

# Cichlid Computer Vision Project – Weekly Progress

Week ending Friday, January 24th, 2025

## Time Log

**Charlie Clark**

### What progress did you make?

- Attempted to fix my WSL 2 environment during the weekend (to no avail).
- Attended Cichlid CV/BioBoost weekly team meeting Monday afternoon.
- Attended publication seminar Tuesday evening.
- Attended Freeman Computational Experts peer meeting Wednesday morning. • Attended weekly HAAG admin meeting Thursday afternoon.
- Met with Kailey to discuss the next steps for the BioBoost project Thursday evening.
- Attended bi-weekly Freeman Faculty-researcher meeting Friday afternoon.
- Continued literature review.
- Reviewed the manuscript submission guidelines for the BioBoost project.
- Continued managing the meeting managers, and collaborated with our higher ed point people, to ensure necessary tasks were completed.

### What are you planning on working on next?

- Continue working with Kailey and Eric on re-running SORT and cropping the tracks into videos.
  - Will be able to perform some computing work on my end next week as well; been refreshing my memory on Windows computing this week to prepare.
- Continue reviewing the manuscript submission guidelines for BioBoost and clarify with Kailey what still needs to be done on the re-write front.
- Continue serving as point person for McGrath and Freeman projects, helping when necessary and delegating when possible.
- Continue literature review.
- Attend all necessary weekly meetings (and some non-required ones when I have free time).
- Re-submit a methods proposal for review by Bree and Jeanette (if last week's submission isn't accepted).

### Is anything blocking you from getting work done?

- **None**

## **Kailey Quesada**

### **What did you do this week?**

- (1) I attended the following 5 meetings:
  - (a) Cichlid Team Working Meeting on January 17th. Project logistics and methods document.
  - (b) Cichlid Team Meeting on January 20th. Went over reID triplet selection and BioBoost.
  - (c) OMSCS Research Seminar on January 21st. The seminar was led by Bree.
  - (d) HAAG Publication Meeting on January 21st. Went over choosing model publications.
  - (e) BioBoost logistics chat with Charlie on January 23rd.
- (2) I worked on the meeting manager role requirements. I listened to the HAAG admin weekly meeting, created the meeting tasks in Microsoft planner, tracked down the required materials (slides, transcripts, notes, recordings, attendance), and wrote up an attendance sheet with meeting links in the project files section on Teams.
- (3) I worked on the BioBoost code since Charlie's computer is down. I combined all YOLO data into one csv, combined all SORT data into one csv, transformed the YOLO coordinate data so that I could match bounding boxes for YOLO and SORT, and matched YOLO and SORT data based on File Name, Frame, and Bounding Box (TopLeft and BottomRight corners). I filtered the matched data to find tracks with more than 25 frames and checked for continuity in the tracks.

### **What are you going to do next week?**

- (1) I need to use a tool like FFmpeg to go through YOLO-generate videos frame by frame to confirm that the missing frames in the tracks are caused by problems with YOLO's output.
- (2) I need to take the SORT code that Charlie used and compare it to the run sort fish detections.py that Bree used. Then, I will need to talk to Charlie to get help re-running SORT with the new code in order to get velocity vectors for TemporalBoost.
- (3) I need to fulfill my meeting manager responsibilities and attend required meetings.

Are there any blockers, things you want to flag, or problems?

- Yes. I still need an email from Dr. Lytle for CS 8903. I will also need PACE access. Most importantly, I need Bree's feedback on the missing frames in the BioBoost project.

