EE/CprE/SE 491 - sddec23-10

Developing a Deep Learning Model to Automatically Detect Microscale Objects in Images and Videos

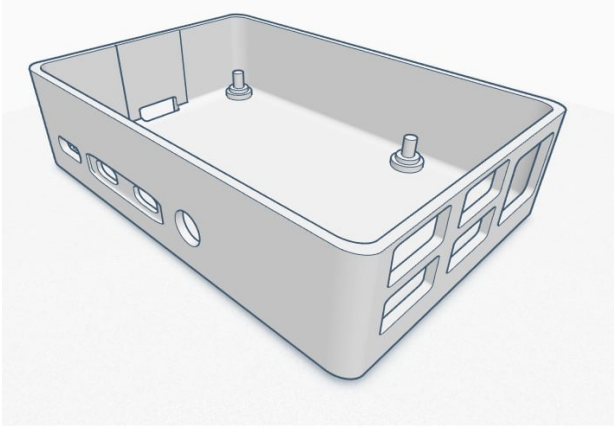
Week 6-8 Report

10/25/2023 – 11/8/2023
Client: Professor. Santosh Pandey
Group number: 10

Team Members:

Katherine Moretina
Ethan Baranowski
Chris Cannon
Matthew Kim

Hardware and GUI

- Flashed new operating system that is compatible with hardware and getting Colab to the Raspberry Pi
- Looked into methods of creating extendable arms to get closer to soybean roots
- Went to hardware store to find materials for tray table and extendable arms
- Found a Raspberry Pi case to edit in SolidWorks
- 

Colab to Raspberry pi

- Pytorch works on the 64bit and Bullseye version. Therefore, Legacy version was needed, and it was downloaded by Raspbin. Also aarch was setted for Raspberry OS.
- Pytorch, Torchvision was downloaded order to fulfill prerequisites of Detectron 2.
- Detectron 2 was successfully downloaded.

Training Faster-RCNN Model

- Fixed all errors with importing data to Google Colab
- Researched and implemented correct configuration values for a practical dataset
    - 1000-10000 iterations (refers to how well the model is fitted to the dataset)
- Achieved a baseline trained model of 300 and 1000 iterations.
- Advanced baseline trained model from 1000 to 5000 iterations. Results are significantly better but do not converge yet.
- Google Colab Pro being used to overcome associated technical challenges.
- Additionally, the training sessions will be held on a lab computer to enable 24hr training sessions. The new lab computer was recently received.

Individual Contributions

Member	Tasks Completed	Hours This Week	Total Hours
Katherine Moretina	• Designed our physical image capturing setup • Found resources on 3d modeling	9	43
Matthew Kim	• Got new SD card from ETG • Installed Raspberry pi 64bit Bullseye version into SD card • Installed Pythorch 2.0.0 • Installed Torchvision 16 • Installed Detectron 2 • Installed OpenCV	12	37

Chris Cannon	• Researched and implemented checkpoint loading. • Wrote a local proof-of-concept program to run the trained model • Began researching anchor box sizes to improve training results	10	38
Ethan Baranowski	• Learned iterative way of training from checkpoints. Now will train the algorithm from every 5000 iteration checkpoints until convergence. • Looked at optimization of training to ensure it is being done correctly.	15	56

Plans for Coming Week

- Download SolidWorks or start 3d modeling on SolidWorks on Iowa State Computers
- Download code onto Raspberry Pi to explore more capabilities
- Finish script to transform LabelMe data into the correct format for Detectron
- Import Detectron 2 to the Raspberry Pi
- Gain access to Google Colab Pro and optimize the training of the model.
- Integrate the baseline model into the Raspberry Pi.
- Implement training on new local lmachine to enhance training accessibility.
- Iteratively train the algorithm with checkpoints every 5000 iterations. Comparative analysis may be returned to later.