Agricultural Optical Sensor Detector Co-Design

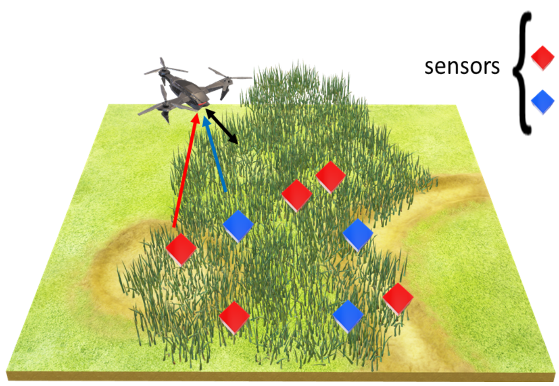
Team 6 (Inter-Departmental Project)

Advisor: Dr. Cherie Kagan (adro@seas.upenn.edu)

April 29th, 2022

IoT4Ag

The Internet of Things for Precision Agriculture
an NSF Engineering Research Center



Team Members:
Anshul Choudhari (anshulc@seas.upenn.edu) (CMPE)
Divyansh Agrawal (divuy2k@seas.upenn.edu) (SSE)
Hyong Min Kim (hyongmin@seas.upenn.edu) (EE)
Nicholas Haas (haasn@seas.upenn.edu) (EE)
Rohit Majumdar (rohitmaj@wharton.upenn.edu) (MEAM | M&T)

Table of Contents

I. Executive Summary

In line with the IoT4Ag effort, Dr. Cherie Kagan's research team at Penn has been engaged with building optical agricultural sensors that change color in response to environmental stimuli. A key barrier to their progress has been the testing required to determine the optimal size, brightness, shape, and color of these sensors such that they are most easily distinguishable by an overhead camera attached to a drone that will eventually image the field where the sensors are deployed. To determine these parameters, testing needs to be conducted with colored pieces of paper (roughly about 1.5x1.5cm) on a leafy background to simulate a real field. Our team's goal was to simplify and automate this testing and deliver some insights about optimal imaging distance and angle as well as sensor color to the group.

To facilitate indoor testing and accommodate the lack of a drone, our system used a cardboard background with faux leaves and we measured the distance between the sensors and the camera horizontally rather than vertically. To click pictures of the setup that would serve as input data, we used a custom designed Raspberry Pi based rover with an adjustable and custom camera mount. This allowed us to click a large number of pictures of the proxy sensor and leaf setup at arbitrary distances and angles with precision and minimal human error. Our rover system moves incrementally closer to the sensors (in adjustable distance increments) and the attached camera sweeps an angle that captures the entire field of view at each increment. At each distance and angle combination, an image is clicked and the collected dataset of images is later processed through a Python script. The image processing routine utilized the Watershed algorithm within the in-built OpenCV library in python. Images collected using the rover were passed through smoothing filters, gray-scaled, and binarized into black and white regions of "foreground" and "background" to obtain contours that could help split the image into distinct segments. These segments were labeled, and filtered further on their polygonal shape to obtain strictly those classified as "rectangles", and were expected to capture proxy sensors.

An inverse exponential relationship was observed between the Watershed segmentation of an image and the distance it was taken at, for each angle studied in the range 60-120 deg. There is however, a seasonality in segmentation – measured by the number of contours identified by the algorithm – as 90 deg images at relatively larger angles were observed to possess more rectangular segments/contours than more inclined angular images at relatively shorter distances. The inverse relationship was seen for both, brown and green boards with leaves. For specific intermediate distances, pink and light green were most easily captured, while for other distances, yellow and red were better captured. Further work in this regard could consider, with limitations on incident angles of the drone cameras to be integrated in the surveillance, which angles and distances are the most constraining mechanically, effectively yielding an RGB range for sensor output that is most sensitive to said setup.

II. Overview and Motivation of the Project

Precision agriculture entails the monitoring of growth environments and health status and therefore maximizes quality and yield. The large-scale implementation of precision agriculture is achievable through remote sensing monitored by autonomous drones. The research team's project seeks to deploy chemical sensors directly in the field that give visible cues in the form of color change - likely with RGB output. These optical sensors will be small and passive, allowing for direct planting onto the leaves of the crops whose health is in question. These visible cues output by the sensors are to be directed based on environmental factors in a crop field and will be read and monitored remotely by drone-mounted cameras. For a more detailed business analysis, refer to Appendix A2.

Our goal was to design and optimize a reliable, repeatable, and flexible method to test the framework of sensing being developed by the research team. Before we began work with the team, they were taking rudimentary iPhone pictures on a tripod. They needed someone there constantly, took images slowly, and were prone to human error. We aimed to create an autonomous system that they could use to collect and computer analyze thousands of images without human supervision to more effectively determine the sensor's characteristics. While the full implementation will be a drone with an RGB camera actively monitoring the sensors using images from above, we moved the axis in question to do the testing horizontally. This made the most sense for indoor tests and enabled users to change many variables easily without the need for a large vertical distance.

Ultimately, our efforts were aimed at helping the research team by creating a system that could outlast our involvement with the project and serve as a tool that can be used for testing throughout the development process. This required paying special attention to the specific requirements and constraints faced by the team. If done correctly, it could save the team a large amount of time and effort by allowing them to fine tune their sensor design without worrying about acquiring expertise in an unrelated area, image processing and robotics.

III. Technical Description

A. Specifications

Working on this project, there were certain specifications that were at first asked of us by the team for which our results were to be obtained. This was primarily in the type of proxy sensors and the nature of the tests because of their goal to manufacture these sensors. These sensors were 1.5 cm x 1.5 cm large and were constructed out of colored construction paper. The tests were performed with three different settings - proxy sensors on brown construction cardboard with leaves (Figures A7 and A8), proxy sensors on green paper without leaves (Figure A11), and proxy sensors on green paper with leaves (Figure A12). Further, while our testing was conducted at distances ranging from 4 feet to 20 feet in 6 inch increments and across a 40 degree field of view swept 10 degrees at a time, it was important for our system to be capable of conducting tests at arbitrary distances and angles in the future. Ease of use, automation, and mobility were also prioritized. Specifically, the rover we designed can be manually carried to different locations and will also work outdoors. The image processing algorithm needed to be able to reliably extract squares from the image, even if just showing relative changes in recognition with varying distances and angles.

B. Iterations

We began this project by looking at the feasibility of implementing a rail-track system (Figures A2 and A3). Whether we were going to machine this rail and track system or purchase a pre-made version online was not certain. However, the system would not have been adequately mobile, and the range would be limited by the length of the rail track, preventing the team from testing arbitrarily large distances. Further, eventual outdoor usage of the system would be challenging due to power requirements.

We pivoted to using a robotic car with tank treads (Figure A4). The tank's initial design used a Raspberry Pi camera mounted on two small servo motors designed to tilt along two axes. Due to poor resolution, this was replaced with a new camera that required us to redesign large parts of the car and, including a new base for the camera and a universal tripod mount to accommodate any further changes to camera type without the need for a complete redesign each time. We also changed the vehicle's wheels to rubber wheels instead of tank treads since the treads were hard to use outdoors, struggled on some testing surfaces due to lack of friction, and were incapable of moving in precise increments.

We initially began probing standard industry-practice edge detection algorithms – Canny and Sobel – to isolate proxy sensors from the background. However, due to interference from leaves, it was difficult to achieve that, and we quickly pivoted to using the WaterShed Algorithm in python's in-built OpenCV library. This enabled image segmentation into contours that could then be studied for their shape, area, and color, to approximate the capture rate for proxy sensors.

C. Societal, environmental, or economic considerations

The design was influenced more by economic considerations than by societal or environmental considerations, but this is primarily due to the fact that our product will be used internally by the researchers led by Dr. Kagan. The way in which we decided to design our testing apparatus was relatively cheap considering some of the other options, and this allows both the ESE department and the research team to use their funds elsewhere while our product does its job collecting and analyzing data. However, this is not to say that the team will not be heavily influenced by these, as they will ultimately be implementing a large-scale agricultural product on farmlands that will affect populations as well as the environment.

