Week 4 Document Submission

Jacob Dallaire

September 13, 2024

1. Paper

Hosna, A., Merry, E., Gyalmo, J. *et al.* Transfer learning: a friendly introduction. *J Big Data* **9**, 102 (2022). https://doi.org/10.1186/s40537-022-00652-w

SUMMARY

The introduction of the paper on Transfer Learning (TL) highlights the challenges of data scarcity and the inefficiencies of traditional machine learning (ML) approaches, which often treat tasks in isolation. TL addresses these issues by enabling the transfer of knowledge from a source task to a target task, thereby enhancing learning efficiency and performance, particularly when target training data is limited. The authors emphasize that TL not only facilitates faster and more effective solutions but also raises important considerations regarding the compatibility of source and target tasks, as well as the potential for negative transfer when the tasks are misaligned.

2. Scripts

A script created to process and resize all images to 256x256 size to speed up training.

```
def resize():
    for subdir, dirs, images in os.walk(folder_path):
        for img in images:
            if img.endswith('.jpg'):
                im = cv2.imread(os.path.join(subdir,img))
                if im is not None and im.shape != (256,256):
                    im = cv2.resize(im,(256, 256))
                    cv2.imwrite(os.path.join(subdir,img),im)
```

Updated download script to run in parallel using multithreading.

3. Documentation

Spent much of my time studying transfer learning so I can try to adapt pre-trained models to the task rather than create an entirely new model. Looking into which pretrained models best fit the use case. Examples of models are ResNet101, MobileNetV2, and Xception.

4. Next Weeks Proposal

Implementation of the transfer learning method. Depending on time it takes compare multiple models to select the best base case to further tune.

Weekly Report

Philip Woolley

2024-09-13

Time Log Reponse:

- What Progress did you make in the last week? - Created proposal for pipeline process, presented to Dr. Stroud and during weekly meeting. Resliced 10 cases, performed full segmentation on 1. Literature review of instance segementation methods for teeth and bones.
- What are you planning on working on next? - Continue segmenting training data. Choose model architecture for segmentation task
- Is there anything blocking you? - None at this time

1 Abstract

Abstract

Purpose. To present SPINEPS, an open-source deep learning approach for semantic and instance segmentation of 14 spinal structures (ten vertebra substructures, intervertebral discs, spinal cord, spinal canal, and sacrum) in whole body T2w MRI.

Methods. During this HIPPA-compliant, retrospective study, we utilized the public SPIDER dataset (218 subjects, 63% female) and a subset of the German National Cohort (1423 subjects, mean age 53, 49% female) for training and evaluation. We combined CT and T2w segmentations to train models that segment 14 spinal structures in T2w sagittal scans both semantically and instance-wise. Performance evaluation metrics included Dice similarity coefficient, average symmetrical surface distance, panoptic quality, segmentation quality, and recognition quality. Statistical significance was assessed using the Wilcoxon signed-rank test. An in-house dataset was used to qualitatively evaluate out of-distribution samples.

Results. On the public dataset, our approach out performed the baseline (instance-wise vertebra dice score 0.929 vs. 0.907, p-value< 0.001). Training on auto generated annotations and evaluating on manually corrected test data from the GNC yielded global dice scores of 0.900 for vertebrae, 0.960 for intervertebral discs, and 0.947 for the spinal canal. Incorporating the SPIDER dataset during training increased these scores to 0.920, 0.967, 0.958, respectively.

Conclusions. The proposed segmentation approach offers robust segmentation of 14 spinal structures in T2w sagittal images, including the spinal cord, spinal canal, intervertebral discs, endplate, sacrum, and vertebrae. The approach yields both a semantic and instance mask as out put, thus being easy to utilize. This marks the first publicly available algorithm1 for whole spine segmentation in sagittal T2w MR imaging.

Summary This paper proposes a multi stage automatic segmentation pipeline for vertebrae from sagittal spine scans. I believe that the authors do a good job of describing the steps of their pipeline, as well as how their preprocessing steps improve model performance over the baseline model. The authors use a two step approach of semantic segmentation followed by instance segmentation, which is an approach that I could potentially follow for tooth segmentation in my pipeline.

