Fish Species Detection using Convolutional Neural Network

1st Deeksha Sareen
Faculty of Computer Science
Dalhousie University
Halifax, NS, Canada
dk930654@dal.ca

2nd Gurleen Kaur Saluja
Faculty of Computer Science
Dalhousie University
Halifax, NS, Canada
gr997570@dal.ca

3rd Manraj Singh
Faculty of Computer Science
Dalhousie University
Halifax, NS, Canada
mn697903@dal.ca

***Abstract* – In this paper, we discuss the convolutional neural network built to detect the species of fish given a dataset with fish species and images. We performed certain tasks such as data preprocessing and augmentation. Based on the analysis we did on the dataset we built a CNN to detect the fish species based on the images. We performed parameter tuning and applied several optimization techniques on the given dataset to derive efficient results.**

***Keywords* – Image Classification, Convolutional Neural Networks (CNN), Deep Learning, Fish Species, Optimization**

I. INTRODUCTION

Due to the dwindling populations of the marine wildlife, underwater research of marine animals and plants continues to gain momentum. There has also been an increase in the availability of instruments that are able to capture and record underwater images of these marine animals. This has further boosted the research in marine life and has led to a massive increase in the collection of underwater image data of plants and animals. Machine Learning continues to be at the frontier of image processing and can thus be used to automate the whole process. Underwater fish recognition and classification has gained importance due to these research advancements in marine life. Thus, automating fish recognition and classification would aid the academic researchers and ocean scientists advance in their research.

The use of deep learning techniques, in particular, CNNs or Convolutional Neural Networks have been the most popular in image classification in terms of performance. The goal of this project is to build a CNN model that will be able to classify fish images into their respective species. The data utilized is collected from videos and images using underwater cameras. The aim is to create such a model whose findings can add value to the ongoing research works on fishes by accurately and efficiently classifying cropped fish images.

II. RELATED WORK

The idea of this project is to detect the species of fish from the image dataset. For this, we will use a dataset that has multiple images per species. The major challenge is the similar fish species which are visually very similar to each other and have very minute differences. The unconstrained underwater scenes are also a big challenge due to the change in fish orientation due to movement, background habitat variety, light intensity changes.

In the related work we found online, we can see various solutions possible as follows:

- To create 3D fish models in which features like height, width and thickness are considered. This model was considered to remove the background objects from the data like coral reefs, kelp, seagrass, beds, and other aquatic plants including sessile invertebrates such as sponges, gorgonian, and ascidians. This was also used to ignore the seafloor structure. For the fish classification in the natural environment using image processing and fetching texture pattern and shape of the fish. But in this model, the fish with distinguishable rich texture and shape were targeted.

- Principal Component Analysis (PCA) or Linear Discriminant Analysis (LDA) was used in this model to distinguish the species and classify them. But this approach expected the appearance of each species, and the background environment is linearly independent of each other. This assumption creates problems in the real sea-like environment as the species are similar to each other and the background reefs and plant lives can also be detected as fish. Sparse Representation based classification was used with this model to increase the accuracy and the ability to distinguish the species, but the model's linear nature is unable to model non-linear differences between fish species and their complex background.
- Another approach for the classification of species without constraints is using a hierarchical classification tree with Support Vector Machines (SVM). The decision-making is based on Gaussian Mixture Modeling (GMM). The results shown by this model are better than the PCA model proposed earlier.
- Multilayer deep neural network provides the opportunity to extract the unique, dependent, fish-oriented features, even in the presence of the background noise and the variabilities in the images. This solution proposes to use the Convolution Neural Network (CNN) mapped with K-Nearest Neighbors (KNN) and Support Vector Machine (SVM) trained on the features extracted using CNN under supervised deep learning.

Each solution discussed above is to classify the fish in the marine and distinguish the features of fish from the background noise like coral reefs, kelp, seagrass, beds, and other aquatic plants, etc.

