Week 9L Document Submission

Jacob Dallaire

October 18, 2024

1. Paper

Kim J-H, Kim N, Park YW, Won CS. Object Detection and Classification Based on YOLO-V5 with Improved Maritime Dataset. *Journal of Marine Science and Engineering*. 2022; 10(3):377. https://doi.org/10.3390/jmse10030377

SUMMARY

The text discusses the limitations of existing public image datasets, such as COCO and Pascal VOC, in training deep neural networks (DNNs) for domain-specific applications, particularly in maritime environments where datasets like the Singapore Maritime Dataset (SMD) suffer from issues like noisy labels, inaccurate bounding boxes, and class imbalance. To address these challenges, the authors propose an improved dataset, SMD-Plus, which corrects these annotation errors and implements novel augmentation techniques, including an Online Copy & Paste method, to enhance the training of DNNs like YOLO-V5 for both object detection and classification tasks. The results indicate significant improvements in detection performance, demonstrating the efficacy of the SMD-Plus dataset and the modified YOLO-V5 model for maritime object recognition.

2. Scripts

No scripts this week. Just Data Annotation.

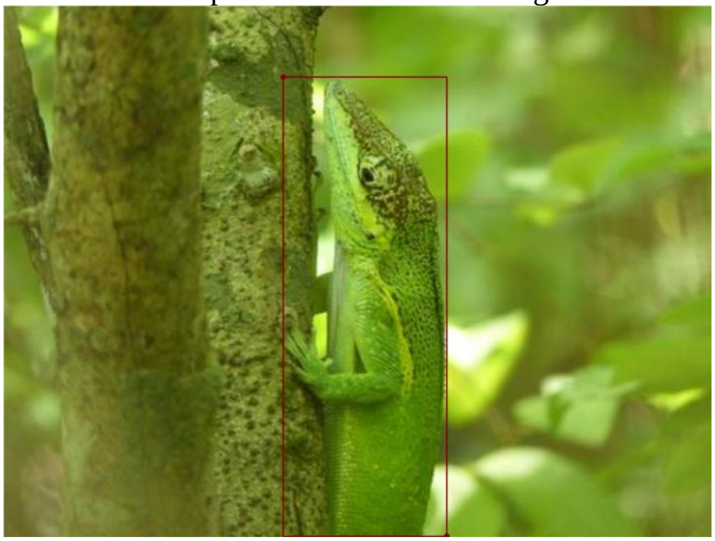
3. Documentation

I had a busy week with other course work as well as handling some personal responsibilities, so between those and the amount of annotations I did I was not able tostart on model training this week.

Annotations

I randomly copied 1000 images to another folder for use of training an object detection model. I was able to generate annotations on 990 of these images using Labelme a lightweight free application for creating segmentations.. In the remaining 10 images I was unable to identify the locations of the Anoles myself and did not annotate them.

Here is an example of one of the bounding box annotations I performed.



The json annotation:


```
{
  "version": "5.5.0",
  "flags": {},
  "shapes": [
    {
      "label": "lizard",
      "points": [
        [
          197.04724409448818,
          50.09842519685039
        ],
        [
          312.40157480314963,
          373.7204724409449
        ]
      ],
      "group_id": null,
      "description": "",
      "shape_type": "rectangle",
      "flags": {},
      "mask": null
    }
  ],
  "imagePath": "36391_233833071.jpg",
  "imageData": %REMOVED FOR READABILITY%
  "imageHeight": 375,
  "imageWidth": 500
}
```

4. Next Weeks Proposal

I will train the object detection model and evaluate if its performance is acceptable. After I will adapt the training and testing scripts of the classification model to incorporate the object detection model into its image processing to reduce the amount of image background information being fed to the model prioritizing just the anole itself as input.

Weekly Report

Philip Woolley

2024-10-18

Time Log Reponse:

- Created proof of concept script to demonstrate 3d registration. Segmented an additional exam.
- What are you planning on working on next? - Continue segmenting training data. Retrain model with additional data and new metric. Change website structure to match request from Bree
- Is there anything blocking you? - Access to "Appearance" tab on wordpress needed to rearrange navigation menu

1 Abstract

Abstract

We evaluate the accuracy of an original hybrid segmentation pipeline, combining variational and deep learning methods, in the segmentation of CT scans of stented aortic aneurysms, abdominal organs and brain lesions. The hybrid pipeline is trained on 50 aortic CT scans and tested on 10. Additionally, we trained and tested the hybrid pipeline on publicly available datasets of CT scans of abdominal organs and MR scans of brain tumours. We tested the accuracy of the hybrid pipeline against a gold standard (manual segmentation) and compared its performance to that of a standard automated segmentation method with commonly used metrics, including the DICE and JACCARD and volumetric similarity (VS) coefficients, and the Hausdorff Distance (HD). Results. The hybrid pipeline produced very accurate segmentations of the aorta, with mean DICE, JACCARD and VS coefficients of: 0.909, 0.837 and 0.972 in thrombus segmentation and 0.937, 0.884 and 0.970 for stent and lumen segmentation. It consistently outperformed the standard automated method. Similar results were observed when the hybrid pipeline was trained and tested on publicly available datasets, with mean DICE scores of: 0.832 on brain tumour segmentation, and 0.894/0.841/0.853/0.847/0.941 on left kidney/right kidney/spleen/aorta/liver organ segmentation.

