Cichlid Computer Vision Project – Weekly Progress

Week ending Friday, November 29, 2024

Summary (please see all detailed attachments below):

Kailey Quesada

1.1 What progress did you make in the last week?

(1) For the Lizard project, I looked into additional tools to compare to B-SOiD. After doing some research,

I decided to go with VAME and started researching VAME.

(2) I installed VAME and tested it on 18 pose estimation files and videos from DLC. I was able to get some

basic preliminary results in terms of training the model and finding motifs. I also spent time on trying to

get VAME's gif function to work, but additional work would be needed to see if we could get that to run.

(3) I wrote a couple required auxiliary scripts for VAME. These were for data preparation and cleaning. I

had to troubleshoot the run instructions in the VAME documentation, so I created an updated run script

as I figured out where the documentation was wrong.

(4) I continued to work on the report for Dr. Stroud and add scripts to the GitHub.

(5) For meetings, I attended the Cichlid CV team meeting, a working meeting for the Cichlid CV team

paper outline, and the publication seminar on November 22nd.

1.2 What are you planning on working on next?

(1) I need to finish looking through the motifs created by VAME. This will require a script for grouping the

motifs together for easier analysis.

(2) I need to write up a user guide for VAME and add it to the GitHub.

(3) I need to run more video examples in B-SOiD.

(4) I need to answer the questions Bree left about the report in the work document.

(5) I need to finish up my writing in the Dr. Stroud report.

(6) I need write my slides for the Dr. Stroud report.

(7) I need write my slides for the final HAAG presentation and practice them.

1.3 Is anything blocking you from getting work done?

None.

<u>Charles Clark</u>

What progress did you make in the last week?

• Wrote a Python script to automate the extraction of video data from .bag files.

o Downloaded the full Lindenthal Camera Traps dataset onto my local machine and tested the script on .bag files with both color and infrared video data.

▪ For .bag files containing color video data, the script works perfectly.

▪ However, for .bag files containing infrared video data, the script runs into an exception… more on this on first page.

• Attended weekly Cichlid CV team meeting on Tuesday

o Presented about our work from end of last week and beginning of this week.

o Discussed Thuan's findings that he had relayed to Bree and me on Saturday.

o Wrote up an initial (tentative) publication outline and picked ICCV as our (tentative) goal conference.

• Continued literature review.

What are you planning on working on next?

• Attend publication seminar this Friday (if there is one).

• Attend weekly Cichlid CV team meeting on Tuesday.

• Attend any additional seminars scheduled for next week.

• Start working on my final report for this semester.

• Hopefully connect with Bree to discuss Thuan's findings, as well how I should proceed with BioBoost work.

Is anything blocking you from getting work done?

- The Python script I wrote to extract videos from the .bag files works perfectly with color video data, but it errors-out when trying to extract infrared video data.
  - This is because of how the original authors encoded the infrared video data.
  - Specifically, they encoded the infrared video data using the 8UC1 format.
    - After some research into the issue, I found that trying to convert 8UC1-encoded data into a grayscale schema like mono8 is like trying to downcast the data, meaning it could lead to information loss.
    - Because of this, the rosbag Python module simply refused to perform the conversion, even though it's likely that the data is in-fact grayscale.
    - I have been unable to find a way to force the conversion, despite countless other people having this same issue, as evidenced by forum posts describing similar issues.
  - Given all the trouble we have had with this dataset, should we consider moving onto to finding a different, more accessible dataset?

Week 15 Document Submission

Kailey Quesada (Cichlid CV & Lizard CV Team)

November 29, 2024

1 Weekly Project Updates

1.1 What progress did you make in the last week?

(1) For the Lizard project, I looked into additional tools to compare to B-SOiD. After doing some research, I decided to go with VAME and started researching VAME.
(2) I installed VAME and tested it on 18 pose estimation files and videos from DLC. I was able to get some basic preliminary results in terms of training the model and finding motifs. I also spent time on trying to get VAME's gif function to work, but additional work would be needed to see if we could get that to run.
(3) I wrote a couple required auxiliary scripts for VAME. These were for data preparation and cleaning. I had to troubleshoot the run instructions in the VAME documentation, so I created an updated run script as I figured out where the documentation was wrong.
(4) I continued to work on the report for Dr. Stroud and add scripts to the GitHub.
(5) For meetings, I attended the Cichlid CV team meeting, a working meeting for the Cichlid CV team paper outline, and the publication seminar on November 22nd.