Models and their advantages and disadvantages:

- Using Principal Component Analysis (PCA) or Linear Discriminant Analysis (LDA) for the fish detection gives high accuracy with the fish which are unique in shape and texture. But for similar-looking fish species, this model tends to give the same results. It's easy to detect fish in a constrained environment with limited species, but in the natural environment, the accuracy of this model solution drops.
- In the next solution, the model is created using Support Vector Machines (SVM) and decisions are made using GMM. In this approach, the model tends to work better than the previous solution. It helps in removing the background noise and fetching the features of the fish. But the limitation is that it cannot distinguish between the fish with similar features. The accuracy drops when there are similar-looking species.
- The final solution and the one we implemented in this project is using a multi-layered deep neural network. Which uses CNN mapped with KNN to extract the features and these features are then used as input to train the SVM model. The advantage of using this approach is that this model automatically extracts the features from the images using

convolutional layers. This automatically removes the background noise and records the small features which help in the classification of similar-looking fish species.

The solution for the final model is to use the convolution neural network in the hierarchical feature combination set up for training the model with the species-dependent visual features which are unique to that species and yet abstract and robust in comparison to the environmental impact on the images of different species and, inter-species variability. This avoids the need to first extract the features from raw data of fish images by implementing a few fragmented image processing techniques. This eventually gives us a single and generic trained architecture with good performance on the dataset or the images which are not even used in the training. This provides a model with low Variance and low Bias even without preprocessing or cropping of the training or testing dataset images. In the paper, the model implementation shows 90% of the accuracy with this solution.

III. METHODOLOGY

Dataset: The QUT FISH Species dataset that is being used in this project consists of 3,960 images collected from over 468 fish species [1]. The dataset has been sourced from Kaggle. Data consists of real-world images of fish captured in 3 types of conditions:

1. controlled
2. out-of-the-water
3. in-situ

These images form the *raw_images* folder of the dataset. The "controlled" condition images consists of fishes with their fins spread, taken against a constant background with controlled illumination. The "in-situ" images are underwater images of fish in their natural habitat and so there is no control over background or illumination. While the "out-of-the-water" images consist of fish specimens, taken out of the water with varying background and illumination conditions. The images are then cropped and included in the *cropped* folder for data augmentation. Cropped images are utilized in this model to avoid overfitting problems. *Figure 1* shows a snippet of the images in the dataset.

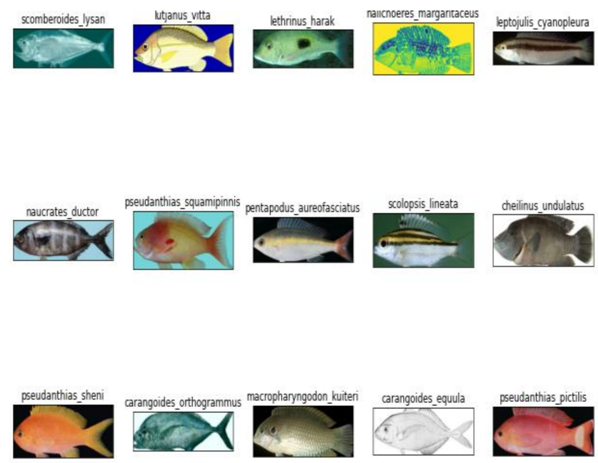

Figure 1: Cropped Image Fish data

Preprocessing: The dataset is loaded in python using pickle library. The labels corresponding to the images are fetched and are placed in label column. The label column is then reshaped to match the images. Further, data augmentation was performed which is discussed in the later sections.

Convolution Neural Network: The Convolution Neural Network model is used for object identification in the image dataset. These objects can be images of animals, fish, text, etc. This Neural network has multiple layers associated with it as shown in the *Figure 2***.** The input to this multi-layer model is the training dataset images. This input data is subjected to a rotating kernel matrix and the computation layer is derived during cross-correlation. An activation operation is performed right after the convolution layer. Followed by a pooling layer as shown in the *Figure 2***.** There can be N total layers in between the First and last layers. Each of these layers comprises a convolution and pooling layer combination.

