Fish Detection and Classification using Mask R-CNN

Monisha K Naik[1], Prof. Rekha B S[2]

[1]*PG student, Department of Information Science and Engineering, RV College of Engineering® Bengaluru, India*
[2]*Assistant Professor, Department of Information Science and Engineering, RV College of Engineering® Bengaluru, India*

Abstract: In this paper we look at Mask RCNN as a mechanism to automatically detect, classify and mask fish in underwater imagery. We describe the outcome of an analysis of a huge dataset of underwater pictures collected by the National Conservancy Organization using camera feeds gathered from boat cameras. The data is diversified; it lacks rotational symmetry across habitats, has particularly large shadows in comparison to the organisms under examination, and large occlusions and objects that are small and not centred in comparison to the whole field of view. The method provided here detects objects in an image efficiently while also producing a high-quality segmentation mask for each instance. Faster R-CNN is extended by Mask R-CNN, which adds a branch for predicting an object mask concurrently with the existing branch for bounding box recognition. Mask R-CNN uses the same two-stage technique, with the first stage being identical (which is RPN). Mask R-CNN outputs a binary mask for each RoI in the second stage, in addition to predicting the class and box offset. Despite the significant disparities in segmentation and classification outcomes when compared to land-based image datasets, the results are comparable to state-of-the-art efforts connected with land-based applications. The system provides effective detection and classification of fish using Mask RCNN with 90% classification accuracy. An automated model is developed by making use of train and test images of fishes in order to identify/ classify the fish species to particular class of its species using Mask RCNN algorithm.

Index Terms - Machine Learning, Object Detection, Convolutional Neural Network, Mask RCNN.

I. INTRODUCTION

Real-time fisheries inspection is required to sustain the marine ecology. Uncultured techniques are proving to be damaging to the marine ecosystem in coastal locations, where the majority of fish are harvested. One of the most significant aspects of water vision is fish recognition. Fish recognition with a vessel's camera, on the other hand, is a difficult task. Fish, unlike other popular object recognition problems like face classification, only take up a small portion of an image captured by a boat's camera. It is a difficult task to get an image classification model to pay attention to the accurate position of the images. We hope to give a solution for fisheries monitoring in this work. Unlike the usual fish recognition challenge, images of fish are provided by cameras mounted on small boats. The study of fish species conservation is important for the protection of species of fish. Each species' diversity and size are significant. As a result, a system that analyses images of fish acquired by boat cameras and instantly detects and classifies them into their species category serves as a solution to resolving the problem. The recent technology collects images from a camera mounted on boats in real time. Its job is to identify and classify the fish in these images into one of the six categories: ALB(Albacore Tuna), BET(Bigeye Tuna), DOL(Dolphinfish), LAG(Opah, Moonfish), YFT(Yellowfin Tuna), and Shark(Silky, Shortfin Mako). The images in the collection were taken under various imaging circumstances. These images are blurry, congested, and the fishes are partially obscured in several of them. This complicates the detection, classification and masking process.

II. RELATED WORKS

Mengfan Wang and Lu Wang [1] proposed using YOLOv2 to detect and classify fish. To categorise and detect images of fish, a CNN based on the state-of-the-art detector termed YOLOv2 is utilised in this paper. This automatic classification and detection system has a MAP of 0.912 and a frame rate of 28.3 FPS, meeting the application's real-time requirements.

Roberto Galeazzi and Jens Christian Andersen proposed using CNN to detect and classify fish in challenging conditions [2]. The first steps toward a system that can parameterize fish groups in underwater images are provided in the paper. The Optical Fish Detection Network (OFDNet), a deep CNN, is used for this purpose. This system uses visual data from underwater cameras to perform fish detection, localisation, and species classification. It is based on cutting-edge deep learning object detection frameworks. OFDNet has been proven to properly detect 66.7 percent of the fish contained in a dataset taken at sea, as well as correctly categorise 89.7 percent of them.

Yogesh Girdhar [3] proposed Vision based Detection of fish Using CNN. This research provides CNN-based solutions based on the YOLO algorithm for real-time fish detection utilising underwater vision. Actual fish video images were utilised to test the suggested method's accuracy and dependability. As a consequence, the network achieved a classification accuracy of 93%, a 0.634 IoU between predicted bounding box and ground truth, and a fish detection rate of 16.7 FPS. It also outperforms a fish detector that uses a sliding window technique and a classifier that is developed using a histogram of directed gradient features and SVM.

B. S. Rekha and G. N. Srinivasan proposed using CNN to decect and classify fish in 2020 [4]. The system in this study uses a three-phase process. The augmentation phase is the first. The detecting phase is the next step. The discovered fish is then classified into its species in the third phase. The detection and classification accuracy of the system is 90 percent and 92 percent, respectively.

First, SSD is performed independently on video sequences recorded by a stereo vision system. The stereo information is then used to verify the accuracy of the SVM classification. The suggested technique passes the precision and mAP tests, indicating that it is adequate for real-time fish detection.

