Cichlid Computer Vision Project – Weekly Progress

Week ending Friday, November 22, 2024

Summary (please see all detailed attachments below):

<u>Kailey Quesada</u>

1.1 What progress did you make in the last week?

(1) For the Lizard project, I looked into B-SOiD by looking at the GitHub, paper, and some seminars on

YouTube from the creators of the tool.

(2) I installed B-SOiD and tested it twice on 22 pose estimation files and videos from DLC. The first time,

there were 53 clusters and second time there were 33 clusters. I was able to get preliminary results in terms

of having B-SOiD label lizard behaviors using unsupervised techniques.

(3) I created a document to start compiling information for Dr. Stroud and a GitHub for the project.

(4) For meetings, I attended both the Cichlid CV team meeting and the publication seminar on November

15th.

1.2 What are you planning on working on next?

(1) For the Lizard project, I want to run B-SOiD again with fewer clusters to see if the results are any more

meaningful than the first two times. I'd also like to look at more video examples of smaller clusters to get a

more full idea of what the tool is picking out in terms of lizard behavior.

(2) For the Lizard project, I need to continue with SimBA and start figuring out how to compare SimBA

with B-SOiD. The biggest barrier is that SimBA requires annotations to identify behaviors while B-SOiD does not.

(3) I need to start adding stuff to the new Lizard GitHub repository.

(4) For the BioBoost project, Bree would like me to revise TemporalBoost to take the highest performing

model instead of the DT. Bree also wants me to look into ID switching and its effect on our research. Since

my focus is on the Lizard project, I may have to wait on this until winter break.

(5) For the DLC + ReID project, I still need to check out Thuan's code and learn about the different models

we will be using with the DLC output. Since my focus is on the Lizard project, I may have to wait on this

until winter break or next semester.

(6) For the DLC + ReID project, I need to help with the abstract and the paper outline for the publication

seminar.

(7) For meetings, I will attend both the Cichlid CV team meeting and the publication seminar.

1.3 Is anything blocking you from getting work done?

Getting some feedback from Bree or the Higher Ed team on the semester-end report for Dr. Stroud would

be very helpful.

Charles Clark

What progress did you make in the last week?

• Read the Lindenthal Camera Traps paper.

• Trained YOLOv5s on the YOLOv5 training dataset extracted last week.

• Verified the YOLOv5s model on the YOLOv5 testing dataset I extracted as well,

calculating the class accuracies of the predictions.

• Attended weekly Cichlid CV meeting on Tuesday night.

• Updated the HAAG website's Meet the Team pages.

• Figured out how to extract videos from .bag files.

o (Attempted) to write a bash script that automates the process, but it wouldn't run… will most likely need to re-write in Python instead.

• Currently attending weekly publication seminar.

What are you planning on working on next?

• Attend weekly Cichlid CV team meeting Tuesday evening.

• Continue working with Kailey and Bree on their project.

o Try to write a Python script that can automate the extraction of the videos from

the .bag files.

o If time allows, continue working through the process Bree has asked me to follow.

• Continue literature review.

• Attend any seminars that might be scheduled for next week.

Is anything blocking you from getting work done?

• Not at this time, no.

Thuan Nguyen

What progress did you make in the last week?

- I discussed with Adam regarding the implementation of the DeepLabCut+ViT model and addressed challenges such as memory issues and replicating his results (>90% when evaluated on test triplets).
- Also tested the DeepLabCut's default transformer ReID model using triplet datasets of varying sizes but observed no significant improvement in accuracy across epochs, with test accuracy plateauing at ~55%.
- I tested DeepLabCut's own re-ID implementation against several videos. Results were similarly poor, whether with 1000, 10000 or 50000 triplets, or with more epochs, or with different videos (See details below).
- Before that, I encountered memory issues with a ViT feature extractor integrated into the DeepLabCut transformer ReID pipeline. I later resolved the memory issue and replicated Adam's DLC+ViT training experiment, achieving high accuracy results similar to Adam's runs.
- In discussion with Bree and Charlie, I analyzed and compared the suitability of PvT, ViT, and TransReID for the fish ReID task, highlighting challenges in applying these models to limited fish datasets.

What are you planning on working on next?

- I still need generate longer videos (where fish appear more often) in order to create more interesting triplet dataset on which to train and evaluate the transformer re-ID pipelines on.

So far, I've evaluated these training runs on 1-minute, 2-minute, or 3-minute videos, but I want to run these models on more diverse videos.

Is anything blocking you from getting work done?

- Not at the moment.

Week 14 Document Submission

Kailey Quesada (Cichlid CV & Lizard CV Team)

November 22, 2024

1 Weekly Project Updates

1.1 What progress did you make in the last week?

(1) For the Lizard project, I looked into B-SOiD by looking at the GitHub, paper, and some seminars on YouTube from the creators of the tool.
(2) I installed B-SOiD and tested it twice on 22 pose estimation files and videos from DLC. The first time, there were 53 clusters and second time there were 33 clusters. I was able to get preliminary results in terms of having B-SOiD label lizard behaviors using unsupervised techniques.
(3) I created a document to start compiling information for Dr. Stroud and a GitHub for the project.
(4) For meetings, I attended both the Cichlid CV team meeting and the publication seminar on November 15th.

1.2 What are you planning on working on next?

(1) For the Lizard project, I want to run B-SOiD again with fewer clusters to see if the results are any more meaningful than the first two times. I'd also like to look at more video examples of smaller clusters to get a more full idea of what the tool is picking out in terms of lizard behavior.
(2) For the Lizard project, I need to continue with SimBA and start figuring out how to compare SimBA with B-SOiD. The biggest barrier is that SimBA requires annotations to identify behaviors while B-SOiD does not.
(3) I need to start adding stuff to the new Lizard GitHub repository.
(4) For the BioBoost project, Bree would like me to revise TemporalBoost to take the highest performing model instead of the DT. Bree also wants me to look into ID switching and its effect on our research. Since my focus is on the Lizard project, I may have to wait on this until winter break.
(5) For the DLC + ReID project, I still need to check out Thuan's code and learn about the different models we will be using with the DLC output. Since my focus is on the Lizard project, I may have to wait on this until winter break or next semester.
(6) For the DLC + ReID project, I need to help with the abstract and the paper outline for the publication seminar.
(7) For meetings, I will attend both the Cichlid CV team meeting and the publication seminar.

