Final Presentation

Senior Design 492: Soybean Parasitic Cyst Detector

—

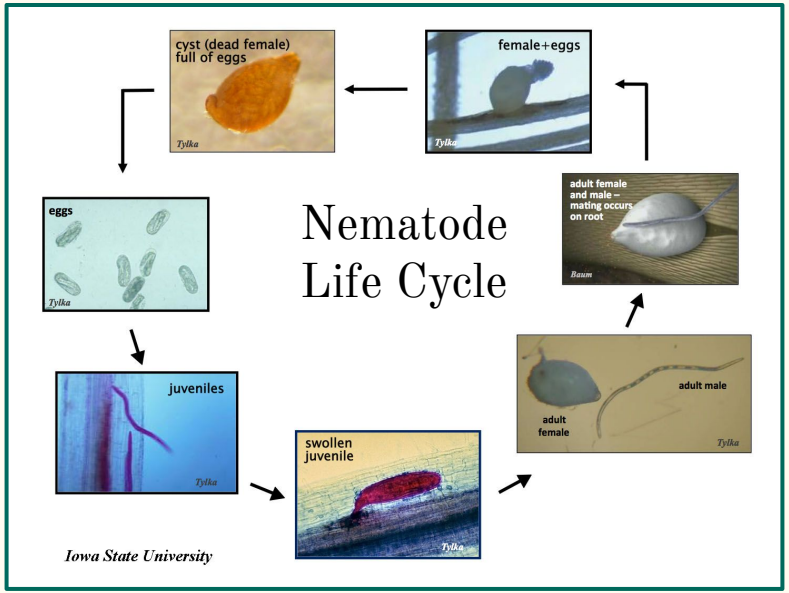
Team: 10

Client: Dr. Santosh Pandey

Chris Cannon, Ethan Baranowski, Katherine Moretina, Matthew Kim

Problem Statement

- Soybean Cyst Nematodes are parasites that reduce annual soybean yield by between 15-30%.
- To determine how many parasitic cysts are on the roots of soybean plants, we developed a deep learning computer vision model designed for small object detection.
- We also created a device to integrate image capturing with the machine learning model.

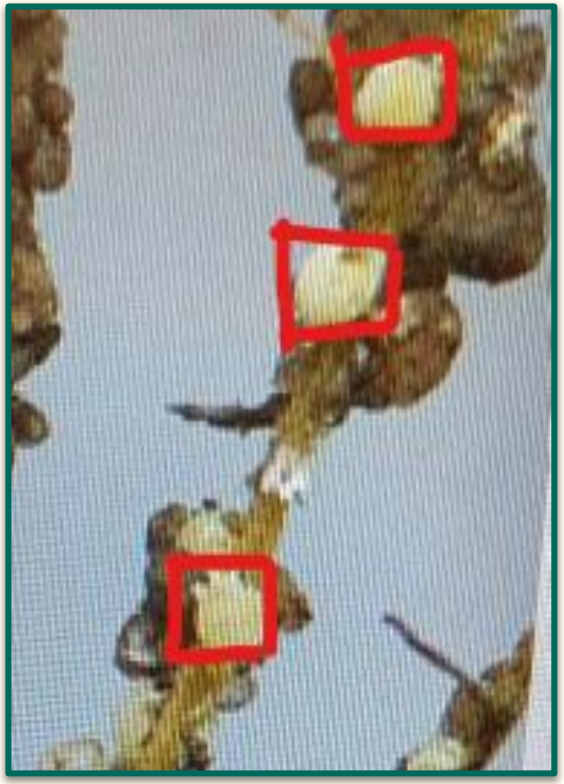


Existing Processes for Counting Cysts

- Processes:
  - Experts visually estimate cyst count on plant roots
  - Plant is sent to lab for microscope analysis
- Importance of Cyst Count:
  - Help researchers develop cyst-resistant varieties
  - Inform farmers for the use of fertilizers and pesticides

Drawbacks of existing processes:

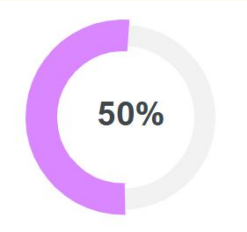
- Expensive
- Requires Experts
- Time Consuming
- Resource-Limited



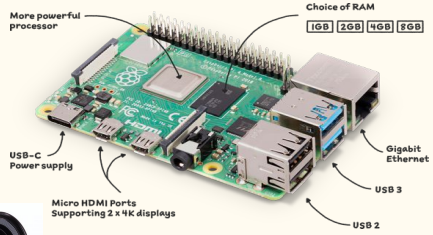
https://www.raspberrypi-spy.co.uk/2015/05/using-an-i2c-enabled-lcd-screen-with-the-raspberry-pi/
https://www.raspberrypi.com/products/raspberry-pi-4-model-b/
https://www.raspberrypi.com/news/new-product-raspberry-pi-high-quality-camera-on-sale-now-at-50/
https://www.freepik.com/free-vector/illustration-paper_2606514.htm#query=document&position=21&from_view=keyword&track=sph

Project Goals

- Have a working computer vision program that counts cysts with more than 50% accuracy
- Have a working prototype of a device for soybean root image capturing
- Integrate the computer vision system so cysts can be counted on the prototype
- Create detailed documentation for the next iteration of this project

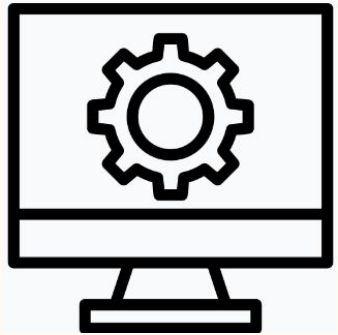


Accuracy

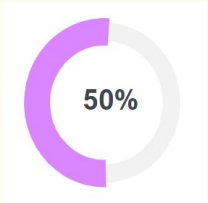


Functional Requirements

https://www.flaticon.com/kr/free-icons/pictogram/5



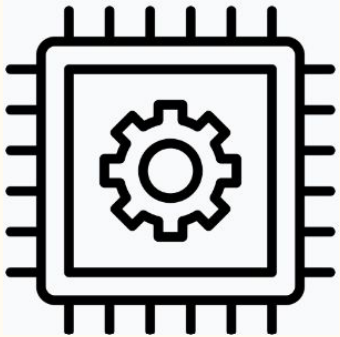
Software



Accuracy

Internet

Time



Hardware

No Damage

Resolution

Portable

Non-Functional Requirements

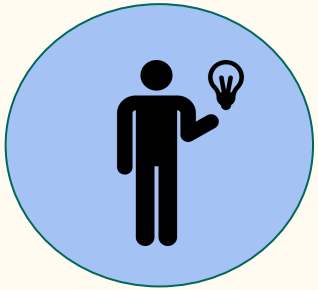
https://www.flaticon.com/kr/free-icons/pictogram/5