1.2 What are you planning on working on next?

(1) I need to finish looking through the motifs created by VAME. This will require a script for grouping the motifs together for easier analysis.
(2) I need to write up a user guide for VAME and add it to the GitHub.
(3) I need to run more video examples in B-SOiD.
(4) I need to answer the questions Bree left about the report in the work document.
(5) I need to finish up my writing in the Dr. Stroud report.
(6) I need write my slides for the Dr. Stroud report.
(7) I need write my slides for the final HAAG presentation and practice them.

1.3 Is anything blocking you from getting work done?

None.

2 Document Submission

2.1 Abstract

Luxem, K., et al. "Identifying behavioral structure from deep variational embeddings of animal motion." Communications Biology 2022. `https://www.nature.com/articles/s42003-022-04080-7`.

Quantifying and understanding how animal behaviors are organized is a significant challenge in neuroscience. Recent technology allows researchers to track animal movements without markers, revealing complex patterns in their behavior over time. The study introduces a new deep learning method called VAME, which can automatically identify and categorize these behaviors into structured groups, helping researchers detect subtle differences in behavior that might be missed by human observation. VAME, or Variational Animal Motion Embedding, is a model designed to analyze the movement of mice in an open-field arena by tracking their body parts using a camera. The model processes the recorded movements to identify patterns, called behavioral motifs, which are repeated units of movement. By using advanced neural networks, VAME can learn and represent the complex dynamics of the mice's behavior in a simpler form. No future work is mentioned.

2.2 Scripts and Code Blocks

This week's scripts can be found here: https://github.com/Human-Augment-Analytics/Lizard-Pose-Estimation-and-Evaluation/tree/main/Behavioral%20Analysis/VAME.

2.2.1 Lizard Project: VAME

VAME requires that pose estimation files be in csv. Therefore, if you have your pose estimation files in h5, you must convert to csv. The code for that is in Listing 1.

```
# Convert H5 Files to CSV
def convert_h5_to_csv(directory_path):
    h5_files = glob.glob(os.path.join(directory_path, ’*.h5’))

    for h5_file in h5_files:
        base_name = os.path.splitext(os.path.basename(h5_file))[0]
        csv_file = os.path.join(directory_path, f’{base_name}.csv’)

        try:
            data = pd.read_hdf(h5_file)
            data.to_csv(csv_file, index=False)
            print(f"Converted {h5_file} to {csv_file}")
        except Exception as e:
            print(f"Error converting {h5_file}: {e}")

# Call Function
directory_path = ’C:/Users/Username/OneDrive/Documents/VAME/data’
convert_h5_to_csv(directory_path)
```

Listing 1: Convert h5 pose estimation files to csv.

Next, the video names must align with the csv file names. In the lizard dataset, they do not. Therefore, you can use the code in Listing 2 to make the names match.

```
# Working Path
working_directory = Path("C:/Users/Username/OneDrive/Documents/VAME/data/")

# Remove Strings that Cause Mismatch of CSV and MP4 Files
for file_path in working_directory.glob("*.mp4"):
    if "_p60_labeled" in file_path.name:
        new_name = file_path.name.replace("_p60_labeled", "")
        new_file_path = file_path.with_name(new_name)
        file_path.rename(new_file_path)
```

Listing 2: Clean file names to allow mp4 and csv files to match.

Finally, you can run your VAME script. There is very little documentation for VAME, and what does exist is missing required parameters, etc. See Listing 3 for the final script that worked for me. Note that the gif function doesn't work right at this time.

```
...

# Go Through Working Directory and Add All Files
for file_path in working_directory_path.iterdir():
    if file_path.is_file():
        if file_path.suffix == ’.mp4’:
            videos.append("C:/Users/Username/Path/To/Files/VAME/data/"+ file_path.name)
        elif file_path.suffix == ’.csv’:
            poses_estimations.append("C:/Users/Username/Path/To/Files/VAME/data/" +
    file_path.name)

# Initialize Project
config = vame.init_new_project(
    project = project,
    videos = videos,
    poses_estimations=poses_estimations,
    working_directory=working_directory_path,
    videotype='.mp4'
)

# Run an Existing Project
# config = 'C:/Users/Username/Path/To/Files/VAME/data/lizard project-Day-2024/config.yaml'

# Transform DLC CSVs to Numpy Arrays
vame.egocentric_alignment(config, pose_ref_index=[0,5])

# Create Training Dataset for Videos with 6 Keypoints
vame.create_trainset(config, pose_ref_index=[0,5])

# Train the Model
vame.train_model(config)

# Evaluate the Model
vame.evaluate_model(config)

# Pose Segmentation
vame.pose_segmentation(config)

# Create Motif Videos
vame.motif_videos(config, videoType='.mp4', parametrization='hmm')

# Run Community Detection
vame.community(config, parametrization='hmm', cut_tree=None, cohort=True)

# UMAP Visualization
fig = vame.visualization(config, label=None, parametrization='hmm')

# Generative Reconstruction Decoder
vame.generative_model(config, mode="centers", parametrization='kmeans')

# Output Video
vame.gif(config, pose_ref_index=[0,4], subtract_background=False, start=None,
    parametrization='hmm',
         length=100, max_lag=30, label='motif', file_format='.mp4', crop_size=(200,200))
```

Listing 3: Run VAME.