## **Eric Iamarino**

### **What did you do this week?**

- Had meeting with Cichlid ReID group to present triplet selection strategy
- Helped draft second iteration of our groups methods document
- Held discussions for transitioning to the BioBoost project
- Reviewed manuscript submission guidelines
- Reviewed current version of BioBoost rewrite

### **What are you going to do next week?**

- Attend BioBoost meeting to find areas I can contribute to the project
- Confirm I have access to necessary tools for project (PACE, Dropbox, website, etc)

### **Blockers, things you want to flag, problems, etc.**

- Need access to website so I can provide updates

## **Abstracts**

### **Charlie Clark**

“SCoralDet: Efficient real-time underwater soft coral detection with YOLO”, Lu et al. (2025; Ecological Informatics)

- Abstract: “In recent years, climate change and marine pollution have significantly degraded coral reefs, highlighting the urgent need for automated coral detection to monitor marine ecosystems. However, underwater coral detection presents unique challenges, including low image contrast, complex coral structures, and dense coral growth, which limit the effectiveness of general object detection algorithms. To address these challenges, we propose SCoralDet, a soft coral detection model based on the YOLO architecture. First, we introduce a Multi-Path Fusion Block (MPFB) to capture coral features across multiple scales, enhancing the model’s robustness to uneven lighting and image blurring. We further improve inference efficiency by applying reparameterization. Second, we integrate lightweight components such as GSConv and VoV-GSCSP to reduce computational overhead without sacrificing performance. Additionally, we develop an Adaptive Power Transformation label assignment strategy, which dynamically adjusts anchor

alignment metrics. By incorporating soft labels and soft central region loss, our model is guided to prioritize high-quality, well-aligned predictions. We evaluate SCoralDet on the Soft-Coral dataset, achieving an inference latency of 9.52 ms and an mAP50 of 81.9. This surpasses the performance of YOLOv5 (79.9), YOLOv6 (79.4), YOLOv8 (79.5), YOLOv9 (78.3), and YOLOv10 (79.5). These results demonstrate the effectiveness and practicality of SCoralDet in underwater coral detection tasks.”

- AI Summary by ChatGPT (4o): “The paper proposes SCoralDet, a real-time object detection framework specifically designed for underwater soft coral detection, addressing the challenges of low contrast, complex structures, and overlapping corals in underwater environments. Built on the YOLOv10 architecture, SCoralDet introduces a Multi-Path Fusion Block (MPFB), lightweight modules (GSConv, VoV-GSCSP), and an Adaptive Power Transformation (APT) strategy to improve detection accuracy and efficiency. The model outperforms state-of-the-art methods, achieving an mAP50 of 81.9% on the SoftCoral dataset with an inference latency of 9.52 ms.”
  - Key Contributions
    - “Innovative Architecture: The Multi-Path Fusion Block (MPFB) captures multi-scale features, enhancing the model's robustness to uneven lighting and image blurring. Lightweight components (GSConv and VoV-GSCSP) reduce computational costs without sacrificing performance.”
    - “Adaptive Label Assignment: The APT label assignment strategy dynamically adjusts anchor alignment metrics, using soft labels and a tailored loss function to improve classification and localization.”
    - “Benchmark Results: SCoralDet outperforms recent YOLO variants (YOLOv8, YOLOv9, YOLOv10) and other single-stage detection algorithms in precision and latency.”
  - Contributions to Knowledge:
    - “Tailored Coral Detection: The paper addresses the unique challenges of underwater environments, such as low visibility and intricate coral structures, through a specialized YOLO-based framework.”
    - “Enhanced Feature Extraction: By integrating the MPFB and reparameterization techniques, the model excels in capturing fine-grained features essential for distinguishing between overlapping and visually similar coral species.”

- “Efficient Real-Time Application: SCoralDet demonstrates the feasibility of deploying high-performing detection models on resource-constrained devices, making it practical for large-scale marine monitoring.”
- Future Research Directions:
  - “Integration with Multi-Modal Data: Incorporating additional data sources like sonar or LiDAR could enhance detection robustness in diverse underwater conditions.”
  - “Application to Other Marine Objects: Extending the framework to detect other underwater objects, such as fish or invertebrates, could broaden its ecological applications.”
  - “Weakly Supervised Learning: Adopting weakly supervised or unsupervised learning could reduce the reliance on annotated datasets, making the approach scalable to new species and regions.”
  - “Optimization for Smaller Targets: Further development is needed to improve the detection of small or j

• Link: <https://www.sciencedirect.com/science/article/pii/S1574954124004795>

### **Kailey Quesada**

Xia, X., et al. “PAB-Mamba-YOLO: VSSM assists in YOLO for aggressive behavior detection among weaned piglets.” *Artificial Intelligence in Agriculture*, 2025.

