Progress Review 7

Individual Lab Report 6

Aaditya Saraiya

Sept 11, 2019

Team E:

Wholesome Robotics

Teammates:

Aaditya Saraiya

John Macdonald

Dung-Han Lee

Aman Agarwal

Hillel Hochsztein

Individual Progress

Capstone Project

Team E is creating an organic monitoring robot which has to autonomously navigate through crop rows. For this phase of the project, my task is to conceptualize and initiate the design process of a plant health monitor tool. The plant health monitor tool is intended to present plant health information correlated with geometric location information after a particular monitoring run at Rivendale farms.

Pipeline architecture conceptualization

As an owner for this work package, the first step was to create a software architecture which would guide the development of this tool. An accelerated systems engineering process was followed to identify the key customer requirements and potential high level features.

The first step was to list down the **customer benefits** associated with this tool. The identified benefits to the customer are as follows:

1) View relevant plant health information efficiently

2) Give user-driven feedback to provide expert supervision to the deep learning based detections.

The second step was to identify the **high level features** which are required from this tool which can be summarized as follows:

1) **2D grid representation of plant health data -** Display plant health information collected correlated to its geometric location. This can be done be representing healthy plants with one color and unhealthy plants with another color.

2) **User in the loop –** Allow the user (aka the farmer) to modify the disease level for a particular pixel. This is done in order to remove false positives/ outliers which may arise due to edge cases with the deep learning pipeline.

Some other high level features were also recognized, which have been termed as optional for now. Depending on the time, these features could be explored in the future.

Optional high level features

1) **Integration with Google Maps Static/ Dynamic API**- This feature will enable the plant health monitor to be deployed on the web.

2) **Generate analytics based on location**- Given a 2D grid with geometrically correlated plant health information, generate analytics and path plans on regions the robot should survey.

Subsystem breakdown

The next step involved breaking down the high level features of the system into sub-systems. The main subsystems identified from this process are as follows:

1) **ROS Bag Parser**- This subsystem takes a ROS Bag as input. This ROS Bag consists of location data and image data associated with their related timestamps. The location data was obtained using an RTK GPS in one of our previous runs. The image data consists of the right and left images from the stereo camera. Depending on a user choice, either the left or right images are taken. The data association step looks at sensor data published at the same time stamp and converts them into a dictionary mapping. The RTK GPS data publish data is higher than that for the image data. Hence, all the images are associated with a location for sure (unless the RTK GPS stops working).
2) **Mask R-CNN Pipeline-** The Mask- RCNN pipeline has been developed as a part of last semesters work. The output labels from the deep network pipeline will be taken as input to the map visualizer.
3) **Map Visualizer-** The map visualizer blocks takes the correlated geometric and image information and converted into a visual representation.

