Week 6 Document Submission

Jacob Dallaire

September 27, 2024

1. Paper

K. Dong, C. Zhou, Y. Ruan and Y. Li, "MobileNetV2 Model for Image Classification," 2020 2nd International Conference on Information Technology and Computer Application (ITCA), Guangzhou, China, 2020, pp. 476-480, doi: 10.1109/ITCA52113.2020.00106.

SUMMARY

The application of machine learning, particularly convolutional neural networks (CNNs), has demonstrated significant potential in image classification tasks, especially in the context of disease verification and face recognition. The study highlights the advantages of pre-trained models, such as MobileNetV1 and MobileNetV2, which leverage depthwise separable convolutions to optimize computational efficiency while maintaining high accuracy rates. By comparing these models against newly developed architectures, the research underscores the importance of model selection and training data size in enhancing performance metrics, thereby contributing to advancements in the field of image classification.

2. Scripts

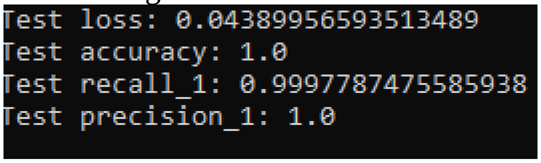
No additional scripts were written just training and evaluation of the model.

3. Documentation

I retrained my model with testing dataset aside. The accuracy, recall, and precision of the model after evaluation on the testing data not used for training were exceptionally high which I found a bit suspicious.

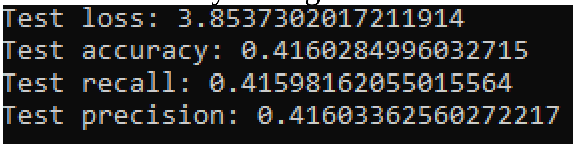
I curated a secondary data set using the non-open licence research grade images from I naturalist as well. These accuracy results were much closer to what would be expected. With a random guess accuracy of 20% the 41.6% result from the testing set was still a significant improvement but not yet reliable.

With testing set:



```
Test loss: 0.04389956593513489
Test accuracy: 1.0
Test recall_1: 0.9997787475585938
Test precision_1: 1.0
```

With a secondary testing set:



```
Test loss: 3.8537302017211914
Test accuracy: 0.4160284996032715
Test recall: 0.41598162055015564
Test precision: 0.41603362560272217
```

4. Next Weeks Proposal

I will continue to investigate the model’s metrics and see if it is possible to produce a higher accuracy. Hopefully I can find out why the unseen data gave such a high evaluation. I will write a script to move training and testing data into sperate folders for repeatability.

Weekly Report

Philip Woolley

2024-09-27

Time Log Reponse:

- What Progress did you make in the last week? - Trained Mask2Former segmenation model using 2 scans and presented results.
- What are you planning on working on next? - Continue segmenting training data. Create a script for postprocessing model results
- Is there anything blocking you? - None at this time

1 Abstract

Abstract

Background

Image segmentation groups pixels with different seman tics, e.g., category or instance membership. Each choice of semantics defines a task. While only the semantics of each task differ, current research focuses on designing spe cialized architectures for each task. We present Masked attention Mask Transformer (Mask2Former), a new archi tecture capable of addressing any image segmentation task (panoptic, instance or semantic). Its key components in clude masked attention, which extracts localized features by constraining cross-attention within predicted mask regions. In addition to reducing the research effort by at least three times, it outperforms the best specialized architectures by a significant margin on four popular datasets. Most no tably, Mask2Former sets a new state-of-the-art for panoptic segmentation (57.8 PQ on COCO), instance segmentation (50.1 APonCOCO)andsemanticsegmentation(57.7mIoU on ADE20K).

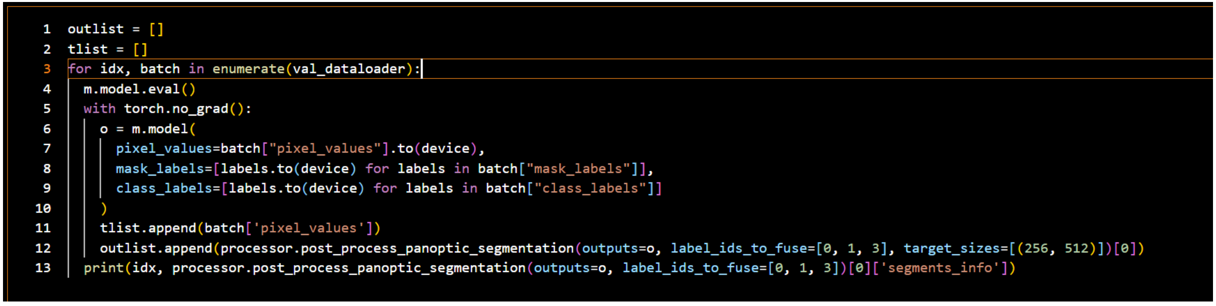
Summary This wiedly reviewed paper from Facebook's AI Team proposes the Mask2Former model architecture, based on the earlier MaskFormer architecture. Mask2Former is intended to be a unified architecture for any image segmentation task, whether semantic segmentation, instance segmentation, or panoptic segmentation. The model differs from MaskFormer in that it uses masked attention, which limits the attention mechanism to the immediate area making up a proposed segment. This leads to significant speedup and performance improvements, as the model does not need to consider information from parts of the image far away from the current segment under consideration when making determinations about a segment. The Mask2Former model shows strong performance improvements over MaskFormer and single purpose-built segmentation models on a variety of tasks and datasets. I am planning to use this model as a core for my pipeline, and so the technical details of this paper will be very relevant to my project.