2.3 Documentation

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

2.3.1 What VAME Does

VAME (Video-based Animal Motion Estimation) finds patterns in animal movement with a focus on finding repetitive behaviors. VAME performs behavior segmentation to identify behavioral motifs. VAME uses K-means and HMM for cluster detection. It uses a variational RNN autoencoder framework for learning the signal and embedding it into a lower dimensional space.

2.3.2 Hardware Requirements

The creators of VAME used a single NVIDIA GeForce GTX 1080 Ti for their paper. For this demonstration, that exact GPU was used. However, other GPUs will also work, with varying results. If the user does not have a local GPU, a compute cluster like PACE can be used instead.

2.3.3 VAME Setup

It is important to note that there are two VAME repositories. Ensure that you are using the newer VAME repository (`https://github.com/EthoML/VAME.git`), as the old repository is no longer maintained and uses deprecated packages. Please see the following steps to setup VAME:

1. Open your anaconda prompt.
2. Use "git clone https://github.com/EthoML/VAME.git" in your file location of choice.
3. Go into the repository with "cd VAME".
4. Create a VAME environment with "conda env create -f VAME.yaml".
5. Activate your new environment with "conda activate VAME".
6. Contrary to the GitHub documentation, you can install using "pip install vame-py". There is no setup.py to install from in the newer VAME repository. If there is a setup.py, you are probably using the wrong repository.
7. Ensure you have the required PyTorch version with "python -c "import torch; print(torch.__version__)". Note that it appears that VAME auto-installs the CPU version of PyTorch. This would mean that, for improved performance, you might want to update this to GPU.
8. Ensure that your data is egocentrically aligned. Since the lizard dataset is already egocentrically aligned, we don't have to change anything.
9. Follow the run instructions here: `https://ethoml.github.io/VAME/docs/getting_started/running/`. Note that these instructions will cause lots of errors, requiring troubleshooting.

2.3.4 Command Line Usage

VAME is available as a command line tool only. Luckily, it is pretty easy to run. Follow these instructions to run your first model:

1. Create your training-script.py. You can use this by following the documentation (`https://ethoml.github.io/VAME/docs/getting_started/running/`) or by using the script provided in the lizard pose estimation repository available in the HAAG organization GitHub. Personally, I wrote code to read the file location and automatically add files to the lists for processing. Alternatively, one can manually list the files they will be using to train with.
2. Create a script for H5 to CSV conversion if necessary. One is provided in the lizard pose estimation repository available in the HAAG organization GitHub.
3. Create a script for cleaning file names. VAME expects the CSV and MP4 files to have the same name, so you will need to remove the "_p60_labeled" part from the MP4 names. A script is available in the HAAG organization GitHub.
4. Activate your environment with "conda activate VAME".
5. Go to the location of your script(s) with "cd path_to_your_script_directory".
6. If necessary, convert to CSV with "python 'H5 to CSV Conversion (Required).py'". You may need to "conda install pytables". Ensure the CSVs have been created.
7. Run your script for cleaning file names or do so manually.
8. Run VAME with "python Run_VAME.py".
9. VAME will create a folder inside the video folder. Delete it if you have errors and need to re-run.

There is very little VAME documentation online. Additionally, most of the VAME documentation that is available is incorrect and deprecated. This means that it is sometimes necessary to look through the source code to figure out what is going on. A few notable differences between documentation and usage are that many of the functions require a parametrization parameter that is not listed as required in the documentation. Additionally, certain parameter types only work with other parameter types, and this can be unclear on the first run. Finally, time was spent on trying to use the gif function provided by the repository, but it wouldn't correctly crop the video if the default video was not used. Additional perusal of the source code could potentially fix this problem.

2.3.5 Cluster Number Selection & Interpreting Results

The number of clusters appeared to be hard-coded to 15 for both k-means and HMM. I think that its possible to do a different number, but I would have to look through the source code. The results can be interpreted by looking through the motif videos created by VAME. I will be writing a script this week to more easily go through these. Preliminarily, it looks like motif 1 picks up on the lizard running and being "chased" by the researcher's hand, and motif 2 looks like the lizard pausing at the top of the frame. I will be looking into this more this week.

