

2.1. REQUIREMENTS & CONSTRAINTS

Functional Requirements

1. Autonomous Drone Flight – Pre-programmed routes with GPS and obstacle avoidance.
2. Plant Growth Tracking – Week-by-week imaging to monitor crop development.
3. Insect Detection – AI-based identification of pest infestations.
4. Harvest Time Monitoring – Visual and data-based estimation of crop maturity.
5. Farmer Dashboard – Centralized platform to view growth, pest alerts, and harvest predictions.
6. Reliable predictions of harvest readiness.
7. Cloud Storage – User management and file storage in the cloud, with authentication and best security practices.
8. Local Storage - Data stored locally, to be synced with the cloud database when online

User Experience Requirements

1. Ability for users to set up and configure drone flight through a cross-platform desktop application.
2. User-friendly UI that enables flight planning, field management, and data transfer after the field scan is completed.
3. Generating reports after each field scan for identified crop diseases, as well as crop growth over time.
4. Allows users to save previous field configurations and reuse them
5. Ability to use the application when not connected to the internet, as well as having data sync when a connection is established.

UI Requirements

1. Desktop application. The UI must be simple enough for any farmer to use, especially those unfamiliar with technology.
2. The UI must be robust enough to display unique field sizes and shapes, as well as corn or soybeans, and a scheduled flight time at the farmer's selected intervals. The central dashboard should allow farmers to see all important information quickly.
3. The UI should be aesthetically pleasing with a color theme, while also being bright enough to be visible on a laptop screen in sunlight.

Database Requirements

1. Local storage when offline, to sync when connected

2. Cloud storage for backup and offloading local storage
3. Data to store:
 - a. Flight Data:
 - i. Flight coordinates
 - ii. Flight time
 - iii. Identified hazards
 - iv. Pathing
 - b. Crop data:
 - i. Crop images
 - ii. GPS coordinates
 - iii. Model results (as described under AI/ML Requirements)
 - c. Telemetry data:
 - i. Battery Usage
 - ii. Field Coverage Time
 - iii. Time of day

AI/ML Requirements

1. The training dataset includes corn/soybean images across various growth stages, with diverse lighting conditions, stress, and environmental conditions.
2. Each image or image tile in the data set must be labeled according to the standardized annotation guide. The labels include:
 - a. Pest presence proxies - a visual sign that suggests insect activity
 - b. Leaf damage severity - ordinal value depends on how severe
 - c. Canopy cover percentage - proportion of the image occupied by green
 - d. Weed presence - binary (weed or no weed) or categorical (broadleaf, grass, none)
 - e. Lodging indicators - evidence of fallen or leaning plants
3. The dataset must be split into train/validation/test sets with no field or time overlap to avoid leakage.
4. Model training must be reproducible with fixed seeds, versioned datasets, and logged hyperparameters.
5. The model should provide explainable outputs (saliency/heatmaps) to highlight regions influencing predictions.
6. All datasets must follow licensing and privacy requirements—used only for research/educational purposes and PII-free.
7. The system should include an active-learning loop to flag low-confidence samples for human review and retraining.

Resource Requirements (Constraints)

1. The drone should cost less than \$750 (at the time of build).
2. The drone should be able to be built with readily available consumer parts.

3. The drone controller software should work with any standard Betaflight drone configuration.
4. Drone design should be able to gather a reasonable amount of data within energy constraints (the average field size of 350 acres on one charge) and support pausing and resuming data collection to recharge the drone for larger fields.

2.2. ENGINEERING STANDARDS

Engineering standards reinforce requirements and safety in creating designs. They are important to adhere to, as they can guide a project when followed, while also cautioning against hazards to keep not only engineers designing it safe, but also users, non-users, and the environment. They also exist to maintain complicity with regulations, morality, and the law.