1.3 Is anything blocking you from getting work done?

Getting some feedback from Bree or the Higher Ed team on the semester-end report for Dr. Stroud would be very helpful.

2 Document Submission

2.1 Abstract

Hsu, A., and Yttri, E. "B-SOiD, an open-source unsupervised algorithm for identification and fast prediction of behaviors." Nature Communications 2021. `https://www.nature.com/articles/s41467-021-25420-x`.

B-SOiD is an open-source algorithm designed to identify and predict animal behaviors quickly and without user bias. It uses advanced machine learning techniques to analyze the positions of limbs captured in video, allowing researchers to extract detailed information about movements and actions in various animal models. This tool improves the speed and accuracy of behavioral analysis, making it easier to study complex behaviors in different settings and across different species. No future work is discussed in the paper. A further reference to a video resource created by the authors is listed in the references section of this weekly report.

2.2 Scripts and Code Blocks

This week's helpful scripts can be found here: `https://github.com/Human-Augment-Analytics/Higher-Ed/tree/main/Personal%20Folders/KaileyCozart/Fall%202024/DLC%20Week%2014`. I created a GitHub for the project on HAAG, and next week I will start adding stuff to it: `https://github.com/Human-Augment-Analytics/Lizard-Pose-Estimation-and-Evaluation`.

2.2.1 Lizard Project: B-SOiD

While it is not needed for the Lizard dataset, it might be helpful to have code to split larger videos into smaller samples. The code in Listing 1 is taken from Thuan and modified. In this case, it is taking a 60 second clip from the original 2-hour-long video.

```python
from moviepy.video.io.ffmpeg_tools import ffmpeg_extract_subclip

input_video = "0001_vid.mp4"
output_video = "0001_vid_60secs_e.mp4"

start_time = 240
end_time = 300

ffmpeg_extract_subclip(input_video, start_time, end_time, targetname=output_video)
```

Listing 1: Getting smaller samples of larger video.

If desired, you can convert DLC .h5 files into .csv files or vise versa. B-SOiD will take either, but, if you prefer a certain file type, you can use the code in Listing 2. Converting to .csv is required for SimBA.

```python
import pandas as pd

h5_file = 'videos/0001_vid_60secs_eDLC_Resnet50_dlc_modelJul26shuffle1_snapshot_125_el.h5'
csv_file = 'videos/0001_vid_60secs_eDLC_Resnet50_dlc_modelJul26shuffle1_snapshot_125_el.csv'

data = pd.read_hdf(h5_file)
data.to_csv(csv_file)
```

Listing 2: Converting h5 to csv.

Finally, it will likely be helpful to have code to convert B-SOiD result CSVs into visualizations of some aspects of the clustering results. See Listing 3.

```python
# Imports
import os
import pandas as pd
import seaborn as sns
import matplotlib.pyplot as plt

# Directory
directory_path = r"C:\Users\username\filepath\Bsoid\control\BSOID"

# Initialize Dictionaries for Counts and Durations
```

```python
behavior_counts = {}
behavior_durations = {}

# Tally Data from Multiple CSVs
for filename in os.listdir(directory_path):
    if filename.endswith(".csv"):

        # Read File
        file_path = os.path.join(directory_path, filename)
        df = pd.read_csv(file_path)

        # Drop 'Unnamed: 0' Column
        if 'Unnamed: 0' in df.columns:
            df = df.drop(columns=['Unnamed: 0'])

        # Ensure Required Columns Exist
        if 'B-SOiD labels' not in df.columns or 'Run lengths' not in df.columns:
            print(f"Skipping file {filename} due to missing columns")
            continue

        # Aggregate
        for _, row in df.iterrows():
            label = row['B-SOiD labels']
            duration = row['Run lengths']
            behavior_counts[label] = behavior_counts.get(label, 0) + 1
            behavior_durations[label] = behavior_durations.get(label, 0) + duration

# Convert Data to DataFrame
summary_df = pd.DataFrame({
    'Behavior': list(behavior_counts.keys()),
    'Count': list(behavior_counts.values()),
    'Total Duration': [behavior_durations[b] for b in behavior_counts.keys()]
})

# Sort Results for Visualization
most_common_behaviors = summary_df.sort_values(by='Count', ascending=False)
longest_duration_behaviors = summary_df.sort_values(by='Total Duration', ascending=False)

# Plot
...
```

Listing 3: Determining most common behaviors and longest duration behaviors.

2.3 Documentation

This week, I am going to cover the process of installing and using B-SOiD's basic features.

2.3.1 What B-SOiD Does

B-SOiD (Behavioral Segmentation of Open-field In Deep Learning) allows users to find behaviors using unsupervised learning - without the need for annotated data. Specifically, B-SOiD finds clusters in animal behavior using pose estimation data from another tool such as DeepLabCut. Seven low-covalence features are used in combination with the t-SNE and GMM algorithms to cluster the data [1]. Finally, the likelihood processing step helps to define poses and occlusions [1].

2.3.2 Data Requirements

Data for B-SOiD is split between control and experimental. You can choose to use just control data or you can use both control and experimental. An example of where you would use both control and experimental would be if you were comparing between two species of lizard. For both control and experimental, approximately 20-40 videos are needed to train a model for simple animal behavior, assuming each video is approximately 15 seconds long.

