TEXAS A&M UNIVERSITY
Engineering

Cotton Drone Analysis

Booth #314

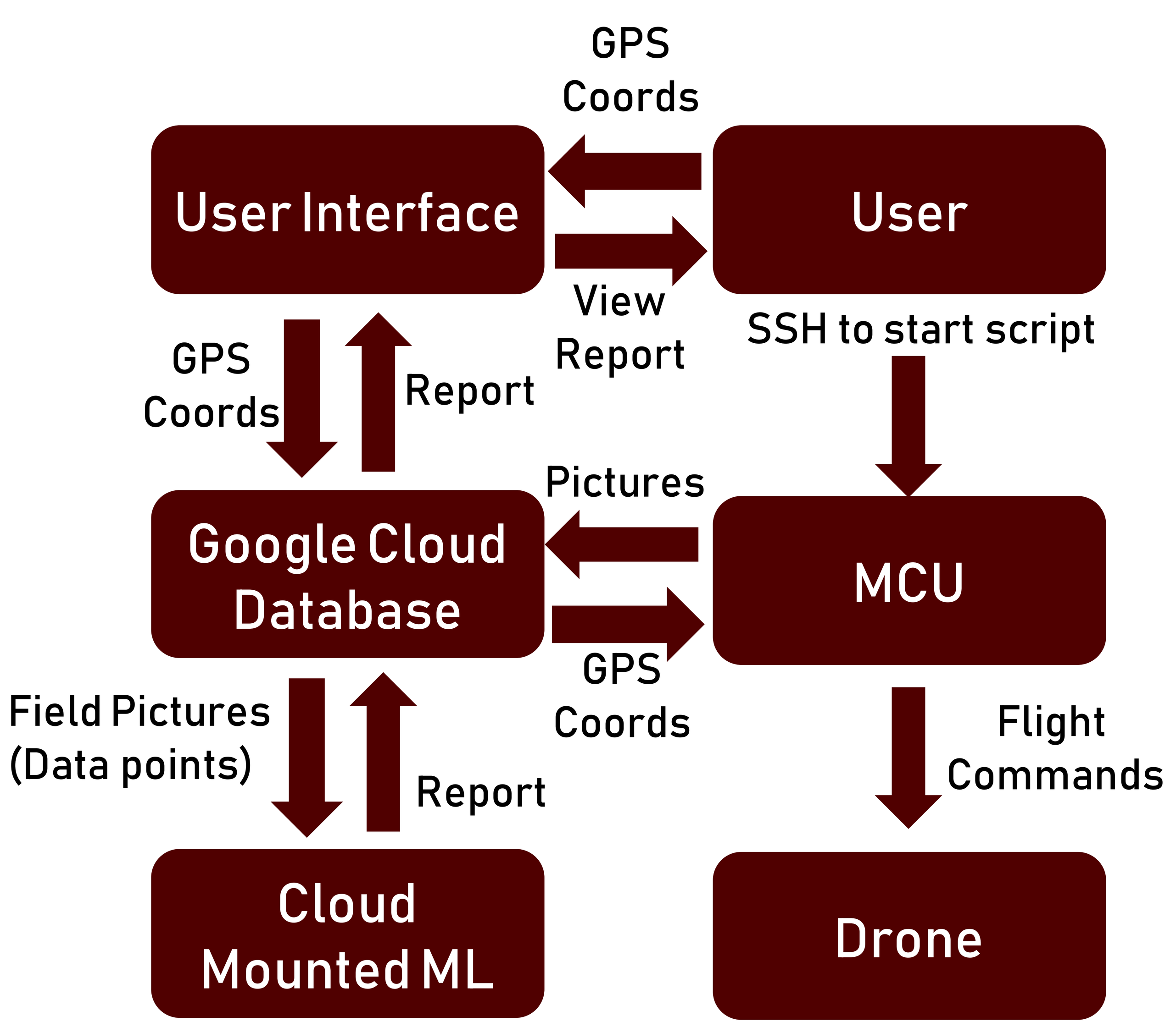
Sam Fabricant, Justin Whitehead, Luis Sanchez, Damilola Owolabi
Faculty Mentor: Dr. Nowka (ECEN)

ECEN

Project Objective:

The objective of this project is to develop an automated system to capture images of cotton fields, analyze them with a machine learning model to predict the crop harvest date and yield, and display the ML analysis on a user interface application. Our goal is to minimize the user interaction and technical prowess needed to collect and process cotton crop data for Texas A&M AgriLife researchers.

System Overview:



The system begins with the user making an account on the user interface app, and enters GPS coordinates of their field which are then sent to the cloud database. When the user is ready to image their field, they SSH into the Raspberry Pi MCU, initiate the flight script, and enter their account user name. The flight script then runs a flight path algorithm, autonomously flies the determined path while imaging the field. Once finished, the drone lands and offloads images back to the database. Next, the machine learning model analyzes data points from the pictures and creates a report predicting yield and harvest date for the cotton field. Lastly, the report is viewable to the user on under their account in the user interface.