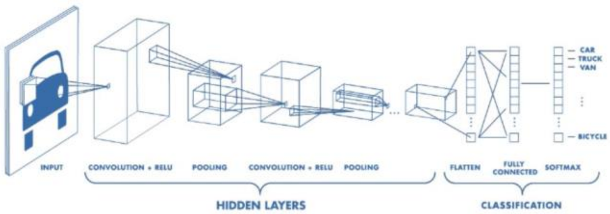

Figure 2: Convolution Neural Network [4]

The dataset was split into training, testing and validation datasets. We chose accuracy as the evaluation metrics because in this project the primary task is object identification and features extraction. Accuracy determines the percentage of correct predictions our model makes in comparison to the total test cases.

In our project, we used two hidden layers comprising convolution and max-pooling layers. The maximum accuracy attained after implementing these layers was 51.4% which was dropping when we used three hidden layers instead of two. The Figure3 shows the convolution layers used in this project. In the convolution layer, the input image is resized and fed as an input to the convolution layer. For this task, there is a kernel that covers and slides over the image while multiplying its values to those covered pixels of the original image. In this project, we used multiple convolution layers. The use of 2nd layer describes the location of the low-level features in the original image. This helps in comparing the small feature differences between species that are very close to each other.

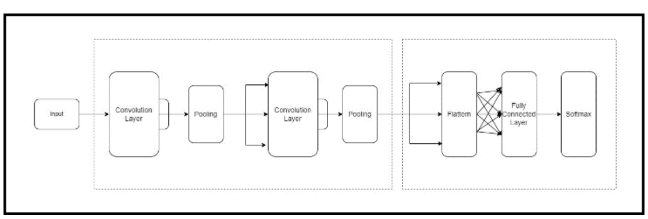

Figure 3: Project Model Architecture [5]

The sample data is as shown in the *Figure 1*. After implementing our model on this data, the output is as shown in the *Figure 6*. This output image shows how the features are highlighted and now these features are easy to read and distinguish the species with.

IV. EXPERIMENTS

We have used the fish species image dataset that contained approximately 4000 images of 468 fish species [1]. We have used the cropped version of images for our model. We conducted experiments where we analyzed the performance of our model by varying the optimizer, batch normalization and learning rate for the model. The dataset we used was imbalanced as can be seen from the image below in *Figure 4*. There were some species with as less as three fish while some species had as many as 319 fish. Due to the imbalance in the dataset we tried to perform oversampling [2] and for this we applied the concept of data augmentation to increase the size of our training dataset in order to improve its accuracy at the end of model building. *Figure 5* shows the snippet of a fish image that has been augmented. The images were rotated, zoomed in and out and resized as a part of augmentation.

