PSEUDO-AMIGA

Phenotyping System for Efficient User-centered Detection On AMIGA

The Project

Prepare a phenotyping system for the AMIGA robot.

Goals:

- Prepare multispectral camera for use on robot
- Develop a pipeline for autonomous capture
- Create a dataset of useful images from the multispectral camera
- Create plant / weed segmentation system

Biggest Initial Problems

Preparing the multispectral camera would be difficult

1. The AMIGA uses a Planet IGS-604HPT-M12 POE connection which is not compatible directly with the MicaSense Rededge-P
2. The skyport kit which the camera was purchased with does not include the POE conversion board which is not sold separately from the camera.
3. The camera is meant for use on a drone and is prone to overheating if in stagnant air
4. The camera is very light sensitive and needs to be calibrated with a reflective panel before use

Mapping the MicaSense to POE

PWR: 1-2 for power, 3-4 for ground

COMM: Only relevant pins are 5-9, being Ethernet TX N/P, Ethernet RX N/P, and the ground.

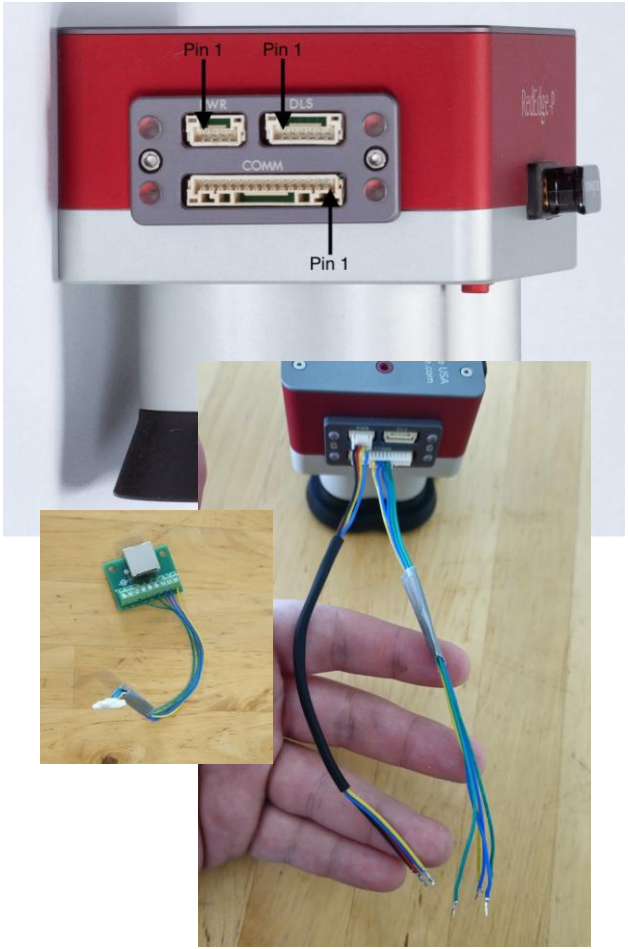
