AI-POWERED SMART FARM ROBOT FOR REAL-TIME CROP AND SOIL MONITORING WITH ADAPTIVE IMAGING AND TERRAIN-AWARE NAVIGATION

Chaiho Wang [1] , Tyler Boulom [2]

[1]Crean Lutheran High School, 178 Compass,Irvine, CA 92618
[2]California State Polytechnic University, Pomona, CA, 91768

ABSTRACT

Today around the world, agriculture faces many challenges like pest outbreaks, plant disease, inefficient resource use, and limited access to affordable monitoring tools for small to medium scale farmers. Agriculture faces persistent challenges such as pest outbreaks, plant diseases, inefficient resource use, and limited access to affordable monitoring tools for small- and medium-scale farmers. Without effective detection and intervention, these issues can result in decreased yields, financial losses, and long-term soil degradation. This project presents a smart farm robot that integrates robotics, advanced sensors, and artificial intelligence to provide real-time crop and soil monitoring. The system is built on a Hiwonder robot base with a Raspberry Pi, and leverages Gemini AI and OpenAI for plant image analysis, Firebase for backend storage, and a mobile application for farmer alerts. Several limitations emerged during development, including inconsistent image recognition under variable lighting and reduced navigation accuracy on damp or uneven terrain. Experimental testing confirmed that lighting conditions significantly impact AI performance, while soil type affects movement precision. Proposed solutions include adaptive preprocessing, LED-based lighting, and terrain-aware navigation controls. By addressing these challenges, the farm robot demonstrates potential as a scalable, low-cost precision agriculture tool. It offers a sustainable alternative to traditional monitoring methods, enabling farmers to make proactive, data-driven decisions that improve efficiency, reduce losses, and support long-term agricultural resilience.

KEYWORDS

Precision agriculture, Smart farming, AI crop monitoring, Agricultural robotics

1. INTRODUCTION

Plant growers and farmers face many problems that pose a threat to the plant's productivity and sustainability. These challenges are labor and time shortage, soil erosion, rising inputout costs, inefficient resource use, unpredictable weather, lack of knowledge of taking care of plants, and so much more. This hinders the plant's health and field, making the food in danger and those in charge of agriculture [1-8]. One very important challenge that is faced by many plant growers and farmers is the lack of knowledge of plant wellbeing. Many people struggle to recognize the condition of their plant, such as detecting early signs of diseases and pesticides, or how to respond effectively to different environments. This inability to monitor and respond proactively to plant needs can lead to reduced yields, financial losses, long term damage to the agricultural land, and even the consequence of death [9-12].

David C. Wyld et al. (Eds): NLTM, CCNET, DBDM, AIMLA, CICS – 2026
pp. 87-99, 2026. CS & IT - CSCP 2026 DOI: 10.5121/csit.2026.160608

For example, the glassy winged sharpshooter that has been spotted in California has devastated agriculture [13]. It is an invasive type of pest, that spreads the pierce's disease, that can wipe out an entire farm if it is not detected early. Without proper tools, many growers will face devastating agricultural loss [14].

Similarly, the powdery mildew, a fungal disease that affects crops, where it goes unnoticed in the early stages [15]. Without the correct and timely identification and solution, it will spread rapidly, affecting not just the yield and harvest of the crop but ultimately killing them [16].

These examples underscore the urgent need for better tools and systems to help growers monitor plant health more effectively and respond before it's too late.

In addition to biological threats, what is more important is the inefficient resource use, such as overwatering or mismanagement of nutrients, exacerbating these challenges, especially amidst unpredictable weather patterns [17, 18]. As consumers demand have shifted into more health products, such as non-genetically modified products or sustainably grown products [19]. This results in the traditional agriculture practice and new market demands. This is not just a problem for farmers; it could affect any plant grower. Monitoring plant health and responding effectively to threats is a universal challenge that requires new innovative and sustainable solutions.

In the methodology at section 5 we saw three different methodologies that tried to solve the challenge of monitoring plant health and supporting farmers with many different views to help them make better decisions. For the first methodology we discussed the TerraSentia robot, which is a small rover that is built mainly for research purposes. It uses cameras like LiDAR, and AI to measure the plants height, stem width, and disease at a very detailed level. This is good for research; however it is too expensive and not adaptable for everyday farmers who need practical and simple solutions. The second methodology we discussed is the use of drones with advanced sensors. Drones can cover large fields quickly and provide an air view of the crop. They are really good at spotting patterns across entire farms, but they also face a lot of limitations. These range from image quality from high altitude, cannot easily capture under canopy conditions, and tall crops breaking their sight. They also require people who need to operate a drone, and it is not always affordable for farmers. The third methodology that we mentioned is satellite remote sensing, this is a good way of providing regional and big scale data that allows monitoring over time. However, satellites will lack the plant level detail, suffering from cloud cover, and unable to provide real time insights. Our robot will improve on all of these approaches by combining close range, under leaf image, which offers farmers affordable, detailed, clear, and real time feedback that will bridge the gap between large scale monitoring and small-scale monitoring.