2.3.6 Model Improvement and Performance Evaluation

The future reconstruction and loss curves are produced by VAME, as shown in Figure 1. I will be looking into this more this week.

2.3.7 Further Usage

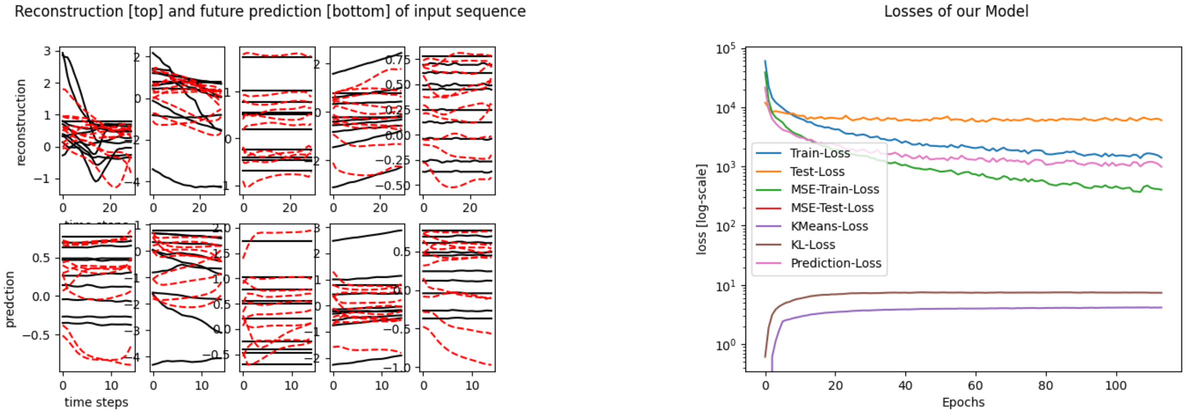
VAME does a lot of the same things as B-SOiD, but without a GUI available. It also has less documentation than B-SOiD, which is unfortunate. After evaluating the motifs, I will be able to give a better recommendation for how useful this tool could be in the future.

2.4 Script Validation (Optional)

No additional script validation information needed this week.

2.5 Results Visualization

To visualize the model evaluation, please see Figure 1.



(a) Future Reconstruction

(b) MSE and KL Loss

Figure 1: Model Evaluation

2.6 Proof of Work

I need to look into the future reconstruction and loss plots to get a better idea of how well the model is performing. It is unclear to me at first glance. I also suspect that VAME will need more data than B-SOiD did, so more dataset may be necessary. Additionally, while the first two motifs seem to be picking up on consistent behaviors, the others do not. I will need to look into that this week.

2.7 Next Week's Proposal

As stated on the first page, this week's goals are the following:
(1) I need to finish looking through the motifs created by VAME. This will require a script for grouping the motifs together for easier analysis.
(2) I need to write up a user guide for VAME and add it to the GitHub.
(3) I need to run more video examples in B-SOiD.
(4) I need to answer the questions Bree left about the report in the work document.
(5) I need to finish up my writing in the Dr. Stroud report.
(6) I need write my slides for the Dr. Stroud report.
(7) I need write my slides for the final HAAG presentation and practice them.

Charles R. Clark
CS 6999 – HAAG: Cichlid CV
Fall 2024
November 27, 2024

Week 15 Report

1. *Time-log Response:*

What progress did you make in the last week?

- Wrote a Python script to automate the extraction of video data from .bag files.
    - Downloaded the full Lindenthal Camera Traps dataset onto my local machine and tested the script on .bag files with both color and infrared video data.
        - For .bag files containing color video data, the script works perfectly.
        - However, for .bag files containing infrared video data, the script runs into an exception… more on this on first page.
- Attended weekly Cichlid CV team meeting on Tuesday
    - Presented about our work from end of last week and beginning of this week.
    - Discussed Thuan's findings that he had relayed to Bree and me on Saturday.
    - Wrote up an initial (tentative) publication outline and picked ICCV as our (tentative) goal conference.
- Continued literature review.

What are you planning on working on next?

- Attend publication seminar this Friday (if there is one).
- Attend weekly Cichlid CV team meeting on Tuesday.
- Attend any additional seminars scheduled for next week.
- Start working on my final report for this semester.
- Hopefully connect with Bree to discuss Thuan's findings, as well how I should proceed with BioBoost work.

Is anything blocking you from getting work done?