2.3.3 Usage Requirements

B-SOiD is available in both MATLAB and Python versions. Because B-SOiD relies upon the t-SNE algorithm for clustering during one of its processing steps, the specific implementation of t-SNE can effect B-SOiD's performance. Specifically, the MATLAB version of B-SOiD is able to rely on a MATLAB implementation of t-SNE with more tweak-able parameters than the Python version of t-SNE [1]. While the Python version of B-SOiD was used for this demonstration, it should be noted that, especially for larger datasets, the MATLAB version will likely outperform the Python version of B-SOiD. Should this direction look promising, it is likely that we could use a MATLAB license from Georgia Tech, since MATLAB does cost money.

2.3.4 Python Setup

The setup for B-SOiD's Python version is mostly straightforward. The instructions are listed in the project GitHub: `https://github.com/YttriLab/B-SOID`. The steps are as follows: download the GitHub repository, create an environment with the requirements listed in the appropriate requirements file, activate the new environment, and run a python script to start the GUI. For this demonstration, Windows was used. The only issue was that tables was required in the anaconda environment. While B-SOiD is recommended for Python 3.7 and 3.8, tables requires Python 3.9 or higher. The solution was to pip install tables instead of using conda to install. I would also recommend installing "pip install imageio-ffmpeg" up front so that you have full functionality later on.

2.3.5 GUI Usage

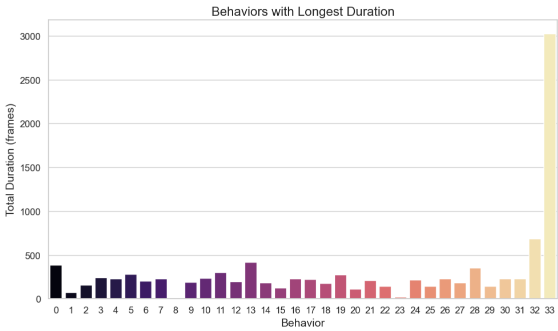
There are both command line and GUI options available for this tool. I will be describing the GUI option here. The first thing to understand about the GUI option is that is assumes a root directory that has two folders, control and experimental. Both control and experimental have the necessary video files and data files. In our case, the videos generated by DLC are in the mp4 format, and the accompanying data files are in the h5 format. Other video formats are accepted by B-SOiD, but mp4 is preferred. For the accompanying data files, either h5 or csv can be used. Additionally, a config.yaml from DLC might be needed in some instances. While videos are not technically required, you will want to include them to be able to visualize the behaviors that B-SOiD is picking up on.

After activating your B-SOiD environment and running "streamlit run bsoid_app.py," the GUI will appear in your browser and the prompts are somewhat easy to follow with some trial and error. Pay attention to what you are checking and unchecking in the toolbar on the left. When you get to the "load previous item" step, make sure that you set your working directory to where the .sav file is located. Typically, this would be under your "output" sub-folder. When outputting the side-by-side videos shown on the B-SOiD GitHub, be sure to edit the default video lengths to get the full video.

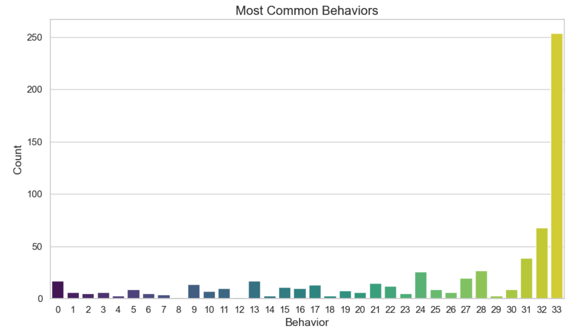
2.3.6 Cluster Number Selection & Interpreting Results

A big part of the resulting clustering is the minimum cluster size range that the user defines during the clustering process. Minimum cluster sizes that are too small can result in an extreme number of clusters that might not break down into useful behaviors. For example, on the lizard data provided, a human annotator might annotate using only two clusters: moving and not moving. However, t-SNE finds 53 clusters when the minimum cluster size is between 0.5% and 1%, which is equivalent to roughly 2 seconds for the smallest cluster. By changing the minimum cluster size to between 1% and 5%, which is equivalent to roughly 4 seconds for the smallest cluster, t-SNE clustering results in 33 clusters. But what do these additional clusters represent? Using Python, we can take the bout_length csv's in "Bsoid / control / BSOID" and determine which behaviors were most common and which behaviors had the longest durations. This can give us some clues to what is going on. Based on the images in Figure 1, we can see that behavior 33 is the most common and the longest in duration. Behavior 32 is the second most common and the second longest in duration. The other behaviors are somewhat less common.

Now, the easiest way to interpret the groupings that t-SNE is finding is to look at the grouping examples that B-SOiD exports to the "Bsoid / control / mp4s / subfolder-name" directory. Alternatively, the "Bsoid / output / bsoid_videos" directory can also be consulted. Very quickly, the more common classes will stick



(a) Comparing Behavior Duration



(b) Comparing Behavior Frequency

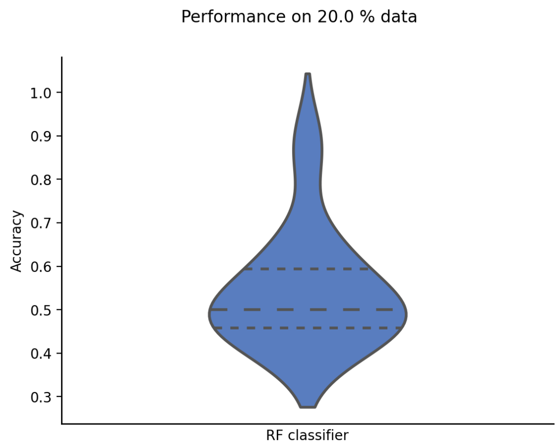
Figure 1: Analyzing Behaviors Discovered by t-SNE

out and, if analysis has gone well, B-SOiD will have grouped similar behaviors together based on the pose estimation data.