Citation

Cheng, Bowen, et al. "Masked-attention mask transformer for universal image segmentation." Proceedings of the IEEE/CVF conference on computer vision and pattern recognition. 2022.

2 Scripts and Code Blocks

This week, I created the VisualizeModelRestults.ipynb notebook. It is a notebook for reviewing the quality of model outputs and comparing them with ground truth segmentations. It also shows the proper way to use the model postprocessor for panoptic segmentation to generate usable masks. As seen in the below codeblock, for each image all instances of non-tooth categories are fused, so that each image has at most one segmentation for Background, one for lower jaw, and one for other bone. This mimics how segmentation is performed and expected in slicer. The category IDs for this dataset are [0: background, 1: lower jaw, 2: lower teeth, 3: all other bone]



```
outlist = []
tlist = []
for idx, batch in enumerate(val_dataloader):
    m.model.eval()
    with torch.no_grad():
        o = m.model(
            pixel_values=batch["pixel_values"].to(device),
            mask_labels=[labels.to(device) for labels in batch["mask_labels"]],
            class_labels=[labels.to(device) for labels in batch["class_labels"]]
        )
        tlist.append(batch['pixel_values'])
        outlist.append(processor.post_process_panoptic_segmentation(outputs=o, label_ids_to_fuse=[0, 1, 3], target_sizes=[(256, 512)])[0])
    print(idx, processor.post_process_panoptic_segmentation(outputs=o, label_ids_to_fuse=[0, 1, 3])[0]['segments_info'])
```

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. T

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)

The code in DataPreprocess.ipynb is a proof of concept that will be rewritten into a .py script, full validation to come at that time.

5 Results Visualization

Here are the model results for the first trained model version.

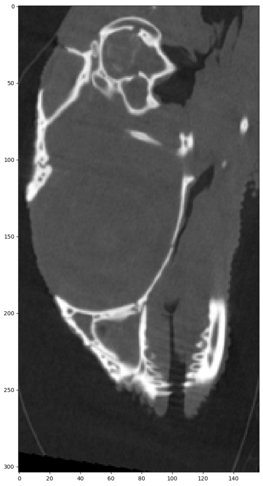


Figure 1: original image

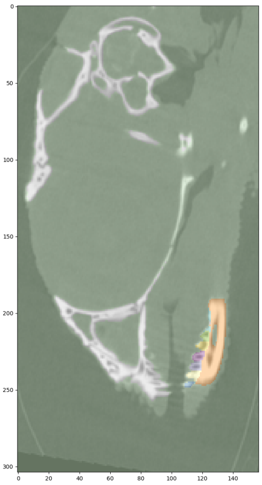


Figure 2: segmentation mask output by model. Each color represents a different instance.

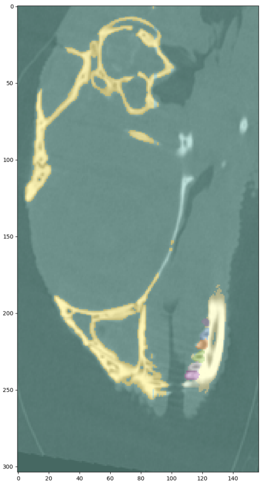


Figure 3: segmentation mask created previously by me and used as ground truth. Each color represents a different instance.

6 Proof of Work

See Code Blocks and Results Visualization sections

7 Next Week's Proposal

- Continue segmenting training data for ML panoptic segmentation model
- Develop script for postprocessing segmentation results
- Keep up with any required blog posts for webmaster role

Week6 report

Ruiqing Wang | CiChild CV team

- What progress did you make in the last week?

1. Successfully set up python and slurm test code in project running on PACE
2. Set up local environment at my local computer which has GPU
3. Run through the project again and confirm the problems
4. Check my PACE allocation and evaluate performance
5. Review papers on DeepLabCut
6. Help assembling paper report submissions and address submission situation.

- What are you planning on working on next?

1. Try google colab and check if this environment works
2. Meet with Cichild CV team to discuss current progress

- Is anything blocking you from getting work done?

