

EE/CprE/SE 491 Weekly Report

11/12 - 11/18, 2025

sdmay26-20

AI Integrated Drone for Smart Crop Health and Growth Monitoring

Faculty Advisor/Client: Md Maruf Ahamed

Team Members:

- Jongmin Lee, jongminl@iastate.edu - AI
- Z Harvey, zdharvey@iastate.edu - Frontend/Backend
- Andres Lopez, andresl4@iastate.edu - Hardware/Firmware
- Alex Pallan, apallan@iastate.edu - Cloud/Security
- Camden Senneff, csenneff@iastate.edu - Frontend/Drone SW
- Caleb Zea, calebzea@iastate.edu - Frontend/Backend
- Srdan Kopunovic, srdank@iastate.edu - Backend/Cloud

Summary of Work:

During the past work period, the team made progress in major categories of the project overall. The frontend team integrated and tested all necessary components to retrieve and send data to the API layer, while also implementing the functionality to select portions of the map through the UI, which will be used to create flight plans. The backend and cloud section successfully deployed the program to the team AWS server, and made progress in setting up the project environment through the AWS service.

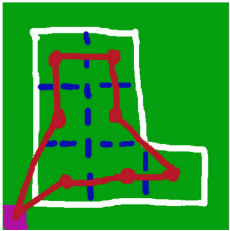
We have created the Python API layer that will connect the backend to the AI service that will process the image data and send it back for the UI to display the identified locations. The next steps for this portion is to properly set up the connection, and test the data flow between the separate components.

Our AI section created a proof of concept that displays the functionality of the trained AI model to identify crops that have root growth around them. This is an important first step as it allows us to test the functionality of our whole system integrated with the different components and services. Once successful implementation, the next step for the AI program will be training it on the data related to our project.

Work Period Accomplishments:

- Frontend
 - Updated API calls to the backend to properly communicate with the database
 - Added basic AWS Cognito information into the frontend app so that user authentication and logins can be added later
 - Added polygon and grid overlay for field, allowing for dynamic field sizing and sets the foundation for flight path, pictures, etc.

AgDrone Control Panel



Crop Health:

0 warnings

[Click for more details](#)



Flight Data:

Battery use: 75%

Flight time: 30 minutes

Field coverage: 90%

Hazards: 0

[Home](#) | [Login](#) | [Flight Path](#) | [Crop Health](#) | [Flight Data](#) | [Crop Pictures](#)

Login page

Username

Password

Submit

[Home](#) | [Login](#) | [Flight Path](#) | [Crop Health](#) | [Flight Data](#) | [Crop Pictures](#)

Flight Data Page

Flight 1

```
{
  {
    "flightId": 1,
    "flightDateAndTime": "2025-12-01T16:00:00"
  },
  {
    "flightId": 26,
    "flightDateAndTime": "2023-01-01T12:00:00"
  },
  {
    "flightId": 27,
    "flightDateAndTime": "2023-01-01T12:00:00"
  },
  {
    "flightId": 28,
    "flightDateAndTime": "2023-01-01T12:00:00"
  },
  {
    "flightId": 29,
    "flightDateAndTime": "2023-01-01T12:00:00"
  },
}
```

[Home](#) | [Login](#) | [Flight Path](#) | [Crop Health](#) | [Flight Data](#) | [Crop Pictures](#)