D. Technical description and approach

Our system was based on a 4gb Raspberry Pi based rover with 32 GB of storage. We began testing with a piece of cardboard, roughly 4ft by 8ft and using colored square pieces of paper cut using an industrial cutter that were 1.5cm on each side. Our testing involved automating the rover and the camera movements to capture the desired number of images. We captured images starting at 5 feet away from the sensor board and moved incrementally further in increments of 6 inches, up until we achieved a distance of 20 feet. At each distance increment, images were clicked at 6 angles, starting with the camera facing perpendicular to the board and incrementing the angle of the camera to the perpendicular 10 degrees at a time, until it was at a 60 degree angle from the board. Lighting conditions were kept constant.

The image recognition process involved using python OpenCV to segment the image into contours. This involve three major steps:

1) Mean Shift Filtering: This is a local homogenization technique that smoothens an image by averaging differences in tonality and shade between extremely proximal pixels. The pyramid mean-shift filter in OpenCV allowed us to toggle the two parameters – spatial distance (between any two pixels) and radius (to homogenize) – which were ultimately set at around 21 and 50 respectively for most analyses.
2) Gray-scaling: The mean-shifted image is then converted to grayscale using the cvtColor function in OpenCV, which further allows us to identify high and low contrast regions for determining edges and background objects later.
3) Otsu's binarization: This technique generates a stark black and white contrast between "foreground" and "background" objects and also gives a preliminary idea of how much of the image the algorithm was able to capture in distinct details.

The binarized image is then used to obtain contours, i.e. bounding walls for segments. The peak Euclidian distance between contrasting pixels in the binarized image is determined and used to find the local maximum so as to divide the region on two sides of it into two distinct segments. The Watershed function uses these inputs to generate segments, identifiable by their unique integer "labels", starting at 0 for each image. Each such label is then studied individually and a "mask" matrix is generated that enhances the values of segments deemed important for us. The findContours function again is used to grab contours for these masked regions. Finally, we implement a shape detection subroutine within each segment to filter out those that may not correspond to our shape of interest, i.e. quadrilaterals, and specifically rectangles. For each label, one (x,y) vertex and the width (w) and height (h) are approximated using the approxPolyDP function in OpenCV, and the resulting polygon is classified as one of "circle", "rectangle", "triangle", or "pentagon", based on the number of edges detected. If the shape of the contoured segment is deemed rectangular, the said contour is visually imposed on the segment as a bounding box, with the label number for that segment also shown for clarity. Finally, the "mean" function is used to approximate the average color, in BGR (the order obeyed by OpenCV), inside each rectangular contour identified above.

After initial image processing, more data was collected wherein the background behind the leaves was covered with green construction paper so as to reduce the contrast between the leaves and the background. Here, distance increased in 6 inch increments from 4ft to 12ft and angles were swept to capture images 10 degrees to the left and right, and 20 degrees to the left and right of the perpendicular for each distance increment.

E. Final status of the project and test results

The analysis of two major setups – leaf-covered boards, one with a brown background and one with a green background – both yielded an inverse exponential relationship between the strength of the Watershed segmentation algorithm and the distance the image was taken at, keeping angle equal across all distances. The segmentation here was measured by the number of rectangular contours recognized, which does not necessarily imply only proxy sensors (true positives). However, it was observed that the number of true positives (hit rate) of the algorithm under equal conditions increased as the contourization increased, implying that a larger number of segments/contours has a higher probability of capturing proxy sensors, while also increasing noise. However, per the solicited needs of the research team, excess noise was more tolerable than a low hit rate in a bid to avoid false positives. The negative exponential relationship is shown in Figures A13 and A14 for brown and green boards respectively.

Another takeaway here was the resulting seasonality in contourization: more inclined angles (tending farther from the perpendicular) at any given distance performed worse in segmentation than a 90 degree image captured at a slightly larger distance.

We also observed an association between specific angle/degree combinations and the hit rate for specific colors of proxy sensors that were identified by the algorithm. As shown in Figure A17, at 7ft 90 deg (i.e. “0 deg” deviation from the perpendicular), light green was most easily detected and about ⅔ of the squares were recognized. Blue and pink squares were also well identified, but orange and yellow lagged, despite the intuitive contrast expected on a purely visual basis. This was conducted against a brown background. On the other hand, experiments with a green background at 4ft from 70-110 deg showed yellow, orange, and blue squares to be captured most easily. This is seen in Figure A18 with the average color identification of segment #141, which is actually yellow.

Another observation concerning contourization may be seen as distances increase (for the same angle) from very proximal values (4/5ft) to 7/8ft. In our setup, there was an increase in contourization moving from 5ft to 6ft, possibly because the FOV widened and allowed for a larger array of sensors and leaves to be captured in general, increasing the hit rate. However, this effect declines with any further increase in distance, as the rover’s vision was limited by the much smaller size of the board, and any advantage from larger FOV was lost to greater distance.

F. Overall evaluation

The seasonality in contourization with distance/angle pairs leads us to believe that system efficiency may be improved by positioning sensors and the accompanying surveillance drone in a setup as vertical as possible for any given distance, since performance at the same distance declines with more slanted angles. However, this effect is offset as the distance increases beyond a certain point, where even capturing a 90 degree image at say 12ft, would be inferior in contourization to a 70 degree image at 7ft. These relationships can be visualized in figures A15 and A16 in the Appendix.

Furthermore, as noted above, yellow and orange sensors were captured better against green backgrounds than when the same distance and angle were captured for leaves on a brown board. One plausible explanation for this difference could be the discrepancy between foreground and background that may arise for yellow and orange hues with a brown background, since they seem to mesh together, leaving the green leaves in the middle sensors of other colors more sensitive to the algorithm. With a green background, this discrepancy is removed and redder hues are captured better.

G. Conclusion

The watershed algorithm can be quite limiting in its performance while recognizing child contours, i.e. sub-contours within larger segments (such as several square sensors inside a square region on the board). However, when it does capture segments, it is able to detect edges and isolate them and their colors quite well. Using the setup of green and brown boards with green leaves, we concluded system improvements stemming from more perpendicularly vertically oriented surveillance, color dependency on the field background and the dominance of yellow/orange hues against greener foundations, as well as the inverse exponential relationship between distance/angle pairs that tend to favor smaller distances and less inclined angles, but can accommodate for slightly larger distances at less inclined angles to perform better than more inclined image capture at said smaller distances. Further work in this regard would focus on color recognition and isolation within captured sensors and how that may evolve with distance and angle independently, instead of just the background of the leaves.

IV. Self-Learning

A. Self Learning

In designing the testing system, the use of Raspberry Pi to control various technologies was an early hurdle for the team consisting of only one computer engineer. System modifications were required to integrate servo and DC motors for controlled, automated testing. The servo motor responsible for sweeping the camera angle required trial and error with different pulse width modulation, and the third party Logitech USB camera was difficult to interface. Initially, we made an effort to use MATLAB and its libraries to set up initial tests for example image processing processes including still center of mass analysis and edge detection of a live video feed. It helped us to narrow down our image processing approach using a more forgiving approach, but it took some effort to become acquainted with the technology. With the acquired data sets, extensive Python code and use of OpenCV and corresponding image processing libraries was required. This required using online documentation of these libraries to achieve desired results while allowing better understanding of the parameters in image processing algorithms and how best to tune them for our needs.

B. Useful courses

Although much of what we worked on required self learning, knowledge from Penn coursework helped tremendously. First, CIS545 and CIS192 gave us background in Image Processing and Python. ESE543 helped us keep in mind the usability of the system and optimize to reduce human effort and error. CIS380 helped with familiarity with terminal commands needed for Raspberry Pi operation. Finally, MEAM101 and ESE292 helped with prototyping and CAD expertise needed for designing the camera mount and base of the car. MEAM201 helped with designing the new axle and making other amendments to the car when we removed the tank treads.