- The Python script I wrote to extract videos from the .bag files works perfectly with color video data, but it errors-out when trying to extract infrared video data.
    - This is because of how the original authors encoded the infrared video data.
    - Specifically, they encoded the infrared video data using the 8UC1 format.
        - After some research into the issue, I found that trying to convert 8UC1-encoded data into a grayscale schema like mono8 is like trying to downcast the data, meaning it could lead to information loss.
        - Because of this, the rosbag Python module simply refused to perform the conversion, even though it's likely that the data is in-fact grayscale.
        - I have been unable to find a way to force the conversion, despite countless other people having this same issue, as evidenced by forum posts describing similar issues.
    - Given all the trouble we have had with this dataset, should we consider moving onto to finding a different, more accessible dataset?

2. *Abstracts:*

"ABD-Net: Attentive but Diverse Person Re-Identification", Chen et al. (2019; 2019 IEEE/CVF International Conference on Computer Vision (ICCV)).

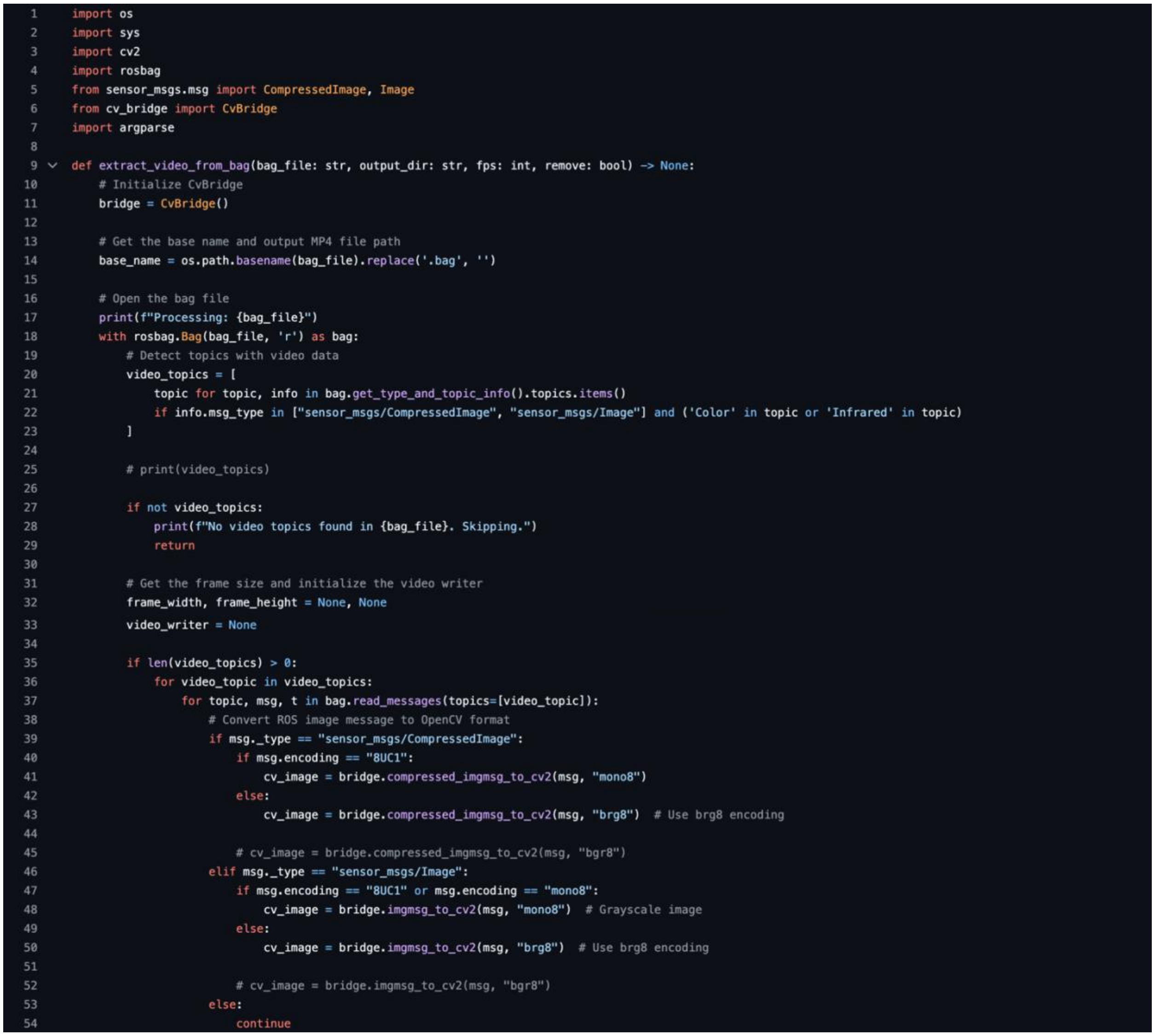
- *Abstract:* "Attention mechanisms have been found effective for person re-identification (Re-ID). However, the learned "attentive" features are often not naturally uncorrelated or "diverse", which compromises the retrieval performance based on the Euclidean distance. We advocate the complementary powers of attention and diversity for Re-ID, by proposing an Attentive but Diverse Network (ABD-Net). ABD-Net seamlessly integrates attention modules and diversity regularizations throughout the entire network to learn features that are representative, robust, and more discriminative. Specifically, we introduce a pair of complementary attention modules, focusing on channel aggregation and position awareness, respectively. Then, we plug in a novel orthogonality constraint that efficiently enforces diversity on both hidden activations and weights. Through an extensive set of ablation study, we verify that the attentive and diverse terms each contributes to the performance boosts of ABD-Net. It consistently outperforms existing state-of-the-art methods on there popular person Re-ID benchmarks."
