# Week 11 Document Submission

Jacob Dallaire

November 1, 2024

## 1. Paper

Weitschek, E., Fiscon, G. & Felici, G. Supervised DNA Barcodes species classification: analysis, comparisons and results. *BioData Mining* **7**, 4 (2014). https://doi.org/10.1186/1756-0381-7-4

**SUMMARY**

In 2003, Hebert et al. introduced DNA Barcoding as a molecular technique for species identification using specific DNA fragments from mitochondrial, nuclear, and plastid genomes, with designated regions such as COI for animals and rbcL for plants. This method addresses the limitations of traditional morphological identification, particularly in complex cases, by enabling accurate species classification even from damaged or immature specimens through short, easily obtainable DNA sequences. The International Barcode of Life project has since promoted DNA Barcoding as a global standard, facilitating the development of various computational approaches, including supervised machine learning algorithms, to enhance the accuracy and efficiency of species classification based on DNA sequences.

Not particularly relevant to vision classification but I am interested in genetics.

## 2. Scripts

Short little script to apply a standard to annotations with point 1 being upper left of the box and point 2 the bottom right of the box.

Additional script change not shown to convert pre-processing to crop images to bounding boxes before training.

```
# Loop through each file in the folder
for filename in os.listdir(folder_path):
    if filename.endswith(".json"):
        file_path = os.path.join(folder_path, filename)

        # Load JSON data
        with open(file_path, 'r') as f:
            data = json.load(f)

        # Process each shape to adjust points for top-left and bottom-right
        for shape in data.get("shapes", []):
            # Extract points
            x1, y1 = shape["points"][0]
            x2, y2 = shape["points"][1]
```

```
    # Calculate top-left and bottom-right
    top_left_x = min(x1, x2)
    top_left_y = min(y1, y2)
    bottom_right_x = max(x1, x2)
    bottom_right_y = max(y1, y2)

    # Update points to be top-left and bottom-right
    shape["points"] = [
        [top_left_x, top_left_y],    # Top-left
        [bottom_right_x, bottom_right_y]  # Bottom-right
    ]

# Overwrite the existing file with updated JSON data
with open(file_path, 'w') as f:
    json.dump(data, f, indent=4)

print(f"Bounding boxes updated in {file_path}")
```

## 3. Documentation

I generated 400 additional annotations. While reviewing some of the data I discovered the labeling tool I used assigned the points based on mouse clicks rather than by standard points of the bounding box. While most labels were created from the top-left to bottom-right sometimes it was easier to label the image starting from a different corner of the bounding box. After discovering this issue, I created a simple script to convert all annotations to the top-left bottom-right standard.

With the correction to bounding boxes and additional training data I retrained the object detection model. Results no longer required padding additional size to the edges. Unfortunately, the output has a preference to being a centered square. The output bounding box will warp its dimensions, change its size, and shift its position to a slight degree, it is not as robust as it needs to be. In the general case most anoles are in the center of the image and the bounding box captures a large portion of the anole, in specific cases where the anole was off center of the image it did not perform as expected.

Figure 1. Detected anoles

In the left image the box has the expected shape and size but did not adjust its position to capture the anoles body and left half the box contents as the background. In the middle image the box adjusted its shape correctly but is slightly oversized. In the right image the box has the correct shape and size but did not adjust its position adequately and failed to capture both the body and head of the anole.

With improved detections I have retrained the classifier, but the testing cycle has not finished at the time of this report.

## 4. Next Weeks Proposal

I will be investigating if there is a logical error in the processing of the annotations that is creating a strong bias to keep the box to the center of the image. The other possibility is there is a bias put in by the by the training data being composed mostly of images with the anole in the center. If the latter is the case, I will need to generate additional training data with affine transformations to create more diversity in the data set.

# Week11 report

Ruiqing Wang | CiChild CV team

- What progress did you make in the last week?

1. Finished labeled videos from DeepLabCut
2. Read user guide on SLEAP and installed it on my local pc and HPC
3. Get annotated data using SLEAP
4. Attend Cichild group meeting and discussed about technical details
5. Review papers on SLEAP and pose estimation
6. Help assembling paper report submissions and address submission situation.