Our proposed solution is a smart farm robot equipped with advanced sensors, AI technology, and machine learning algorithms to monitor plant health, detect threats, and optimize resource use for sustainable agriculture. This robot addresses these challenges faced by plant growers and farmers by providing a comprehensive system for real-time monitoring, analysis and management of crops. The sensors can detect early signs of diseases, pests, plant's insight, soil moisture and nutrient levels. The robot is designed to send alerts through the app, which enables farmers to prepare and prevent major damage before it escalates. Furthermore, it can analyze weather patterns, recommend watering schedules, and analyze data to improve its decision-making and give insights to the user. This solution is really effective because it combines automation and AI to solve many issues simultaneously. Unlike traditional methods, such as manual inspection or the use of general pesticides, this robot minimizes resource waste by targeting specific areas with precision and accuracy. Furthermore, its data-driven approach allows farmers to respond quickly and efficiently to threats, improving their productivity while reducing costs. And when compared to existing solutions, such as station sensors or drones, the smart farm robot has better versatility and mobility, making it able to cover large areas and provide specific insights. Unlike the drone, that could be limited to the low air areas where there are tall plants blocking the pathways of the drone, and drones are very high up it cannot provide detailed and accurate results. Fortunately our robot solves all of the problems of the drone, where it can travel through tight spaces and give the highest level of accuracy and detail. The targeting audience is all plant growers that include small to medium scale farmers who lack an expensive agriculture technology. This solution offers a scalable and sustainable way to enhance efficiency and address the challenges faced by farmers.

For this project we are trying two key blind spots, first is to evaluate the reliability of the farm robot in the real world condition: image capture consistency under variable lighting and navigation accuracy across different terrains. These key blind spots are critical to ensure accurate and efficient plant analysis on the field.

Moving into the first blind spot, we examined the robot ability to capture consistent images for plant health analysis. The AI cognition part depends on high image quality, and inconsistent lighting can really reduce the accuracy in the detection process. To test this, the robot is positioned to photograph the same plant at different times of a day, where we measure morning, midday, and sunset. A control set of images are taken indoors under artificial light. Each light is analyzed for the clarity and the recognition accuracy, and we measure the results as in accuracy percentages. The results showed that midday and indoor conditions performed the best, with the accuracy above 90%, however in mornings and sunset conditions it dropped to around 70% to around 70% due to shadows and low light. We calculated the mean of the accuracy from all conditions and it was around 82%. These results confirmed that light is a major variability and limitation. Possible improvements include switching to a more advanced and high quality sensor, other improvements can include adding a LED ring light to the camera and use preprocessing technique to normalize the brightness and contrast of the light.

The second blind spot that we tested is the robot's movement accuracy along pre-programed paths in different terrains. Navigation is a very crucial component that is a part of our robot, where precise navigation can ensure very consistent plant scans and prevent crop damage by running it over. To test this we program the robot to follow a mark path of one meter intervals across three different surfaces: indoors, dry soil, and damp soil. We measured the deviation between the expected and the actual position at each point of the surfaces. Results came out to show that indoor navigation is highly accurate, where it only has an average devotion of 2 cm. When it comes to surface outdoors the accuracy decreased, where the dirty soil average 6 cm deviation, and the damp soil deviate around8 cm, with some error up to 10cm. The accuracy is due to wheel slippage or the sinking into soft soil. Possible improvement including adding wheel encoders with real time correction, odometry, GPS/IMU fusion, or even adaptive motors that will base on surface condition.

Overall, both experiments did really reveal the importance of adapting the robot to the real world environment. We discovered that the lighting will significantly impact the image analysis, while the terrain will really affect the navigation accuracy. By addressing these blind spots, it will make the robot more reliable and more practical for future precision agriculture.

2. CHALLENGES

In order to build the project, a few challenges have been identified as follows.

2.1. Power Management and Mobility Challenges

The first problem that we had is how can we ensure that the farming robot remains operational in large-scale farming environments without frequent recharging and switching batteries. Since the robot will rely primarily on battery power, where the combined energy consumption of sensors, motors, and AI computations must be carefully managed to maximize battery life, setting a hard limit on the robot. One significant issue is that camera and data transmission can drain the battery rapidly. To solve this, the robot should implement power-efficient scheduling, where sensors are only activated when necessary, rather than running continuously, develop path optimization algorithms that can minimize unnecessary movement and reduce the energy spent on navigation, and last but not least to integrate solar charging capabilities, allowing the robot to extend its operational time during daylight hours. In addition to battery constraints, the robot's mobility across different terrains presents another challenge. Uneven ground, mud, and obstacles like rocks or plant roots could sometimes make our robot get stuck. To overcome this, I could incorporate caterpillar tracks for improved traction. And I can also use sensors such as LiDAR to detect obstacles and adjust to the path.

