CS 4701 Final Presentation: Chromis Counter

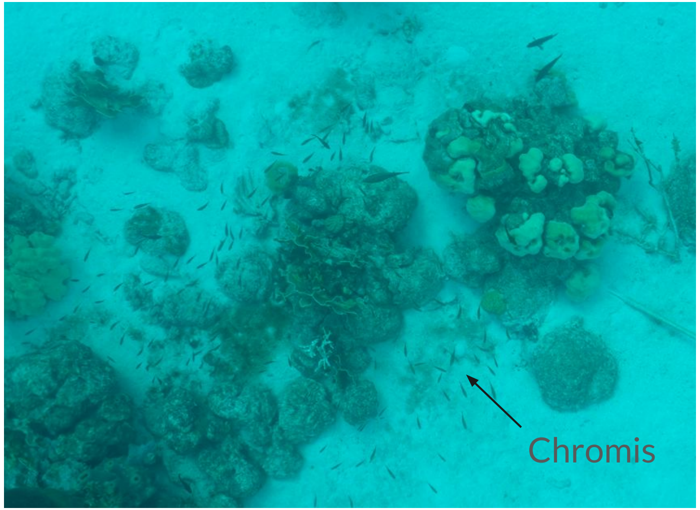
Jenny.M, Jenny.J, Steven.C

Background Information

- The project supports the research efforts of the Hein Lab, where team member Jenny Mao participates, by contributing to a deeper understanding of chromis fish dynamics and their social structures.
- Processes raw underwater transect videos to accurately count the number of fish within each colony, enhancing the efficiency and accuracy of data collection
- By improving our ability to identify and track individual fish, the program provides critical data necessary for comprehensive analysis of chromis fish collective behavior.

Data Description

- Underwater transect videos capturing Chromis fish colonies
- 4K video quality
- Taken with a handheld camera by a diver swimming in the sea
- Durations ranging from 30 seconds to 4 minutes
- Variable number and distribution of fish across different videos



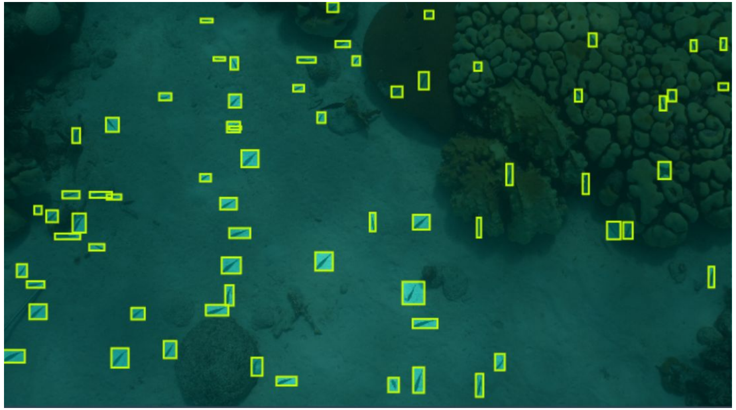
Screenshot of a frame in a video

General Pipeline of the Project

- Goal is to answer the question: How many chromis in each colony in the video?
- Roboflow Annotations
- Model Training
- Model Deployment (Annotating Videos)- Optimized Tracking
- Unsupervised Clustering by frame
- Histogram to count number of chromis

Data Preprocessing

- Video data is separated into frames and uploaded to Roboflow
- Manually created bounding boxes around the chromis fish identified in the frames
- 311 images randomly assigned to to train, validation and test sets based on a 8:1:1 ratio
- Potential bias: all frames come from 3-4 videos