- What are you planning on working on next?

1. Get more annotation data on SLEAP
2. Run DEMO network training and validate results
3. Meet with Cichild CV team to discuss current progress

- Is anything blocking you from getting work done?

N/A

## Paper abstract

Paper: Selfee, self-supervised features extraction of animal behaviors
https://elifesciences.org/articles/76218

Abstract: The authors introduce Selfee, a self-supervised convolutional neural network designed to extract comprehensive and discriminative features from social behavior video frames in an end-to-end manner, thereby mimicking human perception. This approach not only facilitates the detection of subtle anomalous behaviors that may escape human observation but also enables temporal analysis of behaviors, showcasing its potential for various downstream applications in behavioral research.

Methodology: By projecting images into a low-dimensional space that is invariant to various shooting conditions. The extracted features from Selfee facilitate a range of downstream analyses, including t-SNE visualization, k-NN classification and anomaly detection, and integration with autoregressive hidden Markov models (AR-HMM), demonstrating its applicability across multiple model organisms and potential for broader behavioral studies.

Selfee employs a dual-branch architecture of Siamese Convolutional Neural Networks (CNNs) to generate discriminative representations from live-frame video data, utilizing ResNet-50 as its backbone with a modified classifier replaced by a three-layer multi-layer perceptron (MLP) known as projectors. The main branch includes an additional predictor, while the reference branch mirrors the main branch, following the SimSiam framework, which simplifies the training process by optimizing cosine similarity between the predicted and actual representations. During training, live-frames undergo random augmentations, such as cropping and rotation, to enhance the model's robustness and facilitate effective online clustering of the extracted features.

### Scripts and Code Blocks

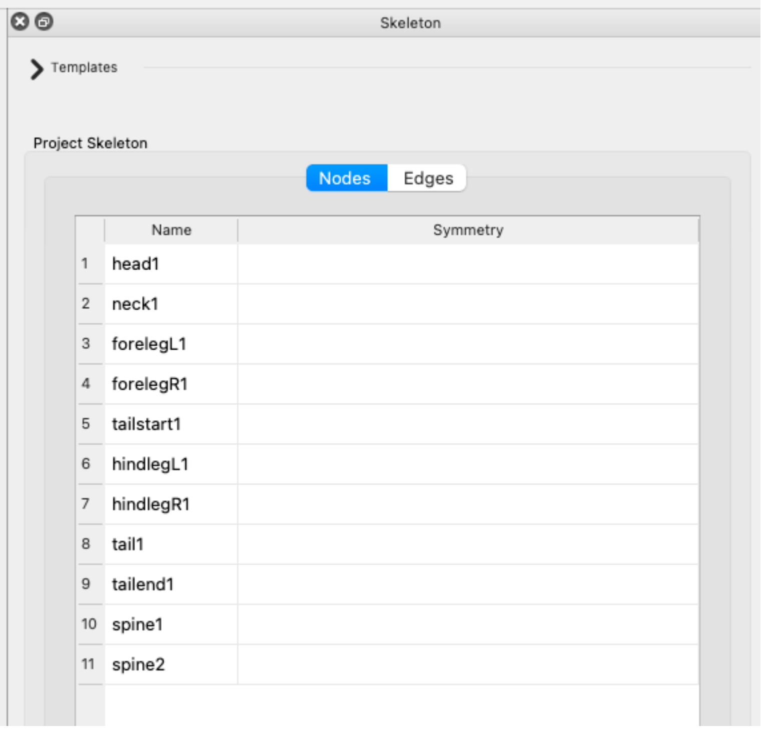
This week I mainly read papers and user guide, install softwares and did data annotation. There are no codes update.