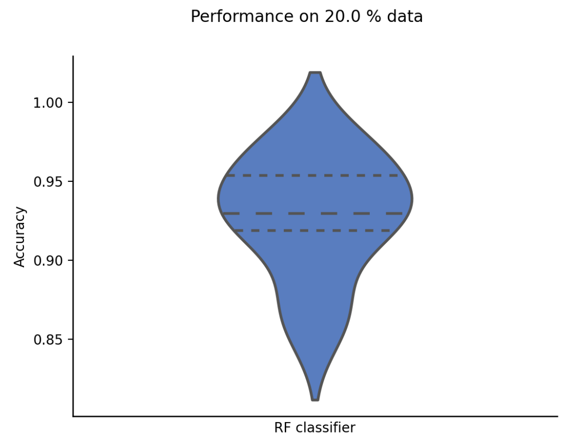
In our case, looking at examples of behaviors 32 and 33, we can see that these are running behaviors. Behavior 30 is a pausing behavior with the limbs splayed and the head straight forward or the lizard being at the top of the course unmoving with no key points visible. Behavior 20 is a pausing behavior with the body curled a bit to the right. Behavior 14 is a pausing behavior with the head cocked to the left. Behavior 13 is a pausing behavior with the right foot higher. Behavior 7 is pausing behavior with the left foot higher. Behaviors 0, 2, and 4 are where the experimenter hand is blocking all points. Behavior 1 is where the model is jittery can is losing some of the key points. Behaviors 27 and 28 are other pausing behaviors with different limb angles. Other clusters represent pausing, occlusion, or movement. Depending on the level of analysis desired and the complexity of the behaviors observed in different groups, the number of clusters should be increased or decreased via the minimum cluster size range.

2.3.7 Model Improvement and Performance Evaluation

To determine if the amount of data you are providing is enough for B-SOiD to consistently assign behaviors to the correct cluster, one can evaluate the performance plots produced by B-SOiD, as shown in Figure 2. A model lacking sufficient data will have low accuracy, like the plot in Figure 2a. In contrast, a higher performing model indicates that the model is able to consistently find an appropriate cluster for the data, as shown in Figure 2b.



(a) Learning on 3 Videos



(b) Learning on 22 Videos

Figure 2: Performance at Identifying Simple Lizard Behaviors

From here, the various plots provided using the B-SOiD GUI or command line can be used to make further improvements to the model. These include visualizations of clustering, directed graphs, trajectory visuals, confusion matrices, and k-fold accuracy plots. Additionally, combined videos can be used to evaluate the progression of different behaviors. A screenshot of one such video can be seen in Figure 3. As the video progresses, each frame is associated with a new point in a cluster and the cluster for that frame is printed to the screen.

Figure 3: Example Frame of Side-by-Side Video

2.3.8 Further Usage

With the inclusion of additional data from, say, another species of lizard, distinctions between lizard movement based on species could be explored using this unsupervised approach. The beauty of B-SOiD is that is has a very easy-to-use GUI compared to other tools of its type while also allowing users to analyze non-annotated data to learn about animal behavior. For datasets such as the Lizard dataset, which has less complicated behaviors to be learned, this tool would be highly applicable.

2.4 Script Validation (Optional)

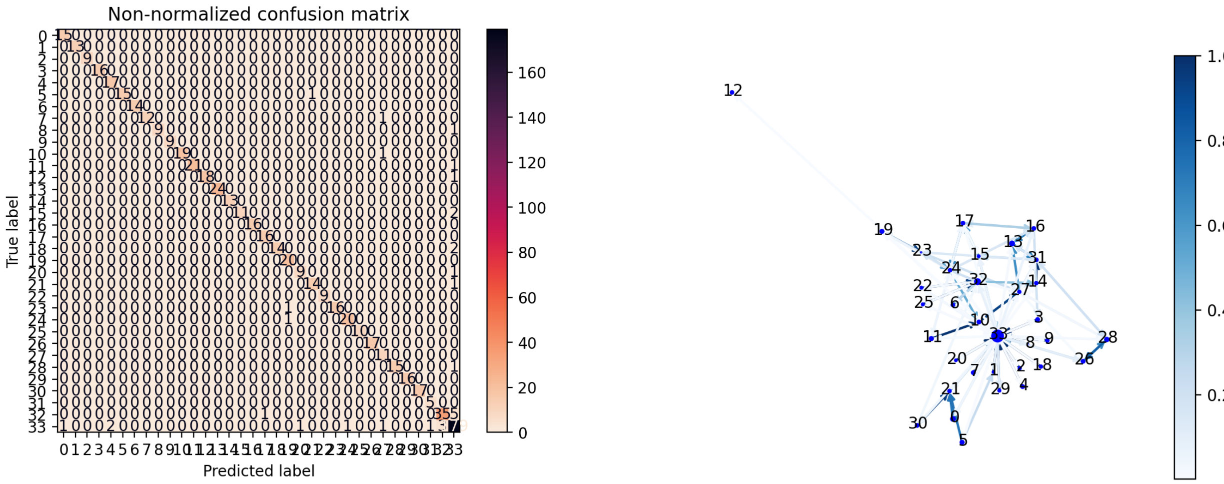
No additional script validation information needed this week.

2.5 Results Visualization

To visualize cluster frequency and behavior duration, see Figure 1 in the Documentation section. To see performance plots, see Figure 2 in the Documentation section. There are a lot of other plots provided by B-SOiD that I would like to look into more. Two of them are displayed in Figure 4.

2.6 Proof of Work

As discussed in the Results Visualization section, there are varios plots. In addition to those in Figures 2 and 4, there are accuracy box plots for the random forests, clustering plots, and limb trajectory plots. Additionally, as previously mentioned, I want to see what the results are like with fewer clusters. I've looked at both 53 and 33 clusters, but adjusting the minimum cluster size range could help me lower the number of clusters even more. Finally, I want to focus on finding examples of each minority cluster from videos so that I can really understand how it is clustering these smaller groups.