2.2. Real-Time Sensor Data Processing

Another major challenge that we have is processing large volumes of sensor data in real time while ensuring accuracy. Given the resource limitations of the Raspberry Pi, running computationally expensive AI models on the robot is really difficult. Because our robot is majoring heavily in cloud based AI systems and requires internet connectivity which may not be feasible in rural farming environments. But one potential solution is to store preprocessed data locally on the Raspberry Pi before sending it to the cloud system for the analysis. Where advanced machine learning techniques like edge computing, SVMs, and decision trees where the preprocess data is stored locally on the Raspberry Pi before sending it to the cloud system for the analysis. Another key aspect is developing a data pipeline that checks for smoothing, outlier detection, and feature extraction to enhance data quality before analysis. Implementing a time-series forecasting model that could also help predict plant health trends based on past sensor data allowing the farmers to prepare before danger strikes.

2.3. Reliable Image Capture

A critical challenge lies in ensuring accurate image capture under varying environmental conditions. Lighting differences, shadows, and obstructions like leaves or soil particles can significantly affect the clarity and consistency of photos taken by the robot's camera. These inconsistencies may lead to inaccurate plant analysis or misclassification by the AI models. One potential solution would be to integrate adaptive camera settings such as auto-exposure and white balance correction. Another improvement could include attaching LED ring lights around the camera for consistent lighting. Additionally, implementing image preprocessing techniques such as noise reduction and contrast adjustment before sending photos to the AI server can help normalize inputs and enhance accuracy.

3. SOLUTION

The core structure of my program consists of five components. The first component of my program is the robot camera system, second component is the Google Gemini server, third component is the backend firebase, fourth component is the app, and fifth component OpenAI.

But the three major components that I integrate into my program are: the photo system taken by the robot, the identification and analysis performed by Gemini AI, and the backend Firebase that stores all the information. For this project, I used Flutter, Android Studios, Raspberrypi to develop the application and use Python to manage the robot's functionality. My program first starts off with the robot moving to the target and capturing a photo of the plant, which is processed through our raspberry pi and sent to Gemini for analysis to determine the plant type and description. Gemini is google's multimodal AI model. Next, The Gemini server is linked to a Firebase JSON file, where it stores both the photos captured by the robot and the analysis results from Gemini in the backend. This entire process takes place within the robot. Once the data is stored, the application, powered by android studios, retrieves the photo from Firebase and sends it to OpenAI, which provides further analysis and feedback about the plan. After we get the feedback from OpenAI, we record and display the result on the phone. The reason that we use two AI models is because after a few tests we noticed that Gemini is not as (efficient/detailed) as OpenAI when it comes to giving feedback and processing texted images. Whereas Gemini does a better job at analyzing raw photos. So that's why we use two AI models, where we can take the pro's of both models while considering the cons. Many people may wonder, why did we use a pre-built AI model like OpenAI and Gemini when we can make it yourself. The answer to that is if we self build we need to input a lot of data, not only that the data we get online is different than real life, there is a huge variability on the data we get online and real life. The second reason that we use pre built-model is because one small change on the object can cause a whole system failure. For example, we train the model to identify and analyze a post it notes. But if we draw a hotdog on the post it notes, the model won't identify it. So we need to train the model to identify the hotdog on the post it note. Now, you think you solve this problem when in reality if you switch the hotdog to french fries, you would need to train the model again, making training the model taking a lot of time and data. The third reason that we use pre-built models it has very dynamic responses where each time has a different response than the previous one.