Pin 5 → RJ45 Pin 6

Pin 6 → RJ45 Pin 3

Pin 7 → RJ45 Pin 8

Pin 8 → RJ45 Pin 7

Pin 9 → RJ45 shield

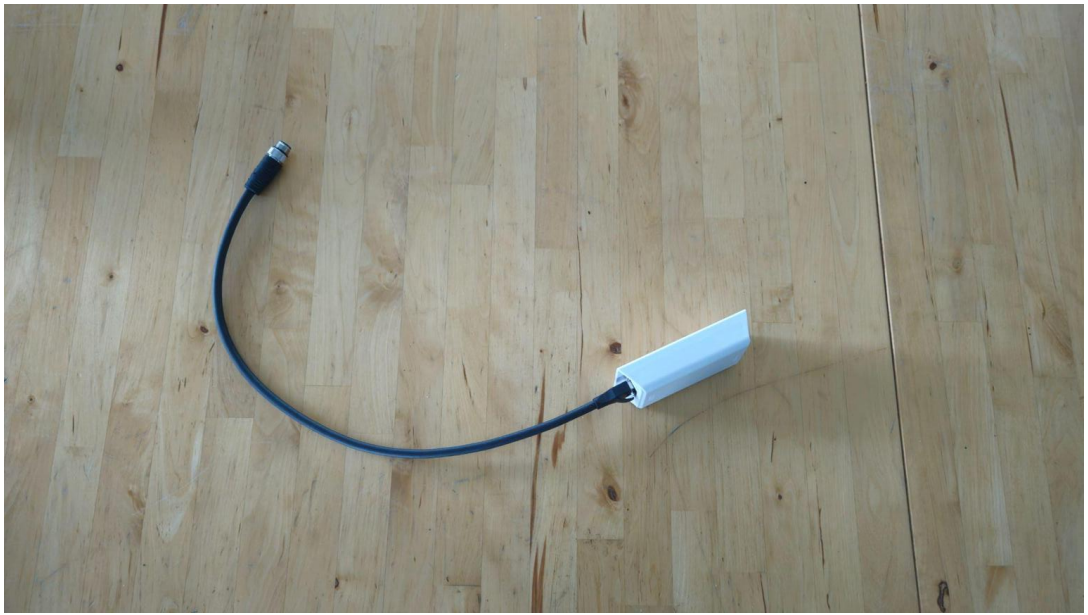


The Hardware

A Coded M12 8 Position RJ45 Ethernet Cable

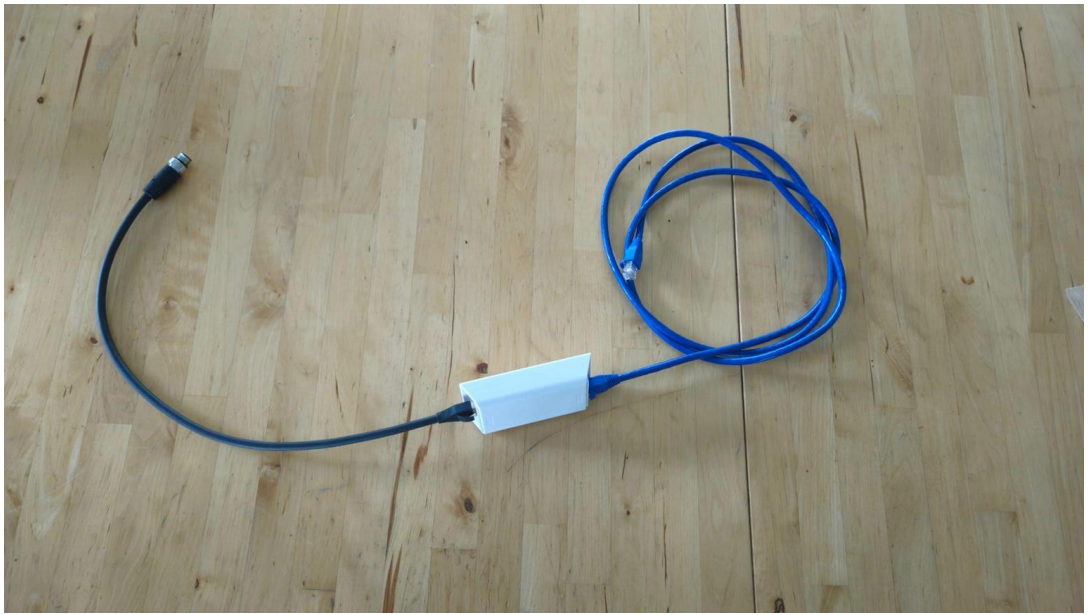
The Hardware

Connected to a POE Texas GAT-12V25W Splitter



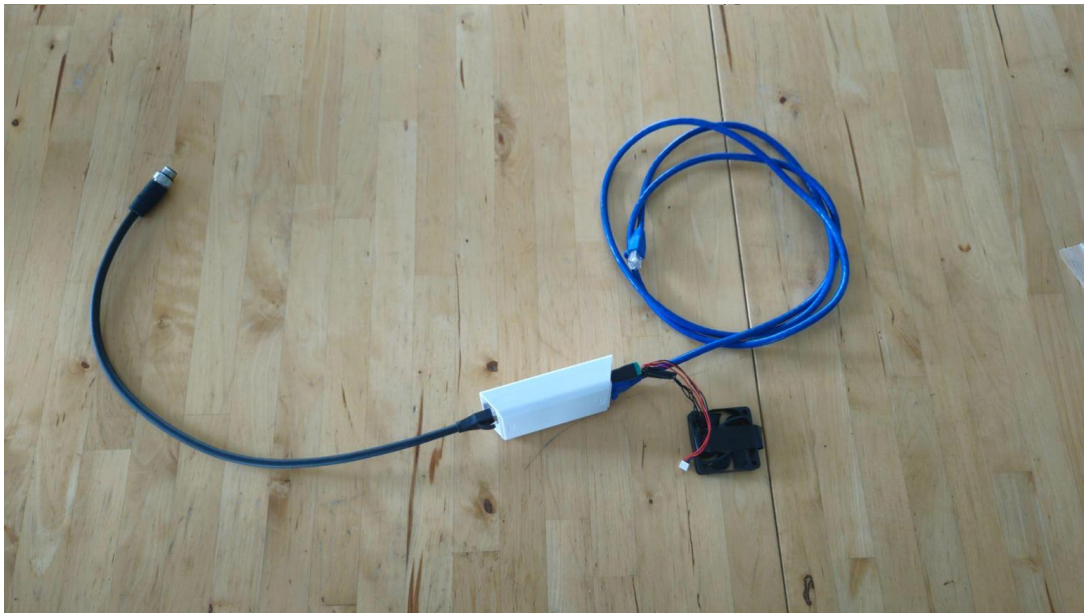
The Hardware

Splitting into Ethernet



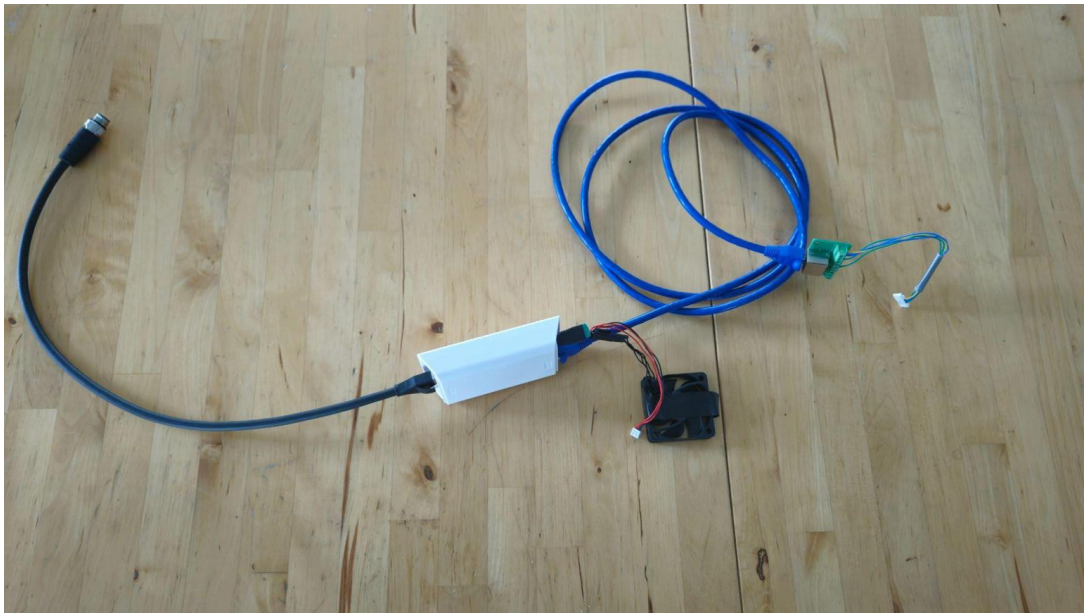
The Hardware

And power (5.5mm/2.1mm barrel with a fan and 4 pin connector in parallel)