Summary This paper describes an evaluation of a proposed automatic segmentation pipeline for medical images. The pipeline contains an algorithmic and a deep learning component, similar to how my proposed pipeline is organized. The authors do a good job of evaluating the pipline outputs, but I would like to see an analysis of each of the components of the pipeline individually. That is the sort of analysis I would like to provide for my pipeline when it is complete, to show how differences at each step can propagate to the final results. It would also have been helpful to have an architecture or workflow diagram for the pipeline.

Citation

Burrows, L., Chen, K., Guo, W. et al. Evaluation of a hybrid pipeline for automated segmentation of solid lesions based on mathematical algorithms and deep learning. Sci Rep 12, 14216 (2022). https://doi.org/10.1038/s41598-022-18173-0

2 Scripts and Code Blocks

Using code adapted from a tutorial for Open3D, I created the registration.ipynb notebook, which has a proof of concept for aligning skull images with full body images, so that the skulls of the full body image can be cropped and resampled.

```
def execute_fast_global_registration(source_down, target_down, source_fpfh,
                                     target_fpfh, voxel_size):
    distance_threshold = voxel_size * 0.5
    print(":: Apply fast global registration with distance threshold %.3f" \
          % distance_threshold)
    result = o3d.pipelines.registration.registration_fgr_based_on_feature_matching(
        source_down, target_down, source_fpfh, target_fpfh,
        o3d.pipelines.registration.FastGlobalRegistrationOption(
            maximum_correspondence_distance=distance_threshold))
    return result
```

```python
import numpy as np
import nrrd as pynrrd
import open3d as o3d
import copy

# Load the NRRD files
source_data, _ = pynrrd.read("vols/KU-herp-63380-wholebody0000 cropped.nrrd")
target_data, _ = pynrrd.read("C:/Users/pgmw9/OneDrive/Documents/KU-herp-75787-wholebody0000.nrrd")

print(source_data.max(), target_data.max())
# Convert the 3D volumes to point clouds
def volume_to_point_cloud(volume, threshold=38000):
    temp = np.asarray((volume > threshold))
    #temp = np.asarray(temp < 60000)
    z, y, x = temp.nonzero()
    points = np.vstack((x, y, z)).T  # Transpose to get points in (N, 3) format
    return points

source_points = volume_to_point_cloud(source_data)
target_points = volume_to_point_cloud(target_data)

# Create Open3D point clouds
source_pcd = o3d.geometry.PointCloud()
target_pcd = o3d.geometry.PointCloud()
print("1")

source_pcd.points = o3d.utility.Vector3dVector(source_points)
target_pcd.points = o3d.utility.Vector3dVector(target_points)
print("2")
```

3 Documentation

The VisualizeModelResults.ipynb notebook is used for creating and viewing images of model output on validation data. Users provide a pretrained model and validation dataset, and this notebook inferences all of the images in the dataset and allows the user to review the output segmentations against the ground truth manual segmentations.

The DataProcess.ipynb notebook is used for converting slicer volume files (.nrrd and .seg.nrrd) into a HuggingFace dataset for use with the pretrained Mask2Former model. Volumes should be added to the "vols" folder, and segmentation volumes should be added to the "masks" folder.

https://www.morphosource.org/projects/0000C1059?locale=enpage=11sort=publication_status_s List of available MicroCT Datasets of anolis lizards that will be used for this project. When infrastructure for data storage is ready I will prepare documentation detailing the downloading and storage process.

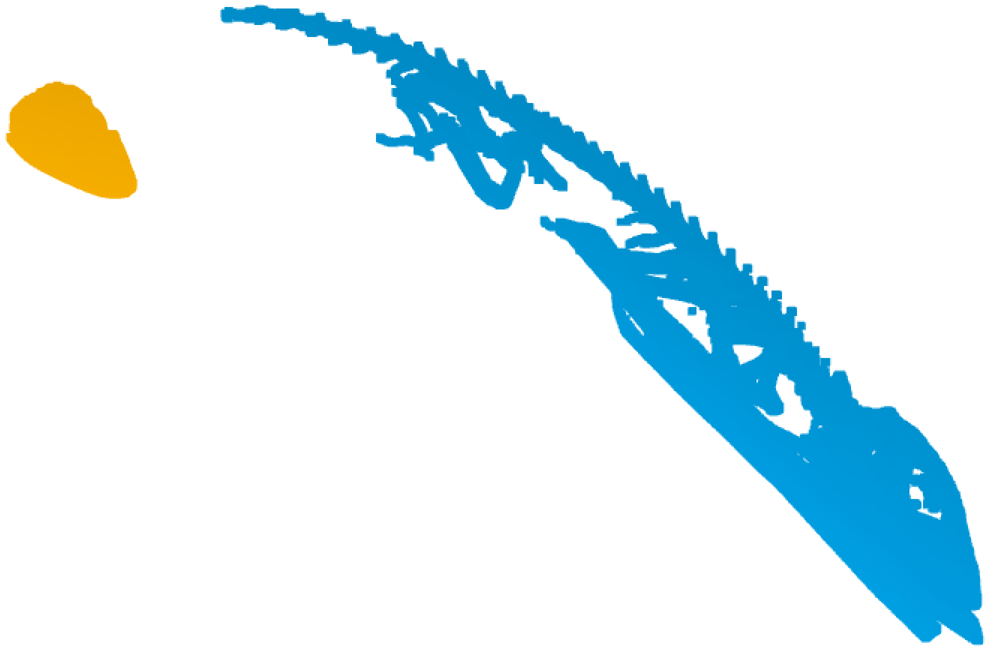
https://slicermorph.github.io/ Documentation for SlicerMorph, an extension of the 3D slicer tool commonly used by Biologists. This is used for loading stacks of .tiff images as a volume in 3d slicer.

