GPIG-A Formative

March 4, 2024

1 An outline of the proposed technical solution

We are concerned with the growth of destructive plants within a forest, these are plants who outcompete their neighbours, make extinct other plants, reduce biodiversity, and alter habitats. We propose a solution that uses drones to monitor and capture image data of a given forest, each image is handed back to a central system where we identify the plants present in the image, and store that along with the latitude and longitude. This system is generic enough to work in any forest, but has the option to train it against specific plants that you are concerned about, such that it can give deeper insights and recommendations. Our user interface allows an operator to:

- See the locations of each drone,
- Command the cluster of drones to move to a new location,
- See which areas have most recently been scanned,
- Chat with a large language model about the data obtained.

The plants within each captured image will be clustered such that similar looking plants are given the same ID. For each plant species that we are especially concerned with, p, we allow the training of a plant classification module $f(s)- > [0, 1]$ mapping a series of images, s belonging to a given id onto the probability that they are of species p. This would allow us reasonable confidence that our plant id maps to a given species. We can then use this to make more informed recommendations.

By tracking the quantity of each identified plant over time, we can see if any are undergoing exponential growth. Using a button on our user interface, this data will get wrapped into a pre-made prompt that can then be sent to a large language model. The pre-made prompt would say which plants are growing and where, if we have it, the predicted plant name, the drone scan history, the location of the forest, and would ask for advice on how to handle it. The operator can then have a conversation with the large language model where they try to plot the best course of action going forward.

2 A brief description of the system that will be prototyped (including assumptions, capabilities, primary functions)

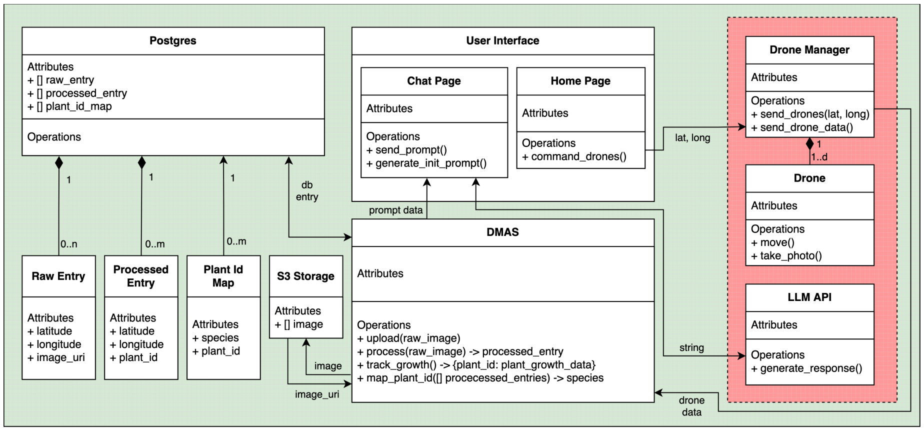

Figure 1: The block diagram of our design. Our system uses a microservice architecture containing our User Interface, Drone Manager, LLM API, DMAS, and Postgres micro services.
Key: Green = will implement, Red = will mock/simulate

As you can see in fig. 1, the following systems will be present in our prototype:

- The datastore and data analysis,
- The plant clustering and id assignment,
- An example plant classifier module,
- The user interface,
- Drone simulation,
- Large language model communication.

The datastore will be a PostgreSQL database containing three tables:

- Raw image entries containing columns for latitude, longitude, camera pose, and the corresponding image URI,
- Processed entries containing columns for latitude, longitude, and plant id,
- A map from plant id to species.

We assume that this should be sufficient for the final product.

We aren't currently sure how exactly we will do the plant clustering and id assignment. If using k-means clustering we would have to carefully define a value of K for each forest, this wouldn't be ideal as it reduces the genericness of our system. We will need to look into clustering techniques that work with variable cluster quantities. Our plant classifier module will take a cluster of images and predict whether they belong to the species Japanese knotweed, to do this we will need to gather a large number of images of the plant to use as training data. We assume that this will be possible.

The user interface will be created using react and we will prioritise the functionality over UX/UI design.

Drone simulation will be done using ROS2 and Unreal Engine We assume that the code in the simulation should run on a real robot in a forest, but this will need to be tested and verified as simulation is only an approximation of the real world and there are many factors that we can't account for. Because of this we will use dummy data feeds to simulate what the drones would send to the actual system. Without significant effort, the images captured in simulation won't be similar to the ones we would capture in real life, therefore, we will create a fake node that can deliver data to our datastore.

Our large language model communication will be handled as follows: Take the data from our system, format it into a prompt, send that to an LLM, forward response to user interface. In the final product this could use a specially trained LLM or a fine-tuned LLM, but for our prototype we will hook into GPT-3.5 using their public API

We have containerised our prototype so that functionality is split into microservices, these can be worked on independently by each team. Each microservice is containerised and the entire project can be ran using docker compose.

3 An overview of the system aspects that the team intends to demonstrate in the group

In our presentation to the customer we intend to demonstrate the following aspects of our system:

1. The coordination of drones to scan specific areas,
2. A simulation of how our system responds when it discovers that plants are growing destructively,
3. A conversation with our large language model.

To demonstrate the coordination of our drones we will start at the home page of our user interface where the user is presented with a map. The demonstrator will then select an area on the map and the interface will provide a pop up asking if they want to direct the drones to that location, the demonstrator will accept. The user interface will then show the drones travelling to the new area, and we will show a monitoring web page that shows us tracking the new plants we have discovered.

To demonstrate how our system responds when it discovers that plants are growing destructively, we will restart the system and feed it with a testing data set where one species has been growing exponentially, and is outcompeting its fellow plants. We will show how the system presents this data to the user and compare it to a situation where the plants do not grow destructively. This will also demonstrate how we test our system.

To demonstrate a conversation with our large language model we will feed our system the same destructive species test data set but this time the demonstrator will select the button that allows them to have a conversation with a large language model about the results. We do not plan on implementing the large language model for the prototype, so this will be done by having a pre-generated script.

4 A summary of progress

5 A brief risk register

Risk	Likelihood	Effect	Mitigation and Contingency
Team member unavailable for meetings	Medium	Low	We will host the meeting when the maximum number of team members are available and provide minutes for those who cannot make it.
Team member unable to complete work	Low	Medium	Deliverables of the projects are assigned to at least two members. We will have stand up meetings to update the team on progress.
Loss of report	Low	High	The report is done in Latex, we will have store this in GitHub and use version control so that we can roll back any mistakes.
Loss of code	Low	High	The code is stored in GitHub, we will use version control so that we can roll back any mistakes.
Team member misunderstands their task	Medium	Medium	We will collaborate when defining what tasks need completing, this means that every member should be aware of what is needed. Having two people per task should also make this less likely.

Team member unavailable for a significant period of time.	Low	High	We will reassign tasks to other team members as necessary/based on task importance. We will also inform the module staff about the situation.

Table 1: Risk Register