Jan D. Almero and Elmer P. Dadios proposed fish detection using classification tree-ANN hybird [6]. Due to the complicated nature of underwater pictures, the persistent tasks, which includes image classification as one of its subtasks, face obstacles. To handle this subtask, a hybrid image classification model based on classification trees and artificial neural networks was developed. The classification tree component extracted a reduced representation of the fundamental dataset, which was produced from a series of captured and processed underwater pictures in a land-based aquaculture setting..

III. SYSTEM ARCHITECTURE

There are four model variants available in the Tensorflow API. I have chosen the Mask RCNN Inception V2 model, which indicates that Inception V2 is utilized to extract the features. Inception architecture refers to a change in feature extraction in a CNN architecture. The distinction is evident in the feature extraction section. Filter concat(Figure 1) and base layer are used in the feature extraction section of Inception v2, as illustrated in Figure 1. The Inception design allows for better utilisation of computer resources throughout the network. When compared to networks that are shorter and narrower, the Inception Architecture offers a significant quality improvement at a minor increase in CPU needs, and it is competitive despite not employing context or bounding box regression. The Inception architecture is based on the use of sophisticated factorization algorithms to alleviate the bottleneck and increase computational complexity efficiency.5x5 pixel convolution layer was factored to 3x3 pixel convolution to improve computational speed in the Inception V2 model, as seen in Figure 3.1[8].

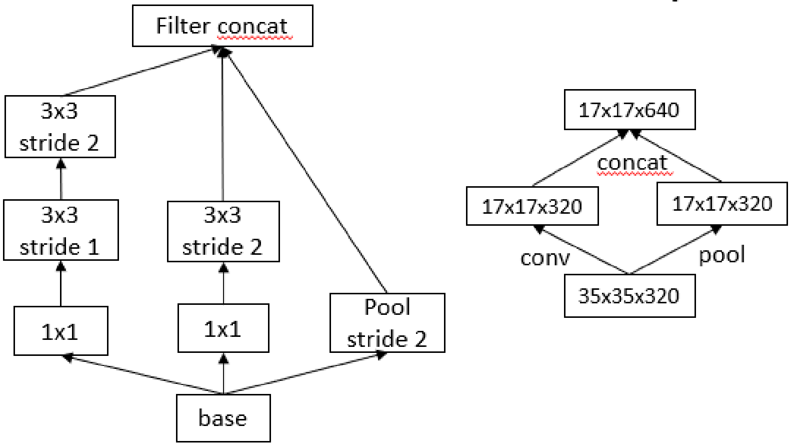

Figure 3.1. Inception v2 Architecture [8]

Mask R-CNN is a model for instance segmentation based on Faster R-CNN. Faster R-CNN is a region-based CNN that gives each object bounding boxes and a confidence score.
Mask R-CNN, like Faster R-CNN, functions in two phases:
Stage 1: There are 2 networks in the first stage.

- Backbone Network, where the Inception v2 architecture is used.
- Region Proposal Network.

To develop a set of area suggestions, these networks are only used once per image. Feature map regions that contain the object are referred to as region proposals.
Stage 2: For each of the proposed regions obtained in stage 1, In the next step, the network predicts bounding boxes and object class. To create predictions, FC layers in networks always require a certain size vector, whereas each proposed region might have any size. To determine the size of these proposed regions, the RoI pool (is quite similar to MaxPool) or RoIAlign techniques are utilised. Mask R-CNN is a variant of Faster R-CNN that includes an extra fork for predicting segmentation masks for each Region of Interest (RoI),as well as object class and bounding boxes.
In the second stage of Mask R-CNN, RoI pool is replaced by RoIAlign, which helps to keep spatial information that is misaligned in the case of RoI pool. RoIAlign uses binary interpolation to produce a fixed-size feature map. The outcome from the RoIAlign layer is later sent to the Mask head, which contains two convolution layers. It produces a mask for each RoI, allowing it to pixel-by-pixel segment an image.