<https://www.sciencedirect.com/science/article/pii/S2589721725000017>.

Aggressive behavior among weaned piglets, such as climbing, nose hitting, biting tails, and biting ears, can harm their health and growth, making it important to monitor these actions for better breeding management. <sup>1</sup> This study introduces a new model called PAB-Mamba-YOLO, which combines advanced techniques to efficiently detect these aggressive behaviors using computer vision. The model shows high accuracy in identifying different types of aggression and operates quickly, making it a valuable tool for farmers to manage piglet behavior effectively. The proposed CSPVSS module is a new design that combines a selective scanning mechanism (VSSM) with a dual-branch structure to improve how features from images are extracted and processed. One branch focuses on maintaining local details using standard convolutional operations, while the other branch captures long-range relationships through more complex processing. By merging the outputs from both branches, the CSPVSS module enhances the overall feature

representation, making it more effective for tasks like image detection. No future work is discussed.

## **Eric Iamarino**

Samuel Hoffmann, Laura Steiner, Andreas H. Schweiger, Alessandro Chiarucci, Carl Beierkuhnlein, Optimizing sampling effort and information content of biodiversity surveys: a case study of alpine grassland, *Ecological Informatics*, Volume 51, 2019, Pages 112-120, ISSN 1574-9541, <https://doi.org/10.1016/j.ecoinf.2019.03.003>.

**Aims:** Current rates of biodiversity loss do not allow for inefficient monitoring. Optimized monitoring maximizes the ratio between information and sampling effort (i.e., time and costs). Sampling effort increases with the number and size of sampling units. We hypothesize that an optimal size and number of sampling units can be determined providing maximal information via minimal effort. We apply an approach that identifies the optimal size and number of sampling quadrats. The approach can be adapted to any study system. Here we focus on alpine grassland, a diverse but threatened ecosystem. Location: Gran Paradiso National Park, Italy. **Methods:** We sampled nine 20 m × 20 m-plots. Each plot consisted of 100 2 m × 2 m-subplots. Species richness and Shannon diversity were quantified for different sizes and quantities of subplots. We simulated larger subplot sizes by unifying adjacent 2 m × 2 m-subplots. Shannon's information entropy was used to quantify information content among richness and diversity values resulting from different subplot sizes and quantities. The optimal size and number of subplots is the lowest size and number of subplots returning maximal information. This optimal subplot size and number was determined by Mood's median test and segmented linear regression, respectively. **Results:** The information content among richness values increased with subplot size, irrespective of the number of subplots. Therefore, the largest subplot size available is the optimal size for information about richness. Information content among diversity values increased with subplot size if 18 or less subplots were considered, and decreased if at least 27 subplots were sampled. The subplot quantity consequently determined whether the smallest or largest subplot size available is the optimal size, and whether the optimal size can be generalized across richness and diversity. Given a 2 m × 2 m size, we estimated an optimal quantity of 54. Given a size of 4 m × 4 m, we estimated an optimal number of 36. The optimal number of plots can be generalized across both indices because it barely differed between the indices given a fixed subplot size. **Conclusions:** The information content among richness and diversity values depends on the sampling scale. Shannon's information entropy can be used to identify the optimal number and size of plots that return most information with least sampling effort. Our approach can be

adapted to other study systems to create an efficient insitu sampling design, which improves biodiversity monitoring and conservation under rapid environmental change.