https://github.com/jmhuie/SlicerBiomech Documentation for the Dental Dynamics module, which is a 3D slicer extension for calculating tooth stress from jaw segmentations. the outputs from my segmentation pipeline will need to be compatible with this module for analysis.

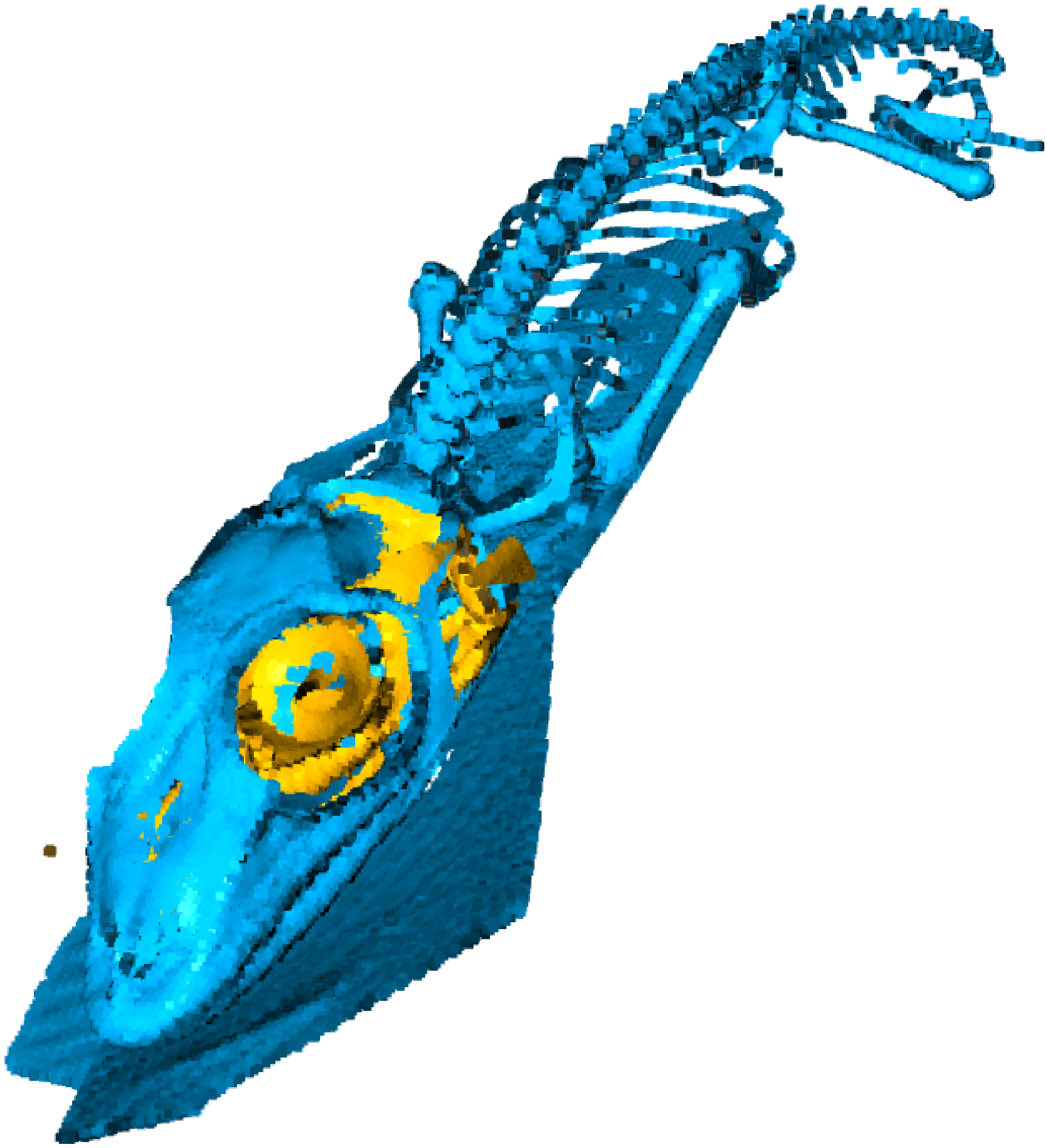
4 Script Validation (Optional)