### Documentation

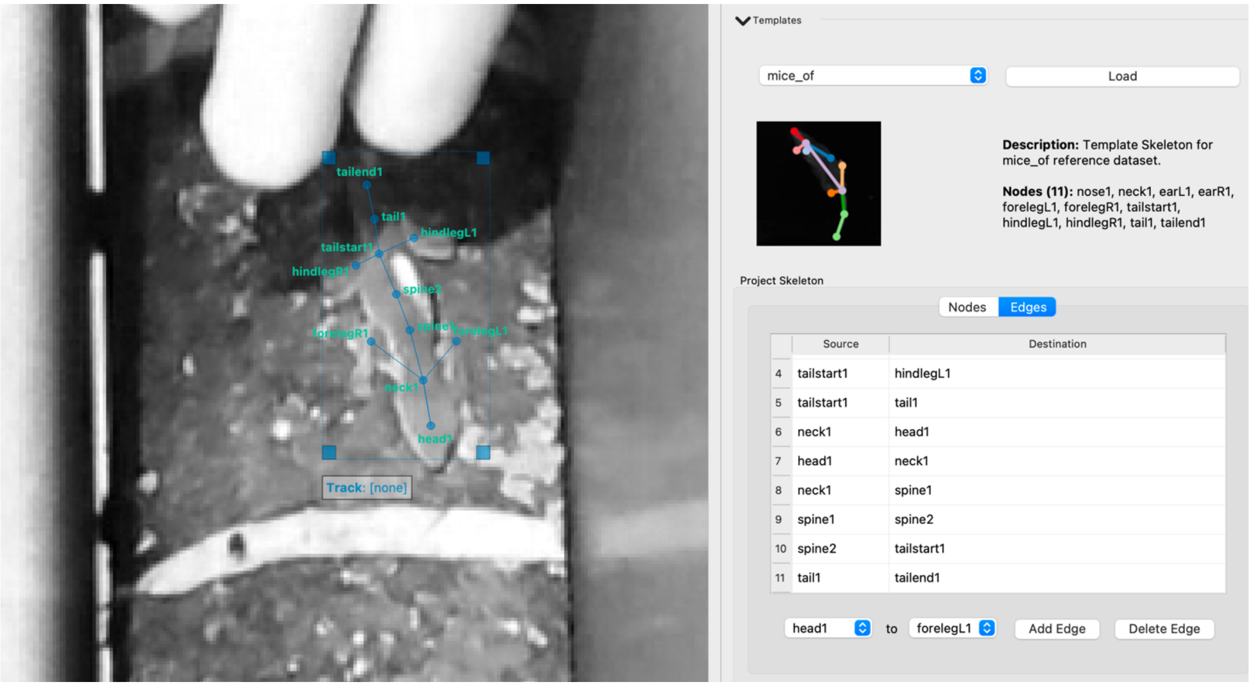
The data annotation in SLEAP route is pretty much similar with DLC. However, the SLEAP software mainly focused on skeleton, instead of label it self. So the annotation is node + edges, basically treated as a graph.
Please check https://sleap.ai/guides/index.html for more details.

### Results Visualization and Code Validation

Here is the node and edge settings for lizard dataset:



|  | Name | Symmetry |
|---|---|---|
| 1 | head1 |  |
| 2 | neck1 |  |
| 3 | forelegL1 |  |
| 4 | forelegR1 |  |
| 5 | tailstart1 |  |
| 6 | hindlegL1 |  |
| 7 | hindlegR1 |  |
| 8 | tail1 |  |
| 9 | tailend1 |  |
| 10 | spine1 |  |
| 11 | spine2 |  |