- *AI Summary by ChatGPT (4o):* "This paper introduces ABD-Net (Attentive but Diverse Network) for person re-identification (ReID), a challenging task in computer vision aimed at identifying individuals across different camera views. ABD-Net combines attention mechanisms with a novel diversity regularization to create robust, discriminative feature embeddings. Specifically, it integrates two attention modules—Channel Attention Module (CAM) and Position Attention Module (PAM)—and employs spectral value difference orthogonality (SVDO) to enforce diversity in learned features. The network achieves state-of-the-art performance on several ReID benchmarks, including Market-1501, DukeMTMC-ReID, and MSMT17."
    - *Key Contributions:*
        - "Compound Attention Mechanism: CAM aggregates semantic information across channels, enhancing feature robustness by emphasizing meaningful channels; PAM captures spatial relationships by focusing on positional features in the image, helping the model focus on relevant regions like body parts."
        - "Diversity Regularization (SVDO): Introduces SVDO, which applies orthogonality constraints to both feature activations and weights to reduce feature redundancy, ensuring diverse feature representations. This ensures the network learns uncorrelated and diverse features, enhancing retrieval performance based on Euclidean distance."
        - "State-of-the-Art Performance: Extensive experiments show that ABD-Net surpasses previous ReID methods on Market-1501, DukeMTMC-ReID, and MSMT17 datasets. It achieves high accuracy in top-1 retrieval and mean Average Precision (mAP) metrics, setting new performance benchmarks."
    - *Contributions to Knowledge:*
        - "Combination of Attention and Diversity: ABD-Net is one of the first ReID models to systematically combine attention mechanisms with