Requires No Formal Training

Under $500

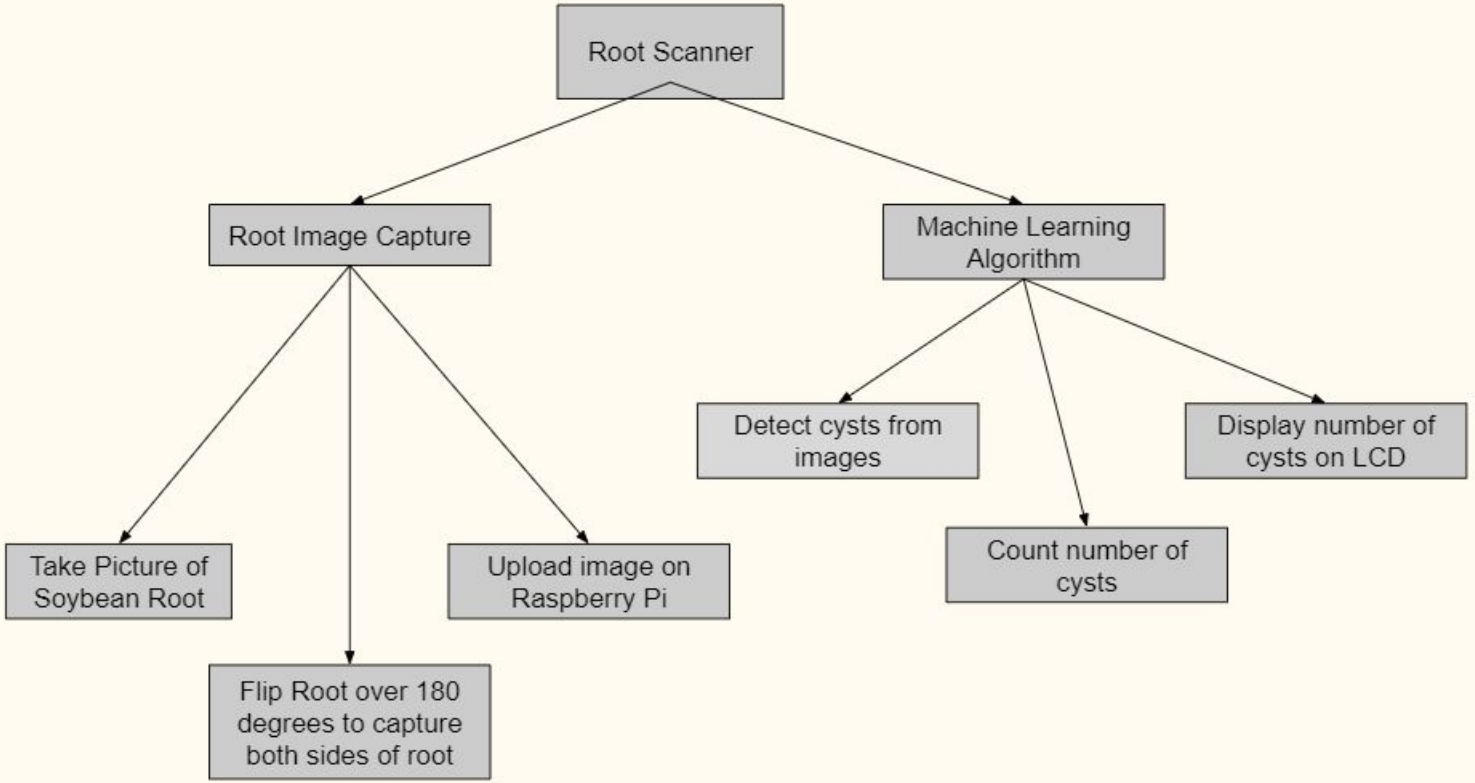
No Crashes During Use

Can be Operated Outdoors



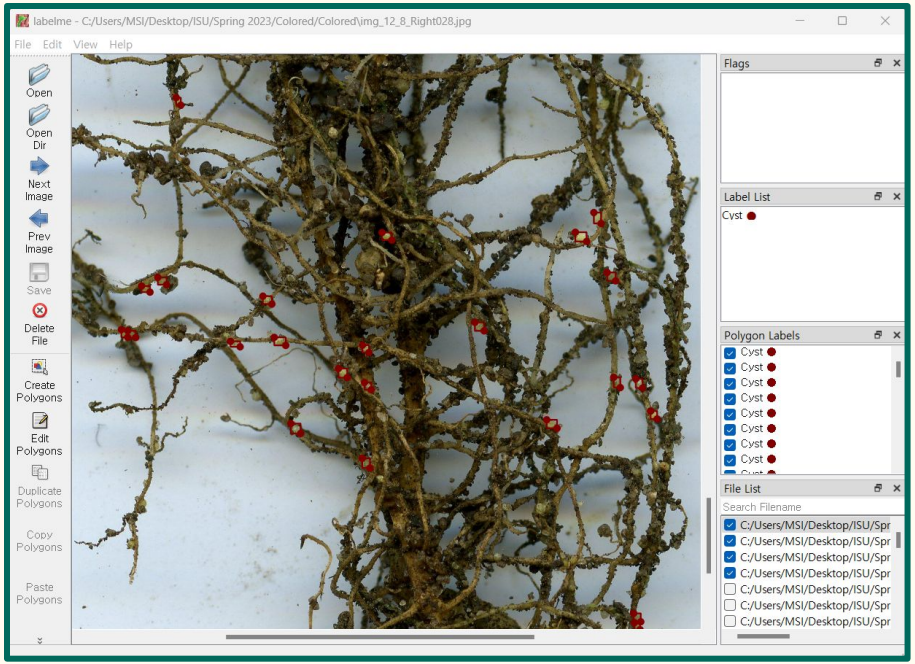
Semi-Automated Operation

Functional Decomposition



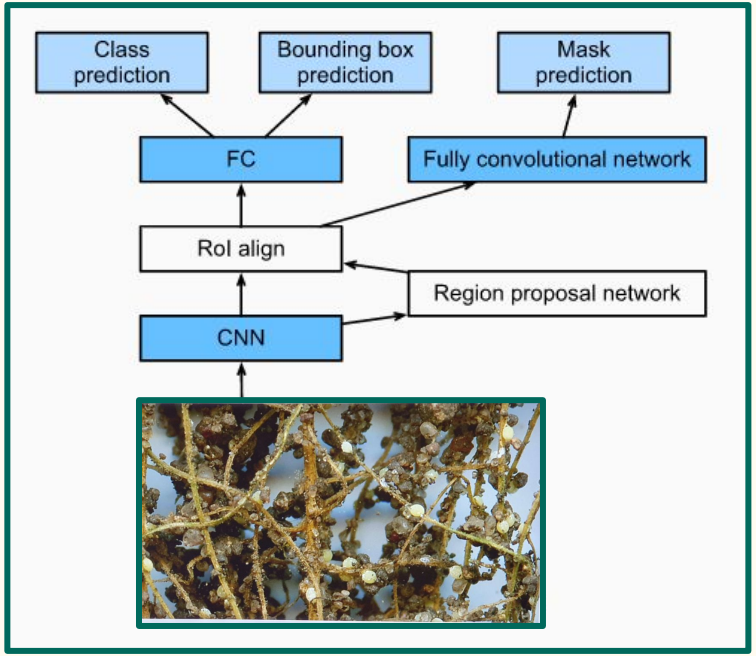
Using Machine Learning to Detect Cysts

- Machine learning is commonly used to detect objects in images
  - Self-driving cars
  - Drone footage
- Computer vision is not generally successful on tiny objects in images.
- Using an algorithm known as Faster R-CNN.
  - Faster Recursive Convolutional Neural Network.
  - 2-Stage Algorithm.