**Description:** Template Skeleton for mice_of reference dataset.

**Nodes (11):** nose1, neck1, earL1, earR1, forelegL1, forelegR1, tailstart1, hindlegL1, hindlegR1, tail1, tailend1

Project Skeleton

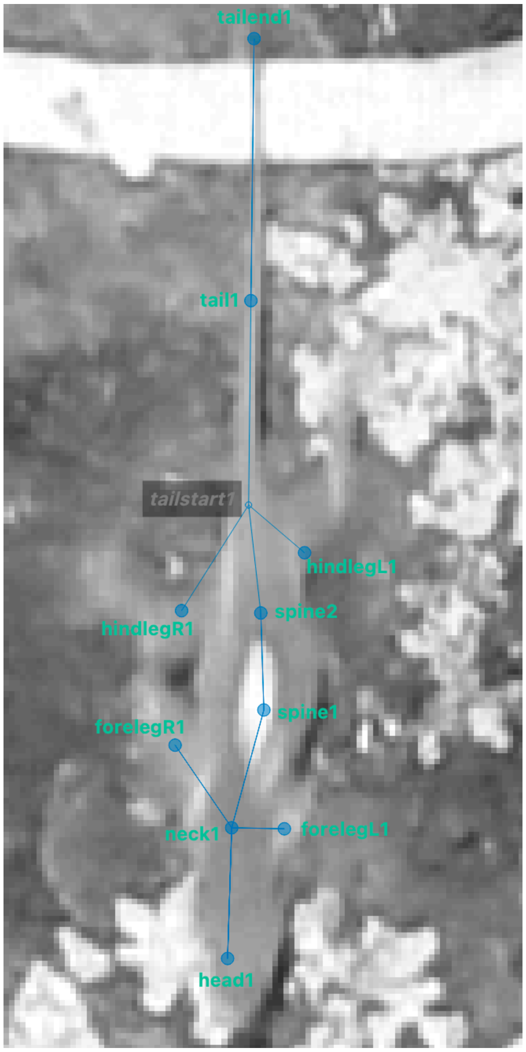
|  | Source | Destination |
|---|---|---|
| 4 | tailstart1 | hindlegL1 |
| 5 | tailstart1 | tail1 |
| 6 | neck1 | head1 |
| 7 | head1 | neck1 |
| 8 | neck1 | spine1 |
| 9 | spine1 | spine2 |
| 10 | spine2 | tailstart1 |
| 11 | tail1 | tailend1 |

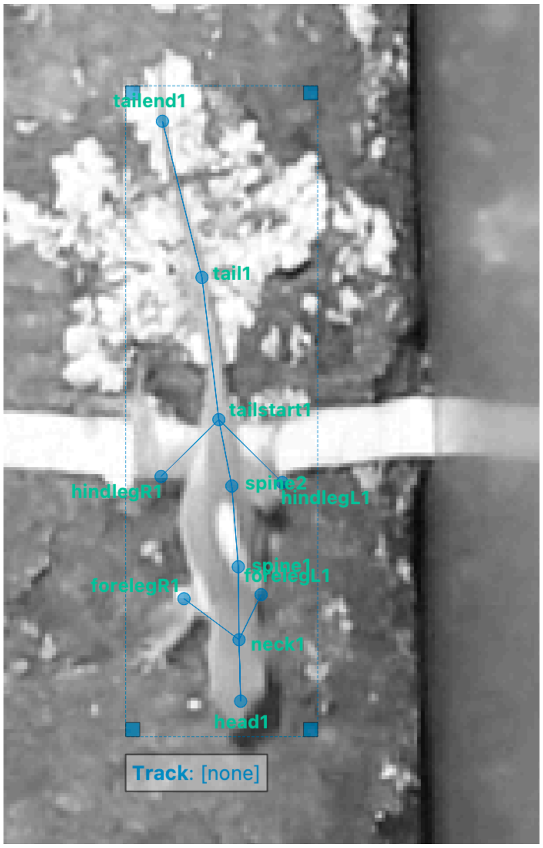
For skeleton design, I basically followed the mouse skeleton template and made small modifications on adding spines bodyparts. Under “Skeleton" interface, it specifically highlighted a node-based system for mapping anatomical points in a subject. The table lists nodes such as "head1," "neck1," "forelegL1," and others, indicating their labeled positions and symmetry attributes, which are key in defining the structure for motion analysis or behavioral studies.