V. Ethical and Professional Responsibilities

A. Professional Responsibility

IoT-based precision agriculture has a universally positive impact on society given that it can promise higher yield and efficiency in raising staple crops with a smaller labor force. This is particularly useful in rural areas with local shortage of labor and in developing countries where lack of expensive agricultural equipment makes it difficult to monitor and effectively use large tracts of farmland by conventional means. In the long term, drone-based monitoring of farmland could fundamentally shift the paradigm of agriculture in the future. The optical sensors to be deployed on the field should be sustainable, mass-producible, and environmentally-friendly.

Our professional responsibility also entailed ensuring the team we were working with was satisfied with our product. In particular, we focused on ease of operation for the research team. We also ensured reproducibility of data: with identical background and lighting conditions, two sets of images collected with the same angle/distance increments should produce very similar results. Finally, we designed a modular system for which key components can be easily removed or replaced on-demand; for instance, our current camera is attached to a universal camera mount that is attached to a servo motor, which the research team should be able to easily replace with a different camera model should they wish to test those in the future.

B. Ethical Issues

We did not run into major ethical issues throughout our project period. Our work did not involve live subjects (human or otherwise), nor did it involve controversial technologies with questionable implications once put into application. However, our project entailed careful assessment of components needed and their potential impact on our surroundings, both the environment and our workspace. We used standard arts and craft supplies to build the backgrounds for testing proxy sensors rather than using real plant samples. In particular, we used fake leaves to emulate plants that the proxy sensors were placed on rather than using real leaves. Using real leaves would have inevitably damaged the environment they were collected from and caused potential, as these backgrounds had to be stored in the senior design lab for months throughout the semester. Furthermore, these leaves would likely have decayed and become a biohazard, while the debris of drying leaves would have polluted the senior design lab and impacted other groups' projects.

VI. Meetings

As we were developing a testing system specifically for the research team, it was imperative for us to stay in close communication with them. We primarily communicated with the research team over email to either provide updates or ask questions twice a week. We also met with the team five times over the course of two semesters to get feedback on our progress and their design requirements. The meetings were planned using When2Meet to get an understanding of the availability of everyone involved. These meetings were extremely beneficial and led to the success of our project. The research team are subject-matter experts on their sensors and have a firm understanding of what they want to test for, so we focused on interacting with them rather than unnecessarily meeting with 3rd party stakeholders or consultants. We plan to continue to follow the team's progress once we are finished with assisting in their testing operations.

VII. Schedule with Milestones

The spring semester milestones (see Table A1) were more intensive than those of the fall, and they allowed us to efficiently and quickly work through the design and testing process. Our ultimate goal was to deliver data to the research team that outlined the effect of color, distance, and angle on sensor visibility. However, due to unexpected challenges with operation of the raspberry pi based rover, the pivot away from the rail track, and difficulties with the image processing code, we were unable to conduct as much iteration as we expected and have been unable to test lighting conditions or do any testing outdoors. Further, image processing with 95% accuracy of square detection could not be achieved in the given timeframe. Therefore, while we could conduct one iteration with a brown background and one iteration with a green background, we could not iterate as much as expected on the image processing algorithm or vary colors, sizes, shapes, and lighting conditions enough.

VIII. Discussion of Teamwork

Operating cross-departmentally spanning majors in computer engineering, electrical engineering, systems engineering, and mechanical engineering, we were able to create a product that leveraged our backgrounds. We internally held weekly meetings to discuss our upcoming work and strategies, and used social media to interact and communicate with each other outside of this designated time.

Anshul's Computer Engineering background played a pivotal role in achieving Raspberry Pi operation, camera integration, and automation of the car to collect images at the desired distances and angles. Divyansh was able to similarly make use of his knowledge of software and programming to implement the final variations of the image processing algorithm and take the lead on the data processing. Nick and Hyong worked to develop testing apparatuses and the construction of the vehicle and worked with the collection of data in the extensive tests. Some initial data processing was also done using MATLAB while in the process of determining the approach most appropriate for final use and implementation in Python. Nick, working from his MEAM coursework, worked to test the final vehicle and engineer a custom base and camera mount. There was also some work from Rohit to fit the new axle to the vehicle in an attempt to make it roll straighter. He also helped the car switch from tank treads to wheels, which involved fabricating a new axle in the Precision Machining Laboratory and fitting it to the new wheels. In all, our team was vastly cooperative and made use of the limited time to design this solution for our stakeholders.

IX. Budget and Justification

In Appendix A1 are tables showing the budget proposed in the fall (Table A2) and the final cost of materials as seen at the end of the year (Table A3).

We delivered a successful project under-budget for a few reasons. Firstly, we shifted from a rail track system to using a car in the spring semester. The Yahboom Tank was 33% cheaper than the rail track, and we also did not need to purchase a race car or selfie stick. The Logitech camera that we ended up using was also significantly cheaper than the cameras that we forecasted in the fall, yet it provided sufficiently high quality and the research team were happy with the purchase. The MDF and faux leaves were $40 more expensive than forecasted in the fall, but that was offset by the significant savings that we saved from using our car a more capital-efficient camera and

X. Standards and Compliance

Our project sought to produce a testing and data-gathering platform for the research team in designing and fabricating the optical sensors. As a result, our final product prioritized operational feasibility for the research team rather than abiding by industrial and engineering standards that apply to products that are marketed directly to the public. Since our project is largely internal and does not affect any mass markets, we were focused on professional responsibilities to the team as outlined above. Due to its internal nature, there were no significant safety or manufacturing hazards in our system.

That being said, we can identify engineering standards that should apply in further work on the initiative by the research team. Within the series of engineering standards collectively called ISO/IEC JTC 1 (Information Technology), subheadings ISO/IEC 7942-1 to 7942-4 are dedicated to computer graphics and image processing. This is to ensure a certain degree of uniformity, flexibility, and robustness across various methods of processing images. This engineering standard encompasses most computing languages including Python, so our image processing algorithm abides by this. Reliability of the image processing algorithm is certainly something the research team should keep in mind in future iterations of testing proxy sensors. Some further engineering standards exist for data security, format, processing, and exchange in smart agriculture, such as P2992 from IEEE. Finally, the research team should be mindful of engineering standards that govern the use and distribution of chemical/optical sensors in farmland based on their environmental-friendliness and biodegradability.

XI. Work Done Since Last Semester

Working in this course over a period of two semesters, we were able to take the additional time to build upon what we had begun in the fall of 2021. We had previously envisioned a rail-track system upon which to build our camera and automated imaging apparatus, but this proved to be rather limited. After returning to the project in January, we - with the help of the research team - decided that it would be best to refocus our efforts on a more modular and mobile system. This entailed a new data collection apparatus consisting of a robotic vehicle driven with a Raspberry Pi. This allowed us to easily test the full desired range of 5 to 20 feet from the proxy sensors. In using this new robotic vehicle, we had to redesign it to fit our needs. This included designing a stronger base upon which to mount our servo, a different wheel system, and a modified camera mount that allowed for multiple cameras to be tested on the same device. There was also significant effort put into designing the tests that were used to collect the data with. It was a process to determine the type and amount of sensors, and apply these to different backgrounds and ambient conditions as requested by the team to ultimately aid their manufacturing process. Accordingly, the image processing algorithm was also implemented this semester to allow for the analysis of the collected data. This was implemented in python, and it employed a segmentation process to pick out the square proxy sensors from the backgrounds they were used on. A large effort also went into actually fulfilling the testing. As our apparatus progressed in the design process, the testing process became less laborious as we fulfilled our goal to automate the process, but it nonetheless took some time to ensure it performed at the capacity required for it to be useful to the research team.

XII. Discussion and Conclusion

Ultimately, our project was able to establish a negative exponential relationship between distance and strength of recognition algorithm. We also established seasonality of this trend based on the angle of incidence wherein more acute or obtuse angles yielded poorer recognition. We also established that for a brown background, light green was most easily recognized whereas orange lagged the most. However, for a green background, yellow, orange and blue were most easily recognized.