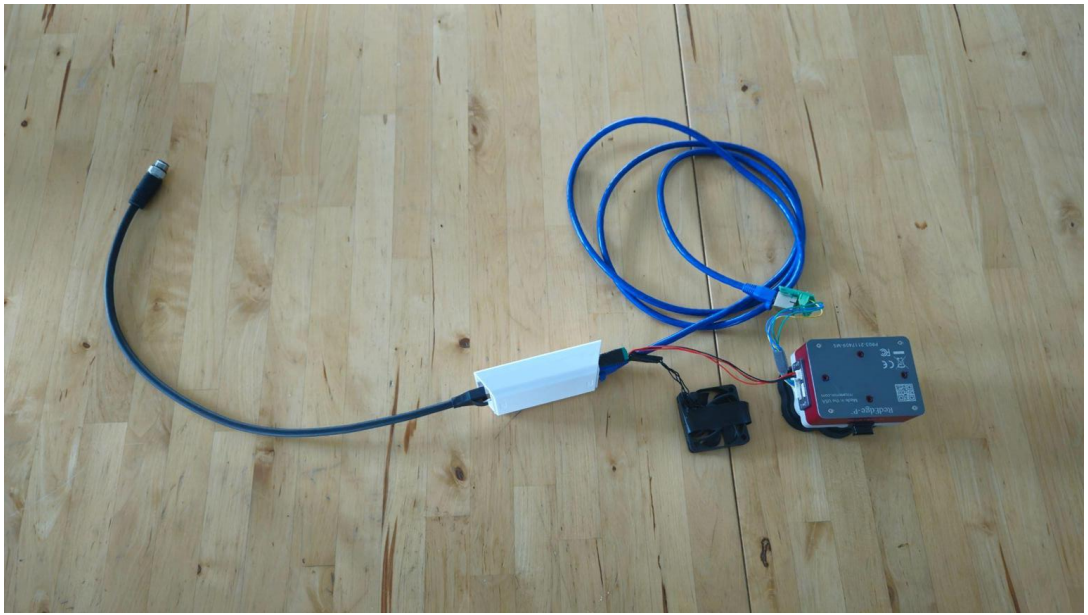
The Hardware

The Ethernet connects into a RJ45 Breakout board, converting it to the 15 pin connection



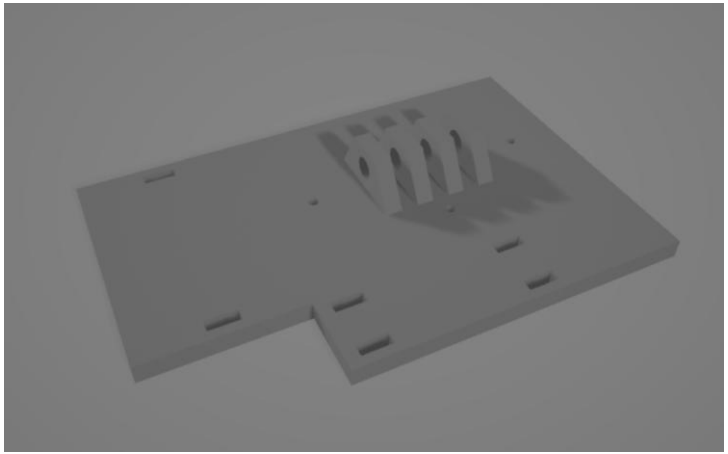
The Hardware

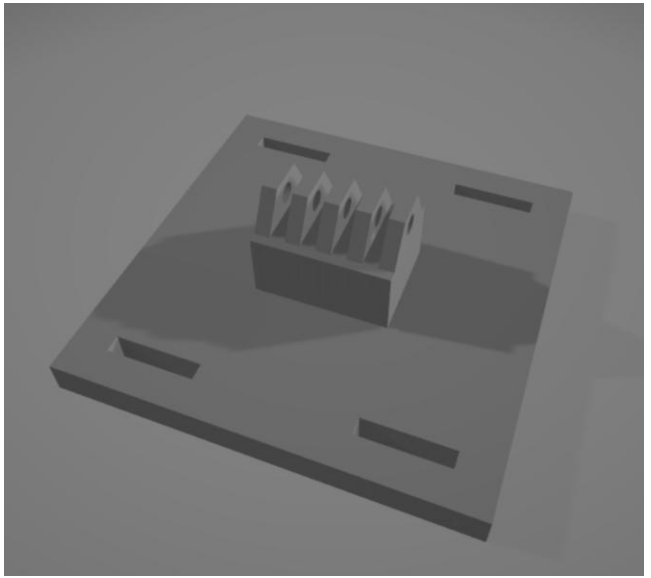
Which finally connects into the side of the camera