[('lutjanus', 319), ('epinephelus', 288), ('halichoeres', 213), ('pseudanthias', 187), ('lethrinus', 160), ('cephalopholis', 133), ('thalassoma', 130), ('bodianus', 125), ('coris', 92), ('cirrhilabrus', 87), ('choerodon', 84), ('anampses', 78), ('carcharhinus', 74), ('scolopsis', 71), ('carangoides', 67), ('stethojulis', 64), ('caranx', 59), ('cheilinus', 59), ('cantherhines', 58), ('plectropomus', 50), ('pseudocheilinus', 46), ('pervagor', 45), ('macropharyngodon', 41), ('oxycheilinus', 41), ('sphyraena', 39), ('pristipomoides', 37), ('bothus', 35), ('variola', 34), ('aphareus', 33), ('meuschenia', 33), ('plectranthias', 32), ('hemigymnus', 31), ('labroides', 31), ('aluterus', 30), ('hologymnosus', 27), ('oxymonacanthus', 26), ('pseudolabrus', 25), ('trachinotus', 25), ('macolor', 24), ('lniistius', 23), ('nemipterus', 23), ('pteragogus', 23), ('acanthopagrus', 22), ('gymnocranius', 22), ('monotaxis', 22), ('seriola', 22), ('labropsis', 20), ('leptojulis', 20), ('suezichthys', 20), ('amanses', 19), ('paraluteres', 19), ('cheilodactylus', 18), ('etelis', 18), ('P1ROZC-Z', 17), ('cheilio', 17), ('notolabrus', 17), ('pseudodax', 17), ('wetmorella', 17), ('acanthaluteres', 16), ('elagatis', 16), ('liopropoma', 16), ('pseudorhombus', 16), ('sillago', 16), ('anyperodon', 15), ('gnathanodon', 15), ('gracila', 15), ('liza', 15), ('scomberoides', 15), ('auxis', 14), ('epibulus', 14), ('pentapodus', 14), ('valamugil', 13), ('achoerodus', 12), ('chromileptes', 12), ('novaculichthys', 12), ('pseudalutarius', 12), ('thryssa', 12), ('aethaloperca', 11), ('chirocentrus', 11), ('eubalichthys', 11), ('gnathodentex', 11), ('monacanthus', 11), ('pseudocaranx', 11), ('acanthistius', 10), ('argyrops', 10), ('hyporhamphus', 10), ('labrichthys', 10), ('novaculoides', 10), ('ophthalmolepis', 10), ('plotosus', 10), ('scomberomorus', 10), ('thunnus', 10), ('CUNWCB-Y', 9), ('crenimugil', 9), ('cymolutes', 9), ('mugim', 9), ('sardinops', 9), ('scaevius', 9), ('selar', 9), ('thysanophrys', 9), ('triaenodon', 9), ('xiphocheilus', 9), ('alectis', 8), ('decapterus', 8), ('euthynnus', 8), ('katsuwonus', 8), ('otolithes', 8), ('pardachirus', 8), ('platycephalus', 8), ('rastrelliger', 8), ('rhabdosargus', 8), ('serranocirrhitus', 8), ('zeus', 8), ('A73EGS-P', 7), ('PQV7DP-S', 7), ('anodontostoma', 7), ('brachaluteres', 7), ('caprodon', 7), ('dactylophora', 7), ('gymnosarda', 7), ('pagrus', 7), ('paracheilinus', 7), ('promethichthys', 7), ('pseudojuloides', 7), ('ruvettus', 7), ('sarda', 7), ('sardinella', 7), ('seriolina', 7), ('symphorichthys', 7), ('Istiophorus', 6), ('acreichthys', 6), ('alepes', 6), ('aprion', 6), ('atule', 6), ('beryx', 6), ('caesioperca', 6), ('dotalabrus', 6), ('herklotsichthys', 6), ('johnius', 6), ('megalaspis', 6), ('naucrates', 6), ('notorynchus', 6), ('rhizoprionodon', 6), ('stegostoma', 6), ('trachichthys', 6), ('trachypoma', 6), ('atractoscion', 5), ('brachirus', 5), ('centroberyx', 5), ('cynoglossus', 5), ('evistias', 5), ('hemiramphus', 5), ('oedalechilus', 5), ('paramonacanthus', 5), ('paraplagusia', 5), ('pinjalo', 5), ('retropinna', 5), ('selaroides', 5), ('wattsia', 5), ('cantheschenia', 4), ('centrogenys', 4), ('centroscymnus', 4), ('cyttopsis', 4), ('diproctacanthus', 4), ('eupetrichthys', 4), ('gempylus', 4), ('harriotta', 4), ('inegocia', 4), ('latridopsis', 4), ('lepidocybium', 4), ('nemadactylus', 4), ('nibea', 4), ('parastromateus', 4), ('protonibea', 4), ('psettodes', 4), ('pseudocarcharias', 4), ('rhincodon', 4), ('soleichthys', 4), ('symphorus', 4), ('uraspis', 4), ('aesopia', 3), ('aseraggodes', 3), ('bathylagichthys', 3), ('chascanopsetta', 3), ('chelidonichthys', 3), ('cymbacephalus', 3), ('negaprion', 3), ('netuma', 3), ('paracaesio', 3), ('samaris', 3), ('samariscus', 3), ('stolephorus', 3), ('zenarchopterus', 3)]