Our limitations were primarily centered around the fact that we could not engineer an algorithm that captured only precisely the proxy sensors and no other segments. Portions of leaves, as well as noise from the background severely limited our ability to detect true positives and this meant that even though data collection had been automated, our processing algorithm could not deliver clear and precise results that would give the research team the precise answer they were looking for. We believe that ultimately, continued iteration would have allowed us to zero in on a more accurate algorithm and would have also helped us better identify parameters that had the largest effect on results. Other potential challenges were the lack of testing conducted outdoors. We were also limited by our ability to create large leaf boards to simulate arbitrarily large crop fields since this remains a manual, labor intensive process.

Our key lessons learned with this project have been that multitasking and parallelizing tasks goes a long way in achieving more substantial results. We believe we could have benefitted immensely from having a dedicated room wherein we could use an entire room-sized wall to act as our proxy crop field. While this was available in Pennovation, we prioritized proximity to campus but this resulted in noisier data. Further, anticipating the low resolution of the raspberry pi camera would have allowed us to begin the integration process of the Logitech webcam sooner, which comprised a bulk of our time. Finally, developing the image recognition algorithm in parallel to the automated rover using manually collected data would have helped us reach a more advanced stage with image processing.

XIII. Appendices

Appendix A1: Reference Figures and Tables

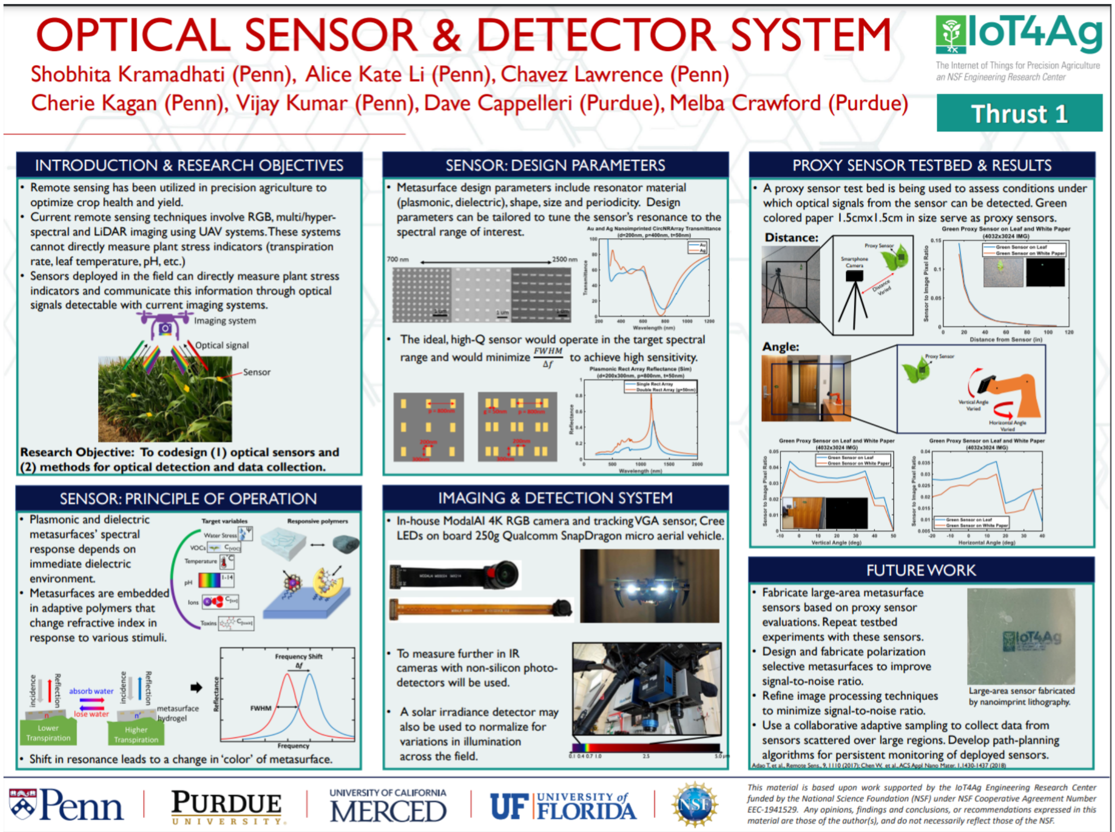


Figure A1: Initial proposition for the project from the research team

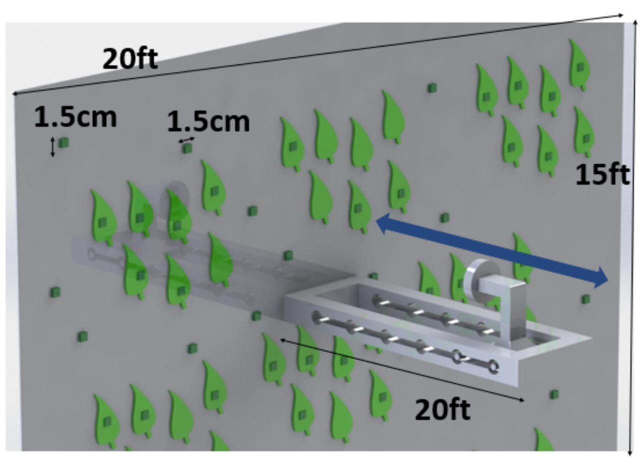


Figure A2: Rendering of initial camera rail system

Figure A3: Rendering of initial camera mount system

Task	**Hyong**	**Nick**	**Anshul**	**Div**	**Rohit**
Purchase key components & assemble rover		1/25			1/25
Conduct initial meetings to outline product requirements	02/05	02/05	02/05	02/05	02/05
Complete image processing code to achieve 95% detection			03/05	03/05	
Automate data collection (30 distance & 6 angle increments)	03/15	03/15	03/15		
Iterations with green background		04/01	04/01		04/01

Table A1: General milestone descriptions and assignments

Item	**Budgeted Cost ($)**
Camera (GoPro Maximum Wow or Intel RealSense)	350
Rail track system (Proaim or Snaptrack)	209
Arduino	50
Race car	50
Selfie stick	15
Any other expenses & materials	100
Total	**$784**

Table A2: Fall estimates for our budget

Item	**Cost ($)**
Logitech Pro Webcam C920	60
Raspberry Pi Kit	100
Yahboom Tank	140
MDF	40
Faux Leaves, Craft Supplies	100
Total	**$440**

Table A3: Actual budget as viewed at the end of the year

Figure A4: Original robot using line tracking to follow a prescribed track

Figure A5: Initial test setup for MATLAB code

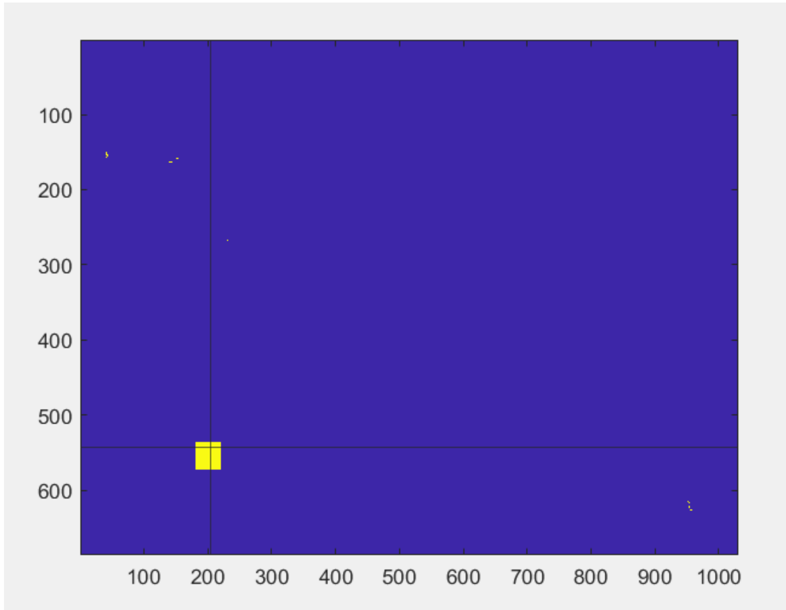


Figure A6: Results of initial MATLAB algorithm showing it’s limitations

Figure A7: Test background for proxy sensors on leaves with brown background


Figure A8: Test background for proxy sensors with different lighting


Figure A9: Resulting image after running the segmentation algorithm on the collected image