## Documentation of Work

### Charlie Clark

- Attempted to fix my WSL 2 environment during the weekend (to no avail).
- Attended Cichlid CV/BioBoost weekly team meeting Monday afternoon.
- Attended publication seminar Tuesday evening.
- Attended Freeman Computational Experts peer meeting Wednesday morning.
- Attended weekly HAAG admin meeting Thursday afternoon.
- Met with Kailey to discuss the next steps for the BioBoost project Thursday evening.
- Attended bi-weekly Freeman Faculty-researcher meeting Friday afternoon.
- Continued literature review.
- Reviewed the manuscript submission guidelines for the BioBoost project.
- Continued managing the meeting managers, and collaborated with our higher ed point people, to ensure necessary tasks were completed.
- Proof:
  - Ecological informatics submission guidelines I was reviewing: <https://www.sciencedirect.com/journal/ecological-informatics/publish/guide-forauthors>
  - BioBoost publication re-write I was reviewing: [Publication.docx](#)
  - Bi-weekly Freeman Faculty-researcher meeting recording: [FACULTY\\_Hume's Leaf Warbler Projects-20250124\\_130222-Meeting Recording.mp4](#)
  - Weekly Cichlid CV/BioBoost meeting recording: [Cichlid CV Weekly Meeting20250120\\_173606-Meeting Recording.mp4](#)

### Kailey Quesada

For a full list of what I did, see the time log above. The most important thing I worked on this week was combining YOLO and SORT data and finding track suitable for our





```
21 track_id += 1
```

Listing 1: Create CSV with All YOLO Data.

Next, the YOLO coordinates need to be transformed so that we can match the YOLO and SORT bounding boxes. For the Lindenthal dataset, the frame width is 1280, and the frame height is 720. We want to take the relative location of the framing box found by YOLO and find the absolute location of the box. Additionally, SORT needs the relative positions of the top left and bottom right of each box.

```
1 ...
2 primary_csv['Abs_Center_X'] = primary_csv['Center_X'] * frame_width
3 primary_csv['Abs_Center_Y'] = primary_csv['Center_Y'] * frame_height
4 primary_csv['Abs_Width'] = primary_csv['Width'] * frame_width
5 primary_csv['Abs_Height'] = primary_csv['Height'] * frame_height
6 primary_csv['Abs_Top_Left_X'] = primary_csv['Abs_Center_X'] - primary_csv['Abs_Width'] / 2
7 primary_csv['Abs_Top_Left_Y'] = primary_csv['Abs_Center_Y'] - primary_csv['Abs_Height'] / 2
8 primary_csv['Top_Left_X'] = primary_csv['Abs_Top_Left_X'] / frame_width
9 primary_csv['Top_Left_Y'] = primary_csv['Abs_Top_Left_Y'] / frame_height
10 primary_csv['Abs_Bottom_Right_X'] = primary_csv['Abs_Top_Left_X'] + primary_csv['Abs_Width']
11 primary_csv['Abs_Bottom_Right_Y'] = primary_csv['Abs_Top_Left_Y'] + primary_csv['Abs_Height']
12 primary_csv['Bottom_Right_X'] = primary_csv['Abs_Bottom_Right_X'] / frame_width
13 primary_csv['Bottom_Right_Y'] = primary_csv['Abs_Bottom_Right_Y'] / frame_height
14 ...
```

Listing 2: Transform YOLO Data.