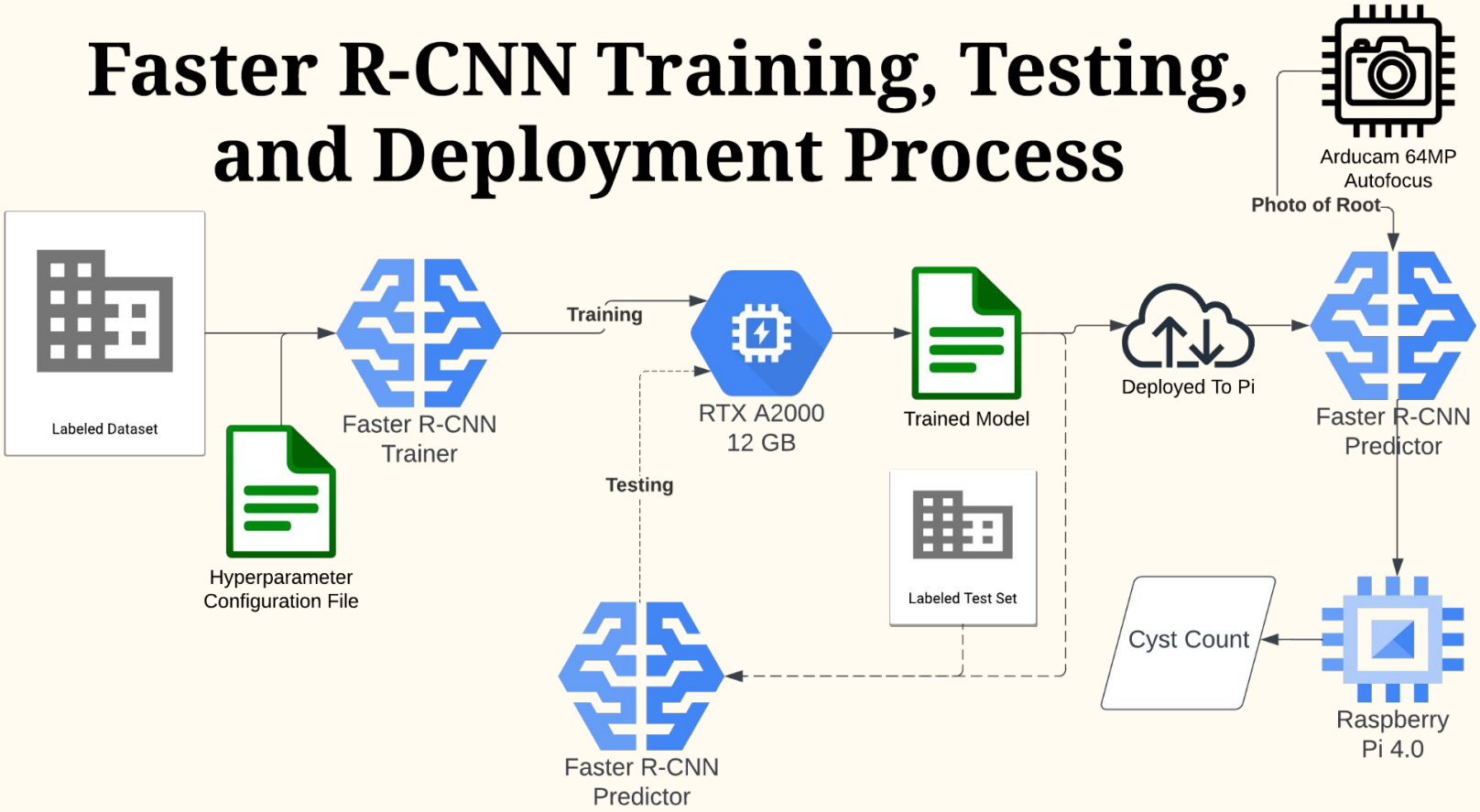
Faster R-CNN

- Compared to other algorithms, Faster R-CNN:
  - Has comparable or better accuracy
  - Has higher accuracy for small object detection
    - For example: Birds in a flock
  - Runs moderately slower
- Two Neural Networks in One:
  - Region Proposal Network
  - Classifier Network
- Two-stage Algorithm:
  - Increased accuracy over one-stage.
  - Second Stage adds RoI (Region of Interest) Pooling layer to filter results.
- Detectron2 Implementation
  - Open-Source Facebook Research Project



For more information how we chose our object detection algorithm, see Appendix slide "Faster R-CNN vs. YOLO vs. SSD"

Faster R-CNN Training, Testing, and Deployment Process



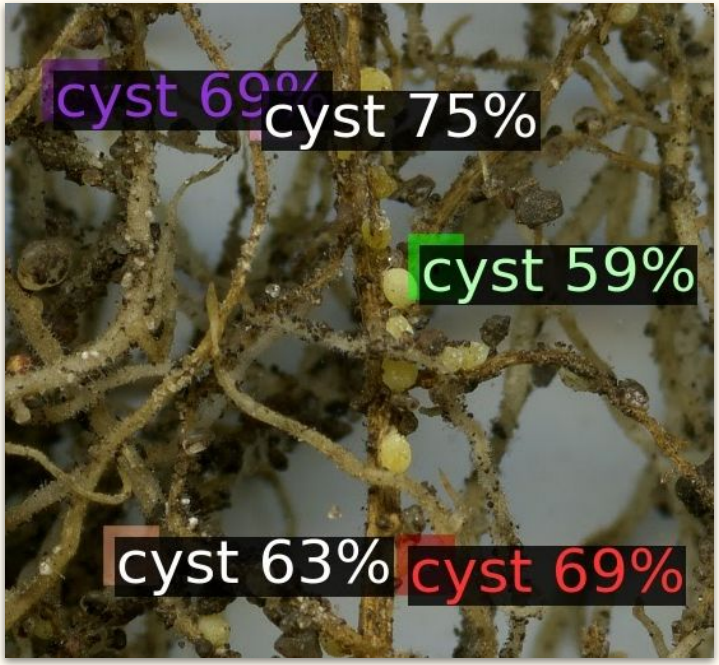
Software Challenges

- Unfamiliarity from team regarding machine learning.
  - Overcame by 3 months of research
- Training an algorithm takes a considerable amount of time
  - Overcame by using a capable local machine to speed up training.
- Small object detection algorithms require advanced user experience to properly train.
  - Standard approaches to solving object detection are ineffective for *small-object* detection.
  - Consulted with multiple experts and tried many different training approaches.

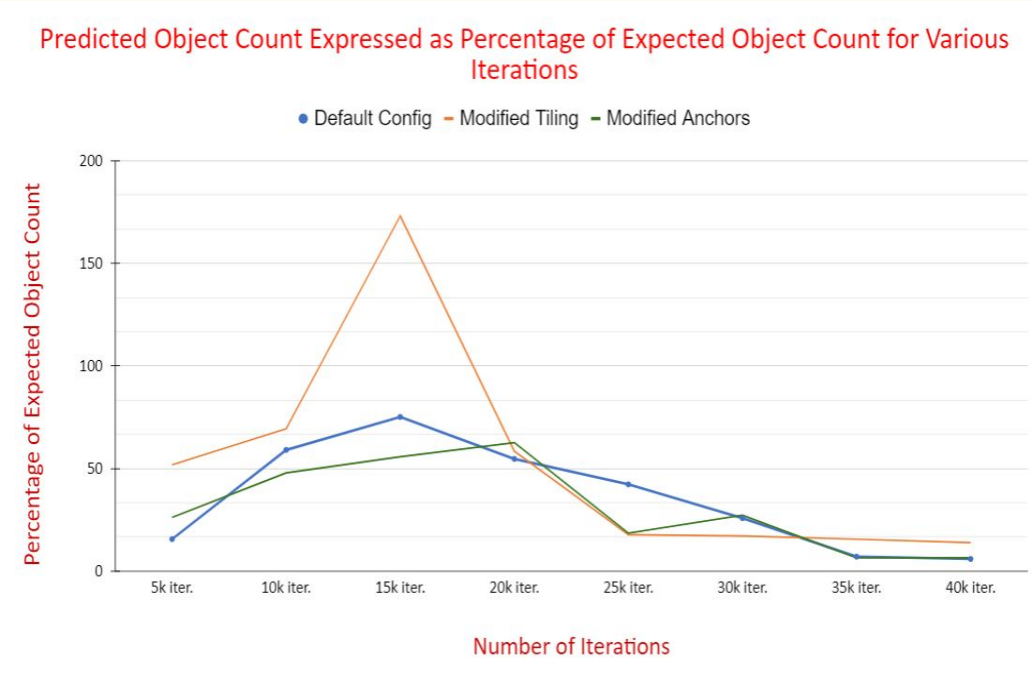
Optimizing Faster R-CNN for Small Object Detection