Citation

Möller, Hendrik, et al. "SPINEPS–Automatic Whole Spine Segmentation of T2-weighted MR images using a Two-Phase Approach to Multi-class Semantic and Instance Segmentation." arXiv preprint arXiv:2402.16368 (2024).

2 Scripts and Code Blocks

This week, I focused on data preparation using 3D slicer, there are no new scripts.

3 Documentation

https://www.morphosource.org/projects/0000C1059?locale=enpage=11sort=publication_status_ssi+
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.

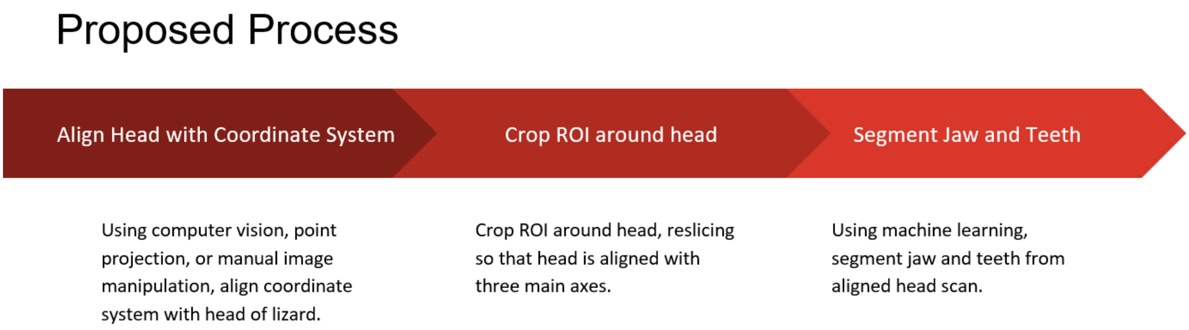
https://github.com/jmhuie/SlicerBiomech Documentation for the Dental Dynamics module, which is a 3D slicer extension for calculating tooth stress from jaw segmentations. the outputs from my segmentation pipeline will need to be compatible with this module for analysis.

4 Script Validation (Optional)

There are no scripts to validate this week.

5 Results Visualization

Here is a slide breaking down the proposed pipeline steps for automatic jaw segmentation



Proposed Process

Align Head with Coordinate System

Crop ROI around head

Segment Jaw and Teeth

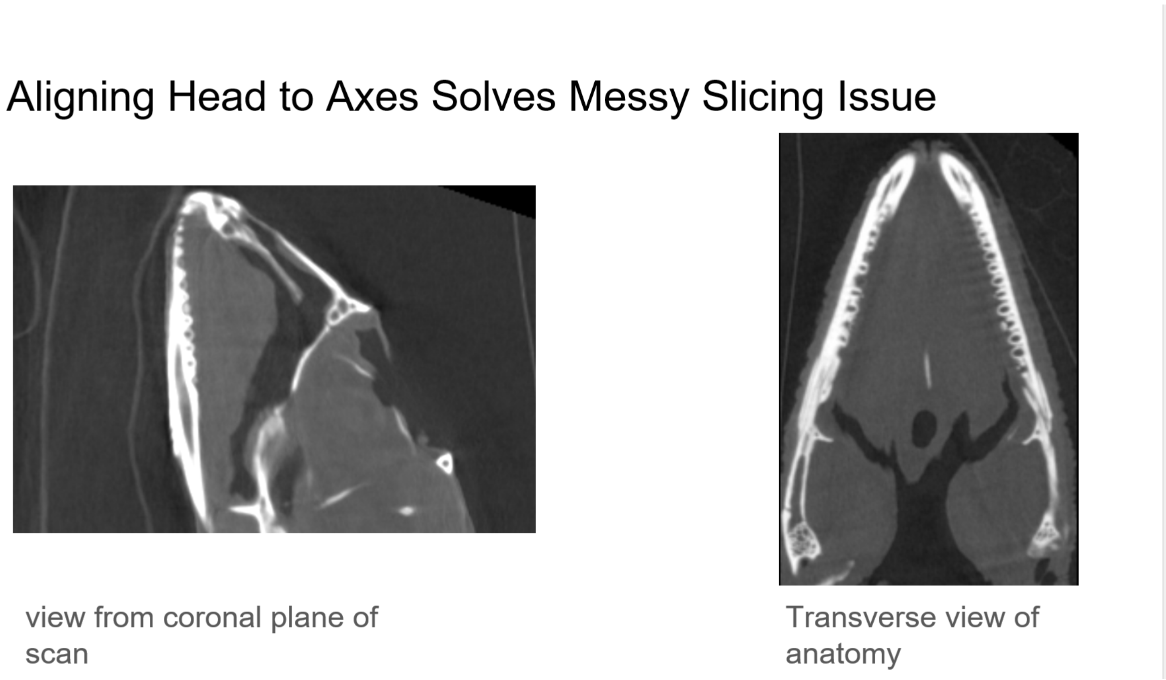
Using computer vision, point projection, or manual image manipulation, align coordinate system with head of lizard.

