FishSense ML Team Final Report

AARUSHI SEHGAL, University of California, San Diego, Department of Computer Science and Engineering
CAMERON NAZEMI, University of California, San Diego, Department of Computer Science and Engineering

Fish are very important to the underwater ecosystem and are an essential food resource, but measuring their activity and biomass in order to protect their population is currently a labour intensive and manual process. Therefore, we implemented a solution using the Intel RealSense camera to detect and calculate the length of fish. This data will be provided to conservation organizations for their research into fish health and biomass. This solution involved algorithms and machine learning implementations for more accurate identification of fish to better calculate their length. The first uses median segmentation and the second method is custom YOLOv4 machine learning model. As a result of the improved ML models, we were able to achieve a 80% accuracy for fish body, head and tail detection. Our contributions help in building a better FishSense module which has long term implications in maintaining the ecosystem for fish and their populations.

ACM Reference Format:
Aarushi Sehgal and Cameron Nazemi. 2022. FishSense ML Team Final Report. 1, 1, Article 1 (June 2022), 9 pages.

1 INTRODUCTION

1.1 Motivation

Our solution consists of two parts, one involving a machine learning implementation for more accurate identification of fish for better biomass calculation and activity tracking. The other part concerns building an efficient low power hardware model to power the system for around 2 weeks, the timespan necessary to collect meaningful data for conservation organization.

1.2 Project Aim

FishSense is a underwater fish detection system that utilizes depth cameras. The FishSense system uses the Intel RealSense depth camera to record fish with high quality video/image resolution and high depth accuracy. The goal is the get the FishSense system in the hands of fisheries and conservation scientists who would take the system underwater to collect and analyze it's captured data. This would provide insight on size, structure, and biomass of the fish. We have a initial working prototype of FishSense and the job of the ML team is build off of the current algorithms towards more accurate fish identification and biomass detection. The ML team implemented a series of computer vision techniques and select the most effective to achieve this goal. The overall end-goal of the machine learning team is to be able to identify fish more accurately using the same hardware currently implemented.

To achieve a successfully enhanced fish detection both outside and inside the water we implemented two main methods. The first method utilizes a Median Segmentation based algorithm. With Median Segmentation we calculated a median/center for the bounding box and then applied BFS/DFS to determine for all the neighbouring

Authors' addresses: Aarushi Sehgal, aasehgal@ucsd.edu, University of California, San Diego, Department of Computer Science and Engineering; Cameron Nazemi, University of California, San Diego, Department of Computer Science and Engineering, .

XXXX-XXXX/2022/6-ART1 $15.00
https://doi.org/

pixels whether they belong to the fish or not and marked them as visited/ not visited. The second method utilized YOLOv4 CNN model to detect fish head and tail. We labeled the fish body, head, and tail of a custom data set of 900+ fish images and trained it on a YOLOv4 machine learning model. The next steps would be to detect the fish head and tail median pixels points for each fish use that pixel information to calculate fish length.

2 RELATED WORKS

Since our project's aim is to enhance the accuracy of fish detection and length estimation, our research consisted of finding scholarly research articles in the field of machine learning some with a focus on fish detection. We have included some notable related works that inspired our work on fish detection and length estimation.

2.1 An effective and robust method for tracking multiple fish in video image based on fish head detection. [1]

This article has the goal of detecting and tracking fish by use of the fish head position. This article gave inspiration for us to utilize head and tail of the fish for measurements.

2.2 Automated Planar Tracking the Waving Bodies of Multiple Zebrafish Swimming in Shallow Water. [2]

This article discusses the difficulties of detecting a Zebrafish and the novel fish body model that is created to overcome this issue. After they use the fish head to continue tracking. This article gave inspiration for us to utilize head and tail of the fish for measurements.

2.3 Yolov4: Optimal Speed and accuracy of object detection. [3]

This article discusses the YOLOv4 model and its many improvements over other models. YOLOv4 has the benefit of a real-time CNN operation making it a fast object detector. This article was about our preferred machine learning model method for detecting fish head and tail.

2.4 Automatic fish detection in underwater videos by a deep neural network-based hybrid motion learning system. [4]

In this article, the researchers discuss the detection of fish underwater using a region-based CNN model. They used optical flow and background subtraction to train this neural network.

2.5 Image-based, unsupervised estimation of fish size from commercial landings using deep learning. [5]

These researchers use an Mask R-CNN model for unsupervised length estimation of fish length.

Overall the FishSense ML team's goal was to try to incorporate ideas to enhance fish detection and length estimation.