5 Results Visualization

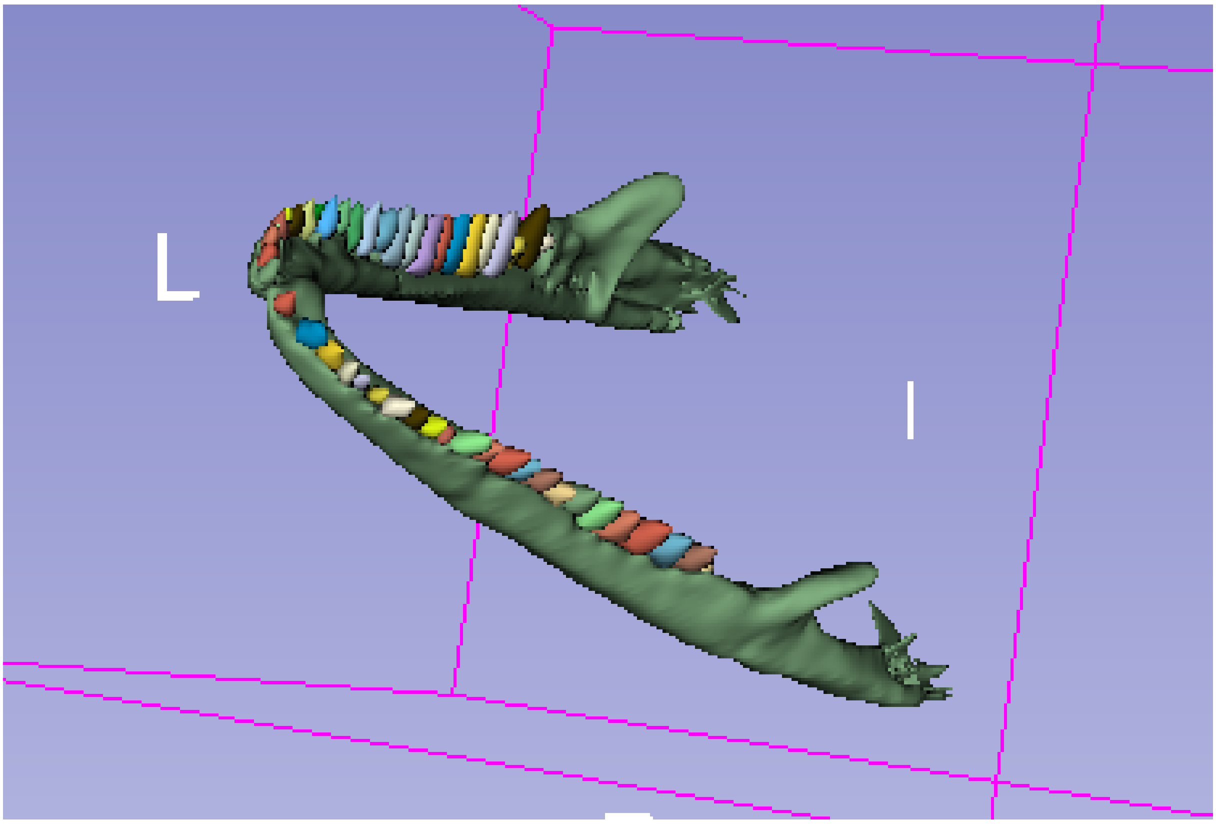
Image showing alignment of skull and body before registration is performed.



Here is an example showing alignment of a skull (yellow) to a full body scan (blue) performed automatically using 3d rigid registration.



Here is another case I segmented manually this week.



6 Proof of Work

Please see Code Blocks and Results Visualization

7 Next Week's Proposal

- Continue segmenting training data for ML panoptic segmentation model
- Develop testing script for 3D image registration for converting coordinate systems
- Reformat blog page as requested by Bree

Week9 report

Ruiqing Wang | CiChild CV team

- What progress did you make in the last week?

1. Trained network on 20 videos and add test videos
2. Evaluate current model and its RMSD
3. Retrieve labeled lizard video on test videos
4. Get X and Y position prediction on multiple trajectory frames
5. Get data analysis on current model performance
6. Attend Cichild group meeting and discussed about technical details
7. Review papers on DeepLabCut and pose estimation
8. Help assembling paper report submissions and address submission situation.

- What are you planning on working on next?

1. Transfer more videos to PACE and create more labeled videos
2. Meet with Cichild CV team to discuss current progress

- Is anything blocking you from getting work done?

N/A

Paper abstract

Student Behavior Recognition System for the Classroom Environment Based on Skeleton Pose Estimation and Person Detection

https://doi.org/10.3390/s21165314

Abstract: The paper proposes a Student Behavior Recognition System for classroom environments using skeleton pose estimation and person detection. The system processes frames from classroom video feeds to collect skeleton data using the OpenPose framework. To improve the accuracy of the pose estimation, the paper introduces an error correction scheme combining pose estimation and person detection to reduce incorrect connections in skeleton data. It focuses on recognizing four main student behaviors: asking (hand-raising), looking, bowing, and being bored. The system uses deep learning for behavior classification by extracting features such as normalized joint locations, distances between joints, and bone angles. Through experimental testing, the system showed improved precision and recall compared to skeleton-only approaches, particularly in crowded classrooms.

Methodology:

Skeleton pose estimation: the system uses the OpenPose framework to detect 2D skeletons in video frames by capturing the key points of human bodies (like joints) and the OpenPose framework identifies 18 joint locations.

Since skeleton estimation can be erroneous in crowded settings (like classrooms), the system proposes a novel error correction scheme by combining pose estimation with person detection. The error correction involves aligning skeleton data with detected persons using bounding boxes, ensuring that joints correspond to the correct person.