We identified four main hyperparameters to optimize:

1. Number of training iterations.
   a. Our Best Values: 20,000 - 30,000
2. Number of predictions allowed per image.
   a. Our Best Values: > 250
3. Feature Extraction dimensions
   a. Our Best Values:
4. Size of the prediction boxes (Anchor boxes).
   a. Our Best Values: 16x16

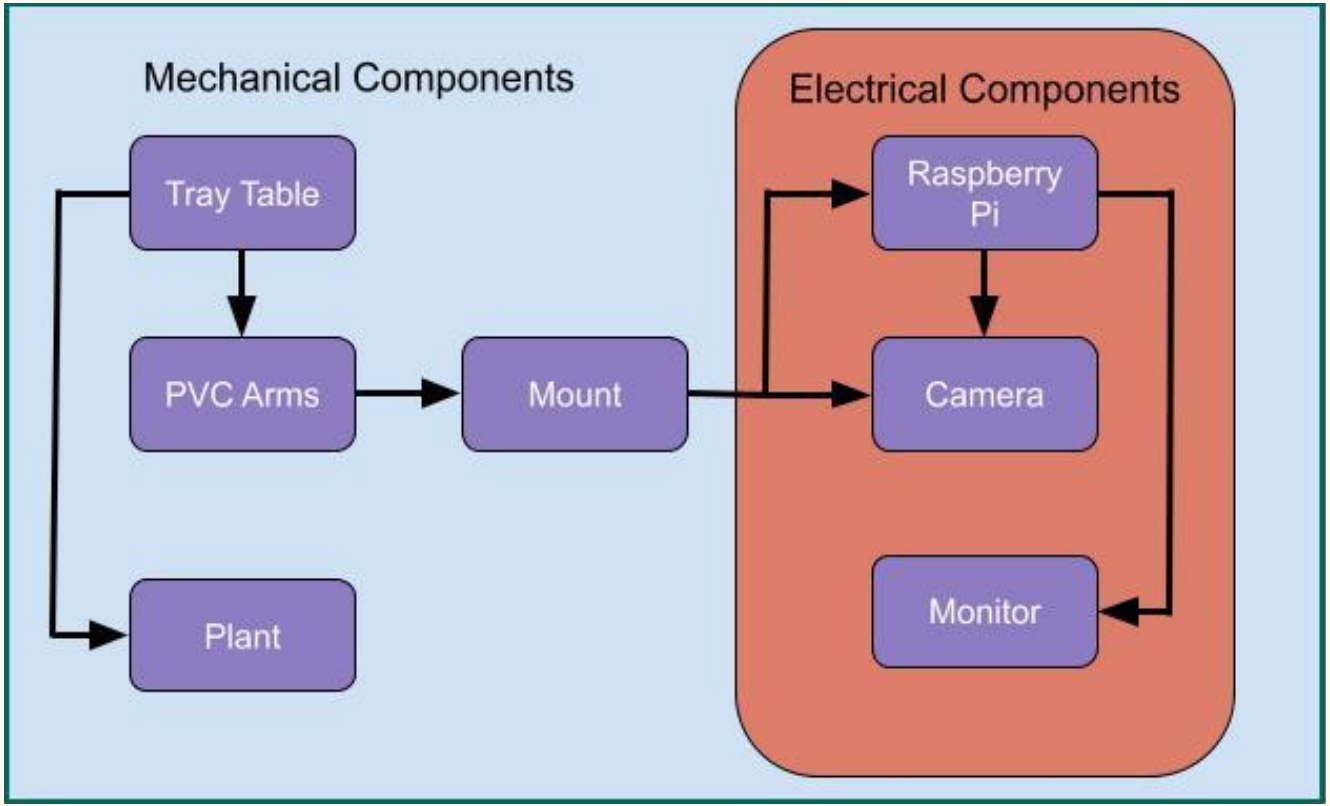


Testing Results: Finding the “Goldilocks” Parameters.

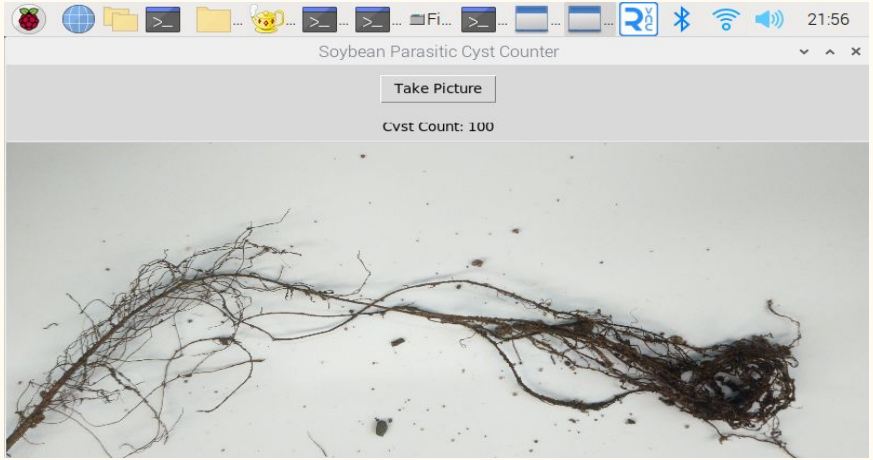


- Average Precision Score of near 0% across every configuration attempted.
- Default Config: baseline results using unmodified Detectron2 hyperparameter configuration.
- Modified Tiling:
  - Max feature size ↓
  - Predictions per image ↑
- Modified Anchors:
  - Prediction box sizes ↓
- Optimal Configuaration:
  - 20k Iterations.
  - Predictions per image ↑
  - Prediction box sizes ↓