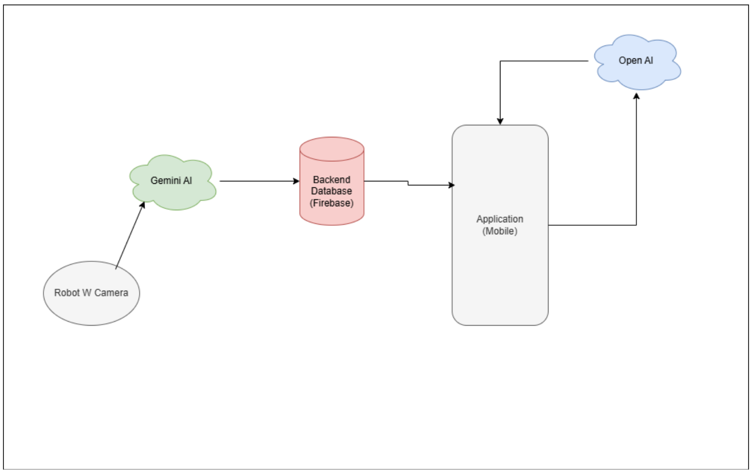
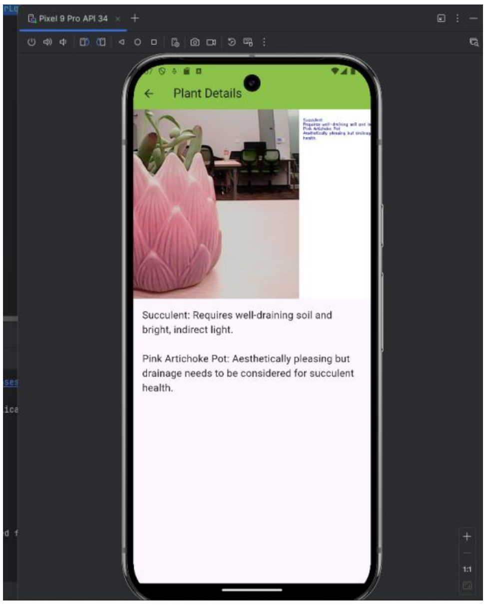


Figure 1. Overview of the solution

The first key component of my program is Gemini AI's plant identification and analysis, which is the primary tool for identifying the type of plant captured by the robot. To implement this, I integrated Google's Gemini AI API, which processes images captured by the robot's camera and pre-processed through a Raspberry Pi. The image, with its metadata, is sent to Gemini for analysis. The AI extracts key visual features, compares them against its database and classifies the plant by providing its common name, characteristics, and care recommendations. Once processed, the results are stored in a Firebase JSON file, making the data accessible for further use in the application.



Succulent: Requires well-draining soil and bright, indirect light.

Pink Artichoke Pot: Aesthetically pleasing but drainage needs to be considered for succulent health.

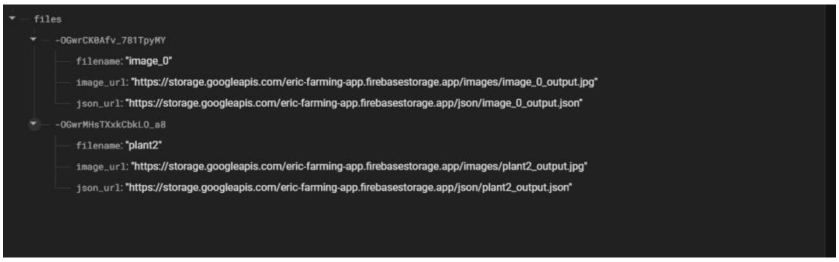
Figure 2. Screenshot of plant details

```
def clean_raw_response(raw_response):

def upload_to_gemini(path, mime_type=None):
    """Uploads the given file to Gemini."""
    try:
        file = genai.upload_file(path, mime_type=mime_type)
        logger.info(f"Uploaded file '{file.display_name}' as: {file.uri}")
        return file
    except Exception as e:
        logger.error(f"File upload failed: {e}")
        return None

def create_gemini_model(model_name):
    """Create a Gemini model instance."""
    generation_config = {
        "temperature": 0.7,
        "top_p": 0.9,
        "top_k": 64,
        "max_output_tokens": 512,
        "response_mime_type": "text/plain",
    }
    return genai.GenerativeModel(model_name=model_name, generation_config=generation_config)

def parse_raw_response(response_text):
    """Attempts to parse the raw response text into JSON."""
    try:
        # Extract JSON content if enclosed in triple backticks
        if "```json" in response_text:
            start = response_text.index("```json") + 7
            end = response_text.index("```", start)
            response_text = response_text[start:end].strip()
        return json.loads(response_text)
    except json.JSONDecodeError as e:
        logger.error(f"Failed to parse JSON: {e}")
        return [{"name": "Raw Response", "fact": response_text.strip()}]