A deep neural network (DNN) is used to classify student behaviors based on the extracted feature vectors.The DNN consists of multiple fully connected layers and uses the softmax function for final behavior classification.The behaviors classified are asking, looking, bowing, and bored. The DNN model successfully identifies these behaviors even in complex scenarios with many students in the frame.

Scripts and Code Blocks

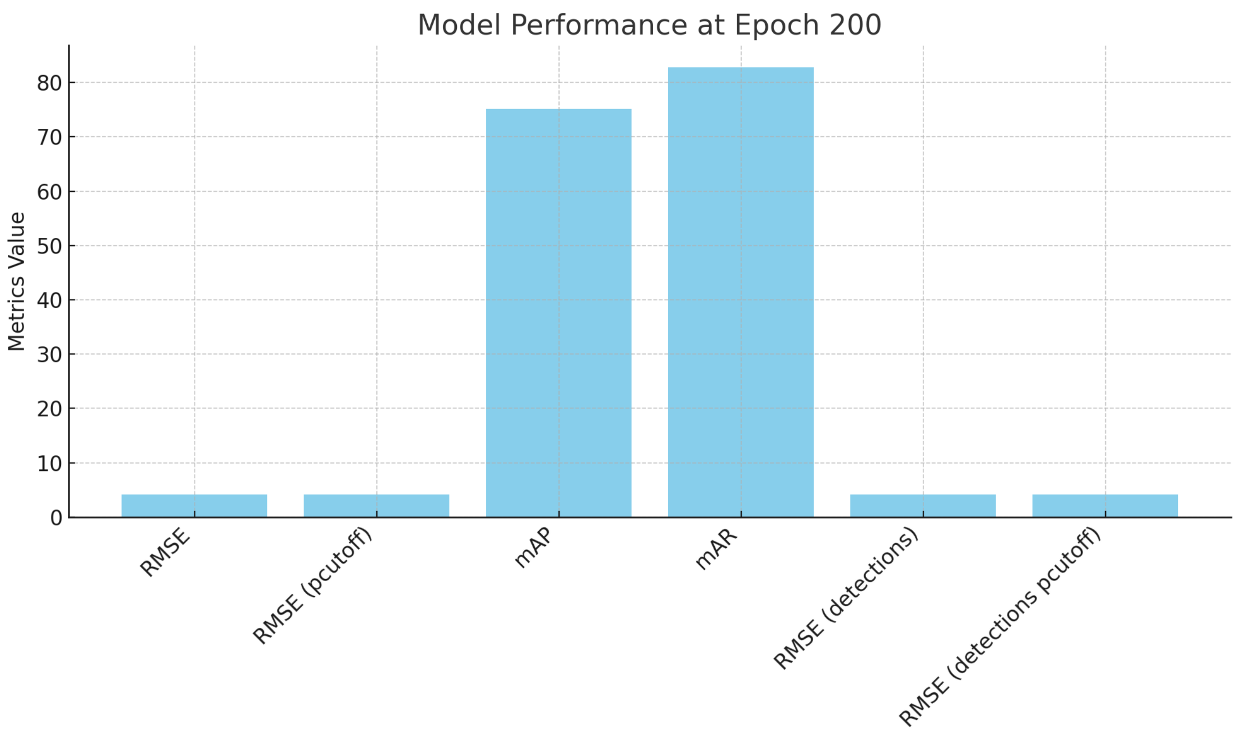
This week I don't have updated code scripts. My main job is on data annotations on various frames and network training. The code I used is almost the same with former codes.

Documentation

The steps are still same to former efforts: mainly evolved data creation, training and network evaluation. All my current code samples were stored in my PACE folder:
/home/hice1/rwang753/scratch/week8

