

# Individual Lab Report 8

Name: Dung-Han Lee  
Team Name: Team E Wholesome Robotics  
Team members: Aaditya, Aman, John, Hillel  
ILR Number: 08

October 10, 2019

# 1 individual progress overview

There were two primary focus area in the past 2 weeks, which is (1) integrate the visualization pipeline and (2) using a simpler model to replace Mask-RCNN

## 1.1 Integrate the Visualization Pipeline

### 1.1.1 Motivation

Until the last progress review, the monitoring pipeline was able to run component-wisely i.e. each component block functions but does not work altogether. The main issue was that ROS sets the default system path to python 2.7 while the deep-net inference pipeline requires python 3.5 for many of its supporting functions.

### 1.1.2 Workaround for Integrated Visualization Pipeline

The issues arise from the line "source /opt/ros/kinetic/setup.bash" in bashrc file, which would change the default python path to python2.7 every time a terminal is launched. It remains unclear how to switch it back once this line is executed in a terminal. However, a workaround is to remove this line in bashrc and only run it when ROS is needed.

Specifically, a .sh file would first open a terminal, run the ros set up, and then use ros package to extract data from rosbag. After that, a different, fresh terminal would be launched to run the deep-net inference pipeline and GUI interface. Besides, to bridge the data between two function blocks, the variables are stored in a temporary file. As a result, the visualization pipeline can now run end-to-end.

### 1.1.3 The Workaround sh File

```
#!/bin/bash

if [ "$#" -ne 2 ]; then
    echo "Usage: arg1: Path to ROS Bag location (end path provided with /) , \
    arg2: Bag name (don't include .bag)"
    exit -1
fi

gnome-terminal -e "bash -c 'source
/opt/ros/kinetic/setup.bash;./map_viz.sh \"$1\" \"$2'"
sleep 50s
gnome-terminal -e 'python3 diagnose_plant_health.py'
```

## 2 Unet: An Alternative Deepnet

### 2.1 Motivation

The previous Mask-RCNN pipeline was closely following an open-source tutorial, although it worked out of the box, it's unclear how to get the max of the library since the repository is pretty complex. On the other hand, the data-set is of 150 images per plant before data augmentation, making it desirable to work with a simple but effective network.

### 2.2 Some Information about Unet

Unet was first introduced in the biology domain to perform semantic segmentation on gray images of cells with very few training samples (30 images). Furthermore, the original task was set up as a pixel-wise binary classification.

The Unet itself consists of a sequence of convolution and de-convolution, effectively its an encode-decoder structure that aims to extract information from the input image and then scale back to original size.

### 2.3 Initial Results

The Unet was trained from scratch on a row of augmented data with 800 images. Note that the training framework has been shifted from TensorFlow to PyTorch as team-members have more experience and control over this framework.

The network was modified to perform multi-class (hole and fungus) segmentation instead of binary class. However, the current output results are very different from the human annotation. A simple print out examination shows that the network is performing bizarrely and future works are needed to debug the network.

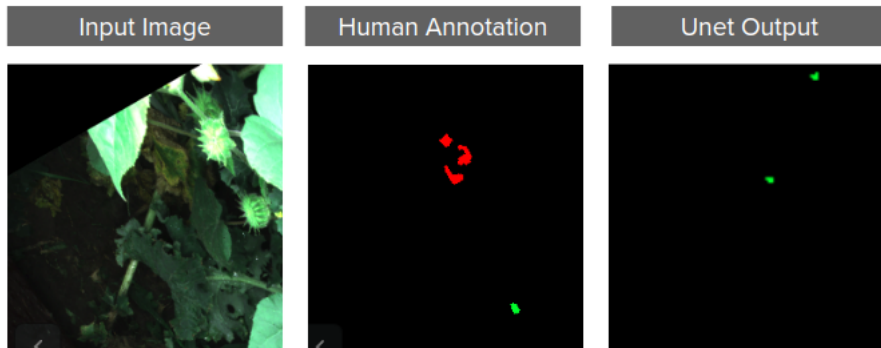


Figure 1: unet initial results

## 3 Challenges

The main challenge came from modifying the architecture of Unet to get desirable results. Specifically in debugging how the network should be structured and formulating the cost function. These challenges can be tackled as John and Dung-Han Lee both have experience with PyTorch, and John can support this task as a similar task has been conducted for him in the last semester.

### **3.1 Future works**

debugging how the network should be structured, specifically making sure that the structure makes sense for a multi-class segmentation task. Also, literature research and searching over stack overflow are needed to make sure a proper cost function is formulated for the task.

## 4 Teamwork

John has been working on debugging the localization with rows of plants with the orientation of the robot, specifically by projecting the RTK GPS measurement on to the heading direction of the robot; John also works on revisiting LiDAR-based navigation since the new farm layout is simpler for the task. John also supported Dung-Han Lee in Unet training.

Aaditya has been working closely with Hillel and Dung-Han Lee for multi-row visualization on a GUI. Specifically, he has designed a new class to make the task more scalable. Meanwhile, Dung-Han Lee has supported him in integrating the pipeline and having the pipeline work on his computer.

Hillel has been working to add on features according to farmers' feedback, as well as fabricating electronic wires for sensors and plant guards for the field test.

Aman has been working a planner for a given farm layout, which would generate the planned trajectory for the robot to follow.