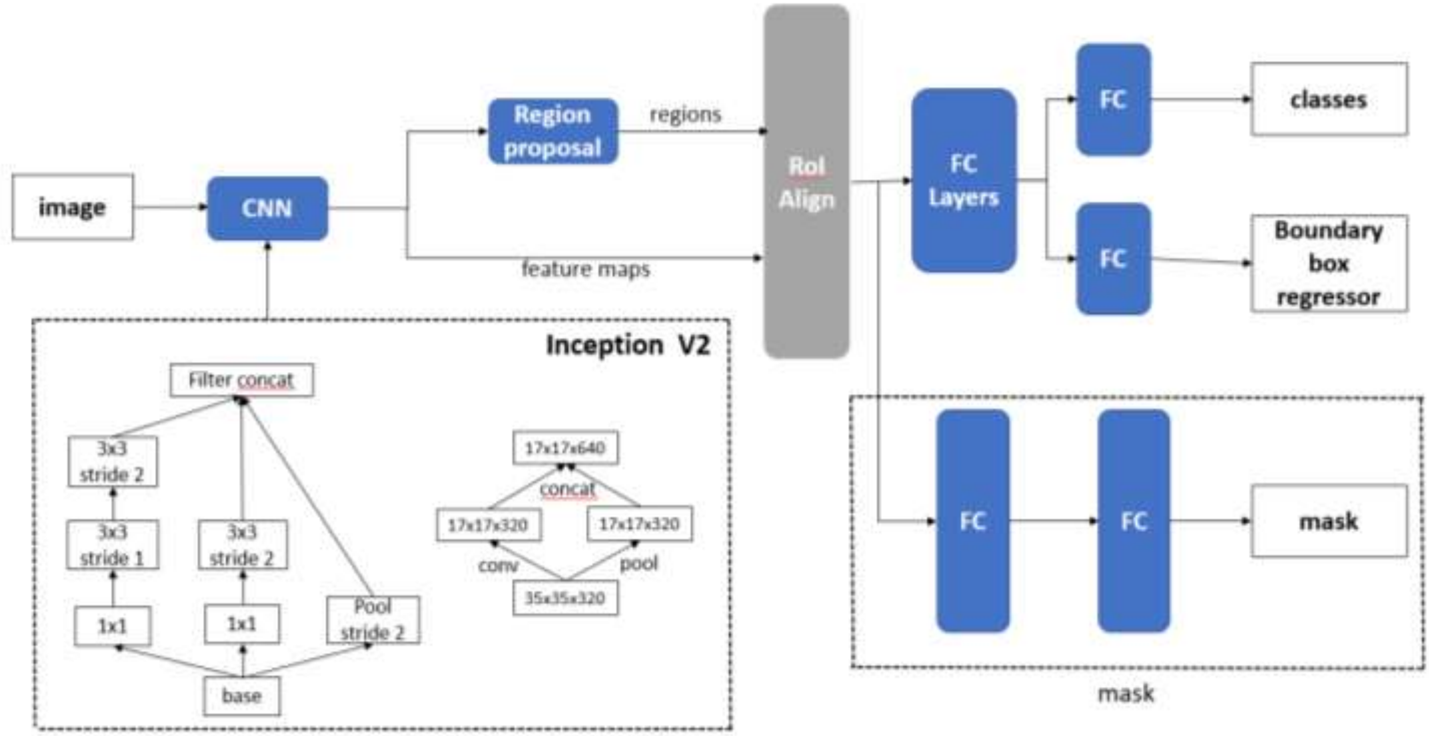

Figure 3.2. mask_rcnn_inception_v2 model

The Mask R-CNN algorithm is developed by adding two key features to the Faster R-CNN architecture:

- •ROI Pooling module is replaced with even more precise ROI Align module
- •A branch is additionally inserted out of the ROI Align module.

The ROI Align outcome is fed into two CONV layers by this additional branch. The mask itself is the output of the CONV layers.

The Mask R-CNN architecture is illustrated in the diagram below:

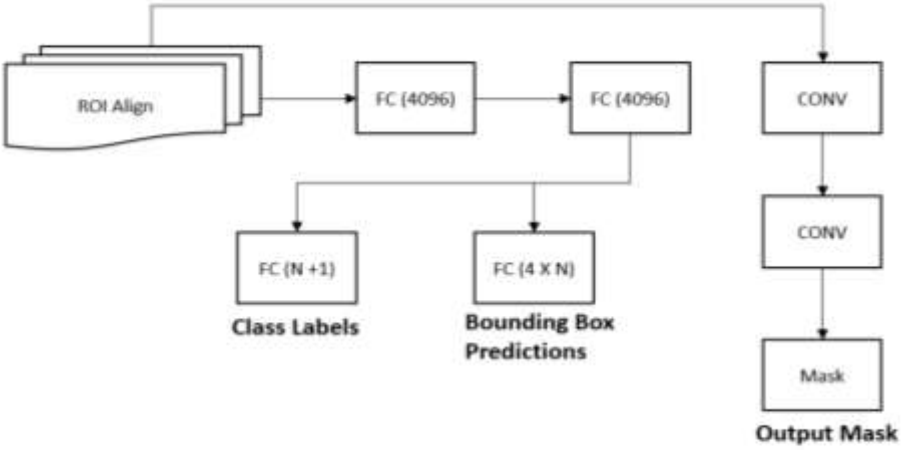

Figure 3.3. The Mask R-CNN model here replaced the ROI Polling module with a more accurate ROI Align module.[11]

The ROI module's output is then sent into two CONV layers. The mask itself is the outcome of the CONV layers. The branch of two CONV layers that emerges from the ROI Align module is where our mask is actually created..

IV. IMPLEMENTATION

Dataset

National Conservancy Organization contributed the original dataset, which was derived from camera feeds received from boat cameras. Different boat environments, different capture time, distortion and occlusion are some of the major challenges faced during the processing of images. The collection contains 3777 images that are classified into six categories as mentioned earlier in the introduction section. The task of categorization is difficult since the dataset is limited and has many variations in imaging parameters. The mentioned dataset is divided into 8:2 ratio for training and testing respectively.

Data Preprocessing: Images are initially resized to 512×512 pixels before training as the model train faster on smaller images as well as the architecture requires that the images are of the same size as the raw collected images may vary in size.