```

Figure 3. Screenshot of code 1

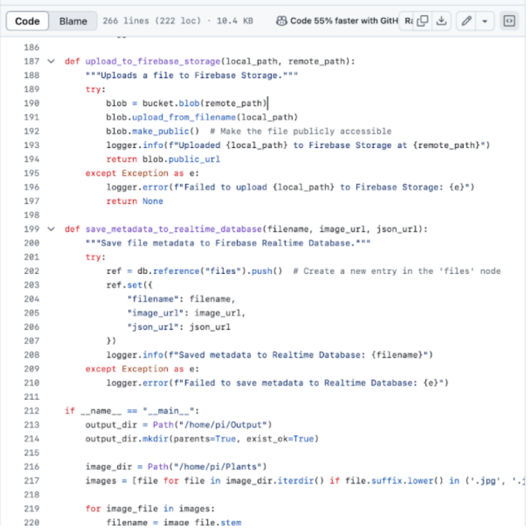
The process begins with the robot capturing an image of the plant, which is then processed on the Raspberry Pi before being sent to Gemini AI for analysis. The upload_to_gemini(path, mime_type=None) uploads the image to Gemini's API and returns a file URI. Once uploaded, the create_gemini_model(model_name) starts a Gemini model with specific parameters, and makes the AI to accurately identify the plant species. After processing, Gemini returns a text response, which is then handled by parse_raw_response(response_text), where it is transferred into a JSON format. This stored data is then stored in a Firebase JSON file, allowing applications to provide the users with valuable insights on plant identification.

Another key component of my program is Firebase, which acts as the central storage system for plant images and analysis results. This component ensures data persistence and accessibility, allowing both the robot and application to retrieve and store information. Firebase relies on cloud storage and real-time database services, enabling seamless communication between the robot and the application. The Firebase JSON database stores structured data, including images and Gemini's analysis, which can then be accessed by the application to request further insights from OpenAI. Without Firebase, there would be no centralized system to manage and synchronize data between the different parts of the program.



```
files
  -OGwrCK0Afv_781TpyMY
    filename: "image_0"
    image_url: "https://storage.googleapis.com/eric-farming-app.firebasestorage.app/images/image_0_output.jpg"
    json_url: "https://storage.googleapis.com/eric-farming-app.firebasestorage.app/json/image_0_output.json"
  -OGwrMHsTXxkCbkLO_a8
    filename: "plant2"
    image_url: "https://storage.googleapis.com/eric-farming-app.firebasestorage.app/images/plant2_output.jpg"
    json_url: "https://storage.googleapis.com/eric-farming-app.firebasestorage.app/json/plant2_output.json"
```

Figure 4. Screenshot of files



```

def upload_to_firebase_storage(local_path, remote_path):
    """Uploads a file to Firebase Storage."""
    try:
        blob = bucket.blob(remote_path)
        blob.upload_from_filename(local_path)
        blob.make_public()  # Make the file publicly accessible
        logger.info(f"Uploaded {local_path} to Firebase Storage at {remote_path}")
        return blob.public_url
    except Exception as e:
        logger.error(f"Failed to upload {local_path} to Firebase Storage: {e}")
        return None

def save_metadata_to_realtime_database(filename, image_url, json_url):
    """Save file metadata to Firebase Realtime Database."""
    try:
        ref = db.reference("files").push()  # Create a new entry in the 'files' node
        ref.set({
            "filename": filename,
            "image_url": image_url,
            "json_url": json_url
        })
        logger.info(f"Saved metadata to Realtime Database: {filename}")
    except Exception as e:
        logger.error(f"Failed to save metadata to Realtime Database: {e}")

if __name__ == "__main__":
    output_dir = Path("/home/pi/Output")
    output_dir.mkdir(parents=True, exist_ok=True)

    image_dir = Path("/home/pi/Plants")
    images = [file for file in image_dir.iterdir() if file.suffix.lower() in ('.jpg', '.

    for image_file in images:
        filename = image_file.stem
```

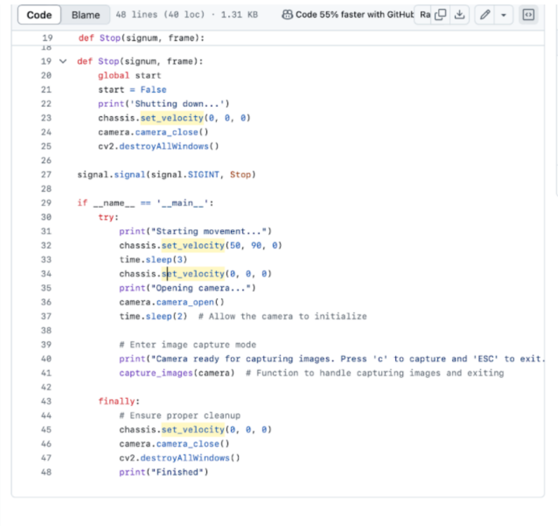
Figure 5. Screenshot of code 2

This shows the process of uploading plant images to Firebase Storage. The upload_to_firebase_storage(local_path, remote_path) function uploads an image file to Firebase Storage, making it publicly accessible. But if the upload fails, it logs an error and returns None. The save_metadata_to_realtime_database(filename, image_url, json_url) function stores essential data—including the filename, image URL, and JSON URL—in Firebase Realtime Database. In the __main__ block, the code sets up directories for storing plant images. These backend Firebase services enable communication between the robot and the cloud. Making sure that images are properly stored and analyzed by Gemini AI for plant identification.