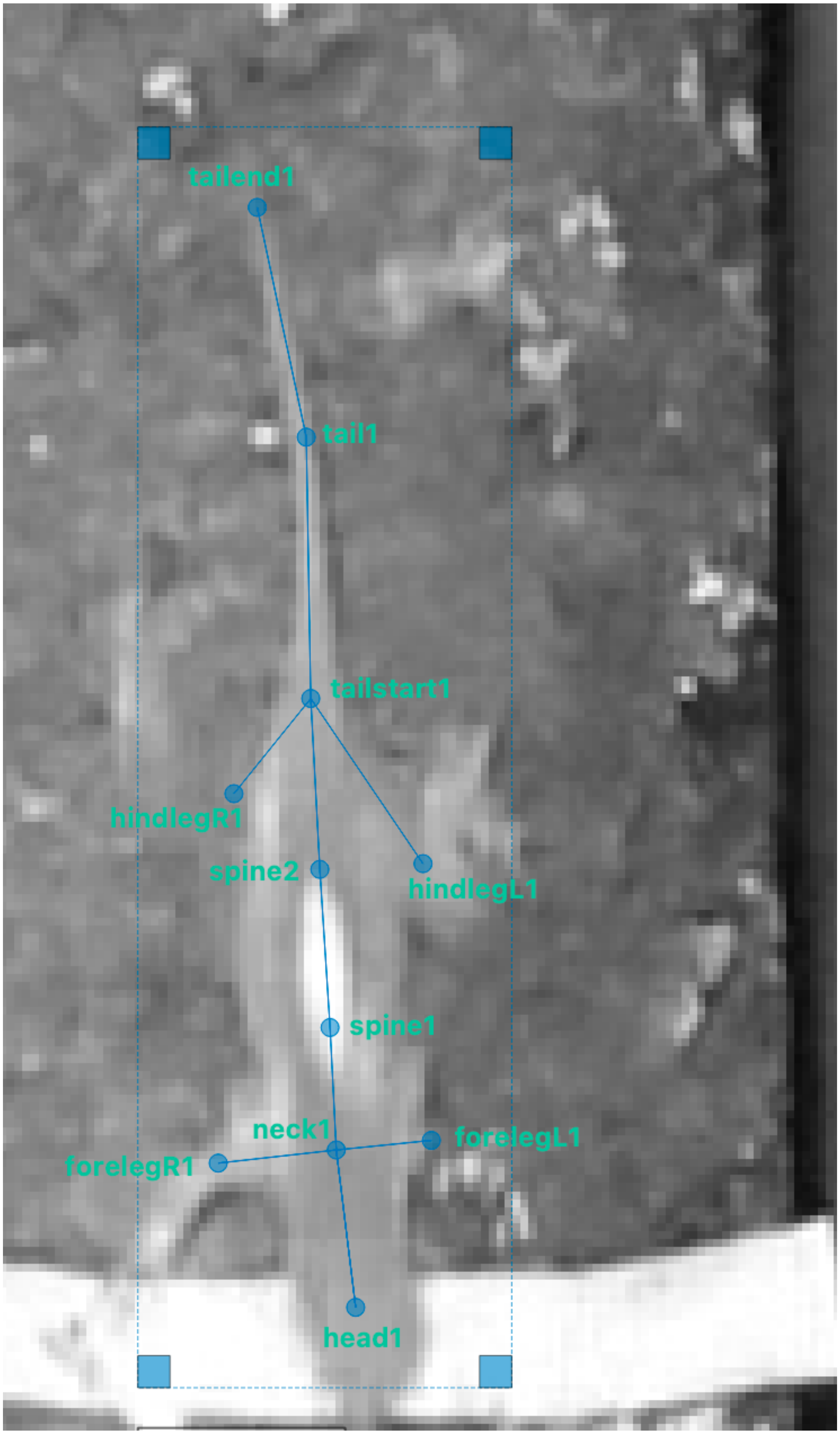
The second image shows the practical application of this skeleton structure within an annotation environment. On the right panel, the "Project Skeleton" section provides a connection table between source and destination nodes, building the skeletal framework needed for defining body movements or postures. This structured approach is essential for ensuring accuracy in movement analysis or automated tracking.

## Proof of Work and code validation

Below are some data annotation samples I did at the GUI:

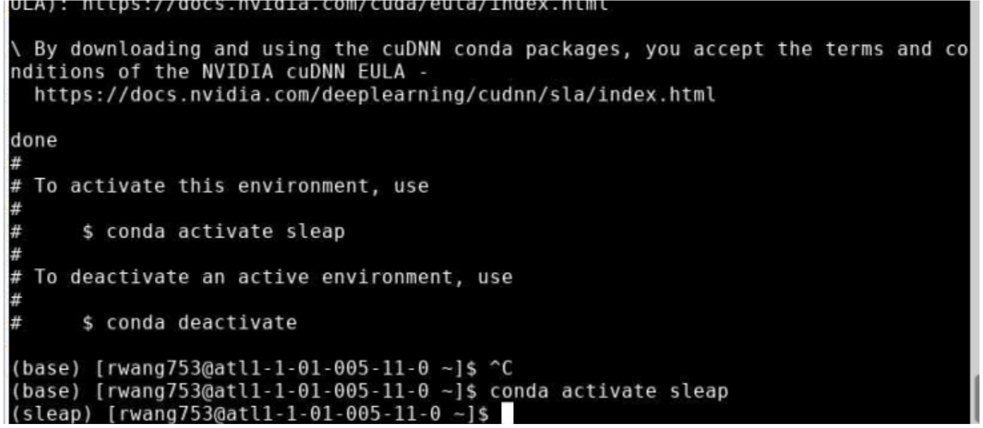

tailend1
tail1
tailstart1
hindlegL1
spine2
hindlegR1
spine1
forelegR1
neck1
forelegL1
head1


tailend1
tail1
tailstart1
hindlegR1
spine2
hindlegL1
spine1
forelegL1
forelegR1
neck1
head1
Track: [none]


tailend1
tail1
tailstart1
hindlegR1
spine2
hindlegL1
spine1
neck1
forelegL1
forelegR1
head1

Key anatomical points such as "head1," "neck1," "forelegL1," "forelegR1," "hindlegL1," "hindlegR1," "spine1," "spine2," "tailstart1," "tail1," and "tailend1" are labeled. These nodes are connected to form a skeletal framework that follows the animal's body structure from head to tail, representing the major joints and points of articulation.

Here is the screenshot for successfully download and installed it in ICE:



```
ULA): https://docs.nvidia.com/cuda/eula/index.html

\ By downloading and using the cuDNN conda packages, you accept the terms and co
nditions of the NVIDIA cuDNN EULA -
  https://docs.nvidia.com/deeplearning/cudnn/sla/index.html

done
#
# To activate this environment, use
#
#     $ conda activate sleap
#
# To deactivate an active environment, use
#
#     $ conda deactivate

(base) [rwang753@atl1-1-01-005-11-0 ~]$ ^C
(base) [rwang753@atl1-1-01-005-11-0 ~]$ conda activate sleap
(sleap) [rwang753@atl1-1-01-005-11-0 ~]$ 
```

The procedure is as follows:
First you should have anaconda installed in your current environment:
[https://www.anaconda.com/blog/a-faster-conda-for-a-growing-community](https://www.anaconda.com/blog/a-faster-conda-for-a-growing-community)
Set the solver in the base environment (and skip the Mambaforge installation):

```
 conda update -n base conda
conda install -n base conda-libmamba-solver
conda config --set solver libmamb
```

Finally install the pakage:

```
conda create -y -n sleap -c conda-forge -c nvidia -c sleap -c anaconda sleap=1.3.3
```

### Next Week's Proposal

1. Get more annotation data on SLEAP
2. Run DEMO network training and validate results
3. Meet with Cichild CV team to discuss current progress

## Week 11 Document Submission

## Lizard X-RAY Landmark Group

**Mercedes Quintana**

What progress did you make in the last week?

- Continued to work on website
- Trained models on both image sets
- Fixed issue from last week
- Talked with Dr. Stroud about possible publishing options
- Starting to package code to be published

What are you planning on working on next?

- Start grid searching to find best model parameters
- Create frontend for final product
- Continue to update the website

Is anything blocking you from getting work done?

- Nope

**Abstracts:**

URL: https://besjournals.onlinelibrary.wiley.com/doi/full/10.1111/2041-210X.13373

ML-morph: A fast, accurate and general approach for automated detection and landmarking of biological structures in images

Morphometrics has become an indispensable component of the statistical analysis of size and shape variation in biological structures. Morphometric data have traditionally been gathered through low-throughput manual landmark annotation, which represents a significant bottleneck for morphometric-based phenomics. Here we propose a machine-learning-based high-throughput pipeline to collect high-dimensional morphometric data in two-dimensional images of semi-rigid biological structures.

The proposed framework has four main strengths. First, it allows for dense phenotyping with minimal impact on specimens. Second, it presents landmarking accuracy comparable to manual annotators, when applied to standardized datasets. Third, it performs data collection at speeds several orders of magnitude higher than manual annotators. And finally, it is of general applicability (i.e. not tied to a specific study system).