Data Augmentation: It's a technique for creating various training data from existing data artificially. This is performed by transforming examples from the training data into fresh and unique training datasets using domain-specific techniques. Data augmentation employing horizontal flipping of images during training period is used to lessen the need for Mask RCNN for larger training data.

Image Annotation: MIT's Computer Science and Artificial Intelligence Laboratory (CSAIL) [13] developed Labelme, an image labelling tool. It has tools for labelling the edges of objects. Numerous polygons will be generated around the fish when the images are annotated. Figure 4.1 illustrates an example of this utility. For one image, it will save vertex coordinates in a JSON file. Because this is a manual process, there will be some errors when annotating photographs. However, it will have no bearing on the model's overall rating.

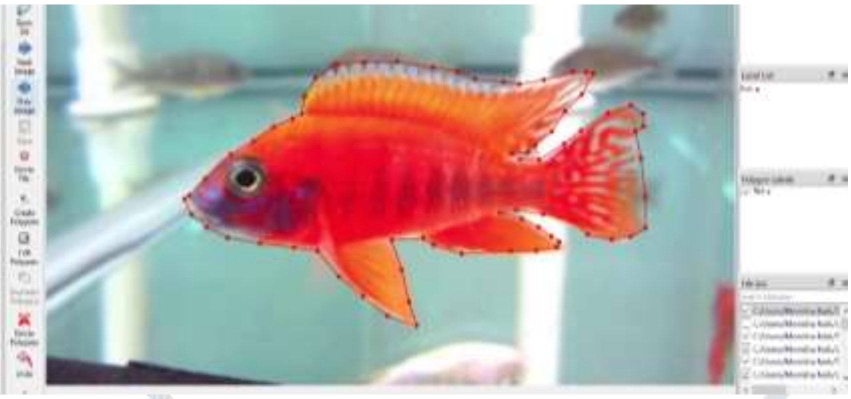

Figure 4.1. User Interface of Labelme tool

Training

To train the model we basically need two things and those are firstly the image and second is the mask(exact pixel wise notation) of the image. To create masks, we have used a tool called Labelme where the masks for each image is drawn manually and this labelled data is converted into json format. This json file has all the specifications of the mask created. As this kind of data cannot be fed directly to the model, the data(class name, mask etc) is converted into machine readable(binary) format in the form of TFRecord file for training and testing data, a compressed representation of the image, the bounding box, the mask, and so on, so that the model has all of the information in one place when it's time to train. This binary data is in then fed to the model for training.

Mask R-CNN extracts features from images using the Inception V2 architecture, similar to how Faster R-CNN extracts feature maps from images using the ConvNet. The initial stage is to extract features from an image using the Inception V2 architecture. The following layer, RPN, uses these features as an input. A region proposal network is now applied to the feature maps acquired in the previous phase (RPN). This essentially predicts whether or not an object is present in that region (or not). Those regions or feature maps are then acquired, that the model predicts will include some object in this step.

The RPN-derived regions could have a variety of shapes. As a result, pooling layer is used to transform all of the regions into the same shape. The class label and bounding boxes are then predicted by passing these regions through a fully connected network. A mask branch could be added to the existing architecture once we get the RoIs. This function returns the segmentation mask for each object-containing region.

The following are some of the training parameters:

- Optimizer: Adams
- Classifier: Softmax
- Annotation tool used: Labelme
- Time taken to train: 12-14 hrs
- No. of training steps: 80,000+ steps.

V. EXPERIMENTAL RESULTS AND ANALYSIS

The proposed model is successfully able to detect classify as well as mask the fish present in the image. It is able to classify the fishes into all the six categories it was trained for as mentioned earlier. The model is also detecting, classifying and masking images with multiple fishes.

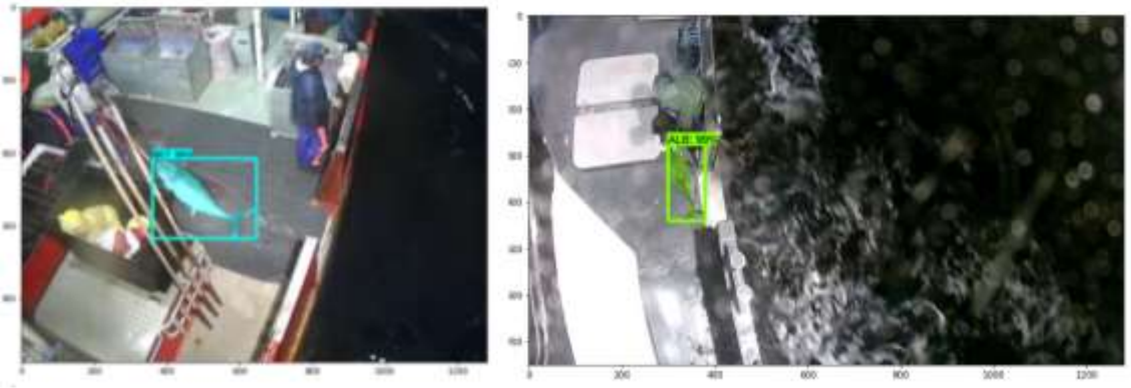

Figure 4.2. Snapshots of detection and classification of fish

Mask R-CNN's multi-task loss function integrates classification, localization, and segmentation mask losses: L=Lcls+Lbox+Lmask, where Lcls and Lbox are class loss and bounding box loss.