(a) Confusion Matrix

(b) Directed Graph

Figure 4: Additional Plots

2.7 Next Week's Proposal

As stated on the first page, this week's goals are the following:
(1) For the Lizard project, I want to run B-SOiD again with fewer clusters to see if the results are any more meaningful than the first two times. I'd also like to look at more video examples of smaller clusters to get a more full idea of what the tool is picking out in terms of lizard behavior.
(2) For the Lizard project, I need to continue with SimBA and start figuring out how to compare SimBA with B-SOiD. The biggest barrier is that SimBA requires annotations to identify behaviors while B-SOiD does not.
(3) I need to start adding stuff to the new Lizard GitHub repository.
(4) For the BioBoost project, Bree would like me to revise TemporalBoost to take the highest performing model instead of the DT. Bree also wants me to look into ID switching and its effect on our research. Since my focus is on the Lizard project, I may have to wait on this until winter break.
(5) For the DLC + ReID project, I still need to check out Thuan's code and learn about the different models we will be using with the DLC output. Since my focus is on the Lizard project, I may have to wait on this until winter break or next semester.
(6) For the DLC + ReID project, I need to help with the abstract and the paper outline for the publication seminar.
(7) For meetings, I will attend both the Cichlid CV team meeting and the publication seminar.

References

[1] A. Hsu, *B soid demonstration and use 4/29/20*, YouTube user metalitia4, 2020. [Online]. Available: `https://www.youtube.com/watch?v=wFZFDpUBPjI`.

Charles R. Clark
CS 6999 – HAAG: Cichlid CV
Fall 2024
November 22, 2024

Week 14 Report

1. *Time-log Response:*

What progress did you make in the last week?

- Read the Lindenthal Camera Traps paper.
- Trained YOLOv5s on the YOLOv5 training dataset extracted last week.
- Verified the YOLOv5s model on the YOLOv5 testing dataset I extracted as well, calculating the class accuracies of the predictions.
- Attended weekly Cichlid CV meeting on Tuesday night.
- Updated the HAAG website's Meet the Team pages.
- Figured out how to extract videos from .bag files.
    - (Attempted) to write a bash script that automates the process, but it wouldn't run… will most likely need to re-write in Python instead.
- Currently attending weekly publication seminar.

What are you planning on working on next?

- Attend weekly Cichlid CV team meeting Tuesday evening.
- Continue working with Kailey and Bree on their project.
    - Try to write a Python script that can automate the extraction of the videos from the .bag files.
    - If time allows, continue working through the process Bree has asked me to follow.
- Continue literature review.
- Attend any seminars that might be scheduled for next week.

Is anything blocking you from getting work done?

- Not at this time, no.

2. *Abstracts:*

"Exploiting Depth Information for Wildlife Monitoring", Haucke and Steinhage (2021; arXiv).

- *Abstract:* "Camera traps are a proven tool in biology and specifically biodiversity research. However, camera traps including depth estimation are not widely deployed, despite providing valuable context about the scene and facilitating the automation of previously laborious manual ecological methods. In this study, we propose an automated camera trap-based approach to detect and identify animals using depth estimation. To detect and identify individual animals, we propose a novel method D-Mask R-CNN for the so-called instance segmentation which is a deep learning-based technique to detect and delineate each distinct object of interest appearing in an image or a video clip. An experimental evaluation shows the benefit of the additional depth estimation in terms of improved average precision scores of the animal detection compared to the standard approach that relies just on the image information. This novel approach was also evaluated in terms of a proof-of-concept in a zoo scenario using an RGB-D camera trap."
- *AI Summary by ChatGPT (4o):* "The paper proposes a novel approach to wildlife monitoring using camera traps enhanced with depth estimation. It introduces D-Mask R-CNN, an extension of the popular Mask R-CNN architecture, adapted for RGB-D images. The inclusion of depth information improves animal detection and instance segmentation, facilitating better population monitoring and ecological studies. The system was evaluated using synthetic datasets and an RGB-D camera trap deployed at a zoo, showing significant improvements in accuracy compared to standard RGB-based methods."
    - *Key Contributions:*
        - "D-Mask R-CNN Architecture: D-Mask R-CNN integrates depth information alongside RGB channels to enhance instance segmentation performance. It uses separate backbones for RGB and depth features, combining them through a feature fusion mechanism, allowing the model to exploit complementary information from both sources."
        - "Depth Estimation in Camera Traps: The paper highlights the utility of depth estimation for distinguishing overlapping or occluded animals, as well as for determining precise distances, which are critical for ecological tasks like population density estimation."
        - "Synthetic and Real-World Evaluation: The authors trained and tested their model on a synthetic RGB-D dataset and an RGB-D camera trap dataset collected at a zoo. Results demonstrate substantial gains in average precision (AP) metrics for both bounding box predictions and segmentation masks compared to the standard Mask R-CNN."
        - "Proof-of-Concept Deployment: An RGB-D camera trap was constructed and deployed in a zoo, showcasing the potential for practical applications in wildlife monitoring."
    - *Contributions to Knowledge:*
        - "Integration of Depth in Wildlife Monitoring: This paper is among the first to demonstrate the benefits of RGB-D instance segmentation for wildlife

monitoring, addressing the limitations of traditional RGB-only camera traps."
    - "Adaptation of Mask R-CNN for RGB-D Data: By extending Mask R-CNN to incorporate depth information, the study introduces a modular approach that can be adapted for other tasks, such as behavior analysis or keypoint detection."
    - "Synthetic Data for Wildlife Research: The use of synthetic datasets to pre-train models for wildlife applications sets a precedent for addressing the scarcity of annotated ecological datasets."
  - *Future Research Directions:*
    - "Dataset Expansion: Future work could focus on collecting larger and more diverse RGB-D datasets, encompassing different environments, species, and lighting conditions, to improve the generalizability of the model."
    - "Stereo-Based Depth Estimation: Exploring stereo cameras with larger baselines could enhance depth accuracy for distant animals, improving monitoring in open landscapes."
    - "Integration with Ecological Models: Extending D-Mask R-CNN for tasks like presence-absence modeling or camera trap distance sampling could provide new tools for biodiversity research."
    - "Automated Behavioral Analysis: Incorporating additional capabilities, such as keypoint detection and pose estimation, could enable studies on animal behavior and interactions."
- *Link:* http://arxiv.org/abs/2102.05607.