The PACE environmental set up I need confirm with school administrator

Paper abstract

Paper: Automatic detection of bumble foot in cage free hens using computer vision technologies: https://doi.org/10.1016/j.psj.2024.103780

Abstract: The study focuses on using computer vision and artificial intelligence to automatically detect bumblefoot, a painful bacterial infection, in cage-free hens. Bumblefoot is common in hens raised on litter floors and can affect their movement and access to food and water. The researchers developed and tested several deep learning models, finding that the YOLOv5m-BFD model performed best in accurately identifying hens with this condition, which could help improve animal welfare in egg production systems. The study aims to create and test a specific YOLOv5 model for diagnosing bumblefoot in cage-free hens and to compare its performance with other versions of the YOLO model.

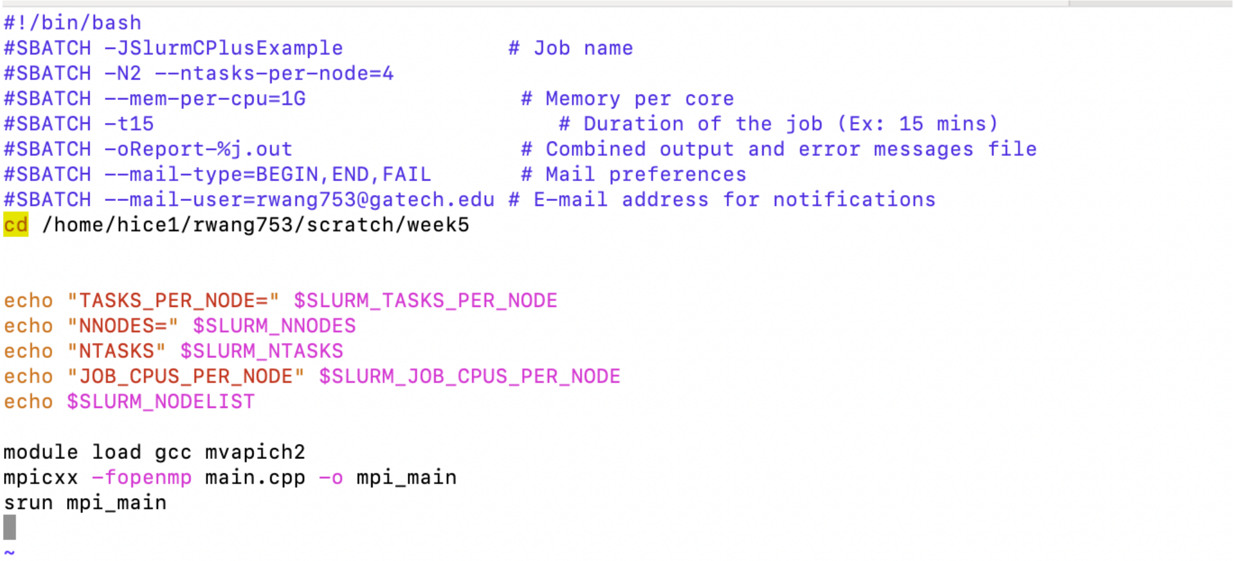
Methodology: The total bumblefoot detection (BFD) datasets were systematically evaluated against three variants of the YOLOv5 object detection models—YOLOv5s, YOLOv5m, and YOLOv5x. The architecture of the YOLOv5-BFD model integrates an input layer, a backbone for feature extraction, a neck that combines a Feature Pyramid Network (FPN) for multiscale feature extraction and fusion, and a Path Aggregation Network (PANet) for effective feature aggregation across spatial scales. In the YOLOv5-BFD architecture, the neck component employs Feature Pyramid Networks (FPN) and Path Aggregation Networks (PANet) to effectively aggregate image features, facilitating the model's ability to generalize and accurately identify bumblefoot objects across varying sizes and scales. The head of the YOLOv5-BFD retains the convolutional structure from YOLOv3 and YOLOv4, consisting of three convolution layers that enable multiscale predictions for object classification, bounding box localization, and confidence scoring.

Scripts and Code Blocks

This week I finally figured out the problem which prevented from submitting jobs on PACE, and I have successfully checked my current CPU and GPU setting. Here is the reference page: https://gatech.service-now.com/home?id=kb_article_view&sysparm_article=KB0042096