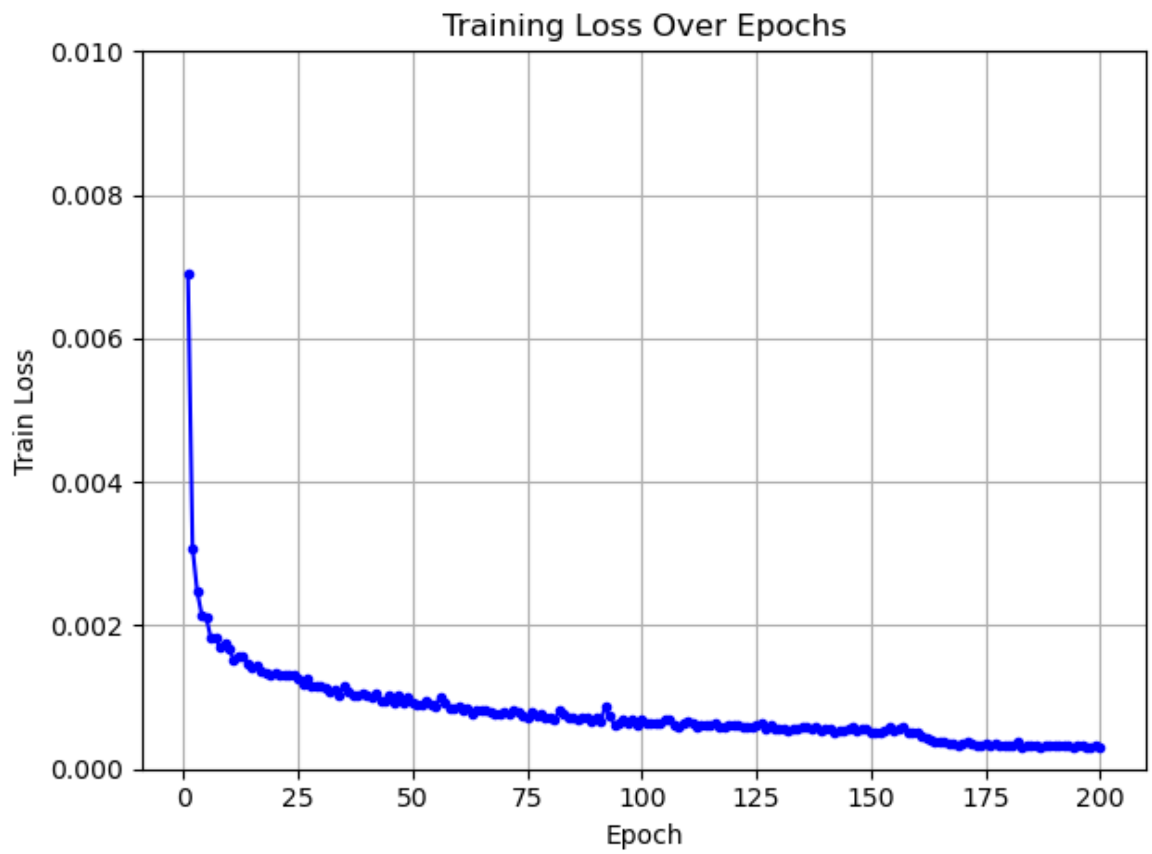
Results Visualization and Code Validation

Here this is the RMSD value for the model at the 200th epoch:



Lower RMSE values generally imply better performance, as RMSE measures the model's prediction error in the units of the predicted value. The mAP value is 75.204, which reflects the model's precision across all classes. A higher mAP score indicates that this model has a good balance between precision and recall. The mAR is 82.778, which is about similar to mAP, the slightly higher may indicated some overfitting.

The below graph shows the training loss during epochs:



From epoch 0 to around epoch 25, there is a steep drop in the training loss. This indicates that the model is quickly learning from the data during the early stages of training. After around epoch 25, the loss starts to decrease more slowly but consistently. From around epoch 150 onward, the loss appears to have nearly flattened, suggesting that the model has reached or is approaching convergence.

Compared with the preliminary work last week, the network training on 20 videos improved a lot and seems to work well. We test it in new random lizard moving video, then we got good videos with lizard labeled and its X and Y position labeled.

