integrated MobileNetV1 model provides a reliable and fully automated tool for the classification of marine creatures without the need for expert assistance.

ACKNOWLEDGMENT

We are thankful to the Odysseo Oceanarium Mauritius for allowing us to photograph the marine creatures in their aquarium in order to create the dataset utilized in this study.

REFERENCES

[1] BBC Earth. "Blue Planet II: The Prequel" [Video]. YouTube, 27 Sep. 2017. https://www.youtube.com/watch?v=_38JDGnr0vA.

[2] UN Food and Agriculture Organization. "7 reasons why we need to act now to #SaveOurOcean." Medium, 5 Jun. 2017. https://medium.com/@UNFAO/7-reasons-why-we-need-to-act-now-to-saveourocean-70671e3da38d.

[3] Newman, T. "Why scientists are searching the ocean for new drugs." Medical News Today, 8 Jun. 2019. https://www.medicalnewstoday.com/articles/325384#Why-look-to-the-sea?

[4] Stuchtey, M. R., Vincent, A., Merkl, A., Bucher, M., Haugan, P. M., Lubchenco, J., & Pangestu, M. E. "Ocean Solutions That Benefit People, Nature and the Economy." World Resources Institute, 2020. https://oceanpanel.org/wp-content/uploads/2022/06/full-report-ocean-solutions-eng.pdf

[5] Luypaert, T., Hagan, J. G., McCarthy, M. L., & Poti, M. "Status of Marine Biodiversity in the Anthropocene." In S. Jungblut, V. Liebich & M. Bode-Dalby (Eds.), YOUMARES 9 - The Oceans: Our Research, Our Future, pp. 57–82. Springer, Cham, 2020. https://doi.org/10.1007/978-3-030-20389-4_4.

[6] Buckley, M., Fraser, S., Herman, J., Melton, N. D., Mulville, J., & Pálsdóttir, A. H. "Species identification of archaeological marine mammals using collagen fingerprinting." Journal of Archaeological Science, vol. 41, 2014, pp. 631-641. https://doi.org/10.1016/j.jas.2013.08.021.

[7] Goodwin, K. D., Thompson, L. R., Duarte, B., Kahlke, T., Thompson, A. R., Marques, J. C., & Caçador, I. "DNA Sequencing as a Tool to Monitor Marine Ecological Status." Frontiers in Marine Science, vol. 4, 2017, p. 107. https://doi.org/10.3389/fmars.2017.00107.

[8] Trivedi, S., Aloufi, A. A., Ansari, A. A., & Ghosh, S. K. "Role of DNA barcoding in marine biodiversity assessment and conservation: An update." Saudi Journal of Biological Sciences, vol. 23, no. 2, 2020, pp. 167-171. http://dx.doi.org/10.1016/j.sjbs.2015.01.001.

[9] Thomas, V. G., Hanner, R. H., & Borisenko, A. V. "DNA-based identification of invasive alien species in relation to Canadian federal policy and law, and the basis of rapid-response management." Genome, vol. 59, no. 11, 2016, pp. 1023-1031. https://doi.org/10.1139/gen-2016-0022.

[10] Harris, D. J., Rosado, D., & Xavier, R. "DNA Barcoding Reveals Extensive Mislabeling in Seafood Sold in Portuguese Supermarkets." Journal of Aquatic Food Product Technology, vol. 25, no. 8, 2016, pp. 1375-1380. http://dx.doi.org/10.1080/10498850.2015.1067267.

[11] Pudaruth, S., Nazurally, N., Appadoo, C., Kishnah, S., & Chady, F. "SuperFish: A Mobile Application for Fish Species Recognition Using Image Processing Techniques and Deep Learning." International Journal of Computing and Digital Systems, vol. 10, no. 1, 2021, pp. 1157-1165. http://dx.doi.org/10.12785/ijcds/1001104.

[12] Proctor, K. "Burneside man stung by 'deadly' fish on honeymoon." The Westmorland Gazette, 20 Oct. 2011. https://www.thewestmorlandgazette.co.uk/news/9314711.burneside-man-stung-by-deadly-fish-on-honeymoon/.