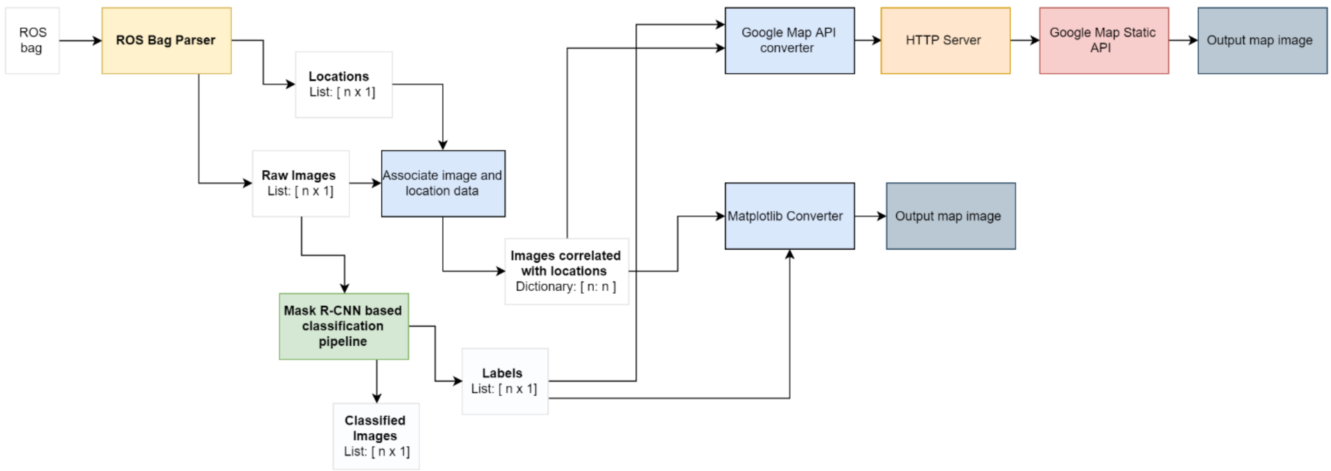

Figure 1: Subsystem diagram for the Plant Health Monitor Tool pipeline

Implementation

This section focuses on briefly summarizing the parts of the Plant Health Monitoring which I have worked on. The steps involved in implementing the components can be summarized as follows:

1) A candidate ROS Bag was chosen. The ROS Bag has both RTK GPS Data as well as image data.
2) The ROS Bag API was used to extract GPS messages and image messages separately. As the data is stored serially in the ROS Bag, the first image data comes up after 6-7 GPS data samples.
3) The CVBridge was used to convert from sensor_msgs/image type to OpenCV images.
4) The code will be adapted in the future to run with independent command line

parameters and will be called with an overarching bash script for generalization.

Challenges

During the design process as well as initial implementation work, there were several challenges which were faced. On further thought, the time delays due to these challenges could be easily prevented.

The **first** challenge was creating a software architecture without looking up the ROS Bag data on hand. During the initial discussions, the conclusions were that we had two separate ROS Bags with location and image data and they were not time-synced. A good portion of my time was spent on researching time-syncing ROS Bags. Some parts of the software architecture design also revolved around having two ROS Bags. However, on further examination of the data, we realized that the ROS Bag had both GPS and image data in one bag. This meant that there was no need to do any time-syncing. **Secondly**, we wished to use the Google Maps Static API and visualize data on the web. However, the Google Map image provided by the API does not correspond to the current state of the farm. Hence, without new data, using the Google Map Static API may not provide the correct semantic context for the farmers. **Thirdly**, the time for integrating software components with multiple people working on the package takes time. A common meeting time between the three involved members has been created in order to facilitate faster communication.

Teamwork

John Macdonald

John worked with Dung-Han Lee to clean the code for the monitoring pipeline. He also evaluated the requirements of the RTK GPS based navigation node.

Aman Agarwal

Aman worked on the initial steps to restructure and clean up code for the controls part of the pipeline. He is the current project manager and focused on organizing the discussions which we have during team meetings.

Hillel Hochsztein

Hillel worked on developing the visualizing module for the Plant Health monitor tool. He created a utility to display healthy and diseased plants in different colors. He also developed a quick GUI tool to allow user to change the value of health marker from user input after showing the relevant image taken for the plant. He also put some initial research work into how the pipeline could be integrated with the Google Maps Static API.

Dung-Han Lee

Dung-Han Lee worked on the cleaning up the labeling code, with John as the reviewer of the code. Secondly, he worked on assessing the deep learning pipeline as a binary classification problem instead of a regression problem.

Future Plans

Team

The future team goals for the upcoming weeks can be summarized as follows:

1) Cleaning up and productizing the control code
2) Get perform evaluation of the recently trained deep learning model on 4 different plants.
3) Create the integrated version of the ‘plant health monitor tool’. This involves integrating the ROS Bag parser, visualizer and the detection pipeline.

Individual

The future individual tasks which are planned for the coming weeks can be summarized as follows:

1) Integration of the Visualizer and the ROS Bag Parser pipeline.
2) Adapting the testing step of the already existing detection pipeline with the Plant Health monitor tool.
3) Research on methods for dynamic exposure handling to solve the under-exposed and over-exposed images problem with the current data collection approaches.