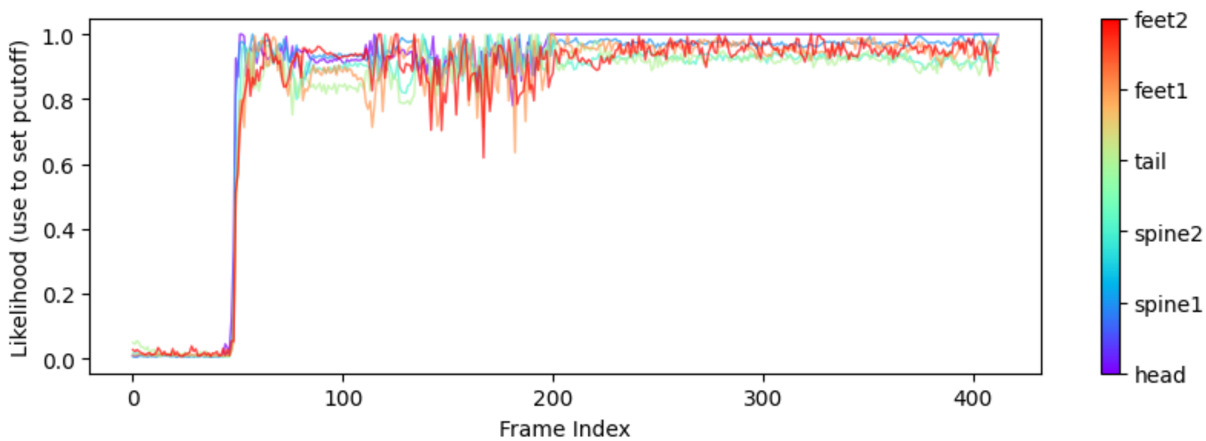


Figure 3: Likelihood of Body Part Detection Over Time

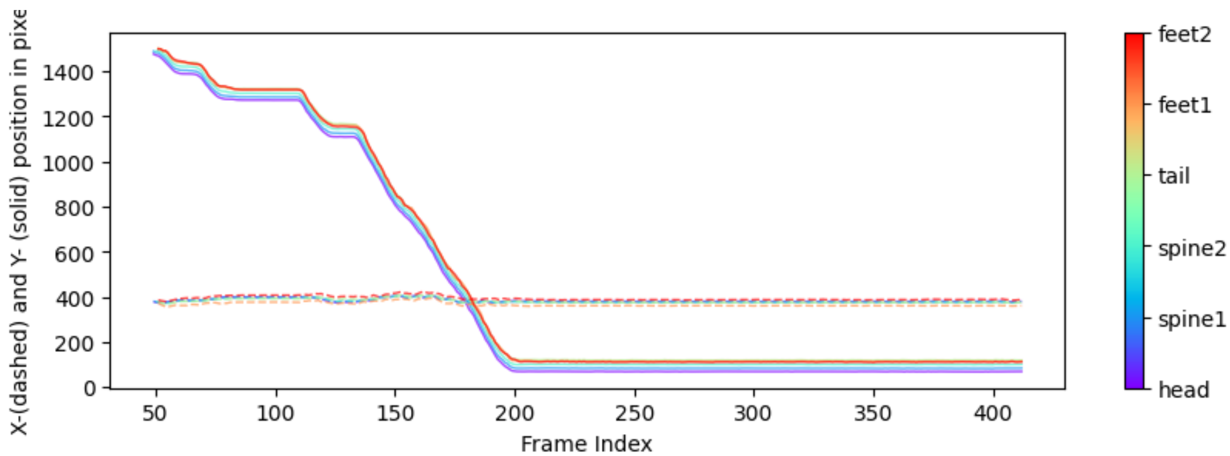


Figure 4: X and Y Position Tracking of Body Parts Over Time

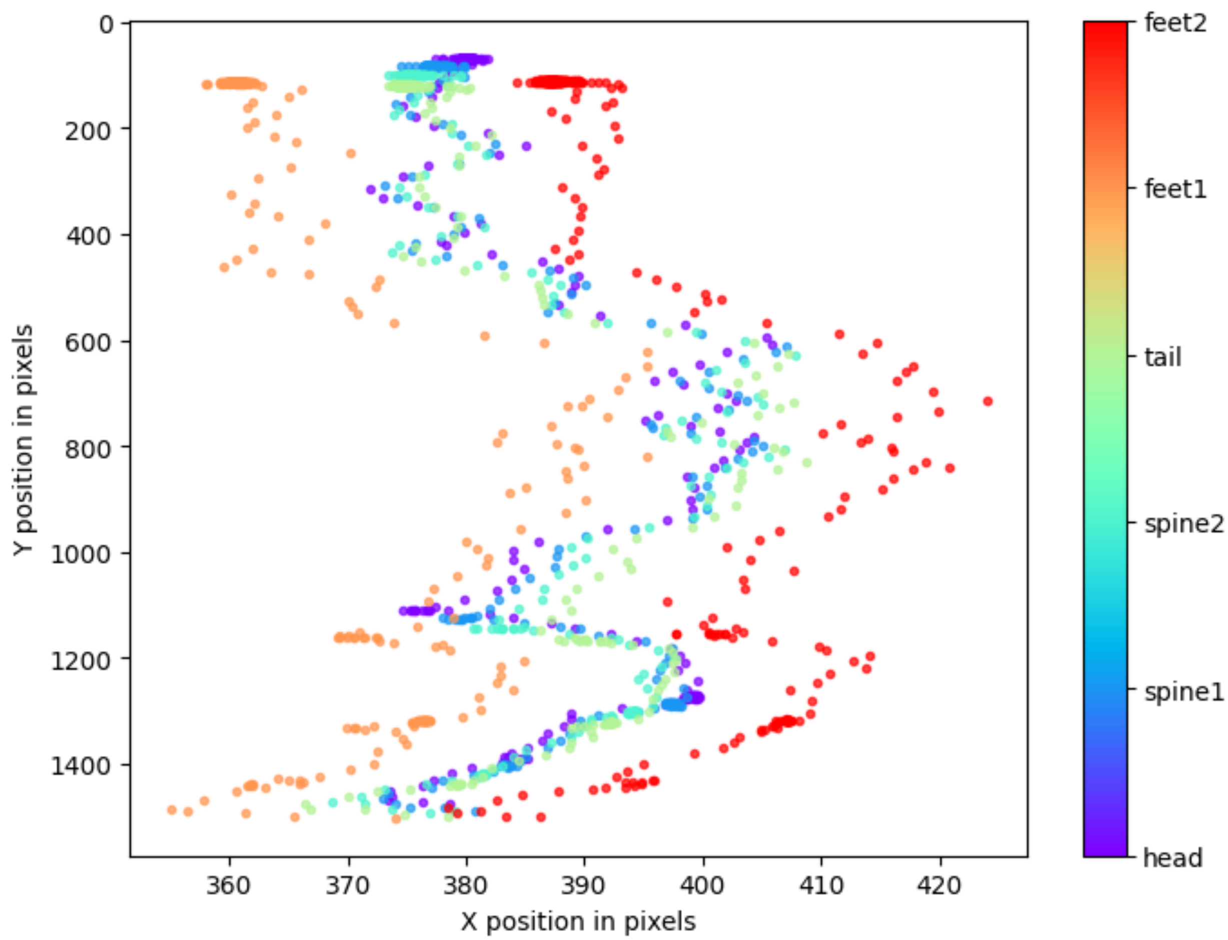


Figure 5: Body Part Trajectory in X-Y Space

In figure 3, this figure illustrates the likelihood values across different body parts (feet, tail, spine, head, etc.) as a function of the frame index. Around frame index 50, there's a sharp increase in likelihood across all body parts. This suggests that the tracking system initially struggles to detect the body parts accurately, but it quickly improves after initial adjustments.

