Individual Lab Report 9

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

October 24, 2019

1 Individual Progress

1.1 Improving the performance of Unet

1.1.1 Motivation

Since there are relatively few data available for the project, it was hypothesized that a simpler deep neural network will better fit the data. However, in the last progress review, Unet failed to produce reasonable results compared to human annotation.

1.1.2 Weighting the cost function

The cost function is the performance metric that used to update the parameters in deep learning models, i.e. a single scalar that guides the update of all the parameters. Image segmentation problems can be treated as pixel-wise classification problems, and usually, a cross-entropy loss is applied for each pixel. It's worth pointing that, since Unet was initially proposed to detect cell boundaries, the author added a weighting function to cost function such that pixels adjacent to cell boundaries received higher weighting.

For hole and fungus detection, it is less important to obtain a precise boundary but to identify their very existence. In the last progress review, Unet failed to pick up most of the holes. And that was attributed to the very imbalanced nature of the data on the pixel level: the pixel-area-ratio of background to fungus to the hole is roughly 1000: 10: 1. In other words, holes tend to be neglected in the training process because their corresponding penalty is simply low enough to be ignored. To address this, a weighted vector is fed into the cost function with weighting inversely proportional to pixel-area of each class i.e. 1: 100: 1000 for background, fungus, and holes. In Fig 1, it's clear that hole areas are successfully learned in this manner.

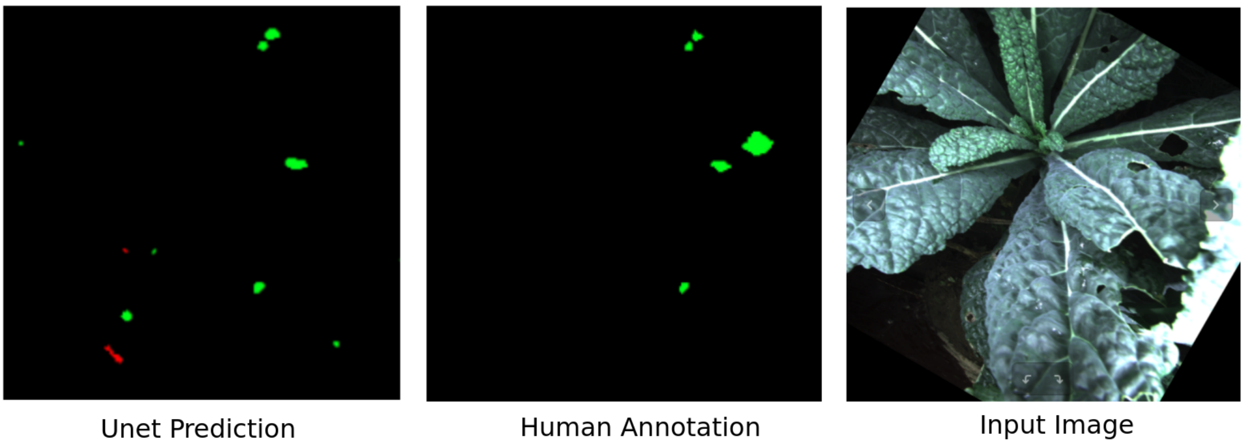

Figure 1: Unet prediction is closer to human annotation

1.2 Improved Results Achieved

For each of the plants of interest, an Unet model is trained from scratched with data associated with that specific plant only i.e. one model for one plant. With the presence of holes or fungus defined as positive, precision $\frac{truepositive}{truepositive+falsepositive}$ and recall $\frac{truepositive}{truepositive+falsenegative}$ are obtained at equal error rate (a point where false rejection rate (defined as % of wrongly neglected sample of all positive samples) equals to false acceptance rate (defined as % of wrongly reported positive-samples of all negative samples). It's clear from Fig 2. that Unet achieves much better results if not comparable results in comparison to Mask-RCNN.

Mask-RCNN				
hole precision	71	82	25	85
hole recall	71	82	33	61
fungus precision	91	82	81	75
fungus recall	84	77	81	75
Unet				
hole precision	75 (+4)	82	31(+6)	94 (+9)
hole recall	85 (+14)	82	38 (+5)	88 (+27)
fungus precision	93 (+2)	94 (+12)	88 (+7)	91 (+16)
fungus recall	93 (+9)	88 (+11)	84 (+3)	91 (+16)

Figure 2: unet outperform Mask-RCNN on all performance metric

2 Challenges

An additional Unet is trained on combined data-set i.e. one single model for all plants of interest. However, hole precision and recall dropped by 10% or more for 2 plants out of 4 plants. This observation suggests that Unet is not able to generalize over different leave texture and holes. Thus it's concerning whether Unet will ever perform well on the newly collected dataset – which consists of 2 new plants that were never presented in the training set. Alternative plans include tuning hyper-parameters of the Unet model or simply trade precision for recall so that farmers can be informed if there's an issue on the field. Also, clustering nearest k plants on the GUI will help, since it acts as an averaging filter. Future works include modifying the current monitoring pipeline to integrate Unet and training a suitable pipeline to classify images on the newly collected dataset.

3 Teamwork

John has been working on row detection based navigation with LIDAR while Aman and Aaditiya were looking into sensor fusion techniques of visual odometry and RTK to get a good yaw estimation for in row navigation.

Meanwhile, Aman has also integrated the map builder into the navigation pipeline. Whereas Aaditya finished exposure testing code for collected data.

Hillel has been working on multi-row clustering for GUI for monitoring pipeline.