3 TECHNICAL MATERIALS

3.1 Terminology

(1) **Biomass** refers to the total mass of organisms in a given area or volume.
(2) **YOLO** or You Only Look Once is a state of the art, real time object detection Convolutional Neural Network model.
(3) **COCO** format is a specific JSON structure dictating how labels and metadata are saved for an image dataset.

(4) **Roboflow** is a software that helps you train any computer vision model while utilizing small dataset of images within a small period of time.
(5) **CVAT** or Computer Vision Annotation Tool is a free, open source, web-based image and video annotation tool which is used for labeling data for computer vision.

3.2 Median Segmentation

Median segmentation is a self devised algorithm where we calculate a median/center for the bounding box and then applying Breadth First Search (BFS), starting from the median pixel, we determine for all the neighbouring pixels whether they belong to the fish or not and mark them as visited/ not visited. At the end we are able to get the fish withing the bounding box and can obtain the fish boundary pixels. Building an $\mathbf{O}(n^2)$ algorithm on the top of that, we obtain two pixels farthest apart from each other and use them to calculate fish length.

3.3 Custom YOLOv4 Model

YOLO or You Only Look Once is a state of the art, real time object detection Convolutional Neural Network model. Object detection in YOLO is carried out as a regression problem and it only requires a single forward propagation through the neural network layers to detect objects. Since YOLO is the state of the art at detecting objects, we decided to utilize this model for detecting fish body, head and tail. For achieving this we first searched the internet for a data set that comprises of fish images at various different lighting conditions, scales, orientations, and visibility. We used Open Images Dataset V6's **[6]** that contained thousands of fish images that were perfect for our project. We then filtered the images and annotated the 1000+ fish images using the CVAT online software. We annotated the fish head, tail, and body. After we completed labeling the images, we initially decided to run our YOLO model on an online software called Roboflow that could help us upload our custom annotations in COCO format, choose the type of model we want to run and then do the training for us, providing us with the final set of weights, mAP and other precision statistics. But, unfortunately we were not able to obtain a good accuracy from this and hence decided to proceed with building our own custom YOLO model.

After building our custom YOLOv4 model to detect fish body, head and tail, we trained the model for 7+ hours we were able to achieve a 80% plus accuracy for the detection.

We aim to further utilize IoU (Intersection Over Union) to find the optimal region from all bounding boxes and then use that to calculate the median points for fish head and tail. After obtaining these pixels we can calculate the fish length.

3.4 Data Collection

We worked on collecting 2000+ images for fish from Open Images Dataset V6 for building our own custom annotated dataset. We then filtered and annotated images using CVAT for fish body, head and tail. These annotations and images were imported in a YOLO compatible format and then later utilized for training our model.

4 MILESTONES

4.1 Milestone 1: Detect Fish Inside Water

4.1.1 Description. Our first milestone was to improve the accuracy of fish identification and length calculation inside water by implementing median segmentation. We calculate a median/center for the bounding box and then applying BFS/DFS we determine for all the neighbouring pixels whether they belong to the fish or not and mark them as visited/ not visited.

4.1.2 Achievements. This was implemented and tested on a simulated fish inside water.Median Segmentation is an effective method to determine fish length for fishes inside water because we are able to successfully utilize depth information and find the edges of the fish. With this, we are able to find the distance between the head and tail to calculate the fish length.

4.1.3 Deliverables.

(1) Deliverable 1: Create and implement algorithm for Median Segmentation.***(Completed)***. We demonstrate the implementation through Figure 1.

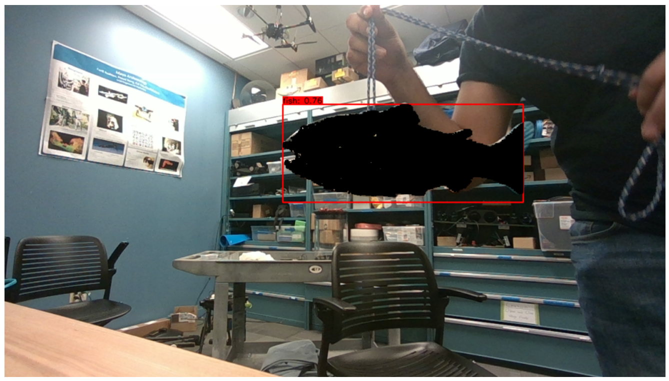

Fig. 1. Implementation of Median Segmentation

(2) Deliverable 2: Adding algorithm to obtain boundary pixels for the fish and marking them in our detection. The implementation is demonstrated through Figure 2. ***Completed***