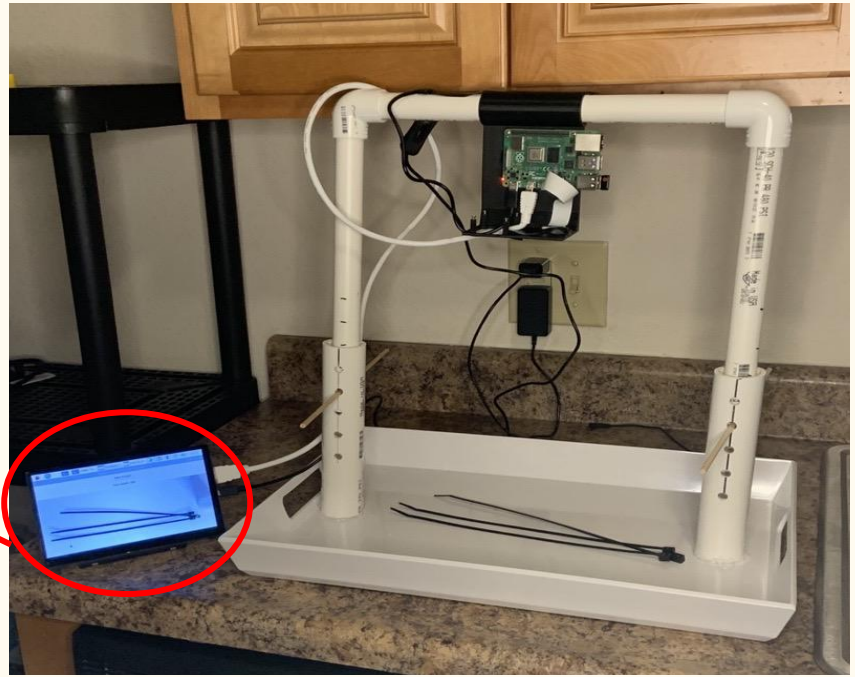
Hardware Detailed Design



Final Hardware and GUI Design

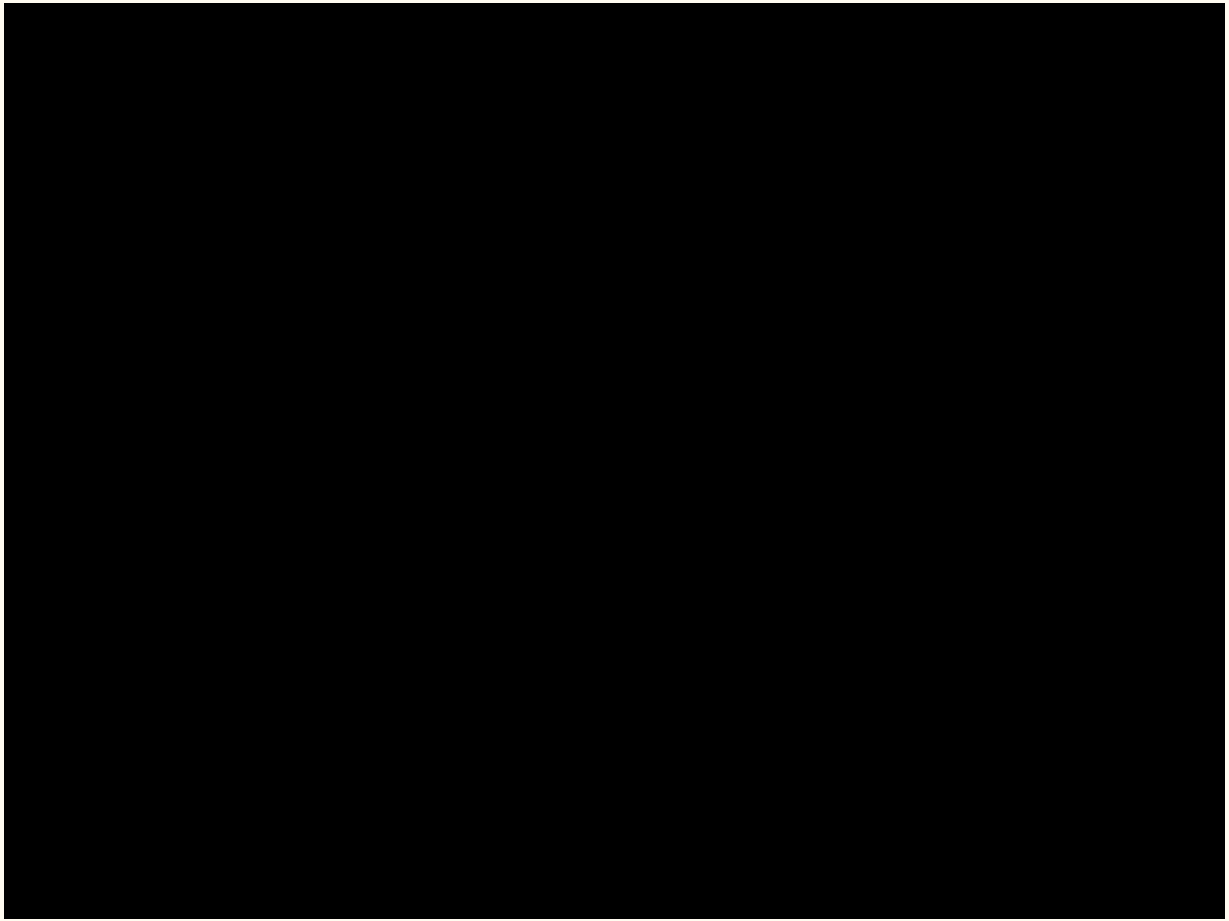


Graphical User Interface



Hardware Prototype

Demo



Dependencies

Detectron2

PyTorch

TorchVision

OpenCV

Constraints

GPU

CPU

Environment

Raspberry Pi 4

Raspberry Pi OS
Debian 11 - Bullseye

Conclusions on the Hardware and GUI

Challenges

- Optimizing custom parts for 3d printing
- Raspberry Pi interfaces change with operating systems

Moving Forward

- Add a PWM fan to the Raspberry Pi
- Updating GUI so it no longer relies on running Linux commands in the code
- Continue market research on high quality cameras

Successful Outcomes:

- First iteration of a hardware prototype
- First iteration of a GUI to accommodate ease of use
- Successfully integrated the software and hardware components

Conclusions on the Software Side

Challenges

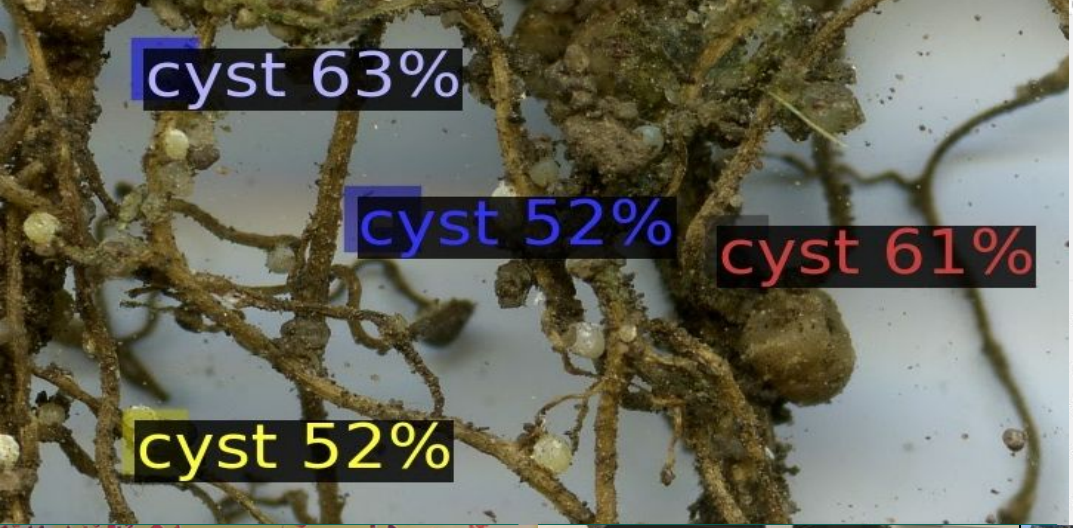
- More restricted than we thought by the timeline of getting our dedicated training machine.
- Customizing complicated, specialized applications to work in new areas is challenging.