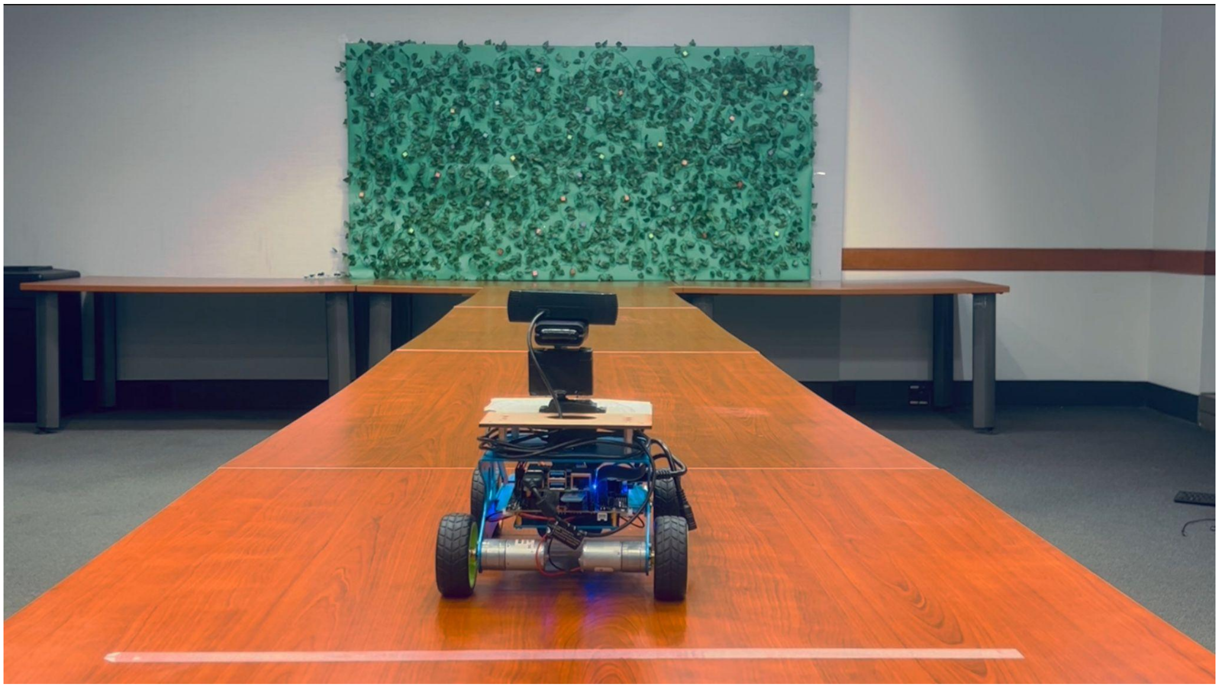


Figure A10: Automated testing system testing the green leaf and green background setup

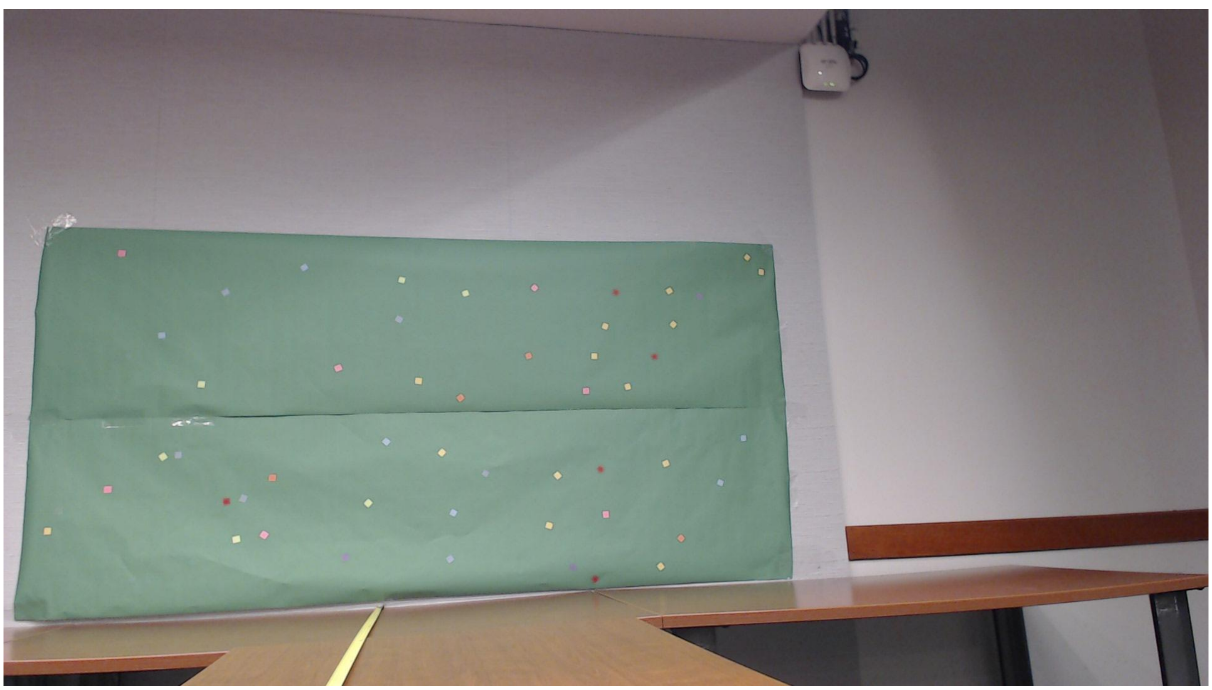


Figure A11: Image from robot showing leafless test setup

Figure A12: Image from robot of the same angle and distance as Figure A10 only with leaves

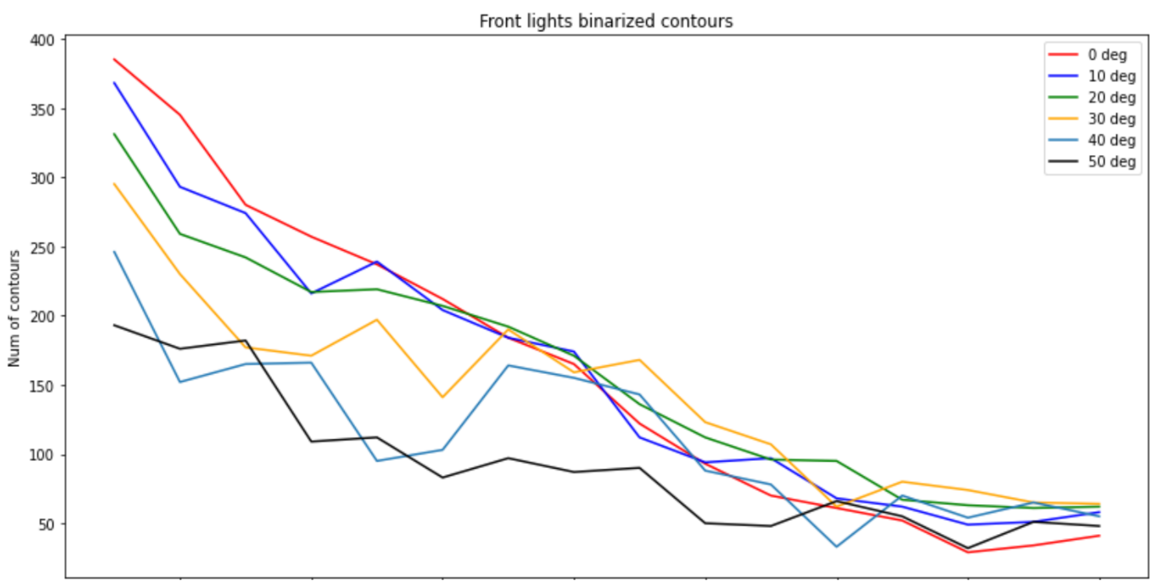


Figure A13: Test results for the output contours from a test varying in angle and distance (brown board)