3. *Scripts & Code Blocks:*

Extract_videos.sh

- A bash script written for my local machine that (attempts) to iterate through a set of .bag files and extract the videos they contain.
  - NOTE: this will NOT work on most operating systems, since it requires an installation of the Robot Operating System (ROS) which itself is only compatible with specific Linux and UNIX distributions.
- *Code blocks:*

```bash
#!/bin/bash

# start roscore
echo "Starting roslaunch server..."
roscore & ROSCORE_PID=$!

# define cleanup script to run at end
cleanup() {
    echo "Terminating roslaunch server..."
    kill $ROSCORE_PID
    exit
}

trap cleanup INT TERM EXIT

```

```bash
# iterate over each .bag file in the directory
TOPIC="/device_0/sensor_1/Color_0/image/data"
for BAG_FILE in "$1/todo/*.bag"; do
    # Extract the filename without extension
    FILE_NAME="${BAG_FILE%.*}"
    echo "Processing $BAG_FILE"

    # start image extraction in background
    rosrun image_view extract_images image:=$TOPIC _filename_format:="$1/done/frames/${FILE_NAME}_frame%04d.jpg" & ROSRUN_PID=$!

    # play the .bag file
    rosbag play "$BAG_FILE"

    # once done playing .bag file, terminate rosrun
    kill $ROSRUN_PID
    wait $ROSRUN_PID 2>/dev/null

    # save frames as .mp4 video with ffmpeg
    ffmpeg -framerate 15 -i "$1/done/frames/${FILE_NAME}_frame%04d.jpg" -c:v libx264 -pix_fmt yuv420p "$1/done/videos/${FILE_NAME}.mp4"

    echo "Saved $1/done/videos/${FILE_NAME}.mp4"
done

# cleanup roscore
cleanup
```

- *Status:* tested (doesn't work, will likely have to re-implement in Python if I want to automate the process more effectively).
- *Data:* Requires that ROS 1 and ffmpeg are installed in the host environment, and that there are existing .bag files containing video data already in the system.

4. *Documentation (non-PhD centered stuff only):*

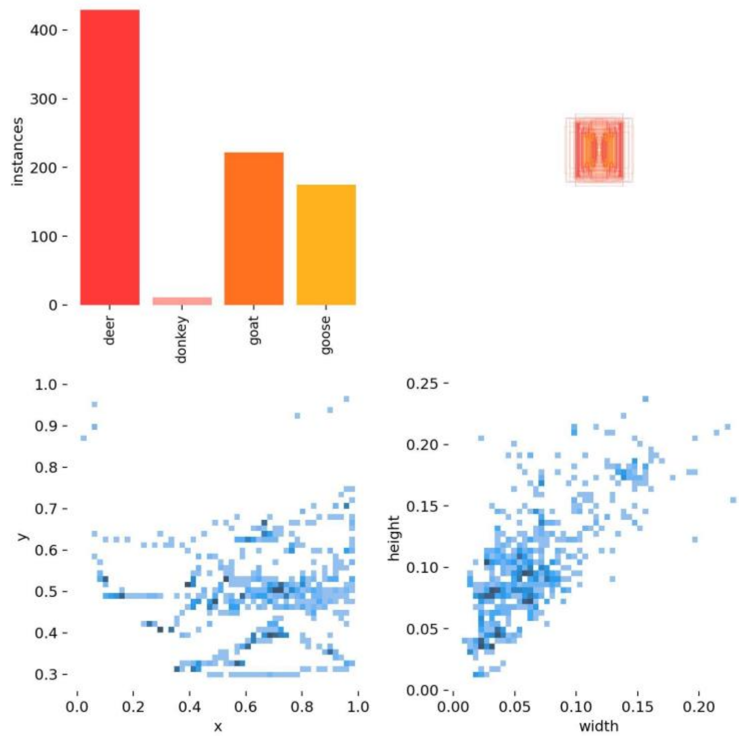
- Read the Lindenthal Camera Traps paper.
- Trained YOLOv5s on the YOLOv5 training dataset extracted last week.
- Verified the YOLOv5s model on the YOLOv5 testing dataset I extracted as well, calculating the class accuracies of the predictions.
- Attended weekly Cichlid CV meeting on Tuesday night.
- Updated the HAAG website's Meet the Team pages.
- Figured out how to extract videos from .bag files.
    - (Attempted) to write a bash script that automates the process, but it wouldn't run… will most likely need to re-write in Python instead.
- Currently attending weekly publication seminar.

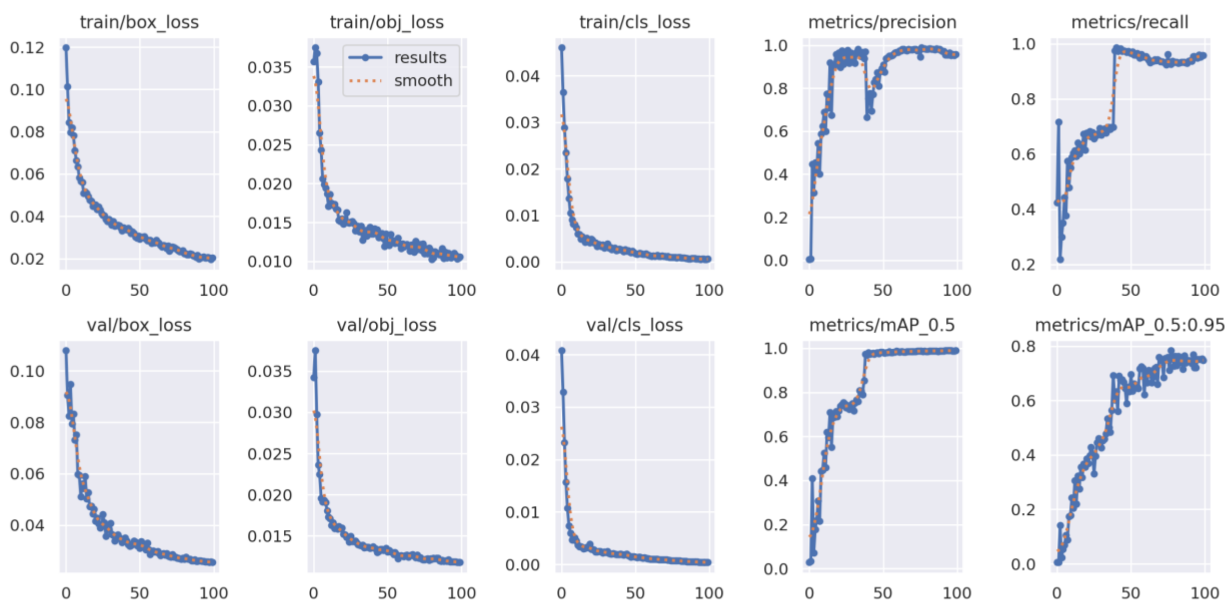
5. *Script Validation (optional):* Code written this week is tested, not functional.

