

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 - Frontend/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 - AI/Drone SW
- Caleb Zea, calebzea@iastate.edu - Frontend/Backend
- Srdan Kopunovic, srdank@iastate.edu - Backend/Cloud

Summary of Work:

Throughout this extended work period, progress made involved the user creation and authentication on the frontend, as well as map refinement to allow users to specify points to create a field for reuse on their account, and some visual improvements on the UI overall.

The drone parts were delivered to the university finally, and the hardware team started assembling the full drone, and installing the required software. The next step is to thoroughly test the flight capabilities and ensure proper component functioning, after which we will begin customizing the flight logs to retain the data in the format needed to forward to the frontend upon flight completion.

The backend system implemented Cognito user authorization, which will be the main form of security protection for the program. Incorporated users within the database in order to track different parameters per account, as well as ensure valid requests and additional error handling. Implemented an option for users to create and save fields for reuse, as well as created the data flow for images between the client, backend, AI service, and database

Our AI service has an implemented API that needs to be tested with the backend to ensure proper processing, and transfer, with the end goal of updating each data point parameter, and storing the images on the team server.

For the frontend focused on laying down the technical foundation for both the Crop Health frontend interface and future model integration. I created the initial React component structure and routing for the Crop Health page and addressed dependency issues that were blocking compilation. I also planned the page layout and placeholder logic to ensure the UI will be ready once ML outputs are available.

- Frontend:
 - Updated AWS Cognito version to support the new SDK
 - Added further Cognito integration
 - Implemented a login/sign-up page, and added secure encryption and decryption of entered sensitive information between the input fields and the backend API call
 - Fixed misaligned CSS due to nav bar changes

Work Period Accomplishments:

- Frontend
 - Set up the initial structure for the CropHealth page, including creating the core React component and preparing the layout where future model-generated insights (growth %, pest/disease indicators, health visualization) will be displayed.
 - Added an additional layer of the Google Maps API, allowing the user to enter their address (or an address near their field) and it will create a google map instance centered at that location.
 - Finalized drawing functionality
 - Added polygon detection
 - Refactored all of the field logic and classes into one component.
 - Implemented AWS Cognito user authentication and integrated it with the backend API calls to ensure better data security
 - Added a shim to the Cognito v3 implementation in order to bypass a dependency depreciation, in order to correctly point the functions to the correct dependency location
- Backend
 - Created the Image model that has a one to one relationship with a unique location per flight data block. The model contains the following attributes:
 - LocationImageId
 - LocationId
 - FlightDataId
 - OriginalImagePath
 - ProcessedImagePath
 - Description
 - Created Service layer functionality for Image Processing, using the following methods:
 - AddLocationImageAsync:
 - Receives an array of LocationImage objects
 - Creates or finds specified image path where images will be stored on server
 - Stores image paths in database
 - GetLocationImageAsync:
 - Sends processed Images to client
 - Sends Image data to client
 - ForwardImagesAsync:
 - Forwards Flight Images from client to AI service
 - AddProcessedLocationImagesAsync:

- Receives processed images from AI service
 - Saves ProcessedImagePath to database
 - Updates data for LocationImage model
- Created API layer for Image transfer between client and AI service
 - Receive Images from client
 - Forward Images to AI service
 - Get status on image processing job
 - Receive processed images from AI service
 - Send processed images and data to frontend
- Created Field Model, used for saving field data per user and reusing fields for surveys. Implemented Service layer for Field data transfer:
 - Creating fields
 - Getting field data based on an ID
 - Getting all the fields data based on user ID
 - Updating a field based on ID
 - Deleting a field based on ID
- Cloud
 - Created User Identity Pool
 - User Pool (created before) is the list of users (much like a directory)
 - Identity Pool is the “roles/permissions” of the end users, allowing them to access AWS services, as well as segmenting individual data so they can only see their own information
 - Cost Optimized the AWS environment and educated others on how to keep cost in mind when developing
 - This included commenting out the terraform for the database when we are not using it for development at this point
- Drone
 - Received drone parts.
 - Assembled Pi hardware and identified existing issues with the Raspberry Pi image generation.
 - Updated image to create a data partition and fixed bugs. This allows us to transfer data between the drone and frontend.
 - Have non-tested image capturing drone software

- AI
 - Analyzed what gaps remain in the dataset for building crop-specific growth detection models. Outlined potential requirements for future datasets (corn/soybean growth stages, pest examples, disease samples) and how these will align with the UI's crop-selection workflow.
 - Investigated approaches for future growth-percentage estimation, including how model outputs might be structured (bounding boxes, segmentation masks, stage classification), and how these outputs will later integrate with the CropHealth UI.