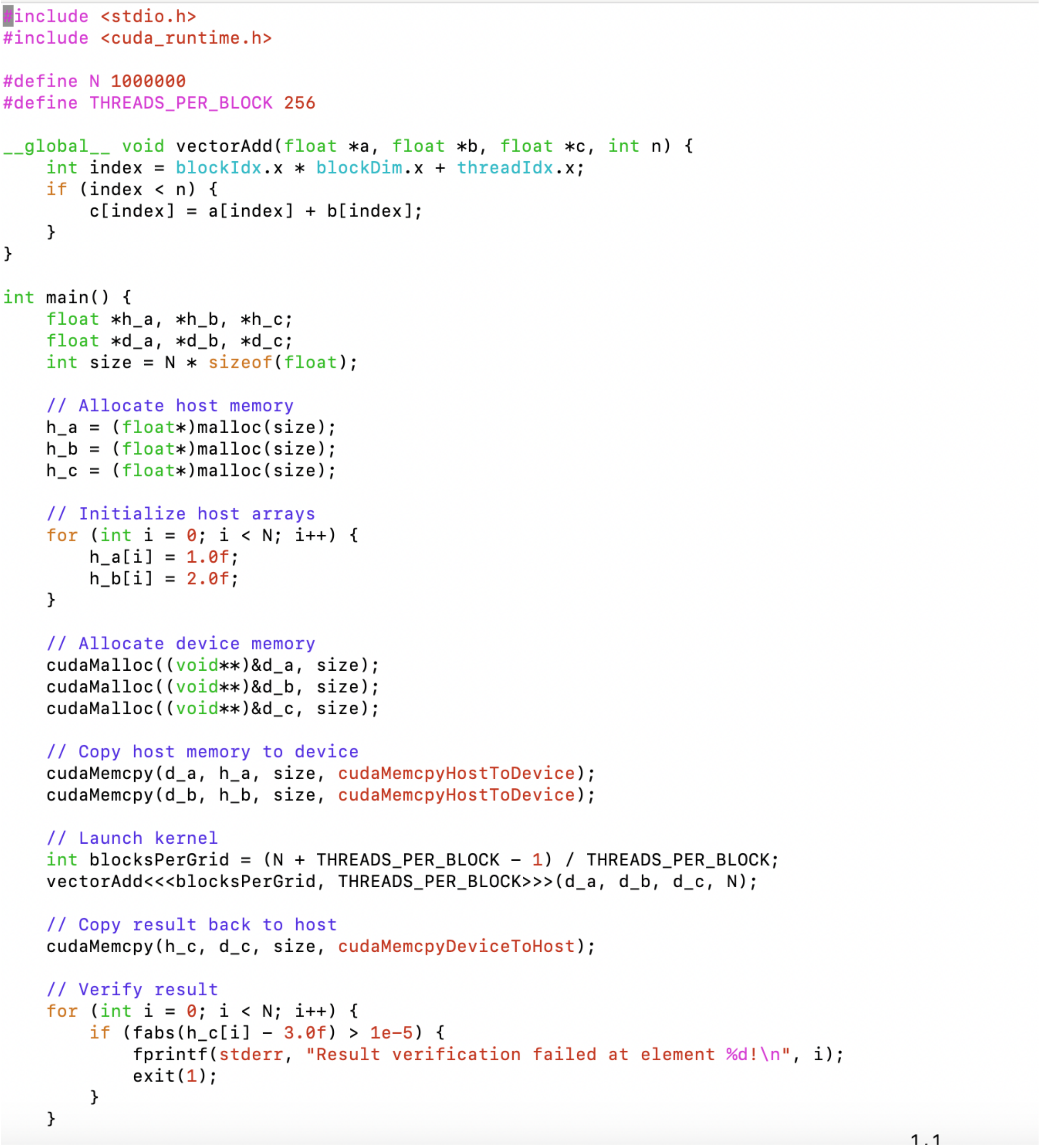
The code I used is from my last week work, here is the link: https://gatech.service-now.com/technology?id=kb_article_view&sysparm_article=KB0042003

The batch code contents as below:



```
#!/bin/bash
#SBATCH -JSlurmCPlusExample             # Job name
#SBATCH -N2 --ntasks-per-node=4
#SBATCH --mem-per-cpu=1G                 # Memory per core
#SBATCH -t15                                # Duration of the job (Ex: 15 mins)
#SBATCH -oReport-%j.out                  # Combined output and error messages file
#SBATCH --mail-type=BEGIN,END,FAIL       # Mail preferences
#SBATCH --mail-user=rwang753@gatech.edu # E-mail address for notifications
cd /home/hice1/rwang753/scratch/week5


echo "TASKS_PER_NODE=" $SLURM_TASKS_PER_NODE
echo "NNODES=" $SLURM_NNODES
echo "NTASKS" $SLURM_NTASKS
echo "JOB_CPUS_PER_NODE" $SLURM_JOB_CPUS_PER_NODE
echo $SLURM_NODELIST

module load gcc mvapich2
mpicxx -fopenmp main.cpp -o mpi_main
srun mpi_main

~
```

The test code was in my git repo, and I also added code in .cu format to check the GPU function: it defines a vector size of 1,000,000 elements and sets up 256 threads per block and creates a CUDA kernel function which adds corresponding elements of two input vectors.