After our YOLO data is ready, we can create a SORT csv. For each file, SORT outputs CSV files with the following fields: frame, track.id, x\_min, y\_min, x\_max, y\_max. We want to append the CSV file name of each video as a field for each CSV. Then, we want to combine all of these files together into one CSV. The unique ID of each frame of a track can be created by using filename\_frame\_trackID.

```
1 ...
2 # Go to Each TXT File in Directory and Add Relevant Details to Primary CSV
3 with open(primary_csv, mode="a", newline="") as primary_file:
4     csv_writer = csv.writer(primary_file, delimiter=",")
5     csv_writer.writerow(["File_Name", "Frame", "Track_ID", "X_Min", "Y_Min", "X_Max", "Y_Max"])
6     for root, _, files in os.walk(path_to_dir):
7         for file in files:
8             if file.endswith(".csv"):
9                 file_path = os.path.join(root, file)
10                file_name_for_csv = os.path.splitext(file)[0]
11                with open(file_path, mode="r") as individual_file:
12                    csv_reader = csv.reader(individual_file)
13                    next(csv_reader)
14                    for row in csv_reader:
15                        if len(row) == 6:
16                            frame, track_id, x_min, y_min, x_max, y_max = row
17                            file_name = f"{file_name_for_csv}"
18                            csv_writer.writerow([file_name, frame, track_id, x_min, y_min,
19                                                x_max, y_max])
```

Listing 3: Create CSV with All SORT Data.

Now, we can combine YOLO and SORT data. We will first round column data. Then, we will match bounding boxes, file, and frame numbers. Basically, a bounding box has to be from the same frame of a file and have a close enough bounding box to be considered a match between YOLO and SORT. The point where X and Y are at their minimum is the top left corner of the bounding box, since the origin is at the top left in computing. Therefore, the point where X and Y are at their maximum is at the bottom right corner of the bounding box. Based on this information, we can combine data.

```
1 ...
2 yolo_df['X_for_matching'] = yolo_df['Top_Left_X']
3 yolo_df['Y_for_matching'] = yolo_df['Top_Left_Y']
4 sort_df['X_for_matching'] = sort_df['X_Min']
5 sort_df['Y_for_matching'] = sort_df['Y_Min']
6 yolo_df['File_Prefix'] = yolo_df['Unique_ID'].str.split('_').str[0]
```

```

7 sort_df['File_Prefix'] = sort_df['File_Name'].str.split('_').str[0]
8 yolo_df['Frame_Part'] = yolo_df['Unique_ID'].str.split('_').str[3]
9 sort_df['Frame_Part'] = sort_df['Frame'].astype(str)
10 yolo_df.set_index(['X_for_matching', 'Y_for_matching'], inplace=True)
11 sort_df.set_index(['X_for_matching', 'Y_for_matching'], inplace=True)
12 matched_df = pd.merge(yolo_df, sort_df, how='inner', left_index=True, right_index=True)
13 matched_df = matched_df[
14     (matched_df['File_Prefix_x'] == matched_df['File_Prefix_y']) &
15     (matched_df['Frame_Part_x'] == matched_df['Frame_Part_y'])
16 ]
17 ...

```

Listing 4: Combine YOLO and SORT Data.

Then, we can filter by the number of frames required for a track to be in the dataset.

```

1 ...
2 frame_count_threshold = 25
3 tracks_above_threshold = frame_counts[frame_counts['Frame_Count'] > frame_count_threshold]
4 ...

```

Listing 5: Filter for Tracks with 25 Frames or More.

And that's it! The finished scripts will be uploaded to BioBoost upon completion: <https://github.com/Human-Augment-Analytics/Bio-Boost>.

## Eric Iamarino

This week, I began transitioning from the Cichlid reID project to the BioBoost project. To help with the transition, I reviewed the manuscript submission guidelines and the current draft of the paper's rewrite. Additionally, I had discussions with Kailey and Charlie to get a better understanding of the BioBoost project's objectives and requirements.

Documents Reviewed:

<https://www.sciencedirect.com/journal/ecological-informatics/publish/guide-for-authors>

[https://gtvault.sharepoint.com/:w:/s/HAAG/ES4iZkwggI1LslNfPp9\\_oVUBDmWtV6CyYaRwi2s0cuU29A?e=aCQQgt](https://gtvault.sharepoint.com/:w:/s/HAAG/ES4iZkwggI1LslNfPp9_oVUBDmWtV6CyYaRwi2s0cuU29A?e=aCQQgt)