[13] Shersby, M. "13 of the most venomous sea creatures lurking in the water." Live Science, 28 Aug. 2023. https://www.livescience.com/animals/13-of-the-most-venomous-sea-creatures-on-earth/

[14] Petsko, E. "Why does so much of the ocean remain unexplored and unprotected?" Oceana, 8 Jun. 2020. https://oceana.org/blog/why-does-so-much-ocean-remain-unexplored-and-unprotected/

[15] Strachan, N. J. C., Nesvadba, P., & Allen, A. R. "Fish Species Recognition by Shape Analysis of Images." Pattern Recognition, vol. 23, no. 5, 1990, pp. 539-544. https://doi.org/10.1016/0031-3203(90)90074-U

[16] Chepkemoi, J. "How Many Fish Live In The Ocean?" WorldAtlas, 25 Apr. 2017. https://www.worldatlas.com/articles/how-many-fish-are-there-in-the-ocean.html

[17] Luo, S., Li, X., Wang, D., Li, J., & Sun, C. "Automatic Fish Recognition and Counting in Video Footage of Fishery Operations." 2015 International Conference on Computational Intelligence and Communication Networks (CICN), 2015, pp. 296-299. https://doi.org/10.1109/CICN.2015.66

[18] Rathi, D., Jain, S., & Indu, S. "Underwater Fish Species Classification using Convolutional Neural Network and Deep Learning." 2017 Ninth International Conference on Advances in Pattern Recognition (ICAPR), 2017, pp. 1-6. https://doi.org/10.1109/ICAPR.2017.8593044

[19] Mandal, R., Connolly, R. M., Schlacher, T. A., & Stantic, B. "Assessing fish abundance from underwater video using deep neural networks." 2018 International Joint Conference on Neural Networks (IJCNN), 2018, pp. 1-6. https://doi.org/10.1109/IJCNN.2018.8489482

[20] Deep, B. V., & Dash, R. "Underwater Fish Species Recognition Using Deep Learning Techniques." 2019 6th International Conference on Signal Processing and Integrated Networks (SPIN), 2019, pp. 665-669. https://doi.org/10.1109/SPIN.2019.8711657

[21] Rico-Díaz, Á. J., Rabuñal, J. R., Gestal, M., Mures, O. A., & Puertas, J. "An Application of Fish Detection Based on Eye Search with Artificial Vision and Artificial Neural Networks." Water, vol. 12, no. 11, 2020, p. 3013. https://doi.org/10.3390/w12113013

[22] Liang, J., Fu, Z., Lei, X., Dai, X., & Lv, B. "Recognition and Classification of Ornamental Fish Image Based on Machine Vision." 2020 International Conference on Intelligent Transportation, Big Data & Smart City (ICITBS), 2020, pp. 910-913. https://doi.org/10.1109/ICITBS49701.2020.00201

[23] Cai, K., Miao, X., Wang, W., Pang, H., Liu, Y., & Song, J. "A modified YOLOv3 model for fish detection based on MobileNetv1 as backbone." Aquacultural Engineering, vol. 91, 2020, p. 102117. https://doi.org/10.1016/j.aquaeng.2020.102117

[24] Conrady, C. R., Er, Ş., Attwood, C. G., Roberson, L. A., & Vos, L. d. "Automated detection and classification of southern African Roman seabream using mask R-CNN." Ecological Informatics, vol. 69, 2022, p. 101593. https://doi.org/10.1016/j.ecoinf.2022.101593

[25] Chen, Q., & Yu, H. "Deep sea high-definition camera system based on marine creature classification technology." 2015 IEEE 16th International Conference on Communication Technology (ICCT), 2015, pp. 78-81. https://doi.org/10.1109/ICCT.2015.7399797

[26] Pelletier, S., Montacir, A., Zakari, H., & Akhloufi, M. "Deep Learning for Marine Resources Classification in Non-Structured Scenarios: Training vs. Transfer Learning." 2018 IEEE Canadian Conference on Electrical & Computer Engineering (CCECE), 2018, pp. 1-4. https://doi.org/10.1109/CCECE.2018.8447682