The Hardware

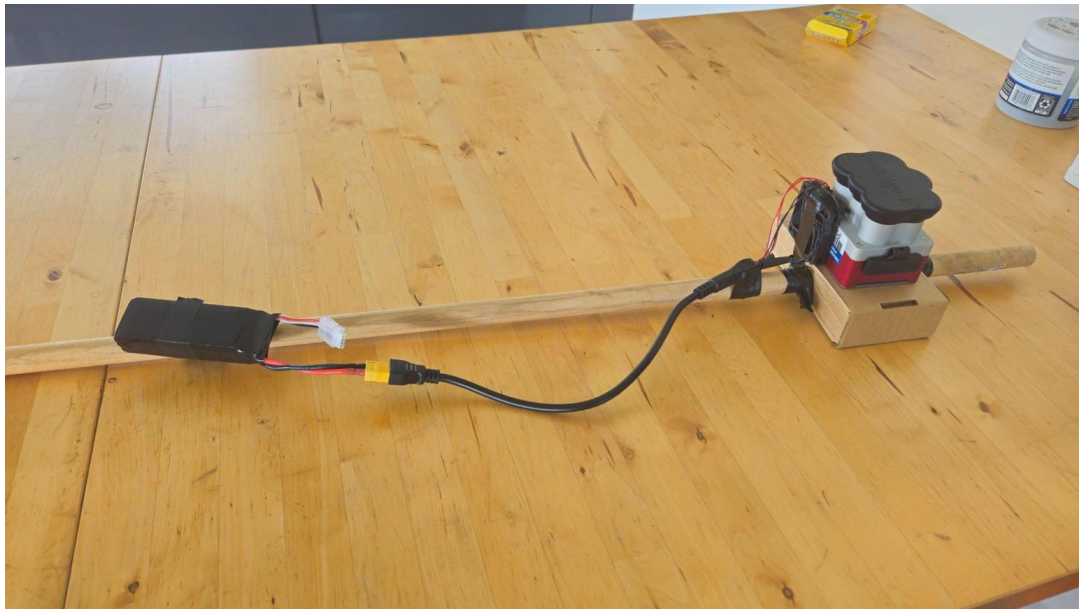
A mount was also CADed and prepared for printing but I wasn’t able to get access to the 3D printer in the highbay in time





The Hardware

Instead the camera was screwed into a cardboard box and attached to a long wooden stick held at shoulder height (approximately the same as the robot)



The Hardware - Overheating

The camera was tested by running it continuously streaming in direct sunlight for 30 minutes, after fifteen it was noticeably hot to the touch and was turned off.

A small PC fan was installed and positioned by the exhaust, fixing the issue completely.

Camera Calibration

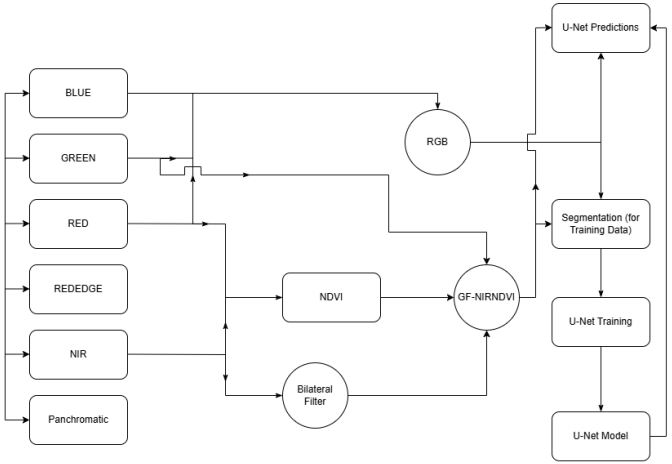
The calibration issue can be easily resolved by bringing the panel closer to the camera, it just needs to be in light.

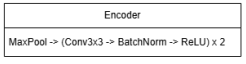
I recommend using a stool or table, something small to get the background in as well.

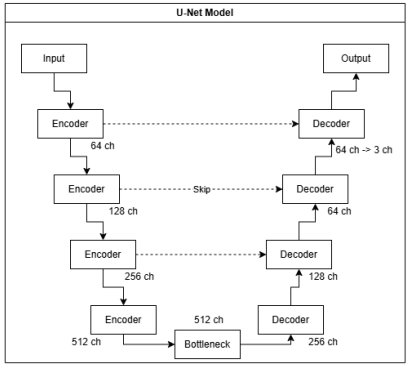
For this I just lowered the camera to the correct height

Segmenting Crops and Weeds

I identified a paper “Segmentation of weeds and crops using multispectral imaging and CRF-enhanced U-Net” and followed its methodology, disregarding the CRF as at higher resolutions it did not improve detection by very much (~1%), and used RGB rather than RGB+NIR to test multispectral imagery itself against the usual spectrum of light. I did opt to use the Green Filtered-NIR NDVI composite image however, as it did seem to have favourable results.