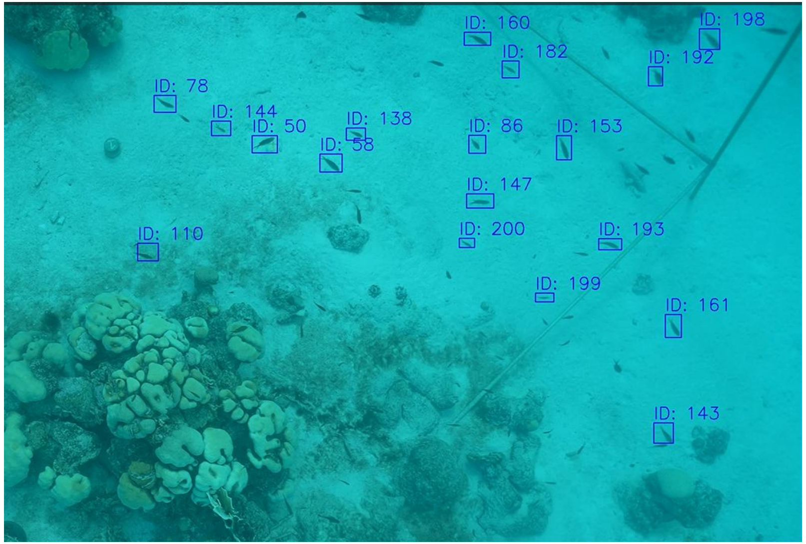


Roboflow annotation setup

Model Training

- Trained an ultralytics YOLOv8 model using the dataset generated from Roboflow
    - YOLO (You Only Look Once): real-time object detection models known for their speed and accuracy
    - Ultralytics YOLOv8: the latest iteration of the YOLO series, designed for high performance in detecting objects in images and videos with improved accuracy in handling small objects and crowded scenes
- Conducted 50 epochs with a batch size of 8
- Deployed on raw videos
- Challenge: algorithm treats fish that disappear (or undetected) and reappear as a new fish

Model Deployment & Tracking



- Used the YOLO model generated to annotate videos
- Mark detected chromis with bounding boxes and unique ID
- Output csv file with chromis information (frame index, ID, x, y, speed, etc.)
- Low accuracy (misdetections)
- One Big Difficulty!

Difficulty 1: Tracking

- Look at fish with id 1. It is not detected several frames and labeled id 85 when reappeared!
- Pair disappeared fish with reappeared fish

Define Loss Functions

- Basic loss function vs Speed-based loss function

$$l_{total}(f_1, f_2) = \lambda_1 l_{position}(f_1, f_2)^2 + \lambda_2 l_{size}(f_1, f_2)^2$$

Basic: $$l_{position}(f_1, f_2) = \sqrt{(x_{f1} - x_{f2})^2 + (y_{f1} - y_{f2})^2}$$

Speed Based: $Expected\ Position = position_{f2} + speed_{f2} \times timeElapsed$, which is

$$E(x_{f2}) = (x, y)_{f2} + v_{f2} \times timeElapsed$$

And the position loss is $l_{position}(f_1, f_2) = \left\| (x_{f1}, y_{f1}) - E(x_{f2}) \right\|$

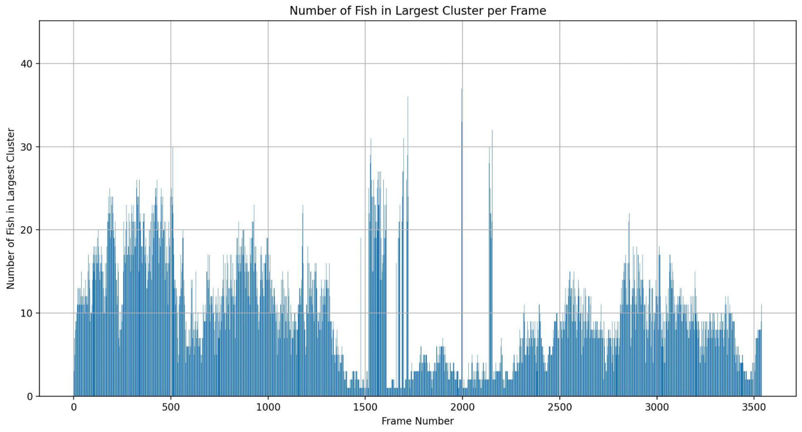
$$l_{size}(f_1, f_2) = \left| size_{f1} - size_{f2} \right|$$