After the sharp rise, the likelihood values remain close to 1.0, indicating that the model becomes highly confident in its body part detection as the video progresses. This chart showed the model is performing well on real-time videos across different body parts.

Figure 4 shows the X (dashed lines) and Y (solid lines) positional coordinates of various body parts (feet, tail, spine, etc.) as a function of the frame index. The graph provides insight into how the tracked positions of the body parts evolve over time. A major drop in Y coordinates occurs around frame index 150-200, suggesting a significant movement (such as lowering, sitting, or crouching) by the subject. Beyond frame index 200, the positions stabilize, implying that the subject remains relatively still after the initial movement phase. The consistent movement patterns of different body parts (colored lines) indicate that they are correctly grouped or aligned during the subject's movement.

Figure 5 shows the trajectories of various body parts (feet, tail, spine, head, etc.) as they move in two-dimensional space. The X-axis represents the horizontal (left-to-right) position, and the Y-axis represents the vertical (up-and-down) position in pixels. The plot shows clear paths of movement for each body part. The trajectories are color-coded, allowing for easy differentiation between body parts. The head and spine markers (cooler colors) show more localized movement, staying closer together and indicating more constrained or stable positions compared to the feet. The feet appear to have a wider range of movement, consistent with walking or shifting positions. The feet trajectories are dispersed, particularly at the top and bottom of the image, suggesting areas of high movement intensity.

Proof of Work and code validation

I would like to provide some screenshot for the test video results:

The video I tested showed that current network is validated and could track lizard body parts in real-time movements.
The screenshot above showed the current detected landmark on walking lizards.
My current work was stored at: /home/hice1/rwang753/scratch

Next Week's Proposal

1. Working on data labels and get advanced network on recognition
2. Test more videos and summarize some lizard behaviors
3. Meet with Cichild CV team to discuss current progress

Week 9 Document Submission

Lizard X-RAY Landmark Group

Mercedes Quintana

What progress did you make in the last week?

- Continued to work on website
- Manicured training dataset for both image sets
- Formatted data correctly to train two models off image sets

What are you planning on working on next?

- Train both image models
- Continue to update the website

Is anything blocking you from getting work done?

- Nope

Abstracts:

URL: https://ieeexplore.ieee.org/stamp/stamp.jsp?tp=&arnumber=9954214

AlphaPose: Whole-Body Regional Multi-Person Pose Estimation and Tracking in Real-Time

Accurate whole-body multi-person pose estimation and tracking is an important yet challenging topic in computer vision. To capture the subtle actions of humans for complex behavior analysis, whole-body pose estimation including the face, body, hand and foot is essential over conventional body-only pose estimation. In this article, we present AlphaPose, a system that can perform accurate whole-body pose estimation and tracking jointly while running in realtime. To this end, we propose several new techniques: Symmetric Integral Keypoint Regression (SIKR) for fast and fine localization, Parametric Pose Non-Maximum-Suppression (P-NMS) for eliminating redundant human detections and Pose Aware Identity Embedding for jointly pose estimation and tracking. During training, we resort to Part-Guided Proposal Generator (PGPG) and multi-domain knowledge distillation to further improve the accuracy. Our method is able to localize whole-body keypoints accurately and tracks humans simultaneously given inaccurate bounding boxes and redundant detections. We show a significant improvement over current state-of-the-art methods in both speed and accuracy on COCO-wholebody, COCO, PoseTrack, and our proposed Halpe-FullBody pose estimation dataset. Our model, source codes and dataset are made publicly available at https://github.com/MVIG-SJTU/AlphaPose

Summary: The paper aims to improve identification of full body multi pose estimation. This paper uses multi-stage pipeline to allow the framework to be able execute in real time.

Scripts and Code Blocks:

I created a script to make two tps files to later be used for training, that combines both image name matching and the training data set that represents the lizards that will best be used for training for future use. I created a txt that has a lizard grade, based on whether it would be good for training, and the manual name called graded_lizards.txt.

Documentation:

Subset_combined_tps.py:

1. Read in tps files for both image sets using the image mapping txt I put together
2. Further remove lizards through graded_lizards.txt
3. Create new tps files for both datasets

Script Validation:

I have no validation steps now.

Results Visualization / Proof of Work:

Here is a screenshot of a small subset of the grading system that I used to find the best data to train the model.

```
0003_dorsal.jpg,good,
0029_dorsal.jpg,bad,feet at two different angles
0031_dorsal.jpg,good,
0036_dorsal.jpg,good,
0038_dorsal.jpg,good,
0039_dorsal.jpg,good,
0040_dorsal.jpg,good,
0046_dorsal.jpg,good,
0047_dorsal.jpg,good,slightly weird front feet
0050_dorsal.jpg,good,
0058_dorsal.jpg,good,
0060_dorsal.jpg,bad,angled toes
0066_dorsal.jpg,bad,angled toes
```

Next Week Proposal:

I plan to keep working on the website to keep it updated with the new meetings and work done. Now that I have two datasets that are ready to train two models, I will train them and have statistics available for our next meeting.