The third essential component of my program is the robot's image capture system, which serves as the initial step and the base of the entire process. The robot uses a camera module to capture images of plants, which are then sent for analysis. The component is programmed using Python, allowing it to adjust different parameters. Without this component, the system would lack the necessary step to perform plant classification and analysis, because it won't be able to capture the picture.

Succulent
Requires well-draining soil and b
Pink Artichoke Pot
Aesthetically pleasing but drainag
health.

Figure 6. Screenshot of the detail



```
def Stop(signum, frame):

def Stop(signum, frame):
    global start
    start = False
    print('Shutting down...')
    chassis.set_velocity(0, 0, 0)
    camera.camera_close()
    cv2.destroyAllWindows()

signal.signal(signal.SIGINT, Stop)

if __name__ == '__main__':
    try:
        print("Starting movement...")
        chassis.set_velocity(50, 90, 0)
        time.sleep(3)
        chassis.set_velocity(0, 0, 0)
        print("Opening camera...")
        camera.camera_open()
        time.sleep(2)  # Allow the camera to initialize

        # Enter image capture mode
        print("Camera ready for capturing images. Press 'c' to capture and 'ESC' to exit.
        capture_images(camera)  # Function to handle capturing images and exiting

    finally:
        # Ensure proper cleanup
        chassis.set_velocity(0, 0, 0)
        camera.camera_close()
        cv2.destroyAllWindows()
        print("Finished")
```

Figure 7. Screenshot of code 3

The code here controls the robot's movement and camera activation for capturing plant images. At the start of the program, the robot begins moving using chassis.set_velocity(50, 90, 0), then stops to take a photo. The Stop(signum, frame) function is an emergency shutdown mechanism, where the setting is start = False, stopping the chassis, closing the camera, and clearing OpenCV windows. In the main execution block (if __name__ == "__main__":), the robot moves forward, initializes the camera, and enters image capture mode. Users can capture images by pressing 'C' on the keyboard. The finally block ensures that once the image is captured, the chassis stops, the camera closes, and all OpenCV windows are properly shut down to prevent memory leaks or errors.

4. EXPERIMENT

4.1. Experiment 1

A possible blind spot is the robot's ability to capture consistently clear images for plant health analysis, especially under low light or uneven sunlight conditions commonly encountered in farm environments.

To test image capture consistency, we will place the robot in various lighting environments throughout the day—early morning, midday, and sunset. At each interval, the robot will capture images of the same plant from the same angle and distance. Each photo will then be analyzed for clarity, contrast, and object recognition consistency using the same AI model. We will also run a control set in a well-lit, indoor setting. This setup allows us to isolate the effects of environmental lighting. Data from the AI's analysis will be compared for accuracy and consistency across lighting conditions to determine if preprocessing or hardware enhancements are needed.

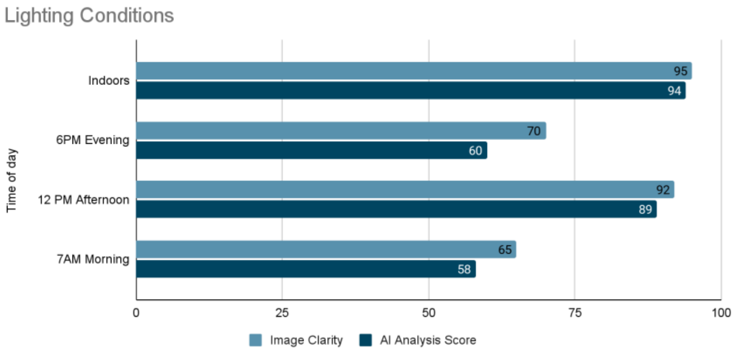


Figure 8. Figure of experiment 1

Our results show that image quality and AI analysis accuracy dropped significantly under low or uneven lighting conditions. In the early morning and late evening tests, images suffered from underexposure or shadowing, leading to lower clarity and reduced plant feature detection accuracy. In contrast, midday and indoor tests performed well, with AI analysis matching known plant health indicators over 90% of the time. The most important factor affecting results was the lighting uniformity. This confirms our hypothesis that inconsistent outdoor light negatively affects plant image recognition. To mitigate this, we recommend adding a consistent lighting source such as LED ring lights to the camera mount and incorporating real-time image preprocessing. These enhancements will improve reliability regardless of time or weather, making the smart farm robot's plant analysis more robust in real-world agricultural environments.

4.2. Experiment 2

Another possible blind spot is the accuracy of the robot's movement along planned paths. If the robot drifts or overshoots, it may miss important scan zones or damage plants.