[27] Demertzis, K., Iliadis, L. S., & Anezakis, V. D. "Extreme deep learning in biosecurity: the case of machine hearing for marine species identification." Journal of Information and Telecommunication, vol. 2, no. 4, 2018, pp. 492-510. https://doi.org/10.1080/24751839.2018.1501542

[28] Song, X., Guo, Y., Zhang, F., Wang, S., Yang, J., & Chang, Y. "Research on marine creature recognition from seafloor videos." OCEANS 2019 - Marseille, France, 2019, pp. 1-4. https://doi.org/10.1109/OCEANSE.2019.8867479

[29] Liu, X., Jia, Z., Hou, X., Fu, M., Ma, L., & Sun, Q. "Real-time Marine Animal Images Classification by Embedded System Based on Mobilenet and Transfer Learning." OCEANS 2019 - Marseille, France, 2019, pp. 1-5. https://doi.org/10.1109/OCEANSE.2019.8867190

[30] Howard, A. G., Zhu, M., Chen, B., Kalenichenko, D., Wang, W., Weyand, T., Andreetto, M. & Adam, H. "MobileNets: Efficient Convolutional Neural Networks for Mobile Vision Applications." arXiv, 2017. https://doi.org/10.48550/arXiv.1704.04861

[31] Rosebrock, A. "ImageNet: VGGNet, ResNet, Inception, and Xception with Keras." PyImageSearch, 20 Mar. 2017. https://pyimagesearch.com/2017/03/20/imagenet-vggnet-resnet-inception-xception-keras/

[32] Chen, B. "Early Stopping in Practice: an example with Keras and TensorFlow 2.0." Towards Data Science, 29 Jul. 2020. https://towardsdatascience.com/a-practical-introduction-to-early-stopping-in-machine-learning-550ac88bc8fd

[33] Janapati, V. "A High Level Overview of Keras ModelCheckpoint Callback." Medium, 31 Oct. 2020. https://medium.com/swlh/a-high-level-overview-of-keras-modelcheckpoint-callback-deae8099d786

[34] Chandra, A. L. "Learning Parameters, Part 2: Momentum-Based & Nesterov Accelerated Gradient Descent." Towards Data Science, 15 May 2019. https://towardsdatascience.com/learning-parameters-part-2-a190bef2d12

[35] Doshi, S. "Various Optimization Algorithms For Training Neural Network." Towards AI, 13 Jan. 2019. https://towardsdatascience.com/optimizers-for-training-neural-network-59450d71caf6

[36] Sharma, P. "Understanding Transfer Learning for Deep Learning." Analytics Vidhya, 1 Mar. 2021. https://www.analyticsvidhya.com/blog/2021/10/understanding-transfer-learning-for-deep-learning/

[37] Marcelino, P. "Transfer learning from pre-trained models." Towards Data Science, 23 Oct. 2018. https://towardsdatascience.com/transfer-learning-from-pre-trained-models-f2393f124751

[38] Brownlee, J. "How to Get Reproducible Results with Keras." Machine Learning Mastery, 14 Jun. 2017. https://machinelearningmastery.com/reproducible-results-neural-networks-keras/

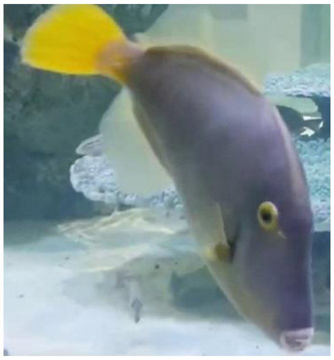
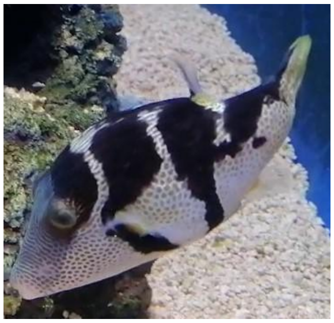
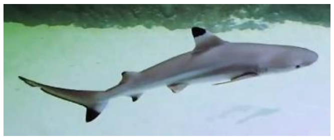
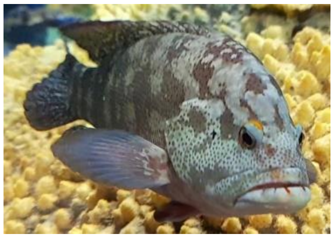
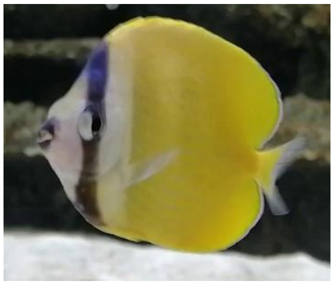
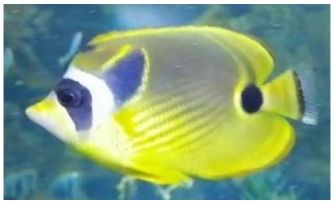
[39] Sharma, P. "Basic Introduction to Convolutional Neural Network in Deep Learning." Analytics Vidhya, 1 Mar. 2022. https://www.analyticsvidhya.com/blog/2022/03/basic-introduction-to-convolutional-neural-network-in-deep-learning/