6. *Results Visualization:* Please find below some automatically generated plots and figures from training YOLOv5s:

```
100 epochs completed in 0.210 hours.
Optimizer stripped from runs/train/exp/weights/last.pt, 14.4MB
Optimizer stripped from runs/train/exp/weights/best.pt, 14.4MB

Validating runs/train/exp/weights/best.pt...
Fusing layers...
Model summary: 157 layers, 7020913 parameters, 0 gradients, 15.8 GFLOPs
                 Class     Images  Instances          P          R      mAP50   mAP50-95: 100%|██████████| 3/3 [00:01<00:00,  1.63it/s]
                   all         82        204      0.985      0.925      0.986      0.783
                  deer         82         95      0.999          1      0.995      0.752
                donkey         82          1      0.957          1      0.995      0.995
                  goat         82         36      0.984      0.917      0.989      0.765
                 goose         82         72          1      0.784      0.966       0.62
Results saved to runs/train/exp
(yolov5) charlieclark@MSI:~/yolov5$
```





Please also find a link to the first video I was able to extract: [20200622220342.mp4]

7. *Proof of Work:* Find below the class accuracies from validating our trained YOLOv5s model on the test set generated last week. An ordered list of class names is also provided to allow the user to match names to class IDs in the first image.

```
Class Accuracies:
Class 0: 1.000 (95/95)
Class 1: 1.000 (1/1)
Class 2: 0.972 (35/36)
Class 3: 0.972 (70/72)
```

```
nc: 4
names: ['deer', 'donkey', 'goat', 'goose']
```

8. *Next Week's Proposal (non-PhD centered stuff only):*

- Attend weekly Cichlid CV team meeting Tuesday evening.
- Continue working with Kailey and Bree on their project.
  - Try to write a Python script that can automate the extraction of the videos from the .bag files.
  - If time allows, continue working through the process Bree has asked me to follow.
- Continue literature review.
- Attend any seminars that might be scheduled for next week.

9. *Questions:* No questions currently.

Week 14 Report

Thuan Nguyen – Cichlid Computer Vision project

Friday, November 22, 2024

Summary

What progress did you make in the last week?

- I discussed with Adam regarding the implementation of the DeepLabCut+ViT model and addressed challenges such as memory issues and replicating his results (>90% when evaluated on test triplets).
- Also tested the DeepLabCut's default transformer ReID model using triplet datasets of varying sizes but observed no significant improvement in accuracy across epochs, with test accuracy plateauing at ~55%.
- I tested DeepLabCut's own re-ID implementation against several videos. Results were similarly poor, whether with 1000, 10000 or 50000 triplets, or with more epochs, or with different videos (See details below).
- Before that, I encountered memory issues with a ViT feature extractor integrated into the DeepLabCut transformer ReID pipeline. I later resolved the memory issue and replicated Adam's DLC+ViT training experiment, achieving high accuracy results similar to Adam's runs.
- In discussion with Bree and Charlie, I analyzed and compared the suitability of PvT, ViT, and TransReID for the fish ReID task, highlighting challenges in applying these models to limited fish datasets.

What are you planning on working on next?

- I still need generate longer videos (where fish appear more often) in order to create more interesting triplet dataset on which to train and evaluate the transformer re-ID pipelines on. So far, I've evaluated these training runs on 1-minute, 2-minute, or 3-minute videos, but I want to run these models on more diverse videos.

Is anything blocking you from getting work done?

- Not at the moment.

Abstract

Deep Learning for Person Re-identification: A Survey and Outlook

Mang Ye, Jianbing Shen, Gaojie Lin, Tao Xiang, Ling Shao, Steven C. H. Hoi

+ Advances in deep learning significantly improved person Re-ID (image and video), surpassing human accuracy on standard datasets like Market-1501.

+ Part-level feature learning, attention mechanisms, and multi-loss training enhance model accuracy and robustness.

+ Issues like dataset size, complex environments, occlusion, and cross-dataset evaluations remain barriers to generalization.

+ Recent progress narrows the gap with supervised Re-ID, but limitations in attention schemes and domain adaptation persist.

+ Research focuses on handling real-world variability, efficient model deployment, dynamic updates, and minimizing reliance on manual annotations.

Work done this week – further details

- I discussed with Adam regarding the implementation of the DeepLabCut+ViT model and addressed challenges such as memory issues and replicating his results (>90% when evaluated on test triplets).
- Also tested the DeepLabCut's default transformer ReID model using triplet datasets of varying sizes but observed no significant improvement in accuracy across epochs, with test accuracy plateauing at ~55%.
- I tested DeepLabCut's own re-ID implementation against several videos. Results were similarly poor, whether with 1000, 10000 or 50000 triplets, or with more epochs, or with different videos (See details below).
- Before that, I encountered memory issues with a ViT feature extractor integrated into the DeepLabCut transformer ReID pipeline. I later resolved the memory issue and replicated Adam's DLC+ViT training experiment, achieving high accuracy results similar to Adam's runs.
- In discussion with Bree and Charlie, I analyzed and compared the suitability of PvT, ViT, and TransReID for the fish ReID task, highlighting challenges in applying these models to limited fish datasets.