State-of-the-art validation procedures show that the method achieves low error levels when applied to three morphometric datasets of increasing complexity, with error varying from 0.57% to 2.2% of the structure's length in the automated placement of landmarks. As a benchmark for the speed of the entire automated landmarking pipeline, our framework places 23 landmarks on

13,686 objects (zooids) detected in 1,684 pictures of fossil bryozoans in 3.12 min using a personal computer.

The proposed machine-learning-based phenotyping pipeline can greatly increase the scale, reproducibility and speed of data collection within biological research. To aid the use of the framework, we have developed a file conversion algorithm that can be used to leverage current morphometric datasets for automation, allowing the entire procedure, from model training all the way to prediction, to be performed in a matter of hours.

Summary: This paper introduces the tool that our pipeline is based on which uses cascade regression trees to automate landmarking of lizard photos.

**Scripts and Code Blocks:**

Last week I had struggled with getting the pipeline to work with new images, and worked with Dr. Porto to change the preprocessing steps to work with an image set that does not need to run through a bounding box detector first. With his help, I integrated his utils.py to work with the new inference script. This updated version will be available in the Github.

**Documentation:**

I have no new code to document as of now.

**Script Validation:**

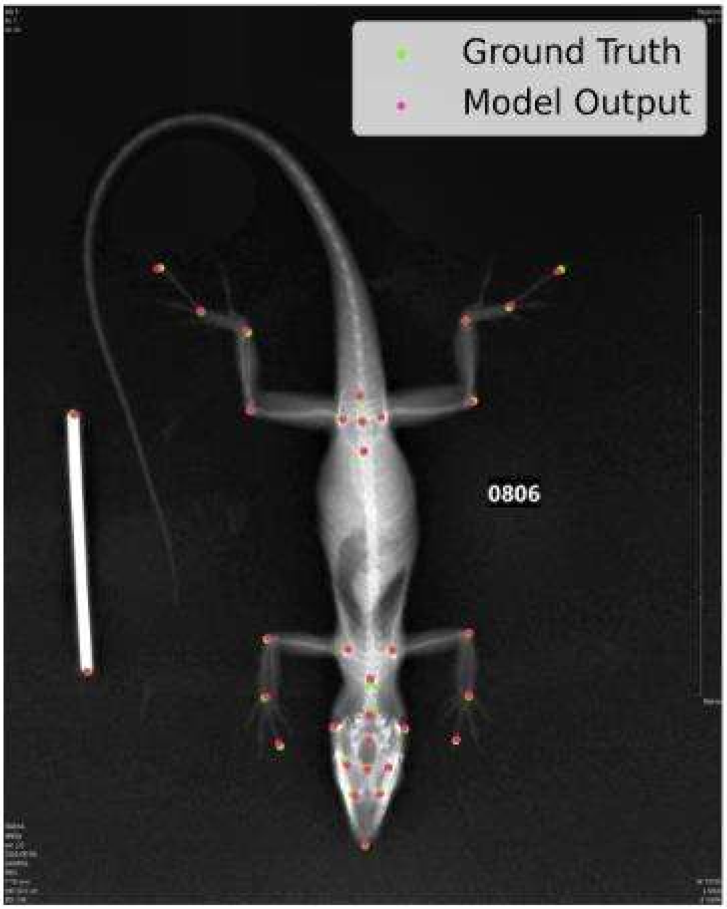
I have no validation steps now.

**Results Visualization / Proof of Work:**

Here are some examples shared in the bi-weekly group meeting of the strange results I was getting from the output of the model. I am sure that the behavior is from how the files are originally prepared to be trained, and am in contact with Dr. Porto who pledged to help.

**Next Week Proposal:**

Both models are working again! Here is an example of an individual lizard from the auto image set.



**Next Steps:**

Next week I am going to work on getting the code documented and workable for someone else. I am going to continue to try and get better performance from both models. I am going to come up with an idea of how we would like our front end for the project to work. I am also going to read through the papers that Dr. Stroud sent us to find out how we would like to present our work in some sort of published fashion.