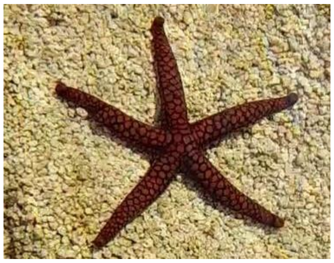
APPENDIX I

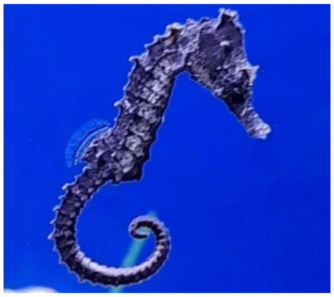
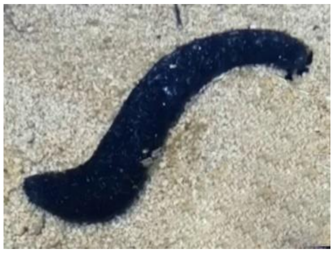
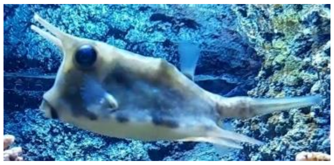
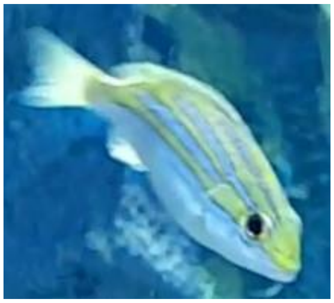
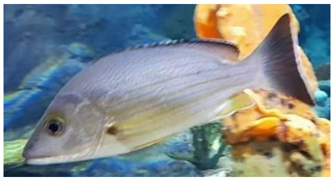
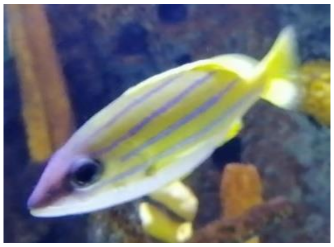
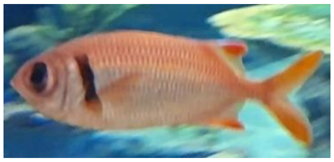
TABLE XII. LIST OF ALL MARINE CREATURES