Crop ROI around head, reslicing so that head is aligned with three main axes.

Using machine learning, segment jaw and teeth from aligned head scan.

6 Proof of Work

See image shown above as well as below example of reslicing head scan for better jaw visualization.



Aligning Head to Axes Solves Messy Slicing Issue

view from coronal plane of scan

Transverse view of anatomy

7 Next Week's Proposal

- Continue segmenting training data for ML instance segmentation model
- Investigate 2d vs 3d segmentation architectures
- Keep up with any required blog posts for webmaster role

Week4 report

Ruiqing Wang | Lizard CV team

Time slot response:

- What progress did you make in the last week?

1. Set up projects environment on Google Colab and DLC GUI on my local computer
2. Creat projects and extract video frames from videos from Dr. Stroud
3. Label Frames and create training dataset using DLC
4. Review papers on DeepLabCut
5. Help assembling paper report submissions and address submission situation.

- What are you planning on working on next?

1. Met with Dr. Stroud and discuss about further methods and recources
2. Diving deeper on videos labeling and data training
3. Check my PACE allocation and evaluate performance

- Is anything blocking you from getting work done?

Preparations and power outage from hurricane

Abstract

Paper: Phenotypic analysis of ataxia in spinocerebellar ataxia type 6 mice using DeepLabCut

Summary:
The study highlights the advantages of utilizing DeepLabCut (DLC) and R, open-source software tools, to automate and enhance the analysis of motor behavior in spinocerebellar ataxia type 6 mouse models. By employing DLC for markerless pose estimation and developing customized analysis scripts in R, researchers were able to track and analyze various body parts during the beamwalk test, revealing specific ataxic abnormalities such as increased hindpaw slips and altered body positions in the ataxic mouse models compared to controls. This detailed analysis provided insights into motor deficits associated with spinocerebellar ataxia type 6, showcasing the potential of combining software automation with experimental data for in-depth phenotypic characterization.

By distinguishing minor and major slips during the beamwalk test in mice, the study successfully validates and characterizes ataxic phenotypes, demonstrating the potential for precise and reliable analysis in comparative studies of mouse models.

Methodology: In the study, DeepLabCut (DLC) version 2.2.1 was utilized to train a neural network for tracking multiple body parts and positions of mice along a beam. The network was trained using 1105 frames from 45 videos representing diverse groups, with 95% of the frames used for training. The trained network, based on ResNet-50 architecture, achieved a test error of 2.77 pixels and a train error of 6.11 pixels, and was capable of tracking various body parts such as the eyes, nose, paws, and tail of the mice.