Plans for Coming Week + Action Items:

- Jongmin Lee -
 - Frontend
 - Finalize the CropHealth page structure by breaking it into modular components, so each can later be connected to backend/model responses with minimal refactoring.
 - AI
 - Document the current model's performance and write a short internal note explaining strengths and weaknesses, and next steps for improvement.
 - Create a mock model-output JSON structure (bounding boxes, labels, confidence, growth percentage) that the frontend can use for early UI integration before the real models are completed.
- Z Harvey - Add connecting pieces with user login, want to store fields that users create in backend, add some more checks (asking the user if they want to finalize their fields, add more direction for first time users, and also have an additional screen to allow them to select their existing field or to add a new one.
- Andres Lopez - Test the image capturing for the drone software I've written then look both into flight automation as well as the machine learning aspect of observing the growth of crops.
- Alex Pallan - Get the ECS infrastructure in place so that when Srdan is ready to containerize the backend application, it is pretty much plug and play. This will include writing the terraform for the ECS cluster, the ECR and the process to upload that code and update the repo, and the task(s) themself(f/ves). Also work with the identity pool to manage keeping the data for each user to keep them separate.

- Camden Senneff - Get the Pi image fully working on the Pi, including the installation of the Pi dependencies we're currently missing and fixing the architecture used in the pilot binary file. By the end of this week, the image will be fully complete for testing directly on the Pi with the production system configuration.
- Caleb Zea - Test frontend -> backend authentication communications and ensure that users can log in and sign up with new accounts. Begin work on drone -> frontend -> backend image transfer
- Srdan Kopunovic - Test Image transfer flow between frontend, backend and AI service. Ensure proper image persistence based on data provided, and optimize the transfer logic and data flow handlers. Test processed image updates are properly handling AI service data updates after fully completing processing.

Pending Issues:

- Jongmin Lee
 - No finalized design guidelines (colors, typography, spacing, component patterns), making it difficult to guarantee visual consistency across new pages.
 - Crop Health page still lacks real model output, so UI testing is limited to placeholder data until the AI pipeline is connected.
- Z Harvey
 - Communicating with the backend properly and integrating my code into how user auth works. No huge issues though, should be easy to resolve. Add a way to delete the way points if a user misclicks. Prompt user for name of field and store that too
- Andres Lopez
 - The image capture code has not been tested yet
 - What machine learning model can take into account the growth of crops? Future Andres will investigate...
- Alex Pallan
 - Working with front/backend to figure out where exactly we are with authentication as far as functionality
 - Making the Identity Pool functional

- Camden Senneff
 - The Pi image generator has some issues (which we were now able to discover since we have the Pi): the binary is being built for the wrong architecture, we're missing dependencies for WiFi and the Pi camera, and the Pi-native libraries aren't getting linked to the binary correctly.
- Caleb Zea
 - Need to merge prior frontend page branches into main so we have a working model of what the frontend is looking like for the senior design presentations
- Srdan Kopunovic
 - Update the file system logic for saving images to server in order to be more robust and properly handle what is needed for our use case.

Individual Contributions:

Team Member	Contribution	Week Hours	Total Hours
Jongmin Lee	established the foundational structure for the Crop Health page, set up routing, and resolved module import issues to support multi-page navigation. I also began outlining reusable UI components and designed placeholder states for future model-driven content.	6	56
Z Harvey	Added additional layer and functionality to the Google Maps API. Added a prompt so that a user can enter an address near their field, and it will create a google maps instance centered at that address. Added drawing functionality so that once the user finds their field they can draw it. Added polygon detection so fields can be any shape and size. Added UI elements to guide user through the process more and a grid overlay button.	7	52
Andres Lopez	Followed libcamera documentation guide for C++ development of setting up and capturing images on a Raspberry Pi camera. The code has not been tested...	5	30
Alex Pallan	Optimized the terraform code to be more cost effective going forward by commenting out the database when not in use. Also laid the foundation for user permissions and data segmentation going forward.	8	48
Camden Senneff	Received parts and ensured everything was present. Got the generated images running on the Pi, and added image resize and data partition generation on first boot. Fixed some issues with the custom services injected into the image – still some more to do.	5	61
Caleb Zea	AWS Cognito basic setup. Updated Cognito from v2 to v3. Implemented a shim to bypass a buffer depreciation in the AWS SDK. Implemented encrypting and decrypting of usernames and passwords between the entry fields and database authentication. Fixed nav bar wrapping and misaligned content body. Reviewed backend API merge request to understand frontend -> backend communication better	8	68
Srdan Kopunovic	Created Model that represents the images per location of a flight, implemented the image transfer service to handle storing images on server and storing to database, and implemented the API layer for communicating with client, and AI service.	10	67