Moving Forward

- Increasing the dataset size - Plant Pathology is still growing samples for this project.
- Refining the model via fine-tuning hyperparameters

Successful Outcomes:

- Fully labeled dataset with ~10,000 instances
- Developed programs to:
  - Consolidate & reformat Labelme dataset files
  - Parse the dataset and load it into Detectron2
  - Train the Faster R-CNN model
    - Acquires default configuration file
    - Allows for tuning hyperparameters
    - Stores checkpoints during the training process
  - Perform prediction on new images
- Identified relevant hyperparameters for small object detection


cyst 63%
cyst 52%
cyst 61%
cyst 52%

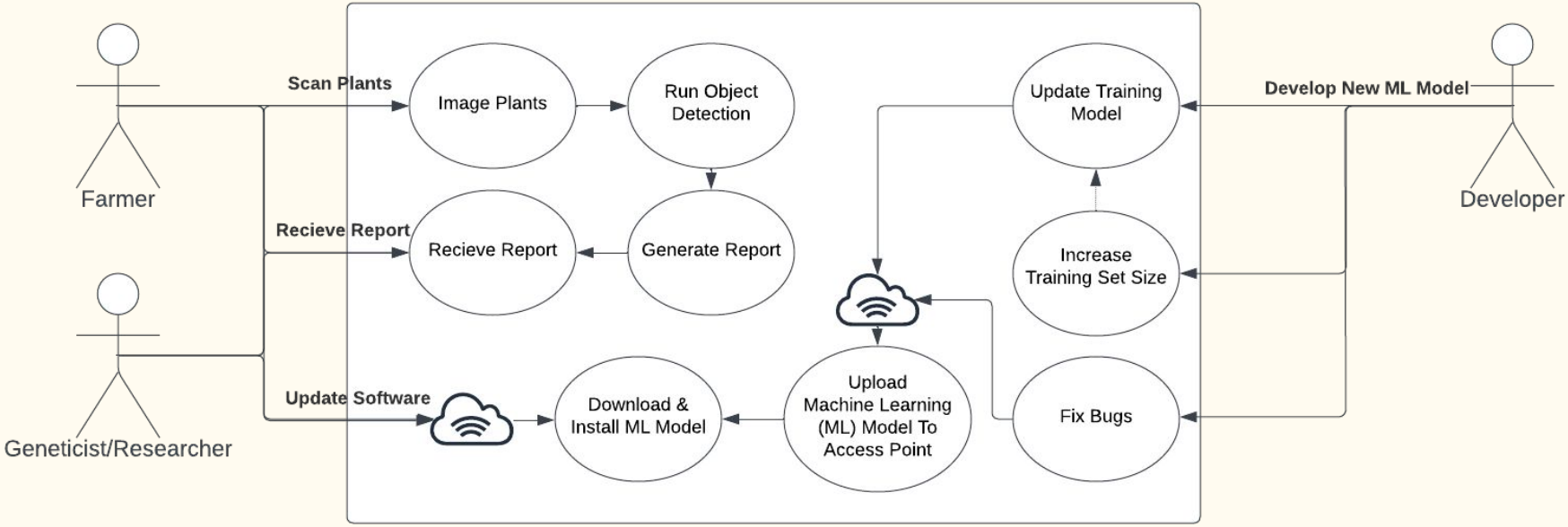

File Edit View Help
Open
Open Dir
Next Image
Prev Image
Save
Delete File
Create Polygons
Edit Polygons
Duplicate Polygons
Copy Polygons
Paste Polygons


0 Eggs
juvenile:0.97

Questions?

Appendix:

Use Case Diagram



Design Exploration

Method	mAP	FPS	batch size	# Boxes	Input resolution
Faster R-CNN (VGG16)	73.2	7	1	~ 6000	$\sim 1000 \times 600$
Fast YOLO	52.7	155	1	98	448×448
YOLO (VGG16)	66.4	21	1	98	448×448
SSD300	74.3	46	1	8732	300×300
SSD512	76.8	19	1	24564	512×512
SSD300	74.3	59	8	8732	300×300
SSD512	76.8	22	8	24564	512×512

Figure depicting accuracy (mAP) and speed (FPS) of the algorithms considered.

Considered 3 high performing object detection algorithms:

- Faster R-CNN
- You Only Look Once (YOLO)
- Single Shot Detector (SDD)

Points of interest for the algorithms:

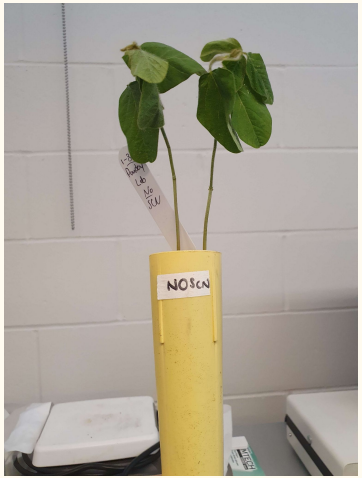
- Classification accuracy
- Algorithm Speed
- Training time
- Region of interest generation
- Small object detection optimization

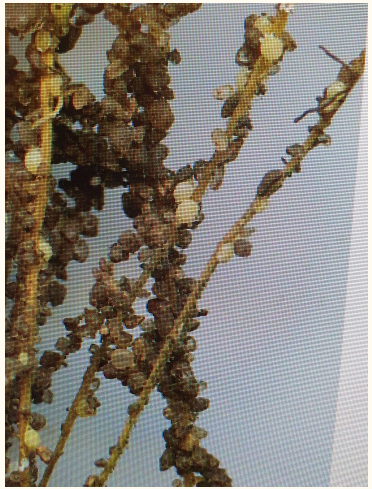
Faster R-CNN vs YOLO vs SSD

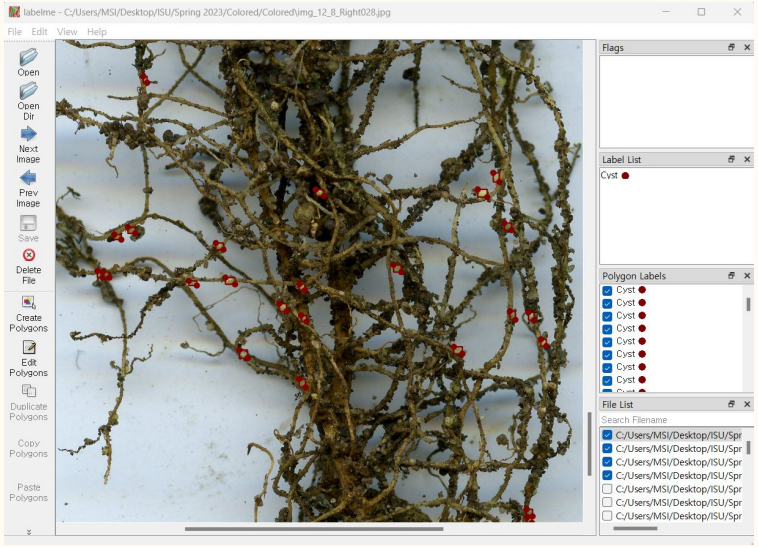
- Sifting cysts off the roots and manually counting
- Small object detection algorithms

Faster R-CNN	You Only Look Once (YOLO)	Single Shot Detector (SSD)
Uses Region Proposal Network to generate regions containing objects for classification.	Uses Anchor Boxes to generate regions containing objects for classification.	Uses Anchor Box Pyramids to generate regions containing objects for classification.
Uses Convolutional Neural Network to classify objects.	Uses Convolutional Neural Network to classify objects.	Uses Convolutional Neural Network to classify objects.
Algorithm Training takes a considerable amount of time.	Algorithm Training takes a moderate amount of time.	Algorithm Training takes a moderate amount of time.
73% mean Average Precision (mAP)	50-65% mean Average Precision (mAP)	75% mean Average Precision (mAP)
Capable of handling high resolution images.	Capable of handling high resolution images.	Capable of handling high resolution images.