A loss is a number that represents how inaccurate the model's estimate was in a specific instance. The loss is 0 if the model's estimate is perfect; otherwise, the loss is bigger. The purpose of training a model is to discover a set of weights and biases that have a low loss across all cases on average.

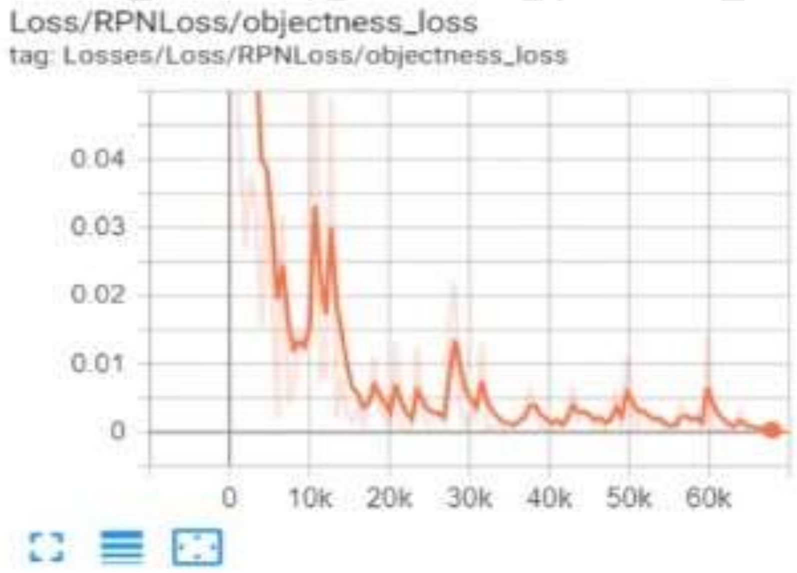

Figure 4.3. Graph representing objectness loss encountered while training

Objectness Loss: A prediction called ‘objectness' is connected with each box prediction. Because it is multiplied by the lass score to give absolute class confidence, it takes the place where prior detectors like RCNN took the confidence that a region proposal contains an item. However, contrary to popular belief, that forecast is an IoU prediction, which refers to how effectively the network believes the box covers the item. The coordinate loss term trains the network to forecast a better box, whereas the objectness loss term teaches it to anticipate a correct IoU. (which eventually pushes the IoU toward 1.0). Figure 4.3 illustrates that the objectness loss is almost 0 after vigorously training the model at around 80k steps.

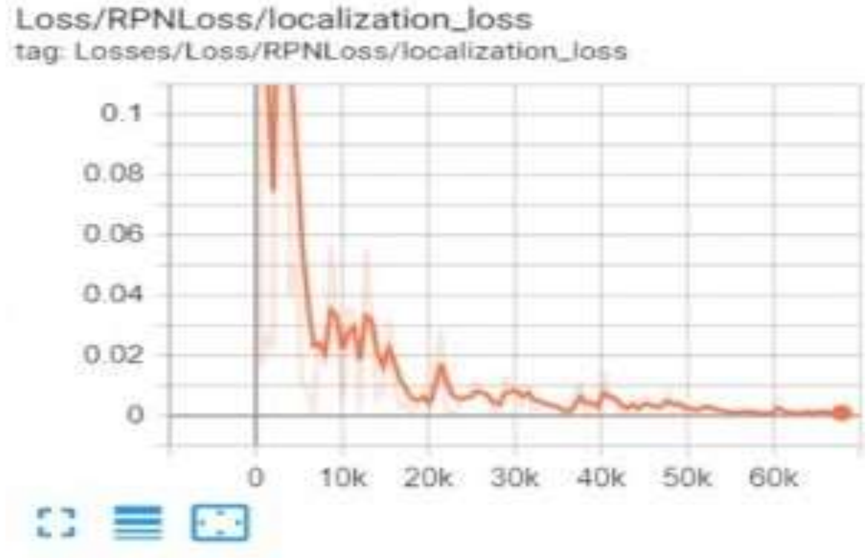

Figure 4,4. Graph representing localization loss encountered while training

RPN Loss/Localization Loss: The architecture of mask rcnn will be having the cnn for getting the region proposals. For getting the region proposals from the feature map there is loss functions. This is the localization loss for bounding boxes for the anchors generated. The sum of classification and bounding box regression losses is the RPN loss. Figure 4.4 illustrates that the localization loss is almost zero at around 80k steps and the localization of the fish is almost perfect, hence the training was halted.