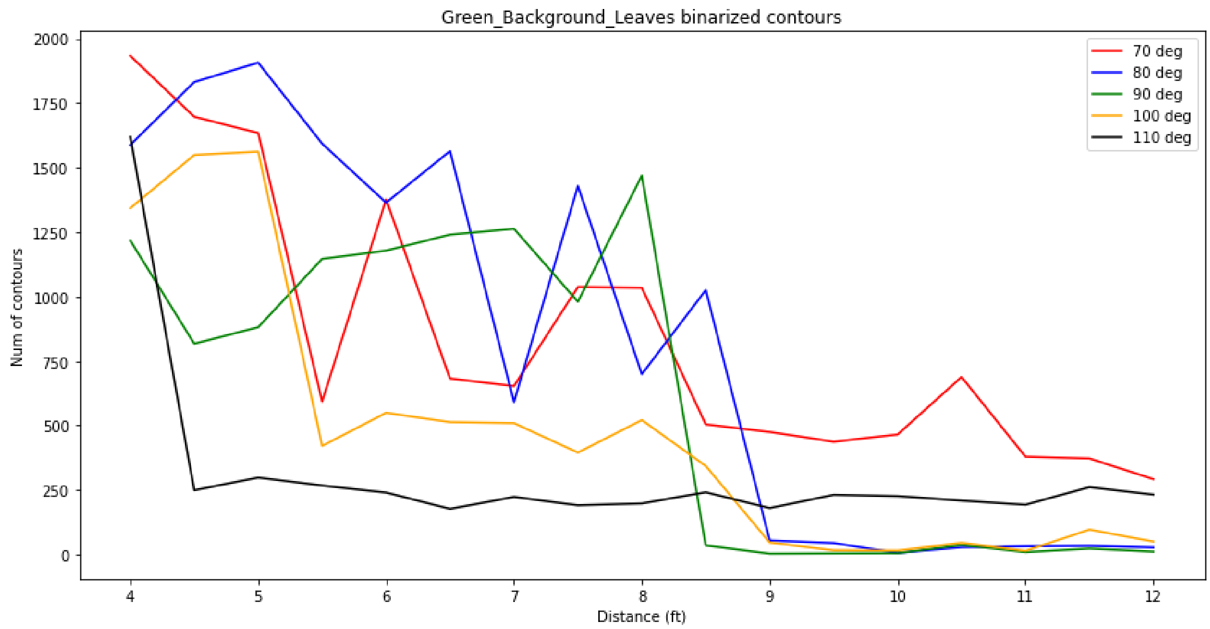


Figure A14:Test results for the output contours from a test varying in angle and distance (green board)

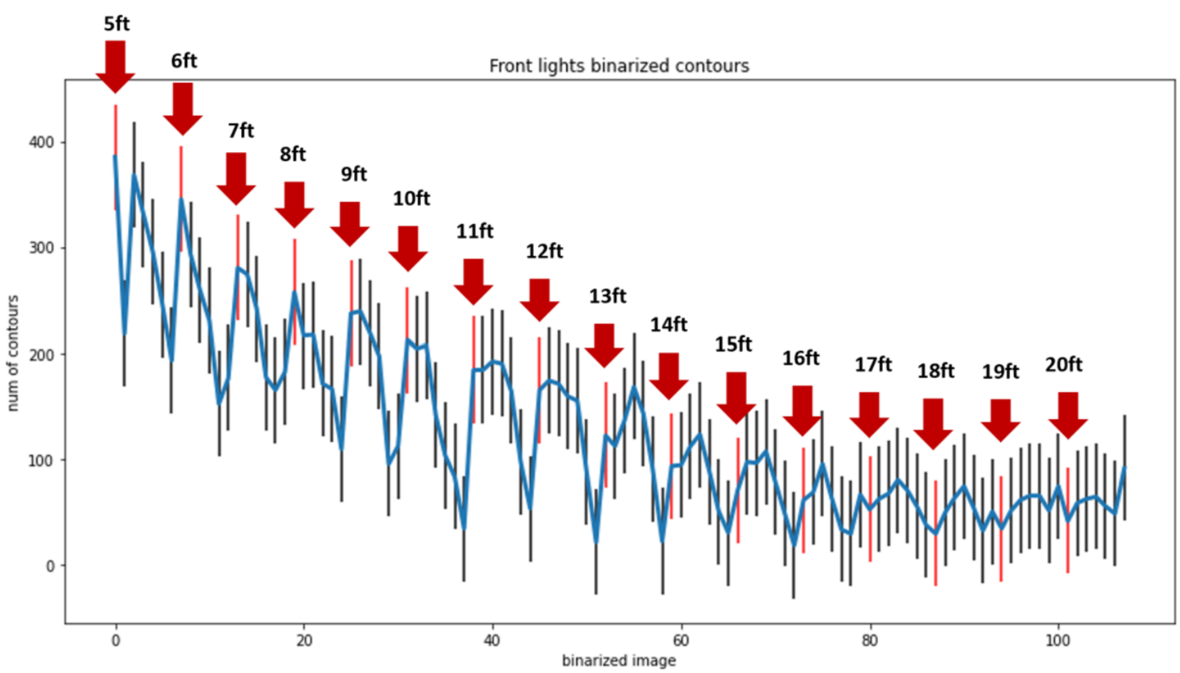


Figure A15: Seasonality in contourization (brown board)

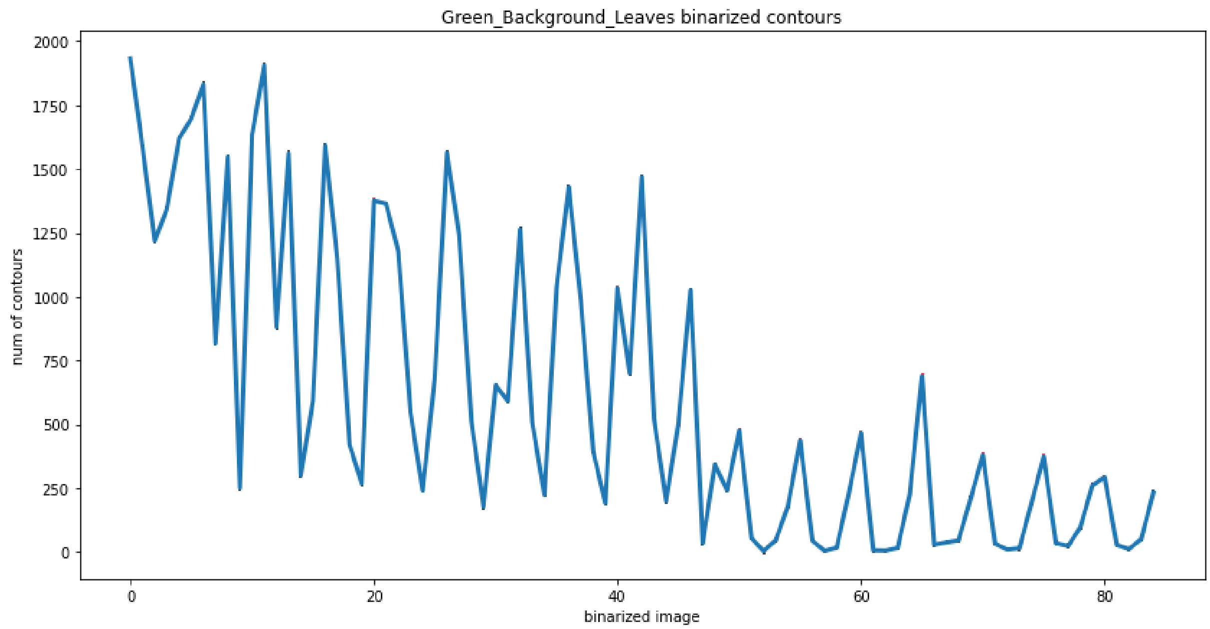


Figure A16: Seasonality in contourization (green board)