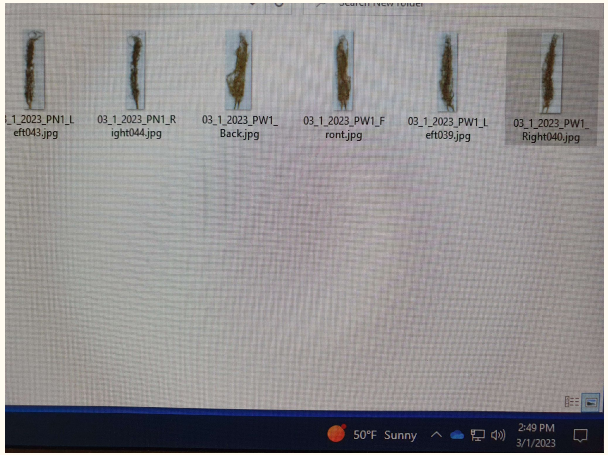
Images: Cyst Nematode Identification Process



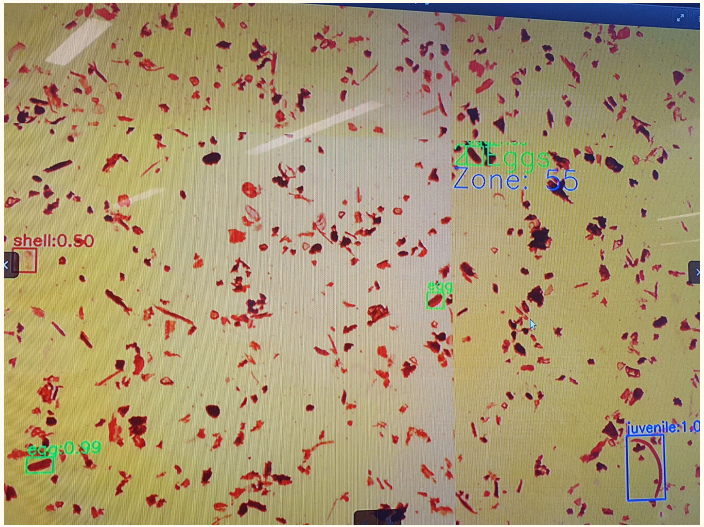


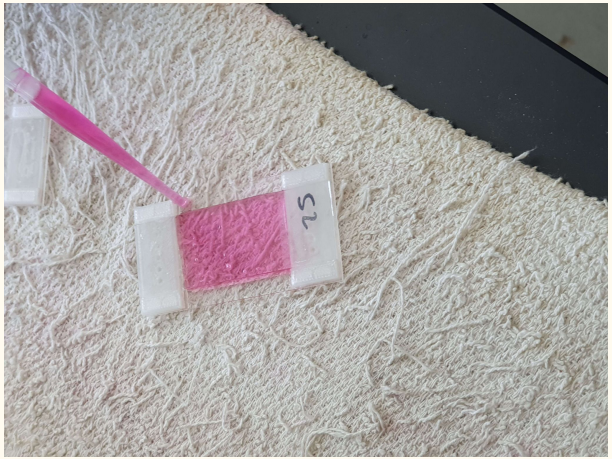


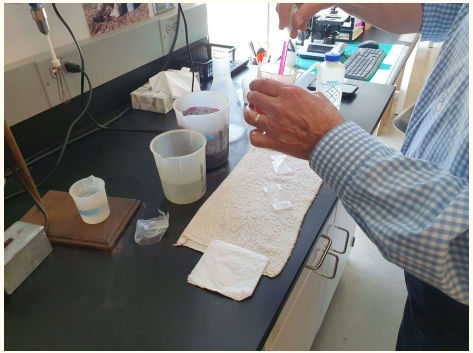
Images: Current Cyst Nematode Identification Process




DELTA
16" DRILL PRESS


shell:0.50
Eggs
Zone: 55
egg:0.99
juvenile:1.0


S1



Mask R-CNN Vs. Faster R-CNN

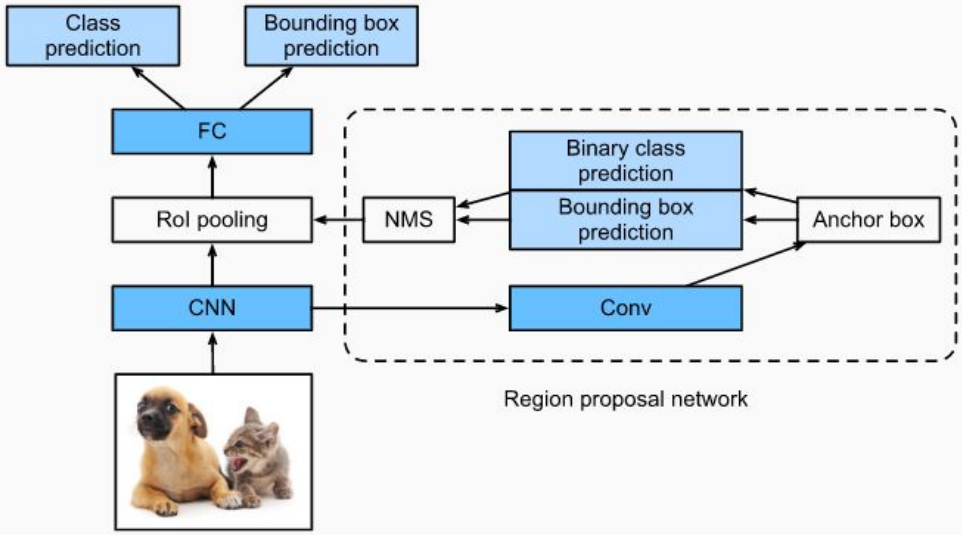


Fig. 14.8.4 The faster R-CNN model.

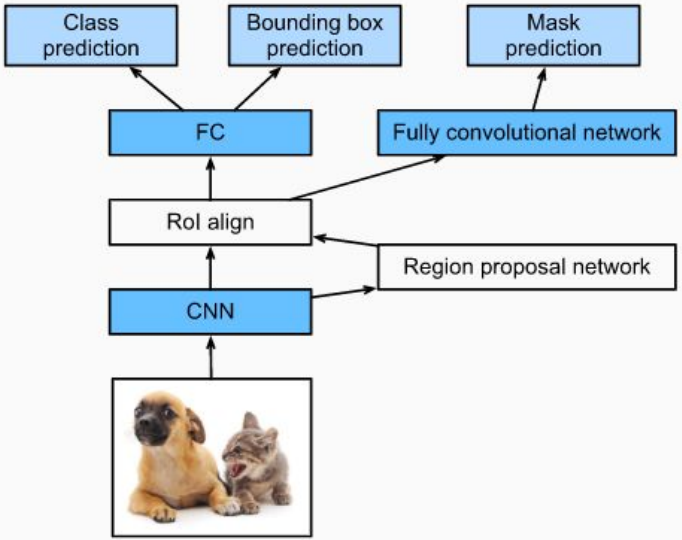


Fig. 14.8.5 The mask R-CNN model.