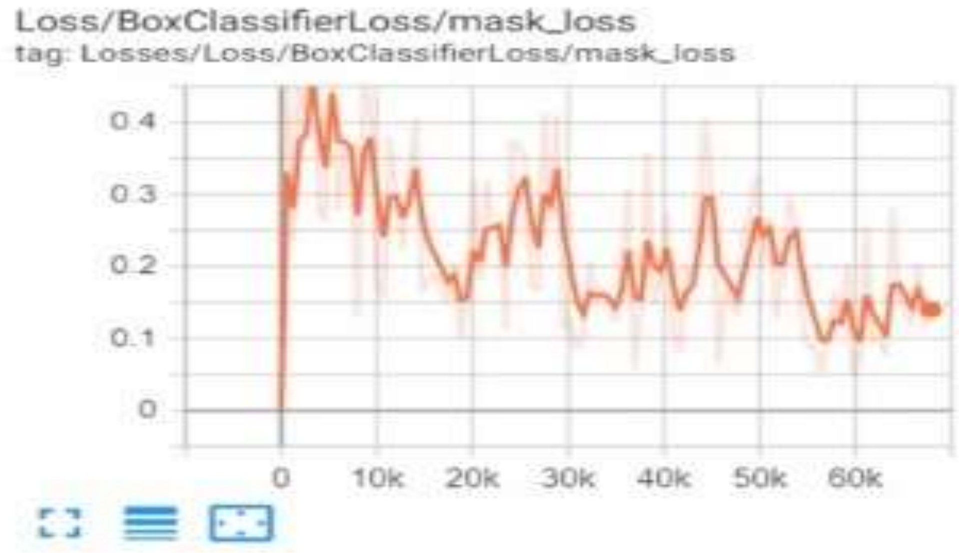

Figure 4.5. Graph representing mask loss encountered while training

BoxClassifier Loss/Mask Loss: This is the loss encountered while the model applies masks to the input image. The loss is seen when the model doesn't correctly mask the fish in its actual shape. The mask loss recorded here is around 0.15 at around 80k steps as shown in Figure 4.5. The training was halted at this point as the object(fish) was getting masked accurately.

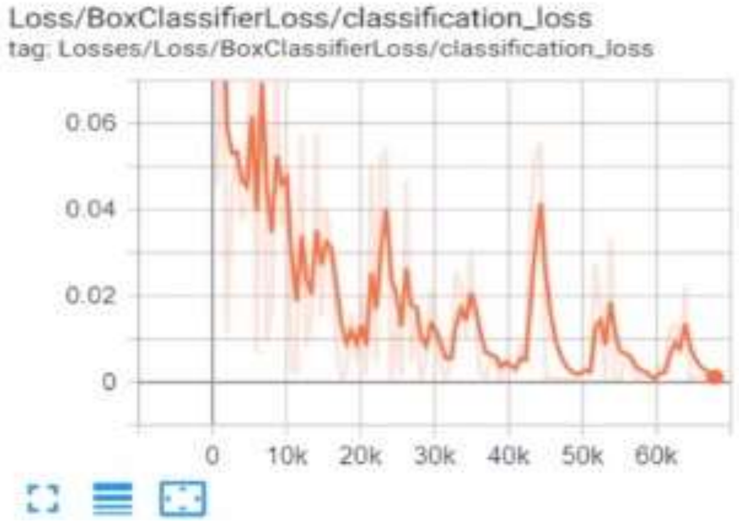

Figure 4.6. Graph representing classification loss encountered while training

Classification Loss: Due to deviations from predicting ‘1' for the correct classes and ‘0' for all other classes for the object in that box, classification loss occurs. As seen in Figure 4.6 that the classification loss is almost 0 after vigorously training the model at around 80k steps.

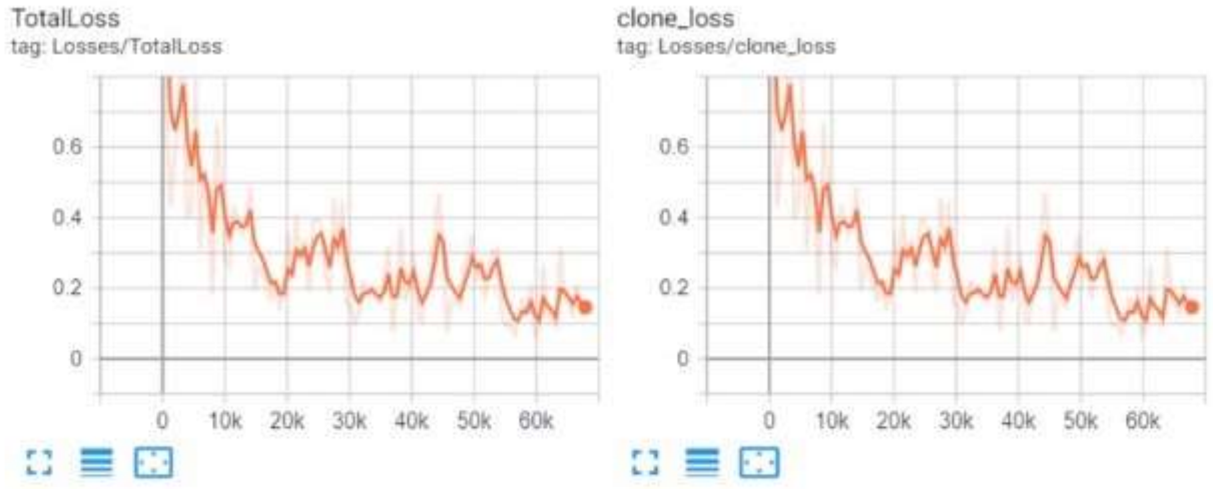

Figure 4.7. Graph representing total loss and clone loss encountered while training

Total loss/Clone loss: Total loss of the model is the combined loss of all the losses encountered while training the model where as clone loss is when TensorFlow will create clones of the model to train on each GPU and report the loss of each clone(matters only when multiple GPU/CPUs are used). As the model is trained on a single GPU/CPU the total loss and clone loss are one and the same and that is at around 0.13 as seen in Figure 4.7. As the total loss is very close to 0 the model’s prediction is considered almost perfect and the training is halted.