        diversity regularization, addressing the issue of feature over-correlation in attention-based models."
        - "Spectral Value Difference Orthogonality (SVDO): The paper introduces a novel orthogonality regularizer that efficiently enforces diversity in feature embeddings, which improves generalization and reduces overfitting."
        - "Application to Real-World ReID Scenarios The model demonstrates superior robustness to occlusion, misalignment, and viewpoint variation, making it highly applicable to practical surveillance and security tasks."
    - *Future Research Directions:*
        - "Generalization to Other Vision Tasks: The authors suggest extending the ABD-Net framework to other computer vision tasks such as object detection or semantic segmentation, where attention and diversity could similarly improve performance."
        - "Efficient Orthogonality Constraints: Future research could focus on developing more computationally efficient orthogonality regularizers to further reduce training time and resource consumption."
        - "Exploration of Multi-Modal ReID: Integrating ABD-Net with additional data modalities like depth, thermal imaging, or temporal data could enhance its performance in more complex ReID scenarios."
        - "Fine-Grained ReID Applications: Applying ABD-Net to other fine-grained identification tasks, such as wildlife re-identification or vehicle re-identification, could broaden its impact beyond person ReID."
- *Link:* https://ieeexplore.ieee.org/document/9008377/.

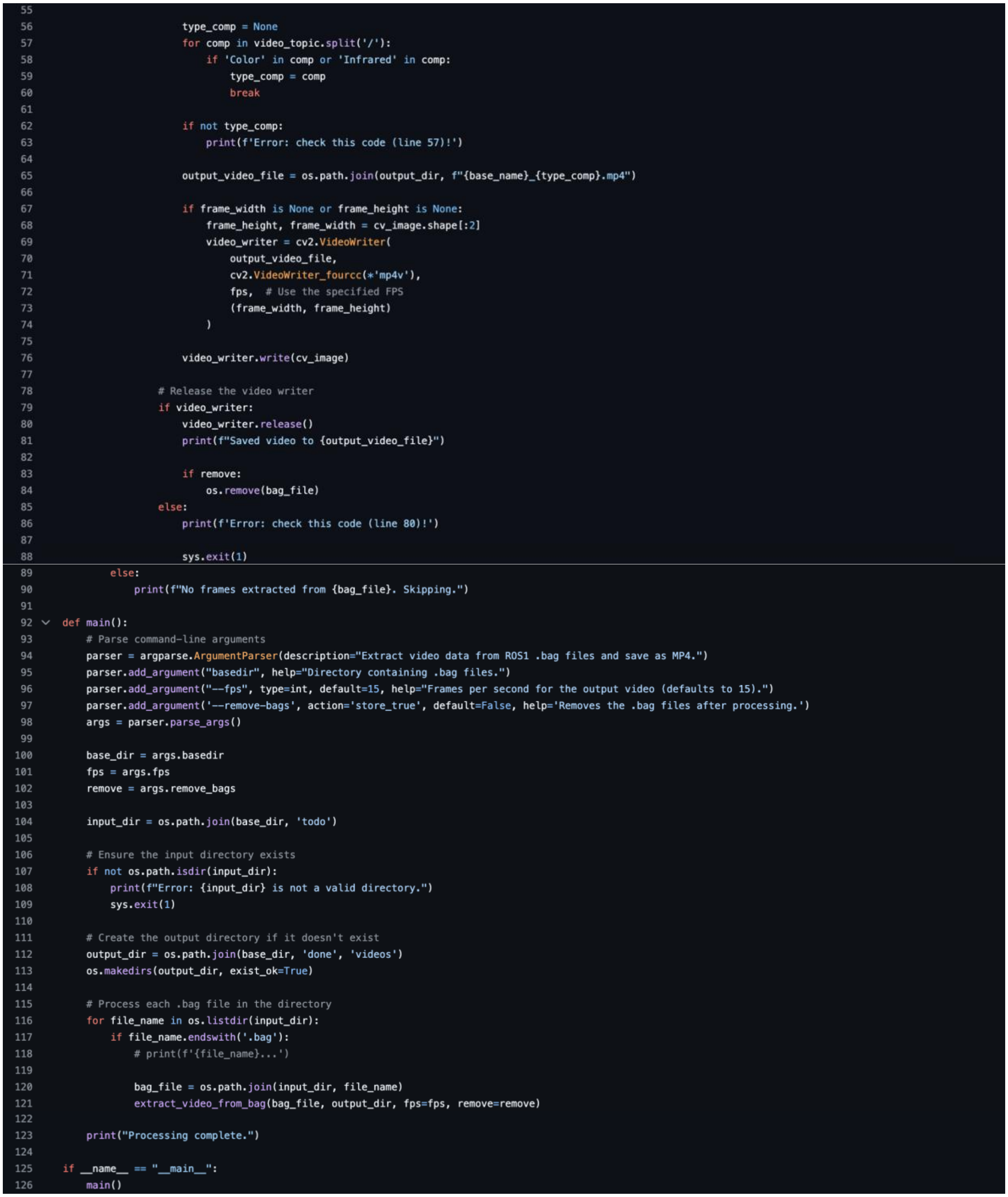
3. *Scripts & Code Blocks:*

extract_videos.py

- A Python script that implements the extraction of video data from a set of .bag files, using the official rosbag Python module.
- *Code blocks (next pages):*



```python
import os
import sys
import cv2
import rosbag
from sensor_msgs.msg import CompressedImage, Image
from cv_bridge import CvBridge
import argparse

def extract_video_from_bag(bag_file: str, output_dir: str, fps: int, remove: bool) -> None:
    # Initialize CvBridge
    bridge = CvBridge()

    # Get the base name and output MP4 file path
    base_name = os.path.basename(bag_file).replace('.bag', '')

    # Open the bag file
    print(f"Processing: {bag_file}")
    with rosbag.Bag(bag_file, 'r') as bag:
        # Detect topics with video data
        video_topics = [
            topic for topic, info in bag.get_type_and_topic_info().topics.items()
            if info.msg_type in ["sensor_msgs/CompressedImage", "sensor_msgs/Image"] and ('Color' in topic or 'Infrared' in topic)
        ]

        # print(video_topics)

        if not video_topics:
            print(f"No video topics found in {bag_file}. Skipping.")
            return

        # Get the frame size and initialize the video writer
        frame_width, frame_height = None, None
        video_writer = None

        if len(video_topics) > 0:
            for video_topic in video_topics:
                for topic, msg, t in bag.read_messages(topics=[video_topic]):
                    # Convert ROS image message to OpenCV format
                    if msg._type == "sensor_msgs/CompressedImage":
                        if msg.encoding == "8UC1":
                            cv_image = bridge.compressed_imgmsg_to_cv2(msg, "mono8")
                        else:
                            cv_image = bridge.compressed_imgmsg_to_cv2(msg, "brg8")  # Use brg8 encoding

                        # cv_image = bridge.compressed_imgmsg_to_cv2(msg, "bgr8")
                    elif msg._type == "sensor_msgs/Image":
                        if msg.encoding == "8UC1" or msg.encoding == "mono8":
                            cv_image = bridge.imgmsg_to_cv2(msg, "mono8")  # Grayscale image
                        else:
                            cv_image = bridge.imgmsg_to_cv2(msg, "brg8")  # Use brg8 encoding

                        # cv_image = bridge.imgmsg_to_cv2(msg, "bgr8")
                    else:
                        continue
```



```python

                    type_comp = None
                    for comp in video_topic.split('/'):
                        if 'Color' in comp or 'Infrared' in comp:
                            type_comp = comp
                            break

                    if not type_comp:
                        print(f'Error: check this code (line 57)!')