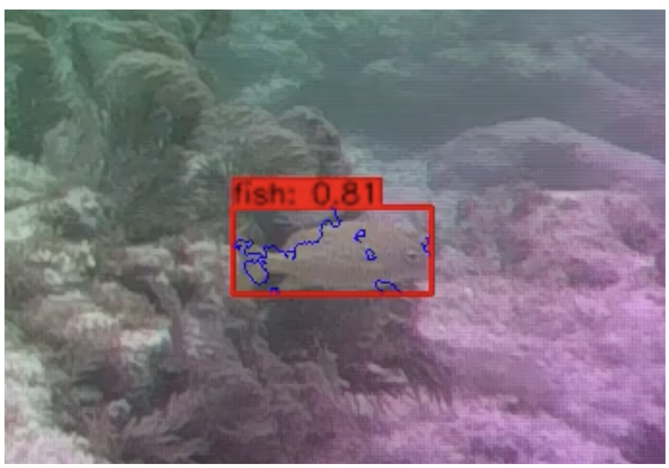

Fig. 2. Boundary pixels detection using Median Segmentation

(3) Deliverable 3: Building algorithm for finding fish length. We utilize an $\mathbf{O}(n^2)$ algorithm which finds the two pixels which are farthest away in the set of boundary pixels obtained from median segmentation and use them to calculate fish length. ***(Completed)***.

(4) Deliverable 4: Test algorithm on image of fish to find fish length. ***(Incomplete)***.
(5) Deliverable 5: Analyze results and determine if Median Segmentation satisfies set benchmarks. Demonstrated below using Figure 3. ***(Completed)***.

Fig. 3. Testing Median Segmentation in different use cases

4.1.4 ***Status***. We achieved our first objective, i.e. detecting fish inside water, during week 4 and 5 as specified in our schedule and have been able to obtain satisfactory results for the same.

4.2 Milestone 2: Detecting Fish Outside Water

4.2.1 Description. Our second milestone was to improve the accuracy of fish identification and length calculation outside water by using a deep learning model to get the fish head and tail pixel points and its distance from the camera to determine the fish's length.

4.2.2 Updates. This milestone has changed from the previous milestone of simply using depth data from the Intel RealSense camera to detect fish length because the Median segmentation approach failed to accurately detect the edges of the fish while it was lying on a flat surface. This is because the depth readings of the fish were too similar to that of the surface behind it, as seen in **Figure 3**.

As you can see when the fish is placed on a flat surface the depth sensor cannot distinguish the floor from the fish. In real life conditions, fish may be measured outside the water on a flat surface and therefore median segmentation would fail to provide accurate fish length.

Therefore we have shifted our Milestone 2 to account for fish length calculation outside water. To do this we will be using a YOLO v4 machine learning model to find the fish head and tail pixel points and find the distance between the two in order to determine the length.

4.2.3 Achievements.

(1) Deliverable 1: Find new way to determine fish length using machine learning. ***(Completed)***.
(2) Deliverable 2: Collect a data set of fish images and store them in cloud. ***(Completed)***. We demonstrate the implementation of the same through Figure 4.

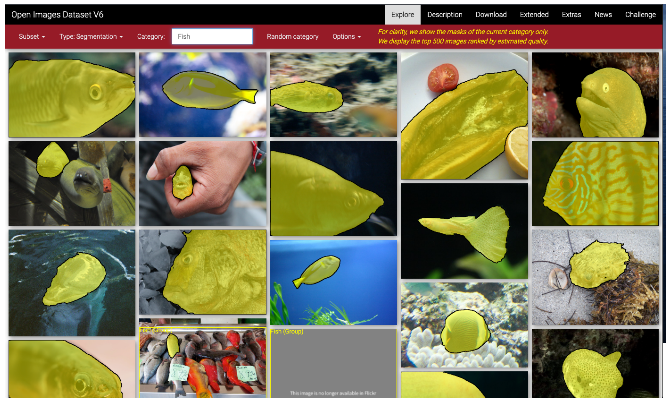

Fig. 4. Open Images Dataset

(3) Deliverable 3: Determine efficient image annotation software for labelling.***(Completed)*** (https://cvat.org/).
(4) Deliverable 4: Annotate/ label head, tail, and body of fish images on annotation software. ***(Completed)***. We demonstrate the implementation of the same through Figure 5.