IEEE Standards

- *IEEE 268-1992:* American National Standard for Metric Practice
- *IEEE/ISO/IEC 32675-2021:* ISO/IEC/IEEE International Standard--Information technology--DevOps--Building reliable and secure systems including application build, package and deployment
- *IEEE/ISO/IEC P24748-6*: ISO/IEC/IEEE Draft Standard - Systems and Software Engineering -- Life Cycle Management
- *IEEE/ISO/IEC 14764-2021:* ISO/IEC/IEEE International Standard - Software engineering - Software life cycle processes Maintenance

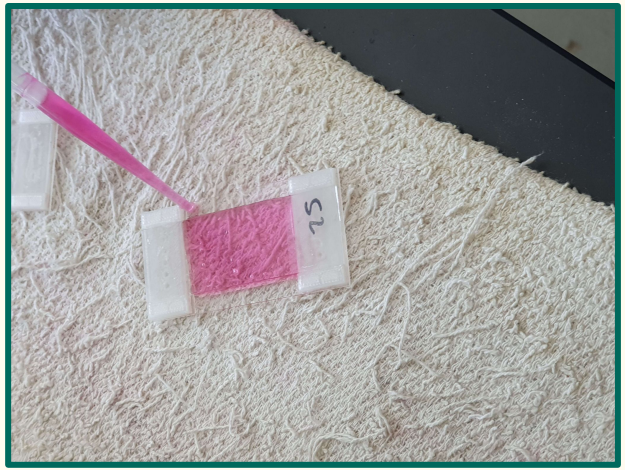
Market Survey

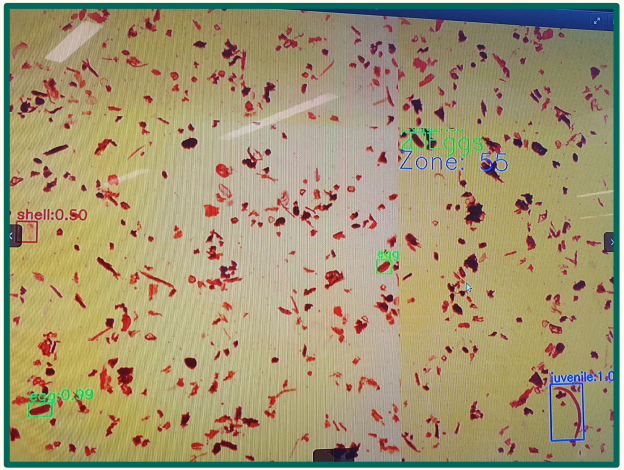
Existing Processes:

- Visual inspection of roots
- Microscope inspection of processed soil samples

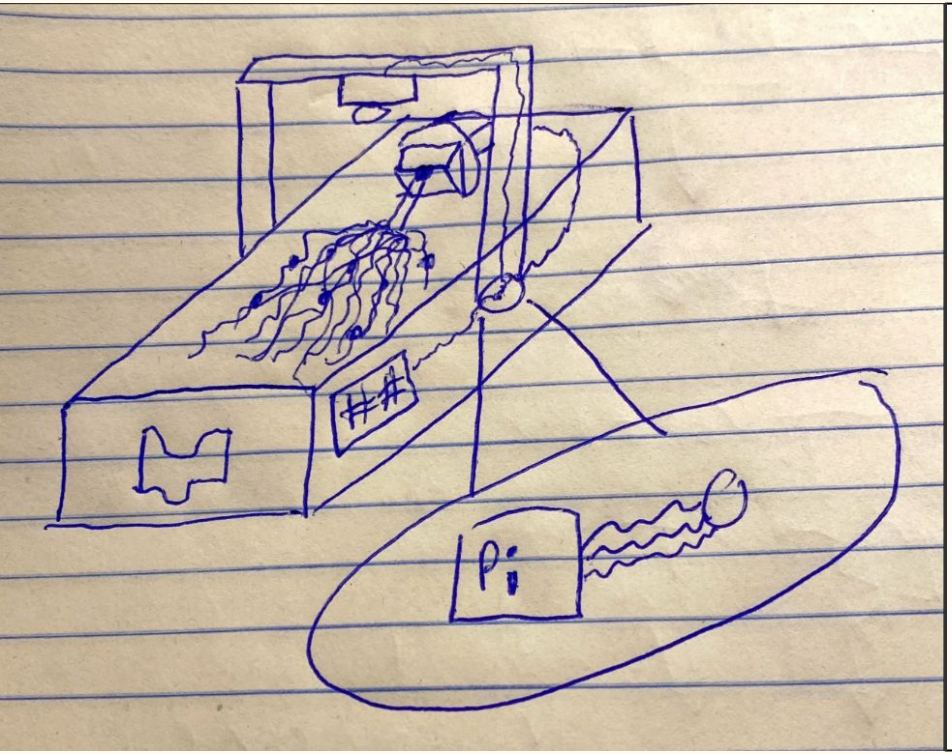
Existing Technology:

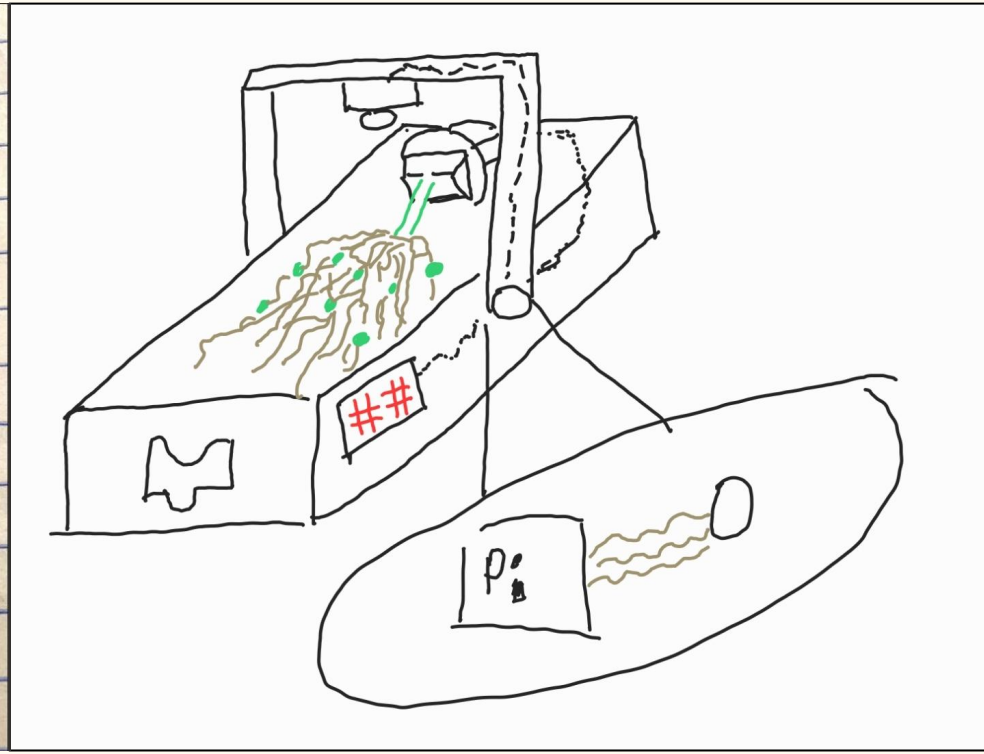
- Various object detection algorithms
  - YOLO, SSD, R-CNN, for example.
- NVidia Jetson
  - High-powered computing for embedded systems using AI.
  - Key strengths are computing power and pre-trained models





Conceptual Sketch





Test Plan

Tests

- Cyst Detector ML Model
  - ＜ 50% Error Range of Output ✓
  - Produces output in＜5s/image when run on Raspberry Pi X
  - No fatal errors occur during operation ✓
- Ensure proper communication between hardware devices
  - Raspberry Pi ✓
  - Camera ✓
  - LCD Output Screen ✓

System Testing Plan

Use unit testing to ensure individual units meet the requirements, and to establish baseline data.

Use integration tests to confirm the data flows appropriately from image capture to cyst-count output.

Compare accuracy of the prototype to the baselines established above.