Figure 4: Imbalanced dataset with species count

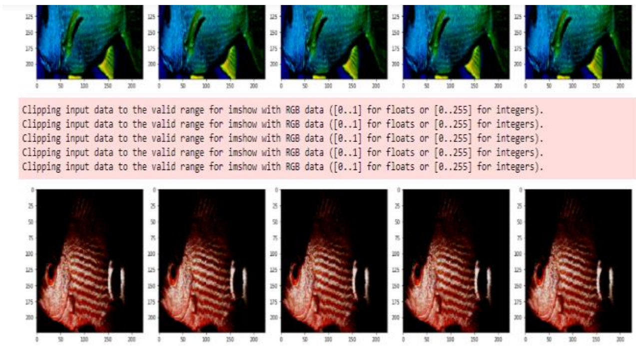

Figure 5: Output of Data augmentation

We began with creating a two-layered CNN with two sets of a convolutional and a max pooling layer which were then flattened and were then passed to the activation function. With Adam optimizer we were able to achieve an accuracy of approximately 51.92% with a learning rate of 1e-3. We have chosen accuracy as a metric of evaluation because we wanted to identify the performance of CNN model and accuracy seemed an appropriate measure to monitor the model's performance. *Figure 6* shows the visualization of the model with best performance with respect to the layers added in our model.

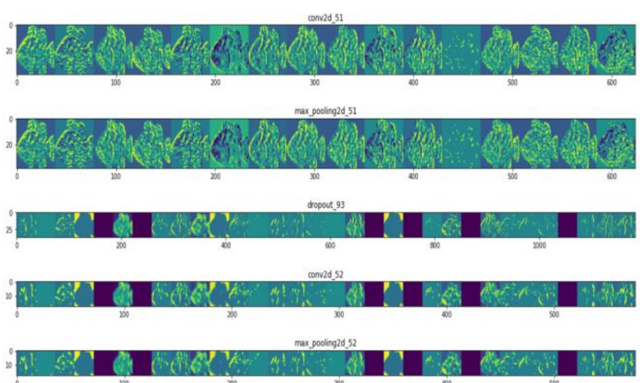

Figure 6: Visualization of CNN model

Below *figures 5, 6 and 7* represents the model accuracy and loss as derived.

```
# Deriving final loss
final_loss, final_acc = model.evaluate(X_test, y_test, verbose=0)
print("Final loss: {0:.4f}, final accuracy: {1:.4f}".format(final_loss, final_acc))

Final loss: 2.0764, final accuracy: 0.5192
```