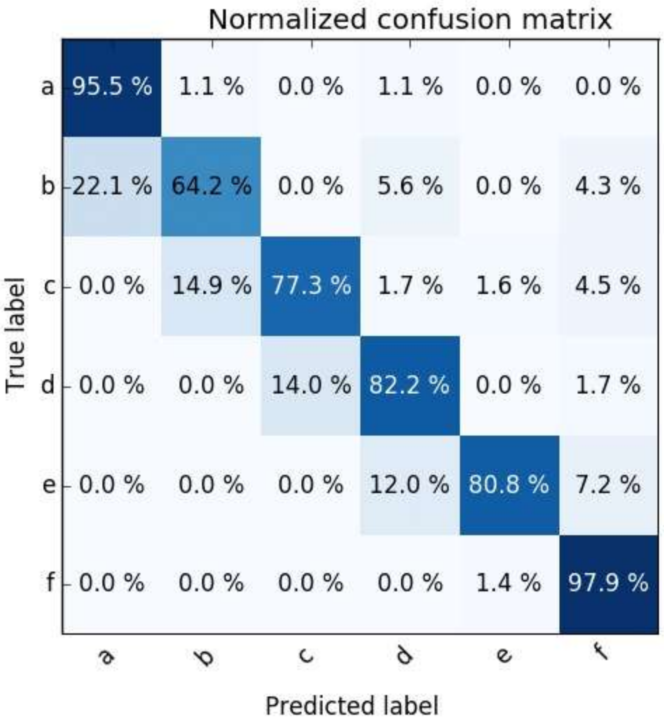

Figure 4.8. Normalized Confusion Matrix

A confusion matrix is a table that summarises the results of classification problem prediction. Count values are used to sum and break down the number of correct and incorrect predictions by class. This is the key to the confusion matrix. When producing predictions, the classification model becomes bewildered, as seen by the confusion matrix. It exposes not only the types of errors that the classifier makes, but also the amount of errors that the classifier makes. In this model a, b, c, d, e and f represents the six classes of fishes considered in this work.

- Each row of the matrix corresponds to a predicted class.
- The matrix's rows correspond to different classes.
- Total numbers of correct and wrong classifications are entered in the table.
- The aggregate of correct predictions is entered into the predicted column and expected row for each class.
- For each class, the total number of incorrect predictions is recorded in the expected row and predicted column for that class's value.

Accuracy of the classifier can be calculated using the following equation-

Accuracy = (TP+TN)/total, where TP is true positive and TN is true negative.

Using this equation, the classification accuracy of the model has come upto 90%.

VI. CONCLUSION AND FUTURE WORK

An automated model is developed by making use of train and test images of fishes in order to identify/ classify the fish species to particular class of its species using MRCNN algorithm. While the model used in this project is the fastest at inference time though it may not have the highest accuracy. As the most light weight model is used for this project. It could be implemented using other models in the suite that are slower, perform in terms of accuracy of detection. Other machine learning approaches may be included in future work, and a thorough comparison of them will be provided. Furthermore, increasing the accuracy of the prediction model by include more software metrics in the learning process is a viable option. Further enhancement can be done with respect to detection and classification in real-time. It could also be implemented on different species of fishes.

.

VII. Acknowledgment

I take this opportunity to thank my guide, Prof. Rekha B S, Assistant Professor, Department of Information Science & Engineering, RV COLLEGE OF ENGINEERING, Bengaluru for her valuable suggestions, support, regular source of encouragement and assistance throughout this project. I also thank senior authorities whose constant encouragement made this possible.

References

[1] M. Wang, M. Liu, F. Zhang, G. Lei, J. Guo and L. Wang, "Fast Classification and Detection of Fish Images with YOLOv2," 2018 OCEANS - MTS/IEEE Kobe Techno-Oceans (OTO), Kobe, 2018, pp. 1-4, doi: 10.1109/OCEANSKOBE.2018.8559141.

[2] J. H. Christensen, L. V. Mogensen, R. Galeazzi and J. C. Andersen, "Detection, Localization and Classification of Fish and Fish Species in Poor Conditions using Convolutional Neural Networks," 2018 IEEE/OES Autonomous Underwater Vehicle Workshop (AUV), Porto, Portugal, 2018, pp. 1-6, doi: 10.1109/AUV.2018.8729798.