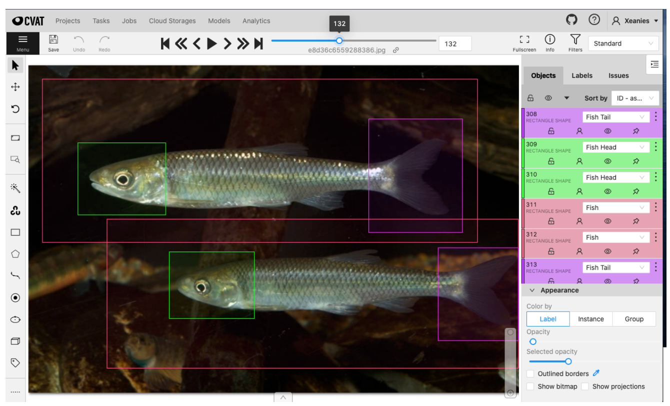

Fig. 5. CVAT annotations

(5) Deliverable 5: Determine model for training fish images and annotation. ***(Completed)*** (YOLO v5 using RoboFlow)

(6) Deliverable 6: Run and test the selected model ***(Completed)***. We demonstrate the implementation of the same through Figure 6.

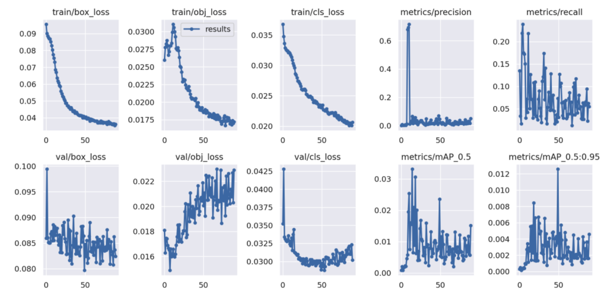

Fig. 6. Roboflow YOLOv5 Model Results

(7) Deliverable 7: Building a new custom YOLOv4 model since RoboFlow couldn't give us the required level of accuracy and precision. Demonstrated through Figure 7. ***Completed***

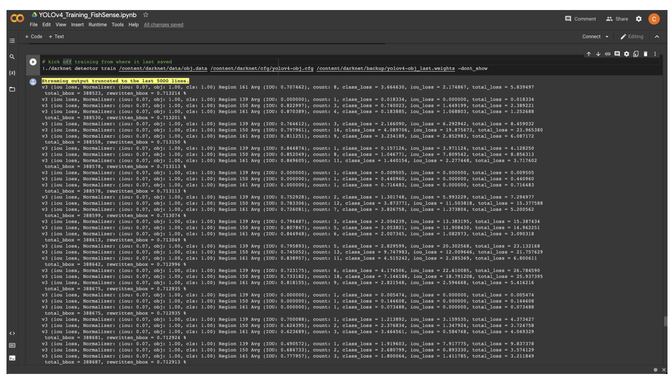

Fig. 7. YOLOv4 code

(8) Deliverable 8: Training the model and getting model performance statistics. Demonstrated using Figure 8. ***Completed***

(9) Deliverable 9: Ensure model operates on new fish images. Demonstrated using Figure 9. ***(Completed)***

(10) Deliverable 10: Calculate fish length using head and tail pixel points along with depth information. ***(In progress)***

4.2.4 ***Status***. We worked on collecting, annotating, filtering and modifying data that needed to be utilized in training our YOLO model for fish detection during week 6 and week 7. We further worked on implementing our YOLOv5 CNN model through RoboFlow during week 8. Since our RoboFlow model didn't perform as expected we worked on building and training our own custom YOLOv4 model during week 9. In week 10 we worked on compiling all our results obtained thus far and analyzing them. We aim to work on improving our YOLO model in the future and finding center pixel points for bounding boxes belonging to fish head and tail, which we will further utilize to calculate fish length.