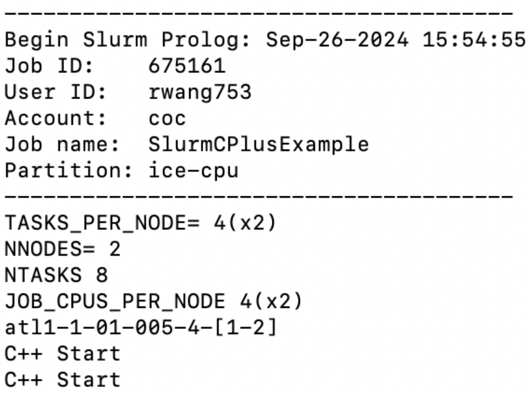
```
#include <stdio.h>
#include <cuda_runtime.h>

#define N 1000000
#define THREADS_PER_BLOCK 256

__global__ void vectorAdd(float *a, float *b, float *c, int n) {
    int index = blockIdx.x * blockDim.x + threadIdx.x;
    if (index < n) {
        c[index] = a[index] + b[index];
    }
}

int main() {
    float *h_a, *h_b, *h_c;
    float *d_a, *d_b, *d_c;
    int size = N * sizeof(float);

    // Allocate host memory
    h_a = (float*)malloc(size);
    h_b = (float*)malloc(size);
    h_c = (float*)malloc(size);

    // Initialize host arrays
    for (int i = 0; i < N; i++) {
        h_a[i] = 1.0f;
        h_b[i] = 2.0f;
    }

    // Allocate device memory
    cudaMalloc((void**)&d_a, size);
    cudaMalloc((void**)&d_b, size);
    cudaMalloc((void**)&d_c, size);

    // Copy host memory to device
    cudaMemcpy(d_a, h_a, size, cudaMemcpyHostToDevice);
    cudaMemcpy(d_b, h_b, size, cudaMemcpyHostToDevice);

    // Launch kernel
    int blocksPerGrid = (N + THREADS_PER_BLOCK - 1) / THREADS_PER_BLOCK;
    vectorAdd<<<blocksPerGrid, THREADS_PER_BLOCK>>>(d_a, d_b, d_c, N);

    // Copy result back to host
    cudaMemcpy(h_c, d_c, size, cudaMemcpyDeviceToHost);

    // Verify result
    for (int i = 0; i < N; i++) {
        if (fabs(h_c[i] - 3.0f) > 1e-5) {
            fprintf(stderr, "Result verification failed at element %d!\n", i);
            exit(1);
        }
    }
```

Documentation

All my current code samples were stored in rwang753/home/hice1/rwang753/scratch
My own github repo: https://github.com/RuiqingW20/HAAG_Research-.git

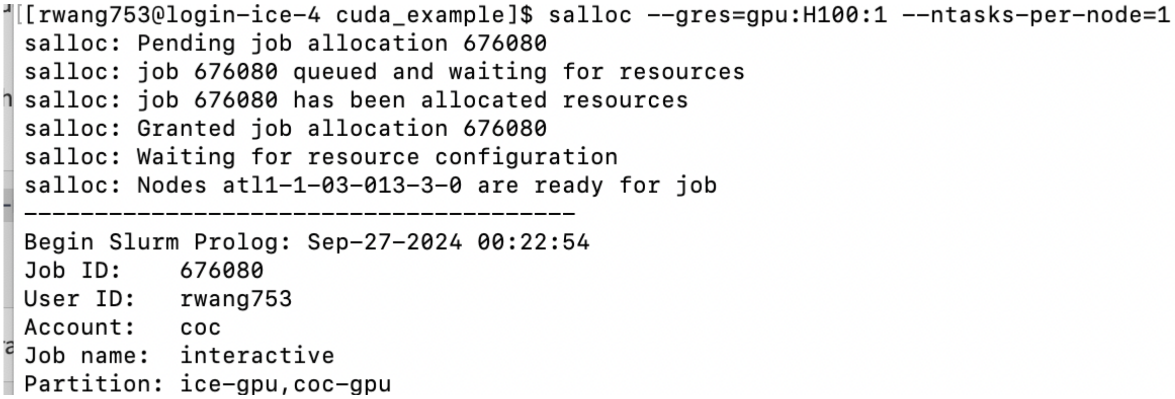
Results Visualization

I tried Batch and interactive mode to submit my job, and both works. For CPU testing, I used MPI to check my requested node and test the function, here is the output:

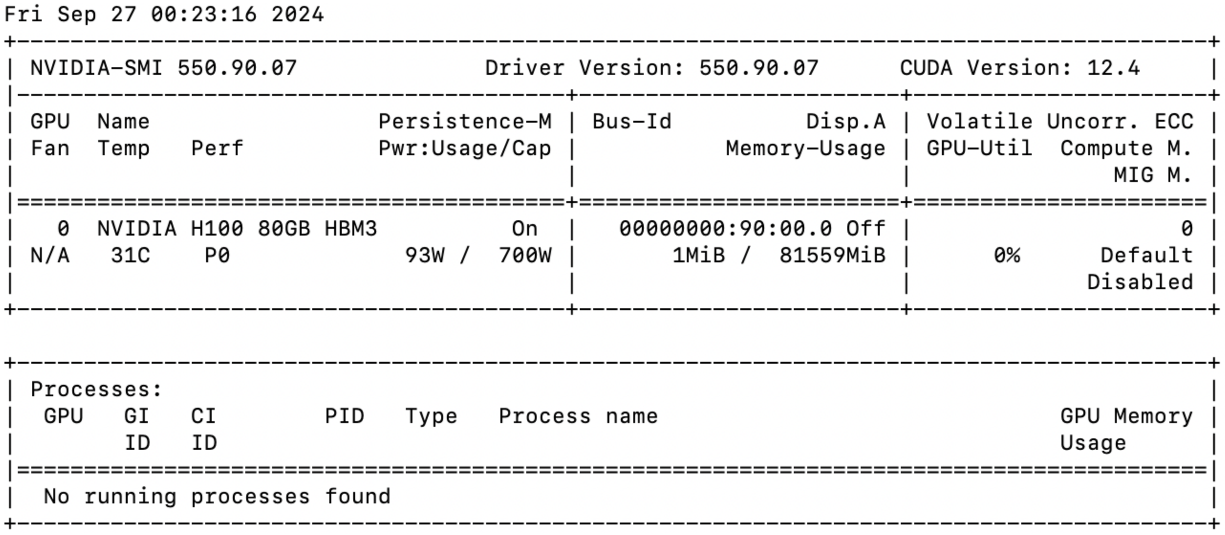


```
---------------------------------------
Begin Slurm Prolog: Sep-26-2024 15:54:55
Job ID:    675161
User ID:   rwang753
Account:   coc
Job name:  SlurmCPlusExample
Partition: ice-cpu
---------------------------------------
TASKS_PER_NODE= 4(x2)
NNODES= 2
NTASKS 8
JOB_CPUS_PER_NODE 4(x2)
atl1-1-01-005-4-[1-2]
C++ Start
C++ Start
```

Besides CPU, I also checked GPU and CUDA usage according to reference, which will be further used in my project.

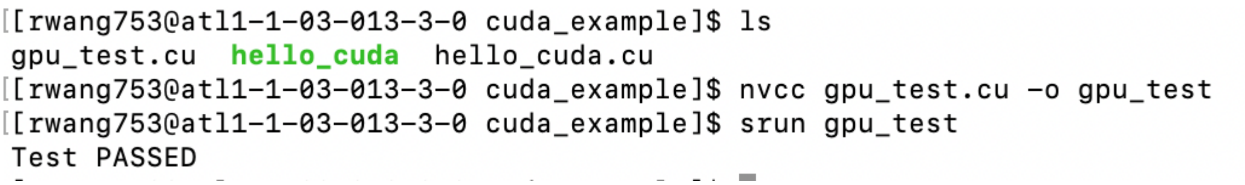


```
[rwang753@login-ice-4 cuda_example]$ salloc --gres=gpu:H100:1 --ntasks-per-node=1
salloc: Pending job allocation 676080
salloc: job 676080 queued and waiting for resources
salloc: job 676080 has been allocated resources
salloc: Granted job allocation 676080
salloc: Waiting for resource configuration
salloc: Nodes atl1-1-03-013-3-0 are ready for job
---------------------------------------
Begin Slurm Prolog: Sep-27-2024 00:22:54
Job ID:    676080
User ID:   rwang753
Account:   coc
Job name:  interactive
Partition: ice-gpu,coc-gpu
```