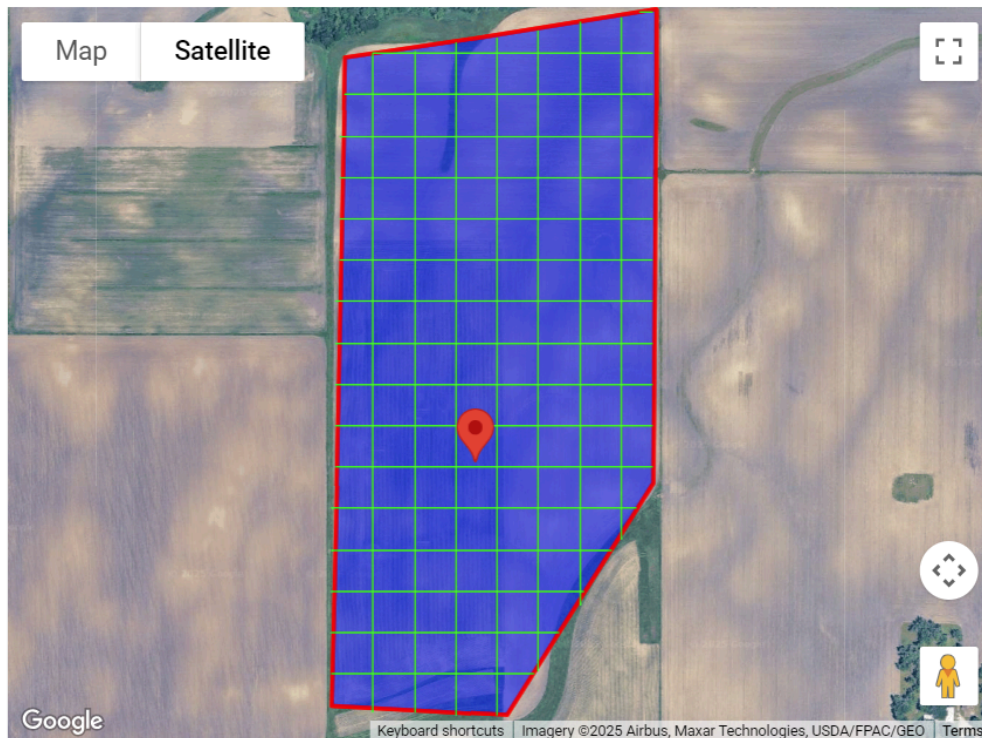
Crop Pictures Page



Image 1

Longitude/latitude here

Field and Flight information



- Backend
 - Created the Python worker service that consumes images that need processed from the backend's queue and reports data back to the backend. Needs automated testing and CI, and integration with the AI work.
 - Implemented the API layer that will communicate with the frontend to receive image data, forward it to the Python service, provide continuous updates on processing status, send completed data to frontend, and save it to the database.
 - Created Integration tests that verify the functionality of the API layer, ensuring all endpoints are returning the expected results, and the data is properly being processed.
- Cloud
 - Worked on getting a database on AWS up and running
 - Shared with the group some documentation and ways to register and log in with accounts using AWS cognito
 - Started working on a concept for a backend to run in AWS to link with the

database

- Drone



- Begun working with a Raspberry Pi and camera to get some testing done on one before the drone parts arrive
- Only have researched camera functionality
- AI
 - Prepared structure for **crop-specific growth estimation models** (corn & soybean).
 - Researched **public UAV datasets** for corn/soybean classification and growth-stage estimation.
 - Organized approach for building a **crop-type classifier** that works with UAV drone images.
 - Evaluated dataset availability issues and explored alternatives (international collaborators, spring–summer data collection).

AgriVision: Crop vs Weed Detector (YOLOv8)

Upload an image

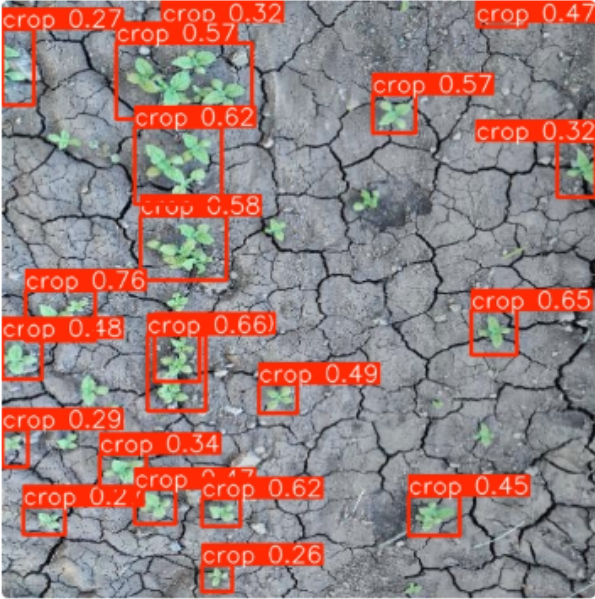


Drag and drop file here
Limit 200MB per file • JPG, JPEG, PNG

Browse files

 agri_0_3964.jpeg 77.3KB





Plans for Coming Week + Action Items:

- Jongmin Lee - I will finalize the public datasets we'll use, start training a baseline corn-vs-soybean classifier on Nova, and organize the experiment setup with clean train/val/test splits. I also plan to connect the model format to the AI microservice, define evaluation metrics, and prepare a simple demo that shows how growth percentage predictions will look.
- Z Harvey - Continue to add onto the grid overlay that I got done this week. It was a huge undertaking but with the understanding I've gained I think that we can do a ton of things with the Google Maps Javascript API. I will be focusing on usability for the user. I want to make it so that a user can enter their address or latitude & longitude, a map instance is created, and then from there they can outline and create the polygon themselves of their field. This polygon they made would be stored in the backend for future reference so they don't have to initialize their field over and over again.
- Andres Lopez - Making headway into capturing an image and testing to see if the image has been captured and the data is not corrupted, while writing the file onto the SD card to move over.
- Alex Pallan - Get a framework for a containerized backend to run using ECS to work well with the postgres database in AWS, make a test program using AWS authentication through Cognito
- Camden Senneff - Complete tests for Python image service and get integrated with the actual APIs. Create a data partition in Pi images and test that the service starts properly on the Pi.
- Caleb Zea - Further comprehend and implement frontend-backend communication (API calls, authentication, etc)
- Srdan Kopunovic - Have a working user system that allows us to separate the data displayed based on the user logged in. Have the frontend fully connected to the AI microservice through the backend API layer.

Pending Issues:

- Jongmin Lee
 - Lack of real UAV crop images from Iowa for accurate growth-stage modeling.
 - Not yet finalized which baseline model will be used (YOLOv8/v9, EfficientNet, ViT, etc.) for classification vs segmentation.
 - Need clarity from the team on what image resolution the drone will capture so preprocessing pipeline matches.
- Z Harvey
 - Getting a second Google Maps instance to pop up, and then dynamically loading the regular one based on the polygon that the user drew. Also have to implement the Google Maps Address API for this to work.
- Andres Lopez
 - Looking at this libcamera library has deep functionality to capture an image
 - Working with the Pi is frustrating having to swap between my mouse and keyboard as well as the Pi not detecting the camera
- Alex Pallan
 - Some trouble getting the postgres database on the free tier for AWS
 - Managing authenticated sessions and specific user permissions for Cognito is a bit more complicated than I thought it would be
 - Linking the backend API to the database running on AWS
- Camden Senneff
 - The Pi image creation is complex, and getting a data partition reliably may be difficult.
 - Image APIs are not yet set in stone, so the Python service may need to be reworked.
- Caleb Zea
 - Reading through the Swagger docs for the API, having never used Swagger before
- Srdan Kopunovic
 - Communicate with the frontend to get the authentication system working properly, and figure out how the API will communicate between the frontend and AI service to continuously provide updates on processing status.

Individual Contributions:

Team Member	Contribution	Week Hours	Total Hours
Jongmin Lee	Analyzed UAV-ready dataset to ensure compatibility with our drone hardware and farmer dashboard. Define a roadmap for implementing growth-percentage detection.	10	50
Z Harvey	Completed the first step in the flight path UI. Was able to overlay additional JS ontop of the Google Map instance, which outlines the field and a grid system so that the user can see their field, the grid, and eventually, the flight path. This is the foundation for the rest of the flight path UI, as there is proof of concept for our ideas. Design 5 document work as well.	8	45
Andres Lopez	Researched the libcamera library for Raspberry Pi to begin implementing writing a mock library for testing when not using the Pi as well as code to capture an image	5	25
Alex Pallan	This week, I added some documentation regarding the Cognito service we are using with AWS. I also made some modifications to the database to reduce cost. I also spoke and worked with Srdan regarding getting the database online and using terraform to output the correct endpoint/instance information.	8	40
Camden Senneff	Built the Python worker service that will consume images from the backend's queue and spin up AI processing threads, then report data back to the backend. Some time working on CI/dev experience improvements in the backend.	8	56
Caleb Zea	Assisting with AWS MFA setup, AWS Cognito, API Testing, fixing broken API requests and routing, design doc 5 work	8	60
Srdan Kopunovic	Worked on deploying backend on AWS server. Worked with frontend to plan out revisions for the existing data models for future needs and to mitigate issues that might exist. Implemented the API requests that will get the images from the frontend, forward to AI service, constantly update status on the processing, and receive images to send back to frontend upon completion	7	57