                    output_video_file = os.path.join(output_dir, f"{base_name}_{type_comp}.mp4")

                    if frame_width is None or frame_height is None:
                        frame_height, frame_width = cv_image.shape[:2]
                        video_writer = cv2.VideoWriter(
                            output_video_file,
                            cv2.VideoWriter_fourcc(*'mp4v'),
                            fps,  # Use the specified FPS
                            (frame_width, frame_height)
                        )

                    video_writer.write(cv_image)

                # Release the video writer
                if video_writer:
                    video_writer.release()
                    print(f"Saved video to {output_video_file}")

                    if remove:
                        os.remove(bag_file)
                else:
                    print(f'Error: check this code (line 80)!')

                    sys.exit(1)
        else:
            print(f"No frames extracted from {bag_file}. Skipping.")

def main():
    # Parse command-line arguments
    parser = argparse.ArgumentParser(description="Extract video data from ROS1 .bag files and save as MP4.")
    parser.add_argument("basedir", help="Directory containing .bag files.")
    parser.add_argument("--fps", type=int, default=15, help="Frames per second for the output video (defaults to 15).")
    parser.add_argument('--remove-bags', action='store_true', default=False, help='Removes the .bag files after processing.')
    args = parser.parse_args()

    base_dir = args.basedir
    fps = args.fps
    remove = args.remove_bags

    input_dir = os.path.join(base_dir, 'todo')

    # Ensure the input directory exists
    if not os.path.isdir(input_dir):
        print(f"Error: {input_dir} is not a valid directory.")
        sys.exit(1)

    # Create the output directory if it doesn't exist
    output_dir = os.path.join(base_dir, 'done', 'videos')
    os.makedirs(output_dir, exist_ok=True)

    # Process each .bag file in the directory
    for file_name in os.listdir(input_dir):
        if file_name.endswith('.bag'):
            # print(f'{file_name}...')

            bag_file = os.path.join(input_dir, file_name)
            extract_video_from_bag(bag_file, output_dir, fps=fps, remove=remove)

    print("Processing complete.")

if __name__ == "__main__":
    main()
```

- *Status:* tested.
- *Data:* Requires a set of ROS1 .bag files stored in a specific file schema, as well as access to a ROS 1 distribution and the official rosbag Python module.

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

- Wrote a Python script to automate the extraction of video data from .bag files.
    - Downloaded the full Lindenthal Camera Traps dataset onto my local machine and tested the script on .bag files with both color and infrared video data.
        - For .bag files containing color video data, the script works perfectly.
        - However, for .bag files containing infrared video data, the script runs into an exception… more on this on first page.
- Attended weekly Cichlid CV team meeting on Tuesday
    - Presented about our work from end of last week and beginning of this week.
    - Discussed Thuan's findings that he had relayed to Bree and me on Saturday.
    - Wrote up an initial (tentative) publication outline and picked ICCV as our (tentative) goal conference.
- Continued literature review.

5. *Script Validation (optional):* Code written this week is tested, partially functional.

6. *Results Visualization:* Please find here a link to a video that was extracted from a .bag file using the aforementioned Python script: 20200623061836.mp4

7. *Proof of Work:* For proof of work this week, I defer to the video for which a link was provided in the previous section.

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

- Attend publication seminar this Friday (if there is one).
- Attend weekly Cichlid CV team meeting on Tuesday.
- Attend any additional seminars scheduled for next week.
- Start working on my final report for this semester.
- Hopefully connect with Bree to discuss Thuan's findings, as well how I should proceed with BioBoost work.

9. *Questions:* Given the difficulty with the Lindenthal Camera Traps dataset, should we just cut our losses and move onto finding a different dataset?