To evaluate movement accuracy, we will test the robot's ability to follow a preprogrammed path marked with flag points spaced 1 meter apart. The robot will perform 10 runs along the path in both dry and damp soil conditions. At each flag, we will record the distance between the robot's expected and actual stopping points. GPS or line sensors may assist, but visual markers and measuring tape will be used to calculate deviation. A control condition on smooth flooring indoors will serve as a baseline. This setup will help identify whether environmental surface conditions impact navigation and help refine motor calibration or path correction algorithms.

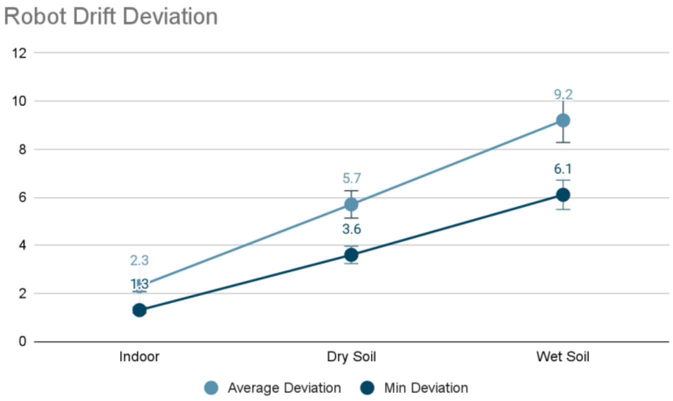


Figure 9. Figure of experiment 2

The results show that the robot's movement is precise on smooth, flat surfaces but suffers increased deviation in outdoor soil environments, especially when moisture is present. In the dry soil condition, average deviation from target positions nearly tripled compared to the indoor control. In damp soil, the deviation increased fourfold. This likely resulted from the wheels losing traction or sinking slightly, causing slippage or momentum drift. These findings confirm that soil conditions directly affect path accuracy and suggest the need for terrain-aware movement controls. Adding wheel encoders with real-time position correction or integrating low-cost GPS/IMU fusion could help minimize error in outdoor conditions. Another solution is adaptive motor power tuning based on surface resistance. Addressing this blind spot is crucial, as even small navigation errors may lead to missed plant scans or unintentional crop damage. Enhancing movement precision ensures that the robot can reliably operate across diverse farm terrains.

5. RELATED WORK

A similar solution/example to my robot is the terrasentia robot, it is a small farm ground rover that is created to move and collect data between rows of crops. It comes with advanced cameras, such as LiDAR, and has an advanced AI program that allows the robot to take detailed pictures such as the plant's height, stem width, and whether they show diseases [21]. This system from the terrasentia robot is pretty useful because it collects the data from the plants and this will reduce the time farmers need to spend checking the crops by hand. However the big limitation is that TerraSentia is mostly built for plant phenotyping research in the different university and research fields. This will mean it will prioritize scientific data collection rather than a simple and easy way that the regular farms can check and use everyday. Furthermore a single Terrasentia robot is extremely expensive and not very adaptable, making it less useless for small to medium farmers who cannot afford such high cost. Then comparing it to our robot, it is cheaper, more flexible, and more practical. Instead of only collecting the research data, our system will detect soil moisture, the need for pesticide, and the different plant diseases, which it can give direct feedback and room for improvements on the APP, and through alerts. This is very friendly and more accessible for farmers and plant growers who need simple and fast solutions to their problems.

The article "Plant disease detection using drones in precision agriculture", a systematic review of UAV plant disease detection found that many drones today are equipped with RGB, multispectral, and advanced sensor that can detect diseases like blight and fungus infection across large amount of lands with high spatial and spectral resolution [20]. While these systems and technology are highly advanced and enable rapid, early disease wipeout, they depend on a very clear image quality, which is really to obtain when they are in the sky. Furthermore, they struggle under occlusions and variable lighting, and they cannot physically treat or remove diseased plants. On the other hand, our robots are designed and targeted to small-scale farmers, where our robots are able to capture close range images that are more precise and clear that gives farmers the most accurate results from one plant to another. Our robot also have pesticides/ disease species inspection/detection, it also applies to targeted to mechanical or non-chemical removal, and most importantly our robot is able to work in any environment terrain, unlike drones, if they get too low to the crops it might get stuck or block the image used to detect the plants, our robot resolve these deadly problems faced by drones.