Figure 7 Final model accuracy

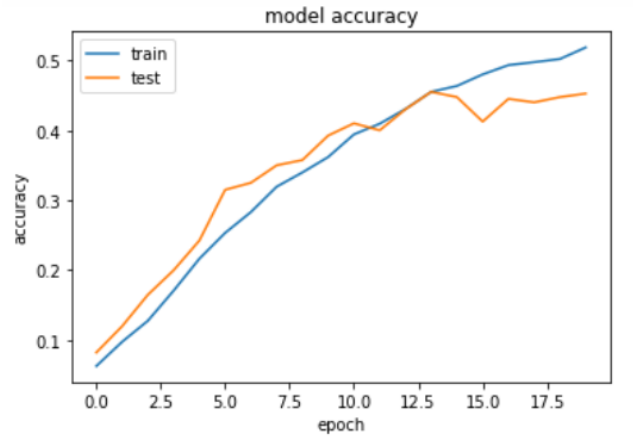

Figure 8 Model accuracy output using Adam optimizer

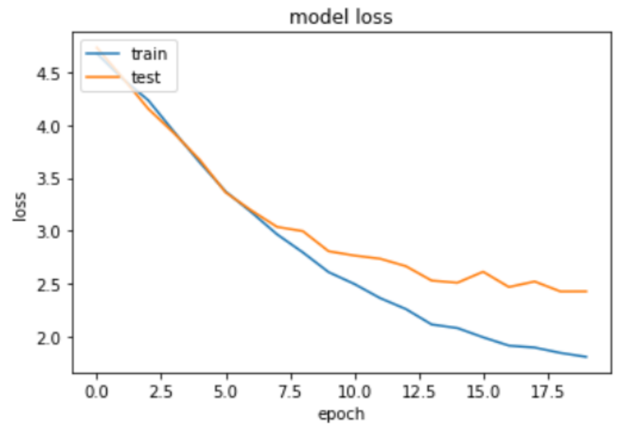

Figure 9 Model loss output using Adam optimizer

By changing certain parameters such as by changing the learning rate on this model we were able to slightly improve the model accuracy and reduce model loss on the test set as shown in the figures below.

We tested our model with RMSprop and SGD optimizers as well. With RMSprop the accuracy significantly dropped to 20% on the test set while the loss was minimized as shown in the figures below (8, 9).

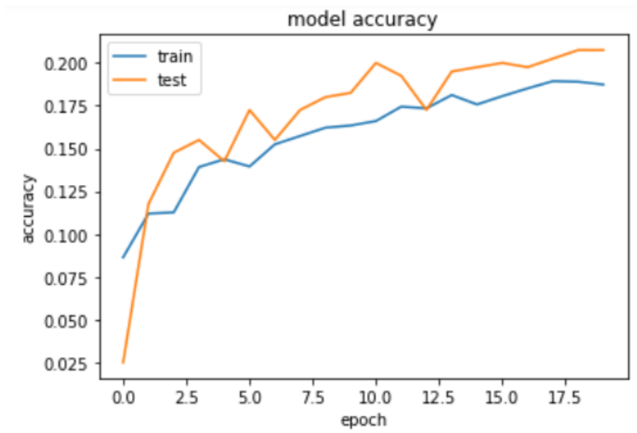

Figure 10 Model accuracy output using RMSprop optimizer

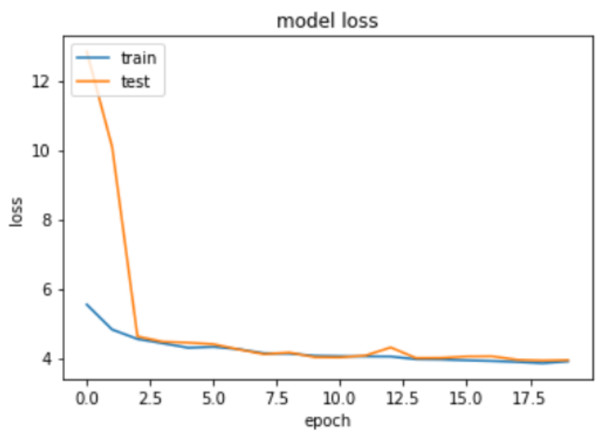

Figure 11 Model loss output using RMSprop optimizer

With SGD the model did not perform quite well on the training data while worked better with the test set, which we believe might be a case of underfitting. The loss was significantly minimized.