[3] M. Sung, S. Yu and Y. Girdhar, "Vision based real-time fish detection using convolutional neural network," OCEANS 2017 - Aberdeen, Aberdeen, 2017, pp. 1-6, doi: 10.1109/OCEANSE.2017.8084889.

[4] B S, Rekha & G N, Dr. Srinivasan & Reddy, Sravan & Kakwani, Divyanshu & Bhattad, Niraj. (2020). Fish Detection and Classification Using Convolutional Neural Networks. 10.1007/978-3-030-37218-7_128.

[5] G. Chen, P. Sun and Y. Shang, "Automatic Fish Classification System Using Deep Learning," 2017 IEEE 29th International Conference on Tools with Artificial Intelligence (ICTAI), Boston, MA, 2017, pp. 24-29, doi: 10.1109/ICTAI.2017.00016.

[6] V. J. D. Almero, R. S. Concepcion, E. Sybingco and E. P. Dadios, "An Image Classifier for Underwater Fish Detection using Classification Tree-Artificial Neural Network Hybrid," 2020 RIVF International Conference on Computing and Communication Technologies (RIVF), Ho Chi Minh, Vietnam, 2020, pp. 1-6, doi: 10.1109/RIVF48685.2020.9140795.

[7] P. Kaveti and H. Singh, "Towards Automated Fish Detection Using Convolutional Neural Networks," 2018 OCEANS - MTS/IEEE Kobe Techno-Oceans (OTO), Kobe, 2018, pp. 1-6, doi: 10.1109/OCEANSKOBE.2018.8559068.

[8] Alamsyah, Andry & Apriandito, Muhammad & Masrury, Riefvan. (2019). Object Detection Using Convolutional Neural Network To Identify Popular Fashion Product. Journal of Physics: Conference Series. 1192. 012040. 10.1088/1742-6596/1192/1/012040.

[9] Shaoqing Ren, Kaiming He, Ross Girshick, Jian Sun: “Faster R-CNN: Towards Real-Time Object Detection with Region Proposal Networks”, 2015; [http://arxiv.org/abs/1506.01497 arXiv:1506.01497].

[10] Mask-RCNN. https://github.com/matterport/Mask_RCNN. Accessed: 2018-04-27.

[11] Girshick, R., Donahue, J., Darrell, T., and Malik, J. Rich feature hierarchies for accurate object detection and semantic segmentation. In Computer Vision and Pattern Recognition (2014).

[12] Girshick, R. Fast R-CNN. arXiv preprint arXiv:1504.08083 (2015).

[13] Ren, S., He, K., Girshick, R. B., and Sun, J. Faster R-CNN: towards real-time object detection with region proposal networks. CoRR abs/1506.01497 (2015).

[14] Shelhamer, E., Long, J., and Darrell, T. Fully convolutional networks for semantic segmentation. IEEE Trans. Pattern Anal. Mach. Intell. 39, 4 (Apr. 2017), 640–651.

[15] Redmon, J., and Farhadi, A. Yolov3: An incremental improvement. arXiv (2018).

[16] Liu, W., Anguelov, D., Erhan, D., Szegedy, C., Reed, S., Fu, C.-Y., and Berg, A. C. Ssd: Single shot multibox detector.

[17] A. Rana, G. Yauney, L. C. Wong, O. Gupta, A. Muftu, and P. Shah, “Automated segmentation of gingival diseases from oral images,” in Healthcare Innovations and Point of Care Technologies (HI-POCT), 2017 IEEE. IEEE, 2017.

[18] T.-T. Do, T. Hoang, V. Pomponiu, Y. Zhou, C. Zhao, N.-M. Cheung, D. Koh, A. Tan, and T. Hoon, “Accessible melanoma detection using smartphones and mobile image analysis,” IEEE Transactions on Multimedia, 2018

[19] J. W. Johnson, “Adapting mask-rcnn for automatic nucleus segmentation,” arXiv preprint arXiv:1805.00500, 2018.

[20] R. Anantharaman, M. Velazquez and Y. Lee “Utilizing Mask R-CNN for Detection and Segmentation of Oral Diseases,” In BIBM, 2018

[21] B. Russell, A. Torralba, and W. T. Freeman, Labelme, The Open Annotation Tool MIT, Computer Science and Artificial Intelligence Laboratory [Online]. Available: http://labelme.csail.mit.ed

[22] Y Li, Y Yu, Z Li, Y Lin, M Xu, J Li, and X Zhou. Pixel-anchor: A fast oriented scene text detector with combinednetworks[C]. arXiv preprint arXiv:1811.07432, 2018.

[23] X Liu, D Liang, S Yan, D Chen, Y Qiao, and J Yan. Fots: Fast oriented text spotting with a unified network[C]. In Proceedings of the IEEE conference on computer vision and pattern recognition, pp. 5676–5685, 2018.

[24] E Xie, Y Zang, S Shao, G Yu, C Yao, and G Li. Scene textdetection with supervised pyramid context network[C]. arXivpreprint arXiv:1811.08605, 2018.

[25] Kaiming He, Georgia Gkioxari, Piotr Dollár, Ross Girshick: “Mask R-CNN”, 2017; [http://arxiv.org/abs/1703.06870 arXiv:1703.06870].