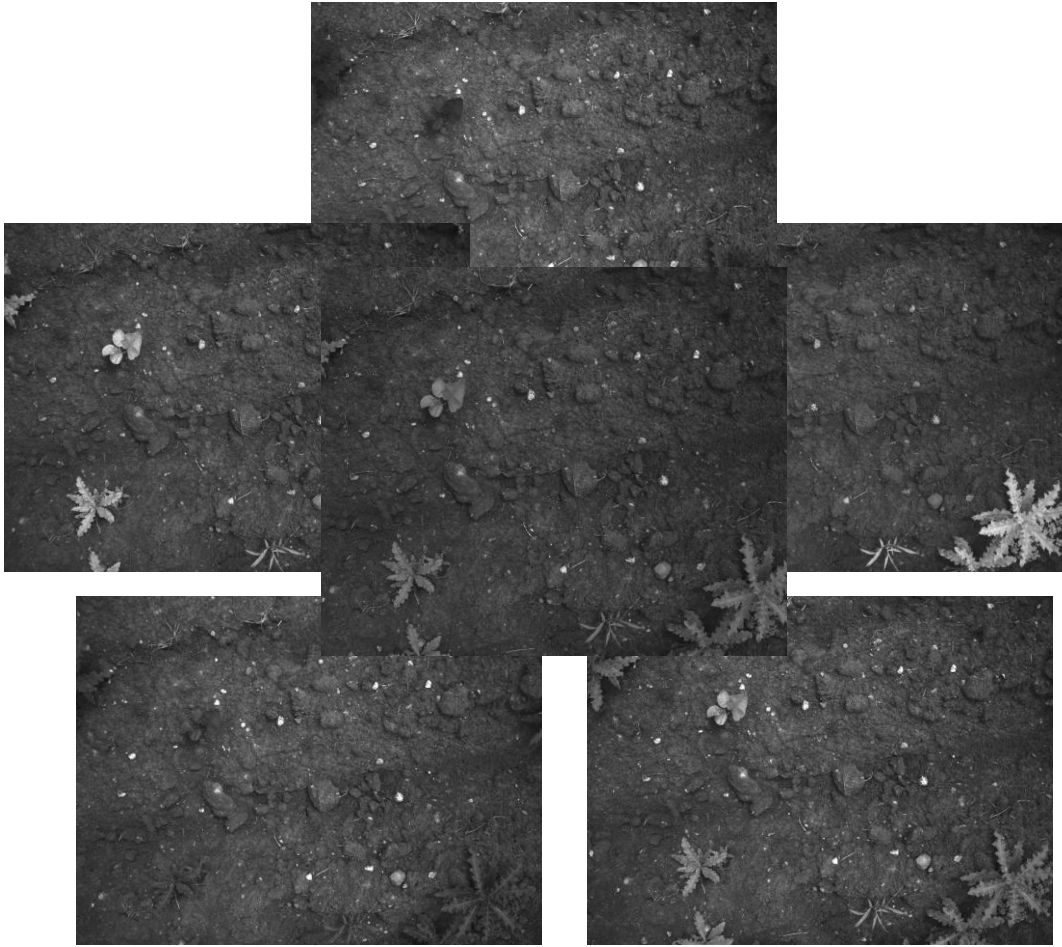


The Pipeline

First Step: Data Collection

Data was collected remotely via the built in wifi connection on the Micasense camera, on several days with multiple lighting conditions and cloud cover. Overall there were over 2900 sets of images collected, consisting of red, green, blue, NIR, rededge, and panchromatic TIFF files.

(./capture.py)



Second Step: Processing Images

Images were processed into two composites, RGB and GF-NIRNDVI images. The wavelengths were not aligned with one another due to the location of the lenses on the camera, so they were first each fitted to the green band as reference before laying them on each other, using the initial calculations done on a reference image and applying that same fitting to every image in the batch.

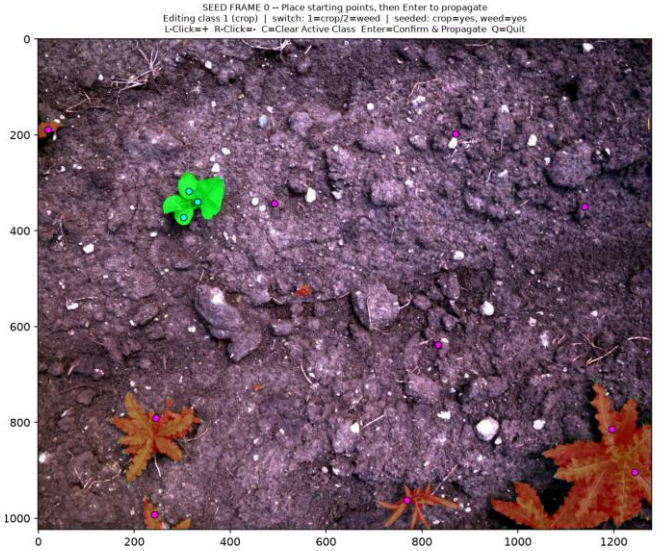
(./tiff_processing.py)

Third Step: Segmentation

With Sam V2 for assistance each frame was hand segmented to classify segments as weeds, crops, and soil.