Scripts and Code Blocks

This week I am mainly focused on video frame extraction and labeling, I created certain code which implemented about it, and it looked like week report #3.

```
[ ] pip install deeplabcut[tf]
```

```
[ ] import deeplabcut
    print(deeplabcut.__version__)
```

```
[ ] config_path = deeplabcut.create_new_project('Lizard_test1', 'RW',
                                            ['content/0604_test/0723_01.MP4', 'content/0604_test/0723_02.MP4','content/0604_test/0723_03.MP4'],
                                            working_directory='/content/project',
                                            copy_videos=True)
```

```
deeplabcut.extract_frames(config_path, mode='automatic',
                          algo='kmeans', crop=False, userfeedback=False)
```

However, the colab don't support GUI labeling and I have to move this step to my local computer.

Documentation

The general route for creating project and data training is as below pipeline:

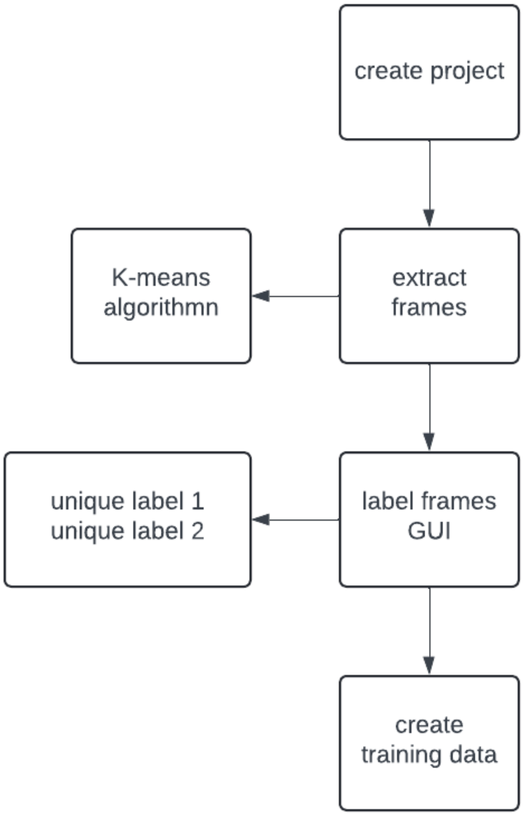
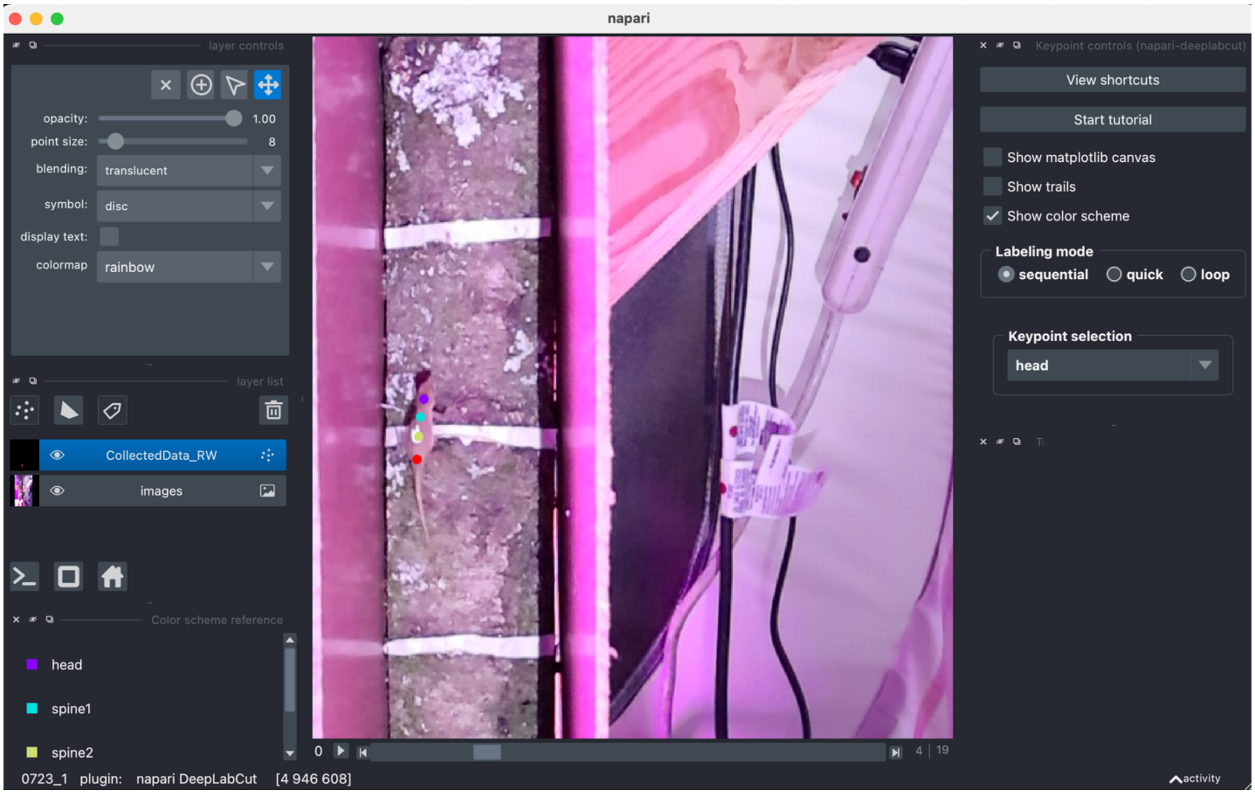