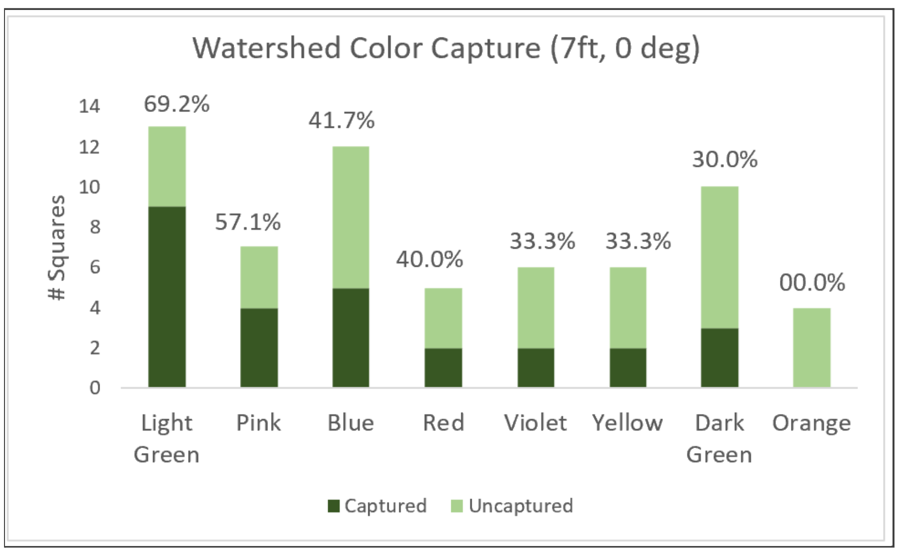
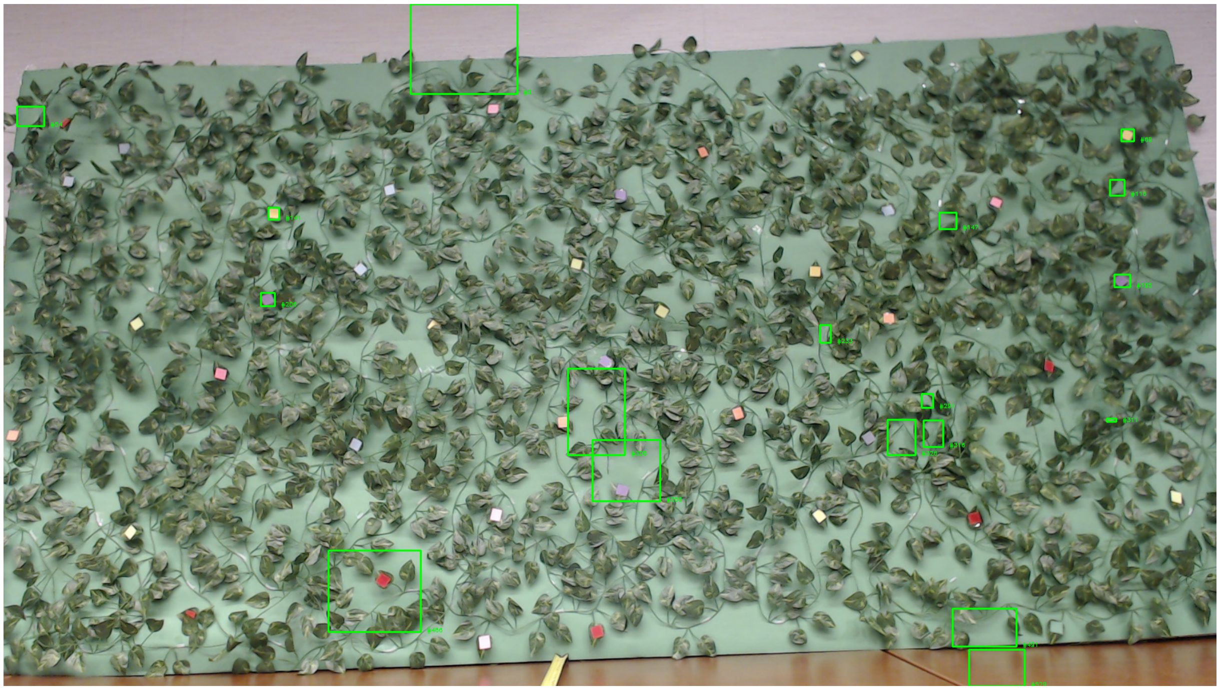


Figure A17- Color capture at 7ft



Average color for segment #141 (yellow)

Figure A18: Rectangular segmentation captures yellow and red more easily on green

Appendix A2: Business Analysis (M&T Requirement)

i. Need & Value Proposition

Modern food production is largely supported by industrial-scale agriculture of staple crops such as corn and soybeans. Precision agriculture maximizes quality and yield of such crops by continuously monitoring their growth environments and health status. Autonomous drones can use remote sensing to implement precision agriculture at a large-scale. Currently, drones in agriculture are largely used for dispensing water and fertilizer. Experimental models under research primarily focus on RGB camera-based image recognition of plants to detect growth defects and pest damage, e.g. yellowing or wilting of leaves. These often require state-of-the-art technology that not every farmer can afford and use sustainably. Furthermore, these models are designed for the imaging of specific plants, they detect damage in progress rather than preventing damage, and they cannot monitor non-visual growth conditions such as pH, humidity, temperature, and pest activity. Hence, an affordable autonomous drone that can detect both visual and non-visual growth conditions and is compatible with all plants should have significant viability for farmers both in the US and globally.

ii. Stakeholders

Monitoring leaf contents, especially in fruit-bearing plants, directly affects the four dimensions of food security: (a) availability, (b) access, (c) utilization, and (d) stability. It directly impacts farmers and the economic and nutritional value of their output. At present, these farmers are being forced to use products like Monsanto fertilizers to help their plants grow, driving them into a vicious cycle of debt. It also improves the non-market valuation of vegetative habitats by reducing the costs and customer frustration related to replacement, landscaping, & protection against pests. This benefits the entire consumption chain, from grocery stores to end-consumers. Helping crops grow optimally is an integral part of tackling the food deficit, as nearly ⅓ of the world did not have access to adequate food in 2020. To that end, the US Government and World Health Organization will be interested in this technology, perhaps even subsidizing costs for farmers in areas of low crop yield.

iii. Market Opportunity & Customer Segments

There are three main types of professionals who would consider an agriculture drone:

1. Farmers who want to fly their own imaging missions
2. Agriculture service providers (e.g. DJI, GoPro) who fly drones for farmers
3. Large food chains farmers of staple crops and fruit-bearing crops that gain most utility from our drones

They would have 40+ acres of land to survey and the crops would be very delicate and spoil quickly without optimal conditions. Our drones would help to alleviate their concerns by precision monitoring either a part of their farmland or the entire vicinity. The US Government could even subsidize farmers' cost in purchasing the drone, as tackling the hunger crisis is a national and international priority. Agriculture service providers could buy hundreds of our drones and millions of our proxy sensors. They would then lease our drones and the sensors to farmers for specific time periods, e.g. one scan of their land or for one week. They can obviously charge a high premium to farmers, but a lot of that premium will likely go back to maintaining the drones as they are exposed to harm from the elements. Lastly, large food chains can purchase some drones for their farmers. For example, McDonalds can purchase our drones to survey their fields, either leasing out the drones to their farmers or using the drones as a quality assurance check. The fast food chain has just recently introduced pioneering regenerative agriculture techniques into the beef industry, so they are clearly looking to optimize their supply chain processes. Further, given the US Government's renewed focus on sustainability in recent years, this could be an avenue for such organizations to win favors with the government.

iv. Market Segment Size & Growth

The agriculture drones market is large and fast-growing. According to GlobeNewswire, the market was worth $530 million in the US alone in 2021 and is growing at a 18.14% CAGR to $3.70 billion market size in 2027. A significant portion of the growth is driven by unmanned aerial vehicles (UAVs) like our drones, which are proving to be a great tool at raising farm yields globally.