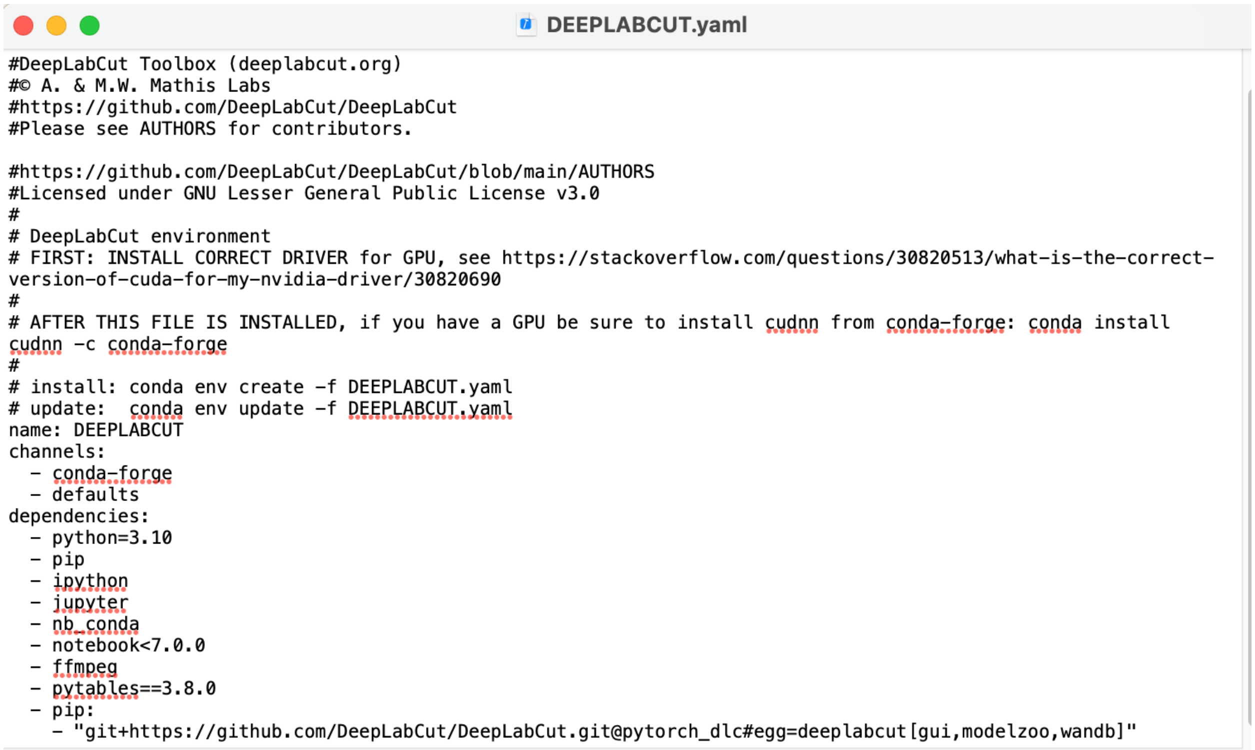
```
Fri Sep 27 00:23:16 2024
+-----------------------------------------------------------------------------------------+
| NVIDIA-SMI 550.90.07              Driver Version: 550.90.07      CUDA Version: 12.4     |
|-----------------------------------------+------------------------+----------------------+
| GPU  Name                 Persistence-M | Bus-Id          Disp.A | Volatile Uncorr. ECC |
| Fan  Temp   Perf          Pwr:Usage/Cap |           Memory-Usage | GPU-Util  Compute M. |
|                                         |                        |               MIG M. |
|=========================================+========================+======================|
|   0  NVIDIA H100 80GB HBM3          On  |   00000000:90:00.0 Off |                    0 |
| N/A   31C    P0             93W /  700W |       1MiB /  81559MiB |      0%      Default |
|                                         |                        |             Disabled |
+-----------------------------------------+------------------------+----------------------+

+-----------------------------------------------------------------------------------------+
| Processes:                                                                              |
|  GPU   GI   CI        PID   Type   Process name                              GPU Memory |
|        ID   ID                                                               Usage      |
|=========================================================================================|
|  No running processes found                                                             |
+-----------------------------------------------------------------------------------------+
```

I requested 1 node on H100 and above is the information of this GPU.
For DeepLabCut usage, the next step will be set up virtual environment in cluster. The software stack link is below: https://gatech.service-now.com/home?id=kb_article_view&sysparm_article=KB0043494
I also run the test of GPU function and it also worked in my allocation:



```
[[rwang753@atl1-1-03-013-3-0 cuda_example]$ ls
gpu_test.cu  hello_cuda  hello_cuda.cu
[[rwang753@atl1-1-03-013-3-0 cuda_example]$ nvcc gpu_test.cu -o gpu_test
[[rwang753@atl1-1-03-013-3-0 cuda_example]$ srun gpu_test
Test PASSED
```

DLC requires Tensorflow as 2.10 above which the cluster already installed, however, for the working environment, here is the yaml file we shall use:



```
#DeepLabCut Toolbox (deeplabcut.org)
#© A. & M.W. Mathis Labs
#https://github.com/DeepLabCut/DeepLabCut
#Please see AUTHORS for contributors.

#https://github.com/DeepLabCut/DeepLabCut/blob/main/AUTHORS
#Licensed under GNU Lesser General Public License v3.0
#
# DeepLabCut environment
# FIRST: INSTALL CORRECT DRIVER for GPU, see https://stackoverflow.com/questions/30820513/what-is-the-correct-
version-of-cuda-for-my-nvidia-driver/30820690
#
# AFTER THIS FILE IS INSTALLED, if you have a GPU be sure to install cudnn from conda-forge: conda install
cudnn -c conda-forge
#
# install: conda env create -f DEEPLABCUT.yaml
# update:  conda env update -f DEEPLABCUT.yaml
name: DEEPLABCUT
channels:
  - conda-forge
  - defaults
dependencies:
  - python=3.10
  - pip
  - ipython
  - jupyter
  - nb_conda
  - notebook<7.0.0
  - ffmpeg
  - pytables==3.8.0
  - pip:
    - "git+https://github.com/DeepLabCut/DeepLabCut.git@pytorch_dlc#egg=deeplabcut[gui,modelzoo,wandb]"
```

The next step will be set up virtual environment and install DLC using this file. I will confirm with school administrator on how to do this.

Proof of Work

My current work was stored at: /home/hice1/rwang753/scratch

Next Week's Proposal

1. Set up environment on PACE
2. Meet with Bree and other team member on current working progress

Week 6 Document Submission

Lizard X-RAY Landmark Group

Mercedes Quintana

What progress did you make in the last week?