Figure 1: DeepLabCut project pipeline part

The DeepLabCut project creation workflow encompasses three primary phases: project initialization, frame extraction, and data annotation. Project setup involves defining parameters and incorporating video sources. Frame extraction utilizes K-means clustering to select diverse, representative frames. Subsequently, the graphical user interface facilitates precise labeling of anatomical landmarks or features of interest, establishing the ground truth for model training.

For frames labeling, it is better set up in GUI selection, here is one example:



opacity: 1.00
point size: 8
blending: translucent
symbol: disc
display text: ☐
colormap: rainbow

CollectedData_RW
images

head
spine1
spine2

☐ Show matplotlib canvas
☐ Show trails
☑ Show color scheme

Labeling mode
● sequential ○ quick ○ loop

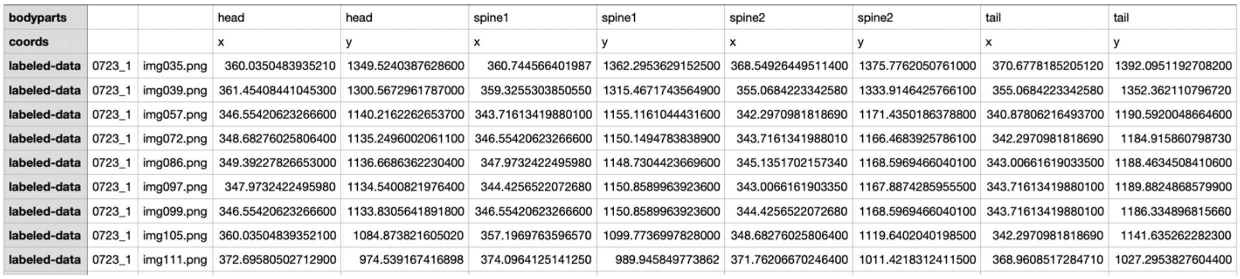
Keypoint selection
head

0 4 | 19

0723_1 plugin: napari DeepLabCut [4 946 608]

Figure 2: frames label of lizard video example (up) and zoomed in lizard labels snapshot

Since the labeling part only involves GUI labeling, it is better to create manually. The body part has been set as head, spine1, spine2 and tail. The labeling output format is exported as a csv file with X and Y coordinates shown:



bodyparts			head	head	spine1	spine1	spine2	spine2	tail	tail
coords			x	y	x	y	x	y	x	y
labeled-data	0723_1	img035.png	360.0350483935210	1349.5240387628600	360.744566401987	1362.2953629152500	368.54926449511400	1375.7762050761000	370.6778185205120	1392.0951192708200
labeled-data	0723_1	img039.png	361.45408441045300	1300.5672961787000	359.3255303850550	1315.4671743564900	355.0684223342580	1333.9146425766100	355.0684223342580	1352.362110796720
labeled-data	0723_1	img057.png	346.55420623266600	1140.2162262653700	343.71613419880100	1155.1161044431600	342.2970981818690	1171.4350186378800	340.87806216493700	1190.5920048664600
labeled-data	0723_1	img072.png	348.68276025806400	1135.2496002061100	346.55420623266600	1150.1494783838900	343.7161341988010	1166.4683925786100	342.2970981818690	1184.915860798730
labeled-data	0723_1	img086.png	349.39227826653000	1136.6686362230400	347.9732422495980	1148.7304423669600	345.1351702157340	1168.5969466040100	343.00661619033500	1188.4634508410600
labeled-data	0723_1	img097.png	347.9732422495980	1134.5400821976400	344.4256522072680	1150.8589963923600	343.0066161903350	1167.8874285955500	343.71613419880100	1189.8824868579900
labeled-data	0723_1	img099.png	346.55420623266600	1133.8305641891800	346.55420623266600	1150.8589963923600	344.4256522072680	1168.5969466040100	343.71613419880100	1186.334896815660
labeled-data	0723_1	img105.png	360.03504839352100	1084.873821605020	357.1969763596570	1099.7736997828000	348.68276025806400	1119.6402040198500	342.2970981818690	1141.635262282300
labeled-data	0723_1	img111.png	372.69580502712900	974.539167416898	374.0964125141250	989.945849773862	371.76206670246400	1011.4218312411500	368.9608517284710	1027.2953827604400

The above file has been uploaded to my github site: https://github.com/RuiqingW20/HAAG_Research-

Results Visualization

The labeled X and Y coordinates has been visualized below:

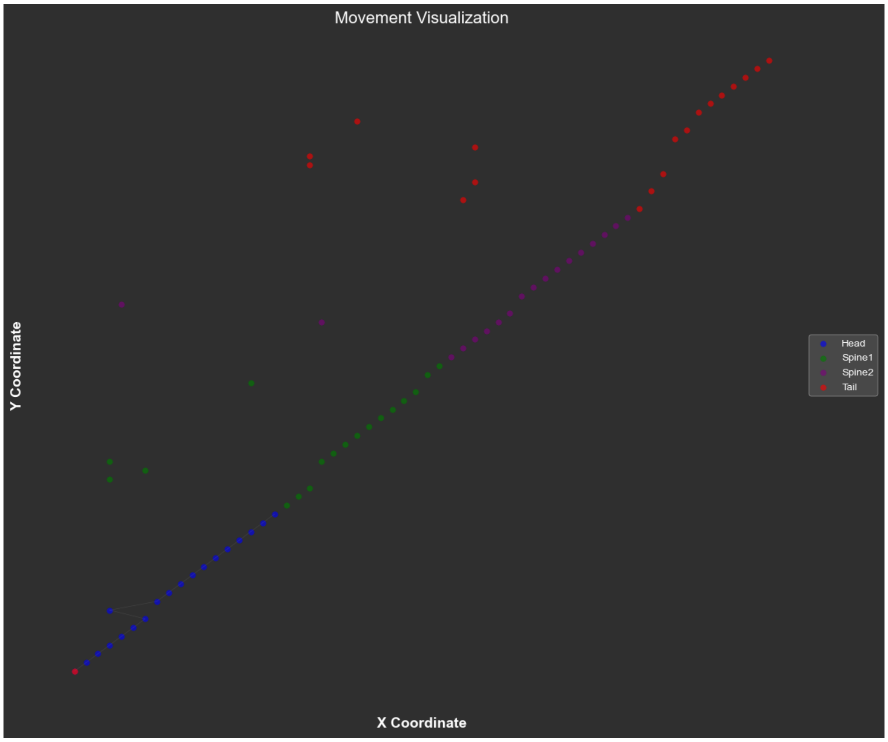


Figure 3: movements visualization

Since the example data is on one track, the Y has almost linear relationship with Y when the lizard move up on the track, which fit the actual movements

Proof of Work:

Please check my github site: https://github.com/RuiqingW20/HAAG_Research-

Below is my local file directory for project set-up:

```
config.yaml                Sep 10, 2024 at 9:37 PM
dlc-models                 Sep 10, 2024 at 12:51 AM
dlc-models-pytorch         Sep 10, 2024 at 10:35 PM
labeled-data               Today at 2:07 AM
training-datasets          Sep 10, 2024 at 10:15 PM
videos                     Sep 10, 2024 at 12:51 AM
```

Next Week's Proposal

1. Meet with Dr. Stroud and discuss about further methods and recourses
2. Diving deeper on videos labeling and data training models
3. Check my PACE allocation and evaluate performance

Week 4 Document Submission

Lizard X-RAY Landmark Group

Mercedes Quintana

Abstracts:

URL: https://arxiv.org/pdf/1703.06870

Mask R-CNN

We present a conceptually simple, flexible, and general framework for object instance segmentation. Our approach efficiently detects objects in an image while simultaneously generating a high-quality segmentation mask for each instance. The method, called Mask R-CNN, extends Faster R-CNN by adding a branch for predicting an object mask in parallel with the existing branch for bounding box recognition. Mask R-CNN is simple to train and adds only a small overhead to Faster R-CNN, running at 5 fps. Moreover, Mask R-CNN is easy to generalize to other tasks, e.g., allowing us to estimate human poses in the same framework. We show top results in all three tracks of the COCO suite of challenges, including instance segmentation, boundingbox object detection, and person keypoint detection. Without bells and whistles, Mask R-CNN outperforms all existing, single-model entries on every task, including the COCO 2016 challenge winners. We hope our simple and effective approach will serve as a solid baseline and help ease future research in instance-level recognition. Code has been made available at: https://github.com/ facebookresearch/Detectron.

Summary: Mask R-CNN is a newly released model from Meta that is the fastest and most accurate model found to date.

Scripts and Code Blocks:

I finished the other visualizations in landmark_skew.py and created a script to run a grid search for the ml-morph toolbox. I also created the website for my subgroup and filled in the team profile.

Found in landmark_skew.py:

Read in test tps data for model and ground truth -> calculate common feature on each X-Ray for a conversion factor to millimeters -> calculate difference between ground truth and model output -> convert to millimeters -> find displacement from landmark -> find direction of error -> estimate a kernel density function of error -> display 3 graphs with matplotlib (kernel density function, histogram of length error and rose plot of angle error.

Documentation:

Landmark_skew.py:

1. Read in data (found in XRAY Lizard Github) for ground and model output
2. Choose a specific landmark to visualize
3. Find the differences between the ground truth and the model output in pixels.
4. Find the difference angles of error
5. Convert and store millimeter conversion using each individual staple.
6. Estimate kernel density function
7. Display with matplotlib

Shape_trainer_grid_search.py:

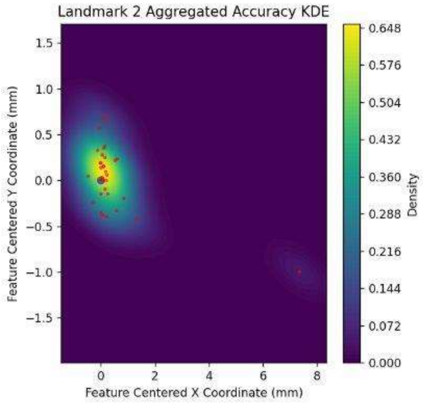
1. Fill out param_grid with desired values for each hyperparameter
2. Use ml-morph to run grid search
3. Look in a newly created performance folder to find a csv with the hyperparameter and the average pixel deviation as well as the model used

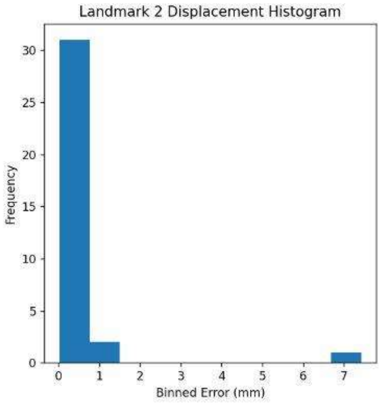
Script Validation:

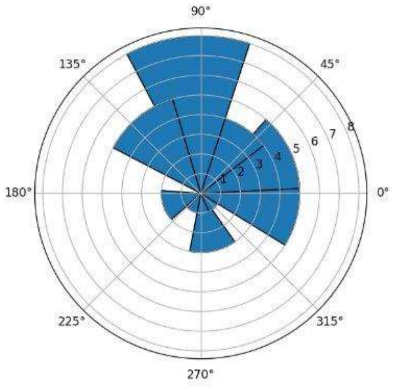
I have no validation steps now.