where $size_f = (x_{f2} - x_{f1}) * (y_{f2} - y_{f1})$

Cluster

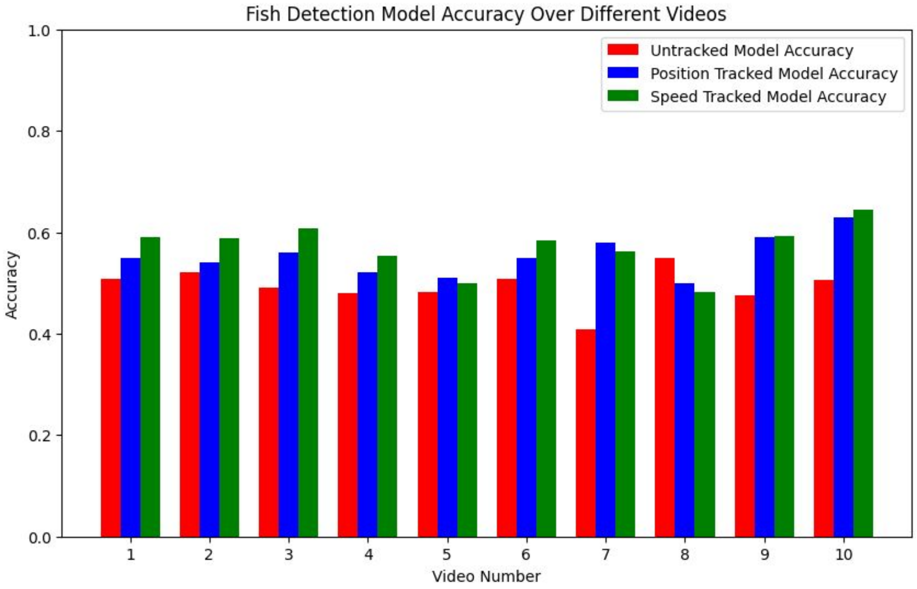
Goal: to estimate the number of fish in colonies by clustering based on spatial location, speed, and direction within each frame

- CSV file from YOLOv8 model containing x, y coordinates, unique track ID, and frame number of each fish
- Calculated speed and direction of fish to enhance features
- Experimented with DBSCAN, MeanShift, and Affinity Propagation → selected DBSCAN
- Performed clustering frame by frame and plotted the size of the largest cluster in each frame on a bar chart