Note: This is a legacy project, where the last team programmed the orthomosaic stitching and processing of images to pull the necessary data points for ML analysis. This process is not included in our project.

Methodology:

Drone Imaging System:

- Autonomous drone flight with Raspberry Pi companion computer running Python-DroneKit API scripts
- Raspberry Pi communicates to drone via serial wire communication. The user initiates flight script via SSH to Raspberry Pi with ethernet cable
- Flight path mapping algorithm developed using developed mathematical GPS functions

User Interface Application:

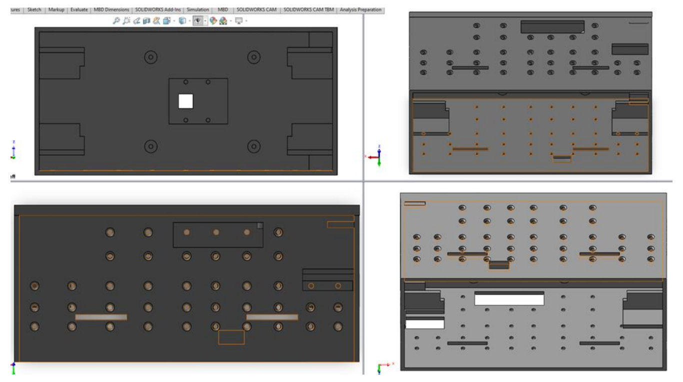
- Website created using Django framework
- User Interface deployed to Google Cloud for global accessibility
- User Interface interacts with the database by uploading commands and extracting ML generated outputs.

Machine Learning Programming:

- Digital twins generated from previous years of data from TAMU AgriLife research team
- AI trained with a dense neural network model in TensorFlow and mounted to Google Cloud.
- Virtual machine instance scans user data on automatic startup and sends data to AI

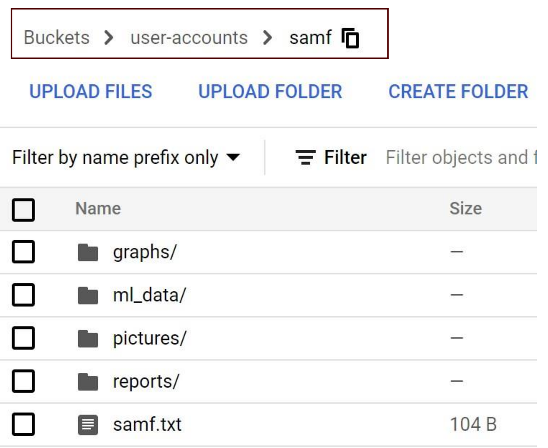
Drone Enclosure and Power Supply:

- Drone enclosure designed on SolidWorks and 3D printed at FEDC. Designed to facilitate Raspberry Pi / Drone interfacing and mount Pi Camera
- Power supply provides Raspberry Pi with over an hour of battery life.



Cloud Storage System:

- The subsystems are connected and share information through a cloud storage database.
- Each registered user has their own folder in the database, that is populated with files when they register their account and input their field GPS coordinates.



Results:

Drone Flight Control and Imaging Programming:

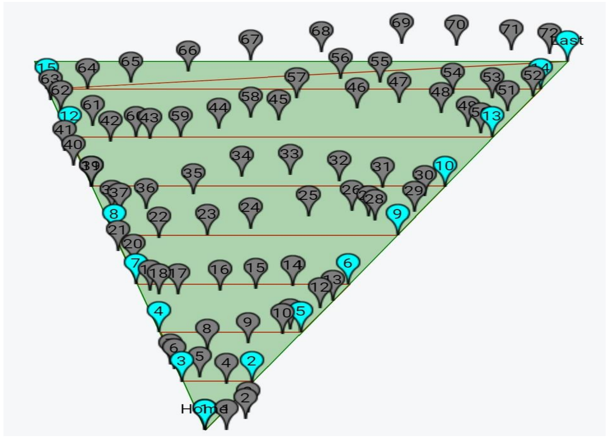
Flight Path Algorithm:

- The flight path algorithm accurately determines a flight path to fly over the entire field given 3 to 5 GPS coordinate inputs

Drone Flight Image Capture:

- The flight script captures images with the Pi camera during flight. The images are triggered based on the drones current location, timed to image the entire field. The figure below shows the location of the drone at the time of each picture.

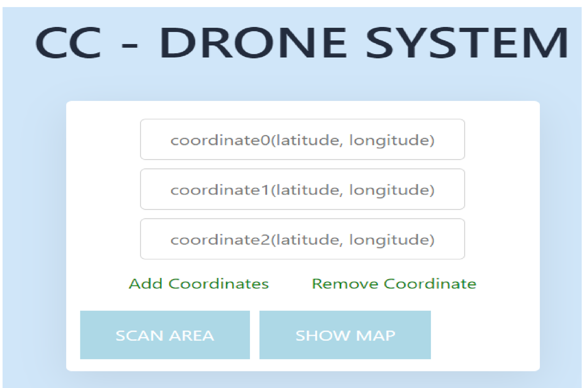
Green Shading: Cotton Field
Blue Markers: Flight Waypoints
Gray Markers: Location of Images