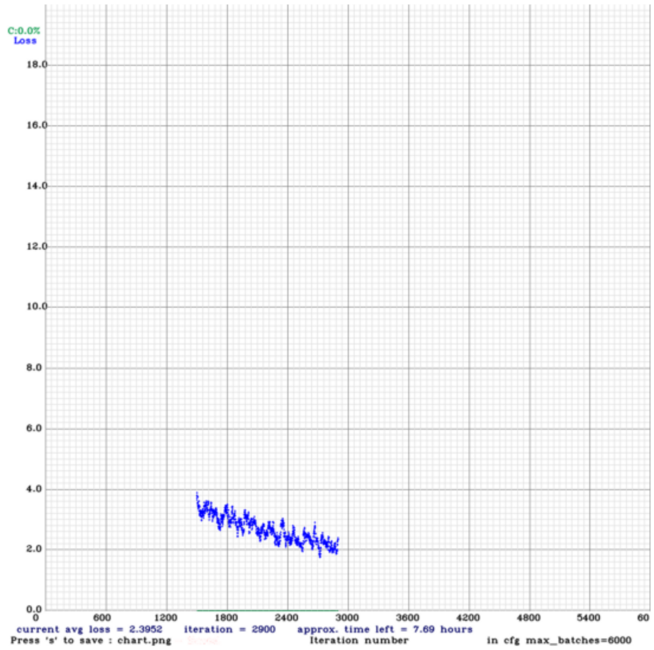

Fig. 8. YOLOv4 mAP Graph

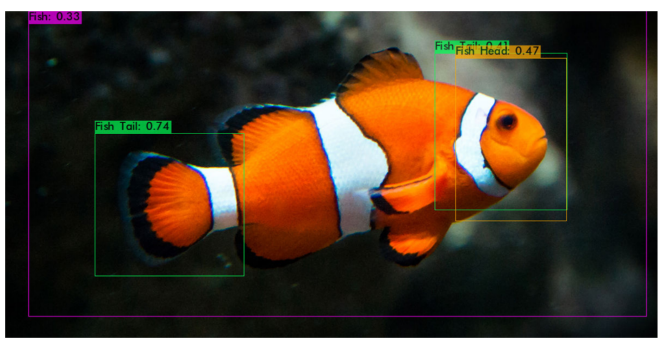

Fig. 9. YOLOv4 detecting Nemo

5 CONCLUSION

In this project, we investigated how various methods for fish length calculation and biomass detection. We started with implementing Median Segmentation for detecting fish and it's boundary pixels and went on to implement YOLOv5 through RoboFlow and building our own custom YOLOv4 model. We explored various techniques and algorithms for achieving our goal and analyzed what works best for which situation. All these methods were inspired by required improved on existing models and seemed promising.

Through implementation we realized that Median Segmentation works well in detecting fish inside water but fails to do outside water on a flat surface due to the same depth for both fish and it's surroundings. Median Segmentation also seemed to lack in accuracy and performance when it came to detecting and marking fish boundary pixels due to the lack of good quality depth data. RoboFlow didn't provide us the desired accuracy for fish body, head and tail detection even after multiple runs which is why we decided to switch to buliding our own custom YOLO model. Our custom made YOLO model provided a 80% accuracy after 7 hours of training with approx. 1000 images.

In conclusion Median Segmentation is a reliable algorithm which can be improved on by gathering more accurate depth data. YOLOv4 also shows a lot of promise since it already achieved a good accuracy in first few hours of training. We aim to further work on utilizing IoU and edge detection to get head and tail pixels for fish in order to calculate the final fish length.

6 REFERENCES

[1] Qian, ZM., Wang, S.H., Cheng, X.E. et al. An effective and robust method for tracking multiple fish in video image based on fish head detection. BMC Bioinformatics 17, 251 (2016). https://doi.org/10.1186/s12859-016-1138-y
[2] Wang, Shuo Hong Cheng, Xi Qian, Zhi-Ming Liu, Ye Chen, Yan. (2016). Automated Planar Tracking the Waving Bodies of Multiple Zebrafish Swimming in Shallow Water. PloS one. 11. e0154714. 10.1371/journal.pone.0154714.
[3] Alexey Bochkovskiy, Chien-Yao Wang, and Hong-Yuan Mark Liao. 2020. Yolov4: Optimal Speed and accuracy of object detection. (April 2020). Retrieved June 8, 2022 from https://arxiv.org/abs/2004.10934
[4] Salman, Ahmad Siddiqui, Shoaib Shafait, Faisal Mian, Ajmal Shortis, Mark Khurshid, Khawar Ulges, Adrian Schwanecke, Ulrich. (2019). Automatic fish detection in underwater videos by a deep neural network-based hybrid motion learning system. ICES Journal of Marine Science. 77. 10.1093/icesjms/fsz025.
[5] Amaya Álvarez-Ellacuría, Miquel Palmer, Ignacio A Catalán, Jose-Luis Lisani, Image-based, unsupervised estimation of fish size from commercial landings using deep learning, ICES Journal of Marine Science, Volume 77, Issue 4, July-August 2020, Pages 1330–1339, https://doi.org/10.1093/icesjms/fsz216.
[6] Open images dataset V6 + extensions. Retrieved June 8, 2022 from https://storage.googleapis.com/openimages/web/index.html