- Continued to work on website
- Found a "mean" shape among images
- Made it possible to save all visualizations
- Set up grid search on PACE

What are you planning on working on next?

- Find the best training data for the model
- Grid search to find the best hyperparameters
- 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=10655207

Generalizable Face Landmarking Guided by Conditional Face Warping

As a significant step for human face modeling, editing, and generation, face landmarking aims at extracting facial keypoints from images. A generalizable face landmarker is required in practice because real-world facial images, e.g., the avatars in animations and games, are often stylized in various ways. However, achieving generalizable face landmarking is challenging due to the diversity of facial styles and the scarcity of labeled stylized faces. In this study, we propose a simple but effective paradigm to learn a generalizable face landmarker based on labeled real human faces and unlabeled stylized faces. Our method learns the face landmarker as the key module of a conditional face warper. Given a pair of real and stylized facial images, the conditional face warper predicts a warping field from the real face to the stylized one, in which the face landmarker predicts the ending points of the warping field and provides us with high-quality pseudo landmarks for the corresponding stylized facial images. Applying an alternating optimization strategy, we learn the face landmarker to minimize i) the discrepancy between the stylized faces and the warped real ones and ii) the prediction errors of both real and pseudo landmarks. Experiments on various datasets show that our method outperforms existing state-of-the-art domain adaptation methods in face landmarking tasks, leading to a face landmarker with better generalizability. Code is available at https://plustwo0.github.io/project-face-landmarker.

Summary: The paper aims to expand the style of face that can be landmarked such as those from cartoons. In order to do this, they use an initial step of a conditional face warper. This leads to a model with better generalizability.

Scripts and Code Blocks:

This week I remade visual_individual_performance.py to save all images to a folder.

Found in visual_individual_performance.py:

Read in test xml for model and ground truth -> Find specific lizard -> Plot over image

Documentation:

visual_individual_performance.py:

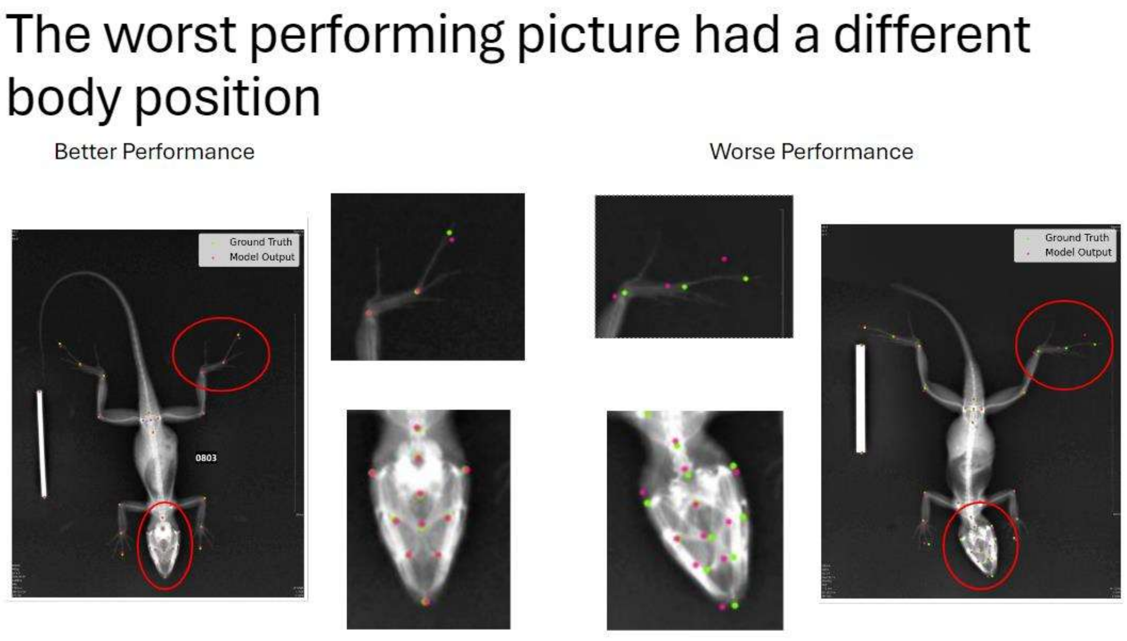
1. Read in test.xml and output.xml with number of image to visualize
2. Code will read in and display lizard image with output and ground truth

Script Validation:

I have no validation steps now.

Results Visualization / Proof of Work:

This week I looked at the test images to find what images were performing well. I noticed that a straight body and straight from feet were the ones that the model was able to classify the best.



The worst performing picture had a different body position

Better Performance

Worse Performance

Next Week Proposal:

I plan to keep working on the website to keep it updated with the new meetings and work done. I plan to talk with Jon to find out what he needs from the model and rework the training data to reflect the type of X-Rays he plans to take. Then hopefully plan to rework the model to incorporate the new data and get better results!