More details on training re-ID model:

DeepLabCut's default transformer re-ID, results plateauing at 55-60%, regardless of triplet dataset sizes, epochs, videos

n_triplets of 1000

Training transformer re-identification model...

Epoch 10, train acc: 0.64

Epoch 10, test acc 0.55

Epoch 20, train acc: 0.64

Epoch 20, test acc 0.55

Epoch 30, train acc: 0.64

Epoch 30, test acc 0.55

Epoch 40, train acc: 0.65

Epoch 40, test acc 0.55

Epoch 50, train acc: 0.64

Epoch 50, test acc 0.54

Epoch 60, train acc: 0.64

Epoch 60, test acc 0.54

Epoch 70, train acc: 0.65

Epoch 70, test acc 0.54

Epoch 80, train acc: 0.65

Epoch 80, test acc 0.54

Epoch 90, train acc: 0.65

Epoch 90, test acc 0.54

Epoch 100, train acc: 0.65

Epoch 100, test acc 0.54

N_triplets of 10000

Training transformer re-identification model...

Epoch 10, train acc: 0.64

Epoch 10, test acc 0.55

Epoch 20, train acc: 0.62

Epoch 20, test acc 0.55

Epoch 30, train acc: 0.63

Epoch 30, test acc 0.55

Epoch 40, train acc: 0.64

Epoch 40, test acc 0.55

Epoch 50, train acc: 0.64

Epoch 50, test acc 0.55

Epoch 60, train acc: 0.64

Epoch 60, test acc 0.55

Epoch 70, train acc: 0.65

Epoch 70, test acc 0.54

Epoch 80, train acc: 0.66

Epoch 80, test acc 0.54

Epoch 90, train acc: 0.64

Epoch 90, test acc 0.54

Epoch 100, train acc: 0.64

Epoch 100, test acc 0.54

N_triplets of 50000

Training transformer re-identification model...

Epoch 10, train acc: 0.61

Epoch 10, test acc 0.55

Epoch 20, train acc: 0.64

Epoch 20, test acc 0.55

Epoch 30, train acc: 0.64

Epoch 30, test acc 0.55

Epoch 40, train acc: 0.65

Epoch 40, test acc 0.54

Epoch 50, train acc: 0.62

Epoch 50, test acc 0.55

Epoch 60, train acc: 0.65

Epoch 60, test acc 0.54

Epoch 70, train acc: 0.67

Epoch 70, test acc 0.53

Epoch 80, train acc: 0.65

Epoch 80, test acc 0.53

Epoch 90, train acc: 0.65

Epoch 90, test acc 0.53

Epoch 100, train acc: 0.64

Epoch 100, test acc 0.53

Test video 4

Training transformer re-identification model...

Epoch 10, train acc: 0.49

Epoch 10, test acc 0.26

Epoch 20, train acc: 0.49

Epoch 20, test acc 0.26

Epoch 30, train acc: 0.51

Epoch 30, test acc 0.26

Epoch 40, train acc: 0.49

Epoch 40, test acc 0.26

Epoch 50, train acc: 0.49

Epoch 50, test acc 0.26

Epoch 60, train acc: 0.49

Epoch 60, test acc 0.26

Epoch 70, train acc: 0.50

Epoch 70, test acc 0.26

Epoch 80, train acc: 0.52

Epoch 80, test acc 0.26

Epoch 90, train acc: 0.48

Epoch 90, test acc 0.26

Epoch 100, train acc: 0.50

Epoch 100, test acc 0.26

Epoch 110, train acc: 0.50

Epoch 110, test acc 0.26

Epoch 120, train acc: 0.51

Epoch 120, test acc 0.25

Epoch 130, train acc: 0.52

Epoch 130, test acc 0.26

Epoch 140, train acc: 0.51

Epoch 140, test acc 0.26

Epoch 150, train acc: 0.51

Epoch 150, test acc 0.26

Epoch 160, train acc: 0.51

Epoch 160, test acc 0.26

Epoch 170, train acc: 0.50

Epoch 170, test acc 0.26

Epoch 180, train acc: 0.50

Epoch 180, test acc 0.26

Epoch 190, train acc: 0.51

Epoch 190, test acc 0.26

Epoch 200, train acc: 0.52

Epoch 200, test acc 0.26

Epoch 210, train acc: 0.51

Epoch 210, test acc 0.26

Epoch 220, train acc: 0.53

Epoch 220, test acc 0.26

Epoch 230, train acc: 0.53

Epoch 230, test acc 0.26

Epoch 240, train acc: 0.50

Epoch 240, test acc 0.26

Epoch 250, train acc: 0.52

Epoch 250, test acc 0.26

Adam's DeepLabCut + ViT model: higher accuracies

<u>For one 30-second video:</u>

Epoch 10, train acc: 0.99

Epoch 10, test acc 0.98

Epoch 20, train acc: 0.96

Epoch 20, test acc 0.86

Epoch 30, train acc: 0.99

Epoch 30, test acc 1.00

Epoch 40, train acc: 1.00

Epoch 40, test acc 0.94

<u>Second video (2 minutes):</u>

Epoch 10, train acc: 0.95

Epoch 10, test acc 0.94

Epoch 20, train acc: 0.98

Epoch 20, test acc 0.96

Epoch 30, train acc: 0.99

Epoch 30, test acc 0.96

Epoch 40, train acc: 0.98

Epoch 40, test acc 0.98

Scripts - Documentation - Script Validation - Results Visualization - Proof of Work

For further details, please refer to the details above.

Next week's proposal

- I still need generate longer videos (where fish appear more often) in order to create more interesting triplet dataset on which to train and evaluate the transformer re-ID pipelines on. So far, I've evaluated these training runs on 1-minute, 2-minute, or 3-minute videos, but I want to run these models on more diverse videos.