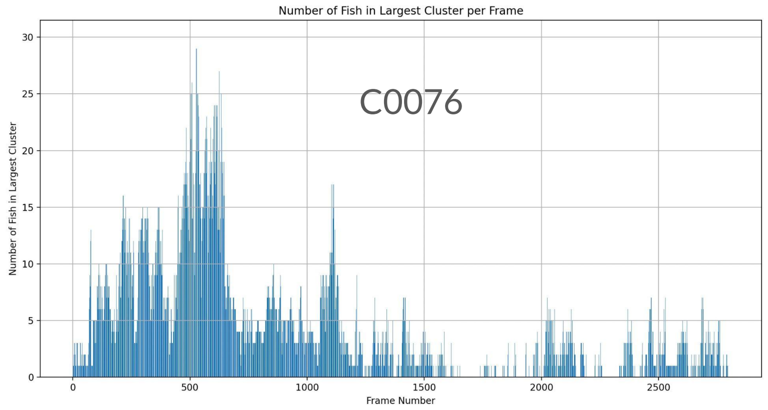
Results

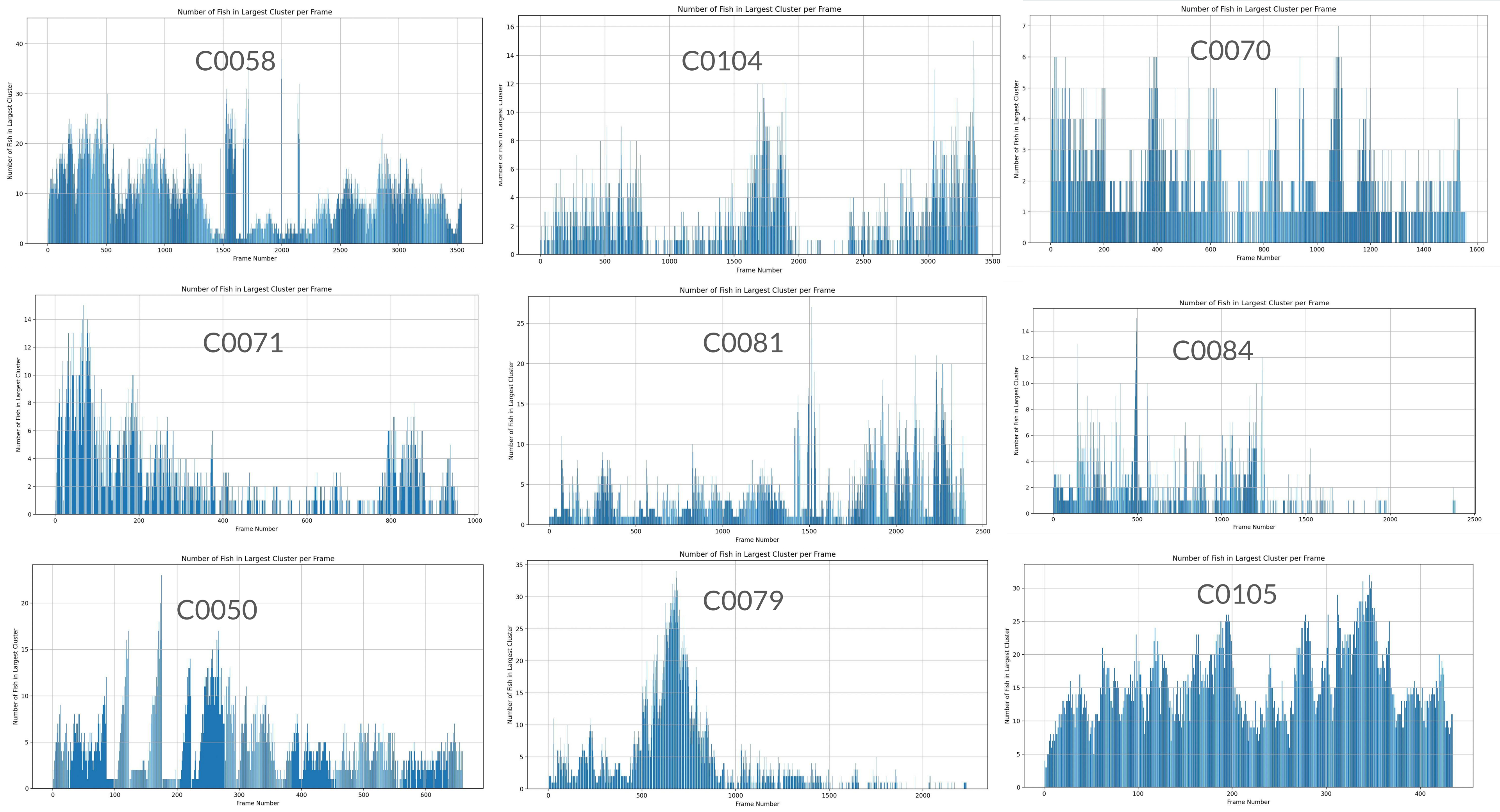
- Three Models
  - Untracked
  - Position Tracked
  - Speed Track
    - Overall perform better than the position tracked model



Results: Largest Clusters vs Frame

- Record the number of fishes in the largest cluster
- Shows that the video captures a large cluster at first denote by the strong peak around frame 500.
- Less fish were detected later
- Verified by human





Number of Fish in Largest Cluster per Frame
C0058
C0104
C0070
C0071
C0081
C0084
C0050
C0079
C0105
Number of Fish in Largest Cluster
Frame Number

Future Direction 1: Improving Tracking & Efficiency

- Better Tracking Algorithm (?)
  - Better Loss function? More complicated? Or simple is better?
  - Rewrite entire tracking algorithm?
- Improve efficiency of tracking helper
  - Current implementation works calls helper every frame

Future Direction 2: Simultaneous MultiCluster

Current focus:

- Primarily deals with frames capturing a single colony
- Reliable estimation of colony sizes for single-colony frames

Challenge:

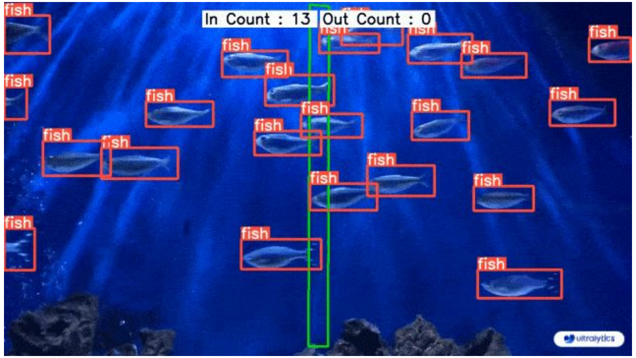
- Multiple clusters in one frame
- Current method only keeps track of the largest cluster

Future steps:

- Extend the method to handle frames with multiple colonies
- Modify existing clustering algorithm to account for multiple significant clusters

Future Direction 3: Counter Optimization

- Cluster by frame vs cluster by video
  - Ideal feature that describes fish throughout the video?
  - Is this possible?
- Object Counting Regions
  - How should object counting regions be defined?
  - Difficulty: Movement of video



Thank You For Listening