Four IEEE standards that we found to apply to our project are as follows:

1. IEEE Standard for Drone Applications Framework
IEEE 1936.1-2021
Description: This standard defines all things related to drones and their potential functions. It defines all of the potential systems that drones with different use-cases would have on board, and provides categories for each different type of drone and how they are defined. Our project most likely fits under the “Agriculture” definition for our application scenario.
2. IEEE Standard Adoption of Moving Picture, Audio and Data Coding by Artificial Intelligence (MPAI) Technical Specification Connected Autonomous Vehicle-Architecture (MPAI-CAV) V1
IEEE/MPAI 3307-2024
Description: This is a standard defining various aspects of an autonomous vehicle that uses AI/ML models to navigate and interpret its environment. It defines the layouts of the sensors, as well as their types. Additionally, it defines different models for how the AI components interact with the other parts of the vehicle, as well as interaction with humans
3. IEEE Standard for a Framework for Structuring Low-Altitude Airspace for Unmanned Aerial Vehicle (UAV) Operations
IEEE 1939.1-2021
Description: This is a standard for structuring the airspace itself for the drone. This is a

more zoomed-out view from the above standard (IEEE 1936.1-2021), which is more about the drone itself. This standard defines how an airspace can be interpreted and split into regions or grids, as well as logistics regarding operating the drone. This standard, in particular, is useful because it also covers creating a mosaic of the landscape, which is a component of our project.

4. IEEE Standard for System and Software Verification and Validation
IEEE 1012-2016

Description: This standard defines the processes and documentation needed to verify that a system is built correctly and validate that it meets its intended purpose. It provides structured methods for testing, quality assurance, and performance assessment across all stages of software and system development. This standard is relevant to our project's vision model, as it guides how we verify that our AI pipeline (data preprocessing, training, inference) functions as intended and validate that the model's predictions—such as pest detection, crop growth, and harvest readiness—accurately reflect real-world field conditions. Following this standard ensures systematic testing, measurable performance metrics, and traceable validation evidence for our AI-based crop monitoring system.

After reviewing the technical details of these four standards, this is how they apply specifically to our project:

- 1) The first standard (IEEE 1936.1-2021) directly relates to our design when section five of the standard comments on different drone application types. In our case, it is information collection, transmission, and processing. There are subsections within this that also directly relate to our case, which is using the drone system to obtain target data using artificial intelligence to classify and analyze the data. The operator should also be able to monitor the drone and correct changes in flight.
- 2) The second standard (IEEE/MPAI 3307-2024) also somewhat correlates with our project, as it discusses the interaction of humans with connected autonomous vehicles. The standard states in section 5.1 that a CAV is defined as a couple of functions that our project will include. The CAV must converse with humans; in our case, a flight path must be taken based on user specifications. The standard also includes that the CAV must plan a route to reach the requested destination, and it will autonomously reach it by taking a few logical steps that are not included in our design specifications, such as building a complete digital representation of the environment.

- 3) The third standard (IEEE 1939.1-2021) is extremely relevant to our design in that it covers standards for unmanned aerial vehicles (UAVs) in low altitude airspace, under a few hundred meters, which is the space our drone will be operating in. It covers the specific data structure and notation that should be used by our drone when subdividing the area where data collection is occurring. It also notes several requirements for how closely the drone should be able to fly within the designated flight plan (for example, altitude must be within 10% of the designed altitude, route curvature no greater than 3%, rate of ascent no more than 10m/s), in addition to safety requirements before the launch of a UAV. These requirements are designed to ensure that the data collected is as reasonably accurate as it can be, which is critical for the level of precision our drone will require for useful results.
- 4) The fourth standard (IEEE 1012-2016) directly relates to our project's vision model and AI data processing pipeline. It establishes a structured framework for verifying that our software components—such as data preprocessing, image classification, and AI inference—are implemented correctly, and for validating that the model's outputs align with real-world crop conditions. This connection is critical since our system's accuracy depends on how well the AI model identifies pest presence, crop health, and harvest readiness based on field imagery. Following this standard ensures that our model undergoes systematic testing, meets performance thresholds, and provides traceable validation evidence that supports its reliability in real agricultural environments.