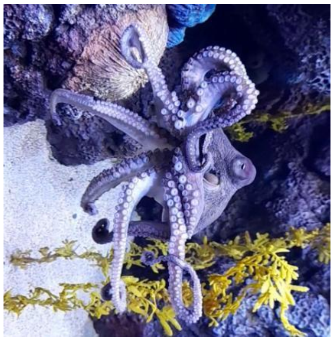
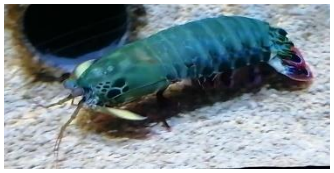
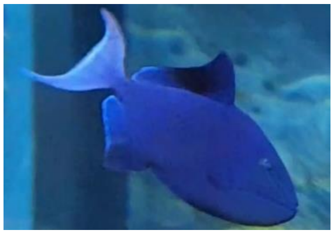
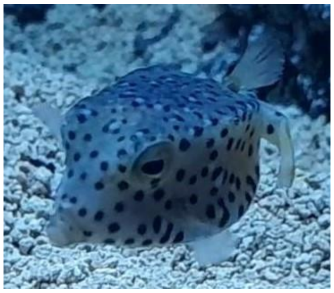
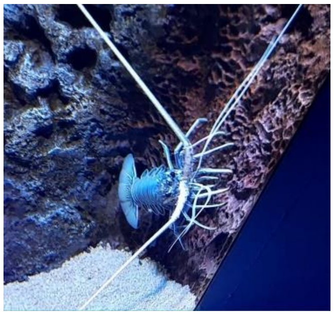
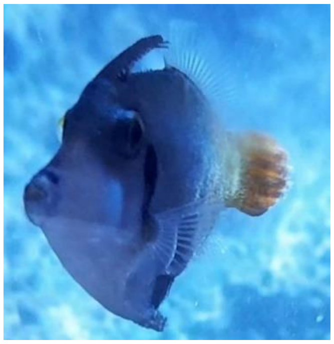
#	Common Name	Scientific Name	Image	Number of Images
1	Scissortail sergeant	Abudefduf sexfasciatus		115
2	Doubleband surgeonfish	Acanthurus tennentii		112

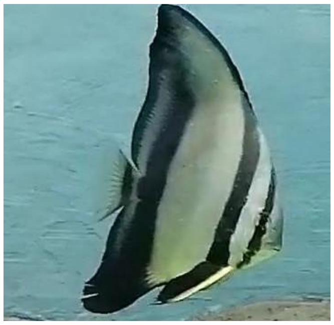
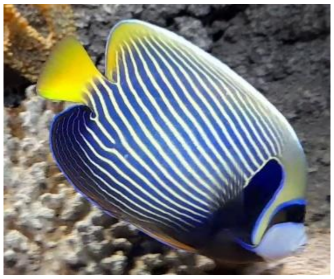
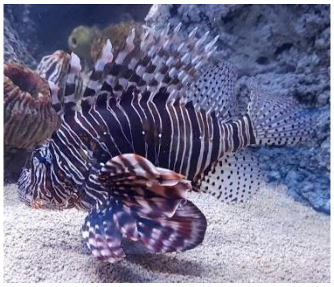
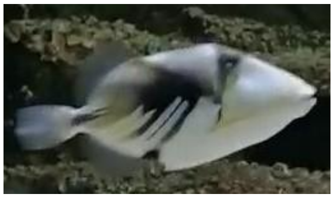
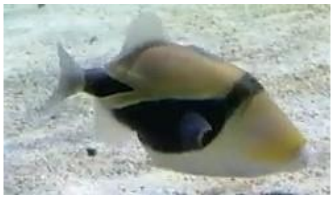
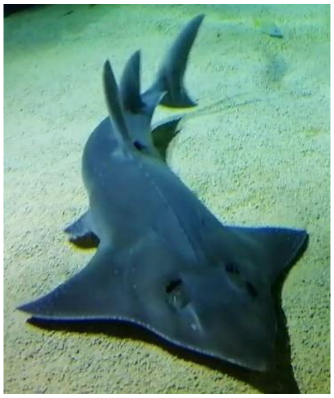
3	Convict surgeonfish	Acanthurus triostegus		116
4	Yellowfin surgeonfish	Acanthurus xanthopterus		122
5	Yellowbreasted wrasse	Anampses twistii		84
6	White-spotted puffer	Arothron hispidus		141
7	Guineafowl puffer	Arothron meleagris		101
8	Blackspotted puffer	Arothron nigropunctatus		128
9	Common jellyfish	Aurelia aurita		100