We are conducting our own top-down market sizing to get accurate numbers. According to a US Government Poll, there were 2.02 million farms in the US in 2020, which has stayed almost constant since 2000. Assume 40% of these farms are ideal for our drone (fruit-bearing crops or staple crops like corn). Assume that only 70% of addressable farmers can afford our subscription model. Assume that 60% of farmers have faith in autonomous drones. Lastly, assume that each farmer wants to do 3 surveys a year, i.e. 1 survey each season, with a survey costing $650.

Market size = 2.02 million x 40% x 70% x 60% x 3 x $650 = **$662 million today**.

Assuming that by 2026 (5 years later) all farmers can afford our subscription model and have faith in it, 2026 Market size = 2.02 million x 40% x 3 x $650 = $1.58 billion. CAGR = **18.9%**.

v. Competition

The US is both the largest and most competitive agriculture drone market. Large companies like Trimble, DJI, and GoPro have all entered the market with their own products at a large-scale. Our closest competitor is the DJI Agrad T16, a water irrigation drone that costs $21,499 for a one-off purchase. Like all its competitors from GoPro and Trimble, it has great reviews, with most of the complaints being related to the pricing. While many GoPro and Trimble models can visualize fields effectively at a similar drone price range to ours, none of the drones are able to detect non-visual health indicators like pH, which are the primary reasons for crop decay in the US today. Hence our drone certainly has a unique angle within this crowded market.

These companies certainly have better brand recognition than IoT4Ag and so farmers would be inclined to purchase from these customers at first. Yet our drone and sensor system is being researched by many highly qualified university faculty and has been granted significant funds, so we will very likely have novel technology that will differentiate us. We should try to lease our drones independently to farmers, yet equally we should try to sell our drones to recognized brands like GoPro for better exposure. This could even provide a great exit opportunity in the future by our vertical acquisition.

vi. Intellectual Property Considerations

We created our car to be used by the research team in their testing process, rather than for commercial usage in itself. Hence, we have limited intellectual property considerations. As mentioned before, the team's final drone will be using a completely different technology compared to typical autonomous drones, due to the novel color-changing sensors that they are developing. Thus, their final drone will likely also have limited intellectual property concerns.

The main potential concern relates to the drone's method of dropping sensors. There are many irrigation drones presently available, like the DJI MG-1S. These drones typically spray water, fertilizer, or seeds over a field by utilizing delivery pumps. Given this is a basic physics principle utilized by dozens of drones, there should not be any major patent concerns for the research team. Nonetheless, they should be cautious to avoid specific patented pumping technologies, like the US10364029B2 - Drone for agriculture patent.

vii. Cost

Our drones are yet to be designed by Dr. Kagan's team, so we have limited visibility into their price beyond looking at the cost of comparable drones. Typical agriculture drones cost $1,500 to $25,000 to purchase and our closest competitor, the DJI Agras T16, costs $21,499 for a one-off purchase. Assuming a 60% profit margin, with professional drones should claim according to Reuters, the DJI Agras likely cost $13,500 to manufacture. Yet the DJI Agras is a water-spraying drone, whereas our drone will simply drop sensors and record their colors. Hence our drone will likely be a bit cheaper to manufacture and sell. We should manufacture for $11,000 and retail for $18,000 for our one-off purchases. High-volume transactions can cost $13,000 per unit. With regards to subscription, we should charge $650 per survey. Assuming we perform 2 surveys per month, it would take us 10 months to achieve our customer acquisition cost payback. This is a very good CAC payback, even for more asset-lite software-as-a-service companies, and should set our company up for fast, profitable growth. The other price options should be $1,300 per week (assume 1 week/month usage), $2,600 per month (assume 1 month used, 1 month unused), and $15,600 per year, calculated in a similar way. Our sensors are yet to be designed, so we have limited visibility into their price. However, as they will have no electronics and shall be made of simple chemo-optical materials, we can estimate that their per-meter cost will be the same as a sheet of paper. Assuming each sensor is 1.5in x 1.5in, a typical 650,000 sensor pack will be approximately 1.5 million in^2 or $1000m^2$. Given the price of paper is $\$0.17/m^2$, that implies a 650,000 sensor pack costs $170 to manufacture. We should sell 650,000 each pack at $300 for a 75%+ profit margin since most customers will already have bought a drone and so will have low price elasticity for sensors. We should seek a similar 75% profit margin for our larger sensor packs.

viii. Revenue Model

We propose either a subscription model or a one-off purchase model depending on the customer. For farmers whom we interact with directly, we will use a subscription model. Farmers will have an option to lease a drone with or without the sensors bundled for various time durations (e.g. 1 survey, 1 week, 1 month, 1 season). A subscription model is preferred for a few reasons.

1. Agriculture drones cost $1,500 to $25,000, or 1/6 of an average American farmer's annual income. We believe that few farmers can pay the significant upfront cost to buy such a drone, so we would be significantly limiting our total addressable market with a one-time purchase model.
2. A subscription model enables us to increase the lifetime value of each customer after a few years.
3. A subscription model will allow us to provide premium options like a services & implementation team for another revenue stream.
4. Farmers are likely to damage drones if they were to buy them from us, making our warranty & renewal costs very high.

For agriculture service providers and large food chains we will sell the drones as a one-off purchase. These companies can then lease out the drones to their farmers and we will collect a 15% royalty on each lease. These companies will likely purchase hundreds of drones for thousands of farmers, so we want to incentivize high-volume transactions by charging a lot for a single drone (e.g. $18,000 per unit) and offering much better prices for larger transactions (e.g. $13,000 per unit when 1000 units are purchased). Selling to these customers is a contentious decision, as their leasing model to their farmers or clients directly competes with our subscription model to the farmer. However, there are a few reasons why we believe that they must be clients. 1) IoT4Ag is still a startup, and many farmers will not consider our drones unless they are backed by a trusted service provider or food chain. This would severely lower our addressable market. 2) These customers will purchase our drones & sensors in bulk, and we earn a royalty from each time the drone is leased, so we will still make good money from each sale. 3) Service providers and food chains seek a profit, yet they will be buying drones from us and paying us a royalty. Hence farmers who lease from these customers will be faced with much higher prices than going directly through us due to double marginalization.

The sensors must obviously be purchased as they are single-use items. All of our customer segments can either buy the sensors separately from the drone (likely to confirm 70% of sales) or together in a bundle pack (likely to contribute 30% of sales). Sensors will be sold in various sizes. The smallest pack will contain 650,000 sensors, which are enough sensors to test a 40-acre field with 4 sensors/m^2. Larger packs will be offered up to a pack of 2 million sensors at a better price to incentivize higher sales size.

Appendix A3: References

IEEE Standards Association (Link)

International Organization for Standardization (Link)

Amazon (GoPro Maximum Wow, Intel RealSense, Proaim, Snaptrack, Arduino, Selfie stick)

GlobeNewswire (Link)

US Government Poll (Link)

DJI MG-1S (Link)

US10364029B2 - Drone for agriculture patent (Link)

Reuters (Link)

Alibaba (Link)

Appendix A4: Notes

Note on finding Contours

Regardless of the choice of underlying algorithm, the foregoing base level of processing is required to prepare the image for segmentation. Contours are defined as the bounding polygons for segments recognized by the algorithm as being distinct based on binarized contrast. The binarized image, hence, provided a foundation to first procure the extremely outlying contours using RETR_EXTERNAL contour hierarchy, which does not consider the “child contours” for outermost “parent contours”. To minimize memory and processing time, we use CHAIN_APPROX_SIMPLE, which reduces redundancy in contour points and retains only the minimum number of coordinates needed for the algorithm to plot a given contour. Both these subroutines are parameters for the OpenCV findContours function, from which we can actually “grab” the contour coordinates using the python “imutils” library.