User Interface Functionality:

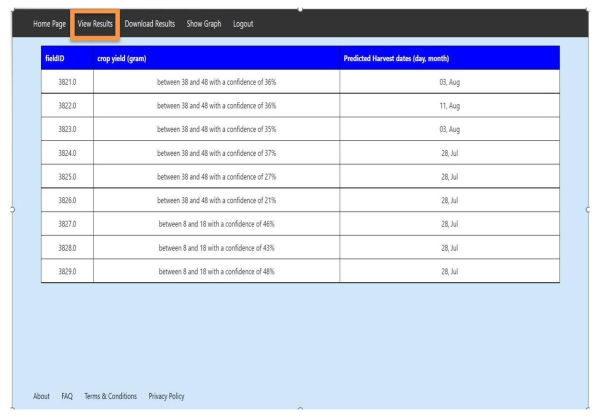
User Input:

- Inputs are received as strings, then sent as text files to the database where it will be used by the drone flight script.



Displayed Results:

- ML generated data are extracted from the database, then displayed to the user interface webpage.



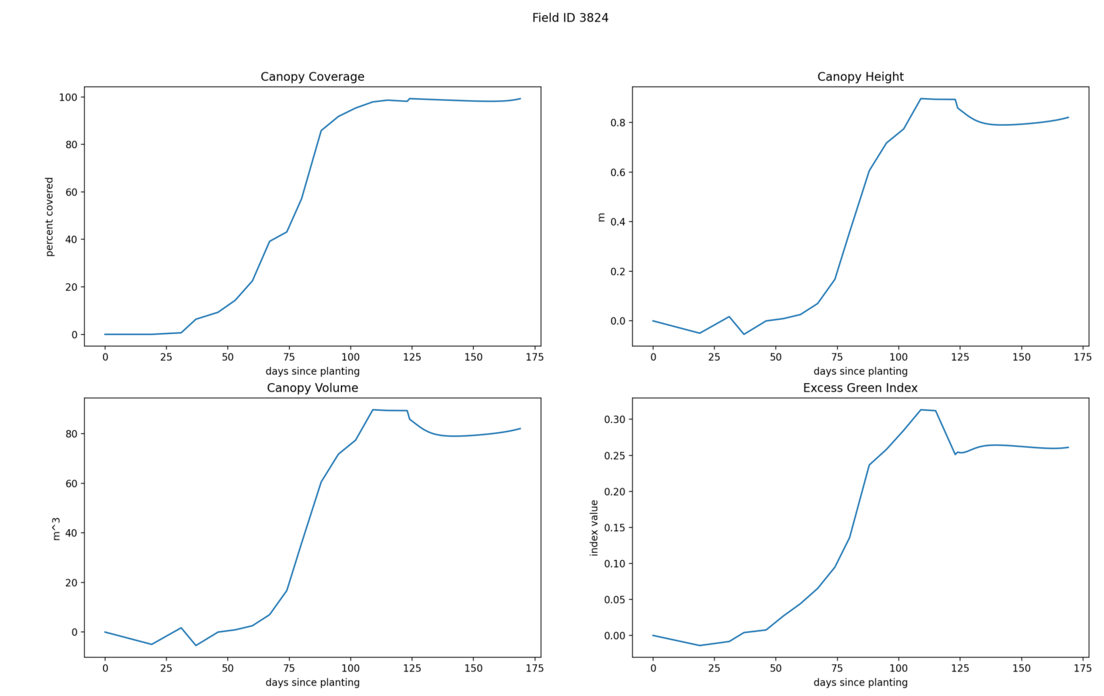
System Cloud-Communication:

- All subsystems share information via google cloud storage (GCS).
- All subsystems can upload and extract data to and from GCS.

Machine Learning Accuracy:

- Maximum ML accuracy obtained was 43%
  - Inaccuracies in original data
  - Oversimplification of a complex problem

Results (continued):



Automation:

- After imaging a field, the flight script will automatically upload images to the data-management cloud (with a Wi-Fi connection)
- Cloud mounted ML automatically scans for user data
- If data is found, a report is generated and is sent to the data-management cloud

Conclusions:

The system has end to end functionality (excluding the orthomosaic analysis). It takes in user field GPS coordinates or crop data files on the user interface and upload them to the cloud database. The MCU/drone can pull those coordinates and image the field. The crop data is automatically processed by the Machine Learning model, running from 1am – 10am. Finally, a report is generated by the ML model and is made viewable and downloadable for the user from the user interface. The user can also view and download graphs of the individual data points. The system does most of this automatically, with the only user inputs being uploading GPS coordinates and/or field data points, initiating the drone flight script, and offloading images from the drone/MCU (if there is no Wi-Fi). The main shortcoming with the overall system is the Machine Learning Model's 42% accuracy.

Acknowledgements:

We would like to say thank you to our faculty mentor Dr. Nowka for his guidance throughout this project. Additionally, we want to extend our appreciation to Swarnabha Roy, for being our FAA certified TA and accompanying us on all of our test flights.