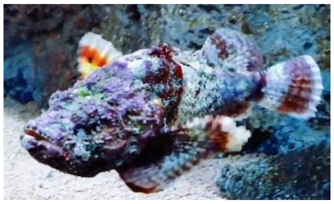
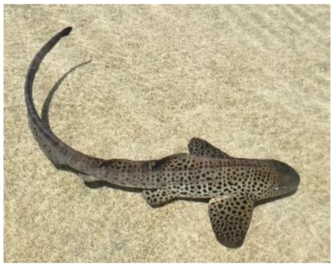
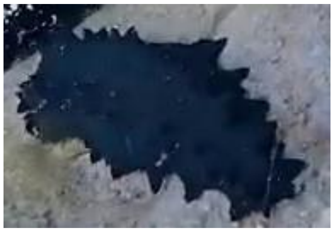
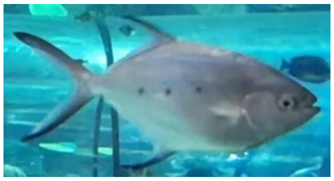
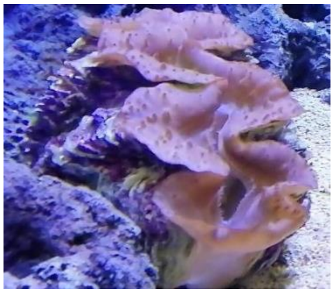
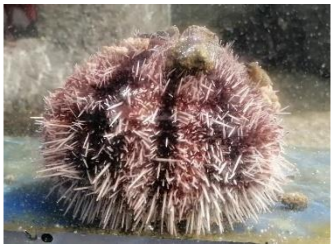
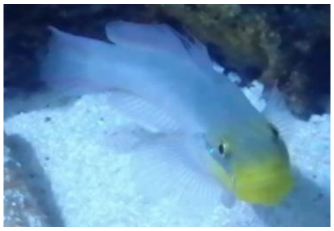
10	Whitespotted filefish	Cantherhines dumerilii		132
11	Valentin's sharpnose puffer	Canthigaster valentini		100
12	Blacktip reef shark	Carcharhinus melanopterus		106
13	Leopard hind	Cephalopholis leopardus		146
14	Sunburst butterflyfish	Chaetodon kleinii		74
15	Raccoon butterflyfish	Chaetodon lunula		92

16	Cushion star	Culcita sp		111
17	Whitetail dascyllus	Dascyllus aruanus		171
18	Thorntail stingray	Dasyatis thetidis		111
19	Red starfish	Echinaster sepositus		93
20	Common mushroom coral	Fungia fungites		111
21	Undulated moray	Gymnothorax undulatus		136
22	Pennant coralfish	Heniochus acuminatus		85

23	Spotted seahorse	Hippocampus kuda		76
24	Black sea cucumber	Holothuria atra		83
25	Longhorn cowfish	Lactoria cornuta		154
26	Bengal snapper	Lutjanus bengalensis		100
27	Blacktail snapper	Lutjanus fulvus		114
28	Common bluestripe snapper	Lutjanus kasmira		114
29	Pinecone soldierfish	Myripristis murdjan		127

30	Big blue octopus	Octopus cyanea		111
31	Peacock mantis shrimp	Odontodactylus scyllarus		113
32	Red-toothed triggerfish	Odonus niger		111
33	Yellow boxfish	Ostracion cubicum		170
34	Painted spiny lobster	Panulirus versicolor		133
35	Orangetail filefish	Pervagor aspricaudus		82

36	Longfin batfish	Platax teira		129
37	Emperor angelfish	Pomacanthus imperator		113
38	Devil firefish	Pterois miles		112
39	White-banded triggerfish	Rhinecanthus aculeatus		88
40	Wedge-tail triggerfish	Rhinecanthus rectangulus		95
41	Giant guitarfish	Rhynchobatus djiddensis		112

42	False stonefish	Scorpaenopsis diabolus		79
43	Zebra shark	Stegostoma fasciatum		129
44	Greenfish sea cucumber	Stichopus chloronotus		79
45	Small spotted dart	Trachinotus bailloni		100
46	Fluted giant clam	Tridacna squamosa		148
47	Striped sea urchin	Tripneustes gratilla		106
48	Blueband goby	Valenciennea strigata		112

49	Moorish idol	Zanclus cornutus		118
50	Indian sail-fin surgeonfish	Zebrasoma desjardinii (Juvenile)		112
51	Twotone tang	Zebrasoma scopas		102