The data was then saved as a mask image with [0,0,0] corresponding to soil, [1,1,1] to crop, and [2,2,2] to weeds.

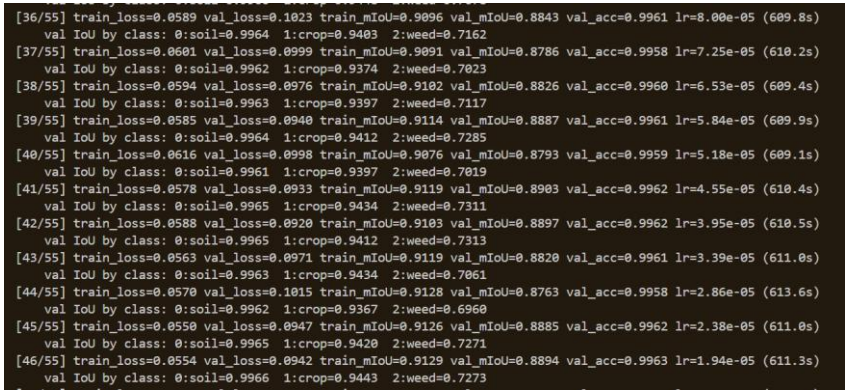
(./segmentation_tool.py)



Fourth Step: Training

Two models were trained, one for RGB images and one for GF-NIRNDVI composite images. The same masks were found to fit both the RGB and GF-NIRNDVI images and were used for both. 2767 pairs were chosen, 2254 for training and 513 for evaluation (~18.5%). The remaining images were set aside for prediction testing. Images were put through augmentation, like random horizontal and vertical flips as well as rotations for varied data.

(./unet_train/train.py)



Fifth Step: Prediction

Prediction takes a checkpoint from the model training and runs it on selected images. It is extremely fast and not resource intensive, able to run on ~200 images in about 20 seconds (not creating overlays)

(./unet_train/predict.py)

Prediction Metrics for Project Success

The following metrics were laid out in the project proposal as a marker for whether the models were useful enough to be used in the field

Minimum Soil IoU: 90%

Minimum Crop IoU: 80%

Minimum Weed IoU: 65%

Which Prediction Model Worked Better?

RGB

Best Soil IoU: 99.66%

Best Crop IoU: 94.67%

Best Weed IoU: 73.13%

Best Mean IoU: 89.03%

GF-NIRNDVI

Best Soil IoU: **99.71%**

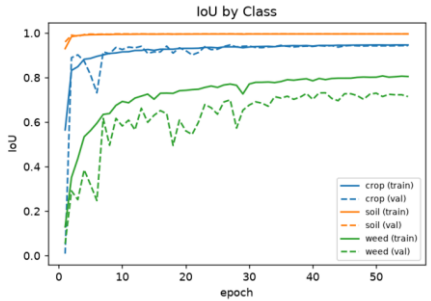
Best Crop IoU: **95.03%**

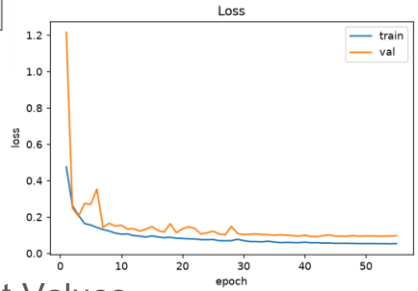
Best Weed IoU: **77.08%**

Best Mean IoU: **90.32%**

Which Prediction Model Trained Better?