After discussing with other members of the group, here are some other standards we found that apply to our project:

1. IEEE International Standard - Software engineering - Software life cycle processes - Maintenance
IEEE 14764-2022
Description: This is a standard describing how to perform the iterative process of development for software. The standard can be applied to any software product, and especially takes into account the maintenance of software into account. The criteria outlined in this standard include planning and maintenance of software, as well as the execution of software maintenance activities. It aims to outline how software maintenance should be planned during software development, and then provides criteria to assess the effectiveness of software maintenance execution. The standard doesn't address the operation of software, backups, recovery, or system administration.
2. IEEE Standard for Information Technology– Telecommunications and Information Exchange between Systems Local and Metropolitan Area Networks–Specific

Requirements Part 11: Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) Specifications

IEEE 802.11-2024

Description: This long-titled standard defines how wireless local area networks operate over short distances. This standard affects how devices like laptops, phones, and routers communicate with each other while on Wi-Fi. The definitions in this standard include the Medium Access Control or MAC layer, which controls how multiple devices share a radio channel without interference with each other. The Physical Layer specifies how bits are transmitted in the air with the help of radio waves. Finally, the standard ensures that interoperability is maintained, in other words, the ability for devices from different manufacturers to be able to communicate with each other reliably. This standard is the foundation for Wi-Fi, addressing how data physically moves wirelessly and how devices work together while on the same Wi-Fi.

3. IEEE Standard for Software Engineering – Software Life Cycle Processes – Maintenance

IEEE 14764-2022

Description: This standard defines the processes and best practices for maintaining software throughout its life cycle. It focuses on planning, managing, and executing maintenance activities after the initial development phase, ensuring that software continues to function correctly and efficiently over time. The criteria outlined include documentation, modification control, error correction, and performance improvements. This standard applies to our project because the AI vision model and drone control software will require regular updates—such as retraining models with new crop data, fixing bugs, and optimizing performance—to maintain accuracy and reliability as environmental conditions and datasets evolve.

Some modifications we plan to make to incorporate these standards apply as follows:

1. IEEE 1936.1-2021 – Drone Applications Framework

This standard defines categories of drone systems and outlines the onboard subsystems required for different applications. To maintain this standard, we will:

- **Categorize our drone under agricultural operations**, following IEEE classification, and define subsystems as navigation, imaging, and environmental sensing.
- **Map system architecture** (hardware components, AI components, and communication modules) to the standard's modular framework.
- **Document system interactions** using standardized terminology from IEEE 1936.1, improving clarity for future integration with other packages.

2. IEEE 1939.1-2021 – Framework for Structuring Low-Altitude Airspace for UAV Operations

This standard provides a structured model for UAV airspace management and coordination. To maintain this standard, we plan to:

- **Design the drone's flight control logic** to comply with airspace segmentation and altitude bands, ensuring safe autonomous operation in low-altitude regions.
- **Integrate a mapping mode** consistent with this framework to standardize how imagery and field data are collected and aligned to geospatial grids.

3. IEEE 14764-2021 – Software Life Cycle Processes: Maintenance

This standard defines the framework for planning and executing software maintenance. To maintain this standard, we will:

- **Establish a formal maintenance plan** early in development, detailing packaging, architecture, and testing.
- **Use version control and CI/CD pipelines** to track maintenance activities and regression tests.
- **Adopt iterative review cycles** for updating both flight software and the AI processing microservice, ensuring reliability.

4. IEEE 1012-2016 – System and Software Verification and Validation

This standard establishes guidelines for verifying and validating software systems to ensure they function as intended and meet user requirements. To incorporate this standard, we will:

- Develop a structured **V&V plan** detailing verification steps (unit and integration tests) and validation procedures (accuracy benchmarking and field evaluation).
- Maintain **traceability** between requirements, dataset specifications, model configurations, and validation results.
- Use standardized **performance metrics** (e.g., precision, recall, and inference latency) as acceptance criteria for validation.
- Document all verification and validation evidence—including test results, error analyses, and retraining outcomes—to ensure transparency and compliance with IEEE 1012 guidelines.