Results Visualization / Proof of Work:

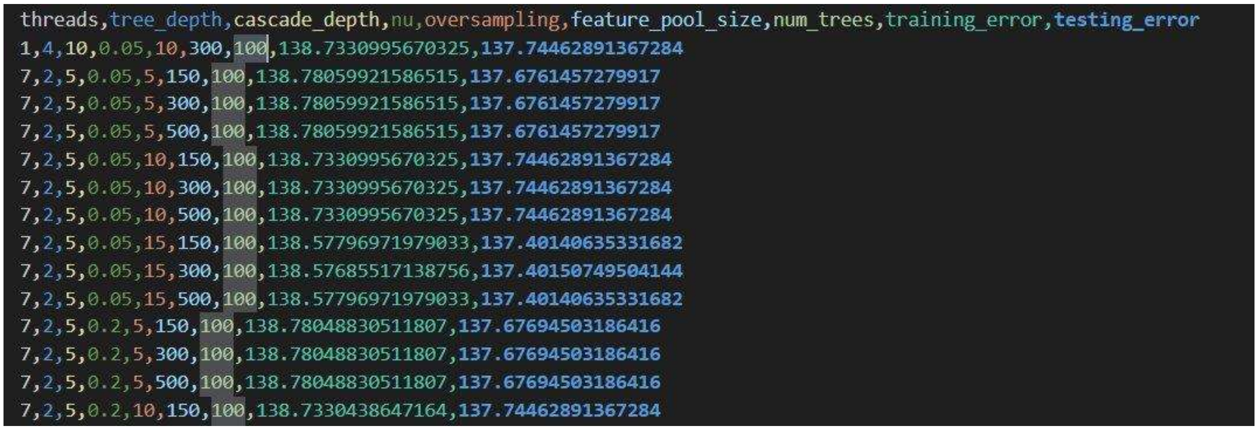
This is the error visualized from Landmark 2.







These are some outputs from the grid search I ran in ml-morph.



```
threads,tree_depth,cascade_depth,nu,oversampling,feature_pool_size,num_trees,training_error,testing_error
1,4,10,0.05,10,300,100,138.7330995670325,137.74462891367284
7,2,5,0.05,5,150,100,138.78059921586515,137.6761457279917
7,2,5,0.05,5,300,100,138.78059921586515,137.6761457279917
7,2,5,0.05,5,500,100,138.78059921586515,137.6761457279917
7,2,5,0.05,10,150,100,138.7330995670325,137.74462891367284
7,2,5,0.05,10,300,100,138.7330995670325,137.74462891367284
7,2,5,0.05,10,500,100,138.7330995670325,137.74462891367284
7,2,5,0.05,15,150,100,138.57796971979033,137.40140635331682
7,2,5,0.05,15,300,100,138.57685517138756,137.40150749504144
7,2,5,0.05,15,500,100,138.57796971979033,137.40140635331682
7,2,5,0.2,5,150,100,138.78048830511807,137.67694503186416
7,2,5,0.2,5,300,100,138.78048830511807,137.67694503186416
7,2,5,0.2,5,500,100,138.78048830511807,137.67694503186416
7,2,5,0.2,10,150,100,138.7330438647164,137.74462891367284
```

Next Week Proposal:

Next week I plan to try the mask R-CNN model with the lizard data. I also plan to work with ml-morph and pace to do a more thorough grid search. I think finding out what Dr. Porto thinks about our data and his toolbox will be really helpful. I also plan to rework the website to have project descriptions.