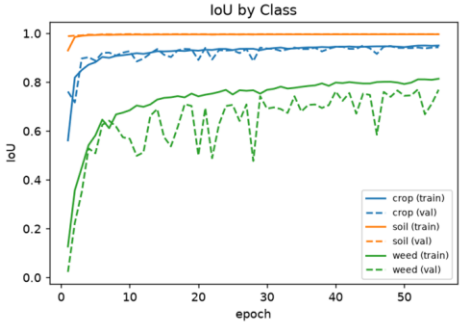
RGB

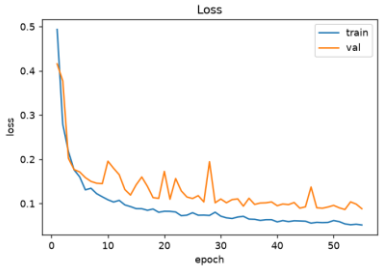




More Stable, Consistent Values

GF-NIRNDVI





Larger Swings but Higher Highs, More Potential

Compared to The Original Paper?

Their GF-NIRNDVI (704x704)

Best Soil IoU: 98.3%

Best Crop IoU: 90.5%

Best Weed IoU: 74.4%

Best Mean IoU: 87.8%

My GF-NIRNDVI (1280x1024)

Best Soil IoU: **99.71%**

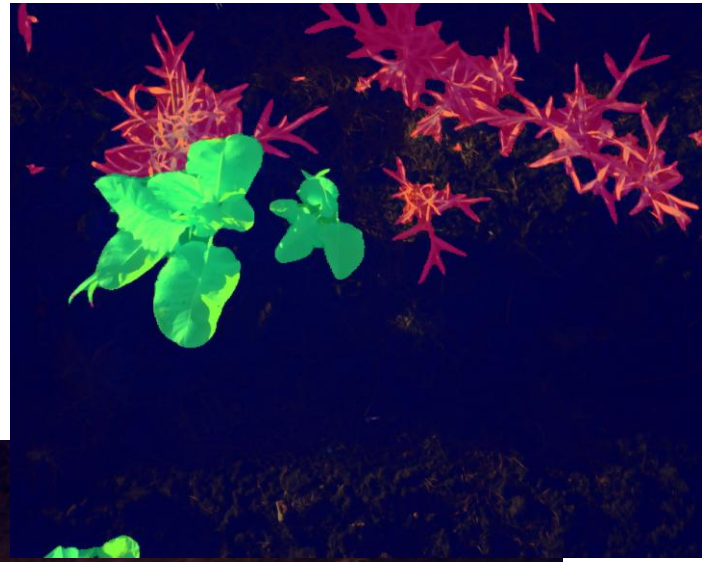
Best Crop IoU: **95.03%**

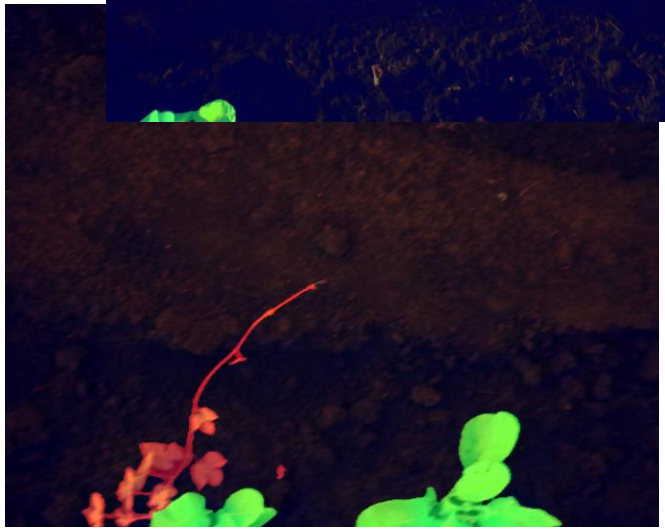
Best Weed IoU: **77.08%**

Best Mean IoU: **90.32%**

Why the improvement?

1. My dataset consisted of almost 3000 images while theirs consisted of about 500
2. My segmentations are likely not as good as theirs, they were done using SAM V2 and some were rushed, missing smaller details, so less strict
3. Higher Resolution means More Data to learn from as well





Next Steps for PSEUDO-AMIGA

1. Integrate the Micasense camera into the AMIGA software + mount it on the robot
2. Convert the pipeline from an external, laptop connected approach to one functioning on the AMIGA system
3. Create a survey function on the robot that paths through the field and sends images down the pipeline (Bands -> GF-NIRNDVI -> Prediction Model -> Mask) and marks problem areas (high weed/crop ratio) on a GPS based UI.
4. Use collected data set to train for other phenotyping subjects (disease, canopy coverage, nitrogen levels)

Questions?