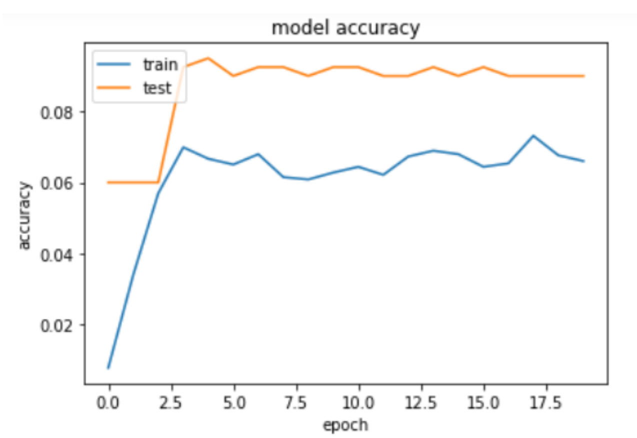

Figure 12 Model accuracy output using SGD optimizer

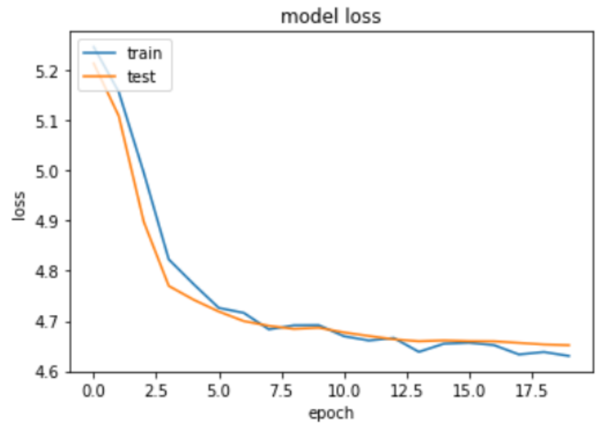

Figure 13 Model loss output using SGD optimizer

We then changed the number of layers in the model which yielded an accuracy of about 35%. Then we added batch normalization layer which yielded the accuracy of about 38%. Further, we modified the learning rate for the model to be 3e-4 where the accuracy turned out to be approximately 51.17%. The highest accuracy was achieved with the model with Adam optimizer, two hidden layers and learning rate as 1e-4 with 51.92%.

III. CONCLUSION AND FUTURE WORK

The CNN model proves its ability to efficiently classify fishes based on their species with an accuracy of 51.92%. The model is fast and can also be built on low resources. CNNs are also better than the classic neural networks with fully connected layers (FC) because they automatically detect the image features i.e., the pixel data without much human intervention. Furthermore, we observed that the model built with two layers and Adam optimizer performed better than other compared models.

Future work can include improving the accuracy of this model and studying the impacts of learning rate on this model as the learning rate 1e-4 and 3e-4 gave nearly same accuracies. Tuning the parameters for these models can add to the scope of future work. Once the accuracy is dealt with, this model can then be used to classify other categories of marine life like corals and plantae as well. Other machine learning models can be created and experimented upon to be used in conjunction with CNN model to increase efficacy, accuracy, and performance. The current classification in the model is done on static dataset. It can be expanded to dynamic datasets.

IV. REFERENCES

[1] S. Srinivasan, "Kaggle," December 2021. [Online]. Available: https://www.kaggle.com/sripaadsrinivasan/fish-species-image-data.

[2] A. Lo Duca, "Towards Data Science," [Online]. Available: https://towardsdatascience.com/how-to-balance-a-dataset-in-python-36dff9d12704.

[3] Salman, A., Jalal, A., Shafait, F., Mian, A., Shortis, M., Seager, J. and Harvey, E. (2016), Fish species classification in unconstrained underwater environments based on deep learning. Limnol. Oceanogr. Methods, 14: 570-585. https://doi-org.ezproxy.library.dal.ca/10.1002/lom3.10113

[4] C. C. Chaterjee, "Towards Data Science," [Online]. Available: https://towardsdatascience.com/basics-of-the-classic-cnn-a3dce1225add.

[5] Draw.io. [Online]. Available: https://draw.io.