Another thing that most people use in agriculture is satellite remote sensing, and this is the method that relies on different satellites like Sentinel 2 to capture images from space. These can calculate and discover different vegetation indices like NDVI or EVI. This basically measures how green or how healthy the plants look [22]. These images and analysis could help farmers and researchers to see patterns in crop growth, water stress, and even possible disease outback across very large areas. The main strength of satellites is that it can cover huge areas at once, something that the robots on the ground struggle to do. However, satellites come with a lot of weaknesses. First their resolution is not very detailed, it usually takes 10-30 meters per pixel, this makes it very hard to see what is happening with individual plants. They also struggle with cloudy days that contain many clouds, which can block the images, and they cannot see under the canopy of crops where the disease and pest actually begin [23]. Due to these limitations satellites are great for the “big picture" monitoring, but not for close up plant health checks. Our robot will fill this specific gap by taking clear, close-range photos under the leaves, using sensors and even adding lighting to make sure that the images are consistent and linear every time. So, this way, satellite can guide farmers in which a field need special attention, and then our robot can go to the detail analyzation and give farmers the exact information they need.

6. CONCLUSIONS

While our farm robot offers a very promising and diverse solution to many of the agricultural challenges to the world, there are also some limitations. One major limitation is its dependency on the internet to access AI services like Gemini and OpenAI. However in many rural farming areas where there is no internet access, this would cause delay or failure of processing data. Or even in worse cases it would cause the robot to stop running. After doing some research it is found that hotspot and starlink can solve parts of the problem, but hotspot is slow and unstable. Starlink does not provide affordable, quick, and global coverage internet. One future improvement would be using edge computing or local data processing, where as it sends information to a local data processing center that would do the job like OpenAI and Gemini. In this case this would allow the robot to operate without an internet connection. The second limitation is accuracy and the reliability of the sensor when it is used when detecting the useful data. Whereas in the current phase it can only provide the basic physical identification and detection.Going in further, the soil moisture sensors, for example, can be strongly influenced by the environment like the soil composition and temperature,this can produce very inconsistent and different results across different places. Furthermore, pesticides are evolving as the different pests are evolving, we must not limit our detection to specific chemicals, in this case robots can generate false negatives. The third limitation is the battery capacity and the overall energy management system. The robot base can only operate around 50 minutes to an hour, and this would require the user to charge after each time, and switch to new batteries. This reduces its ability to cover large fields efficiently. Additionally, the current hardware setup allows and does for basic detection and identification. However, it is not ruggedized for long term outdoor use. Where the exposure to dust, water, and heat will cause damage to sensors and electronics on the robot.

If I had more time and money, many improvements would be made. First, I would like to integrate more advanced and high-quality sensors that are specifically designed for agricultural uses. A great thing about these sensors is that they are more resistant to the different environmental conditions that it will face, and it will highly improve the accuracy of data collection. Second, I would implement a and add a local LLM (like Ollama.) that would have offline capability. This would allow the robot to process the image locally and not needing to have internet connection, which many farms don't have. Third I would also want to expand robot life by including solar panels so it has more efficient and longer battery use. This allows the farmers to leave the robot in the field for a longer time.

This project demonstrates how robotics, AI, and sensors can be combined in use to address many different challenges in agriculture, like monitoring plant and soil health, detecting pesticide, and providing valuable information for improving plant care. While this is a prototype in early stages, it still shows how low cost hardware can be integrated with LLM's and cloud-based AI tools to create something useful for many farmers around the world. In the future, the prototype would evolve into a more developed and more advanced agricultural robot by incorporating local LLMs implementation for offline analysis, high quality and more advanced sensors, and solar panels to increase the battery life. These improvements not only transform the robot from a data gathering device into a decision-making decision assistant but it also helps farmers optimize, help, and monitor their plants and resources.

Now moving on to personal development/level, I really experienced how challenging it is to move from the concept of an idea to the actual implementation of the idea. Building this robot required not only the knowledge of the different sensors, but also knowledge about raspberry pi, programming, AI integration, and also the understanding of practical realities that farmers face like cost, connectivity, and reliability. During this process I also realized that innovation is not just about building advanced technology, but it is more about building advanced technology so that it can be helpful to people who are in need. This project really inspired me to explore how technology can be used to solve real world problems in many meaningful ways.

REFERENCES

[1] Costea, M., et al. "CHALLENGES AND ISSUES FOR FARMERS IN THE MODERN ERA." Research Journal of Agricultural Science 55.2 (2023).

[2] Devlet, Abdulgani. "Modern agriculture and challenges." Frontiers in Life Sciences and Related Technologies 2.1 (2021): 21-29.

[3] Ahearn, Mary Clare. "Potential challenges for beginning farmers and ranchers." Choices 26.2 (2011).

[4] Kachroo, Mohammad Mubashir, and Arshad Bhat. "Agriculture and Farming Community in India: Challenges, Problems and Possible Solutions." Indian Agriculture, Farmer and Labour: Issues and Reforms 1.

[5] Von Loeper, Wolfgang, et al. "Analysing challenges facing smallholder farmers and conservation agriculture in South Africa: A system dynamics approach." South African Journal of Economic and Management Sciences 19.5 (2016): 747-773.

[6] Mpandeli, Sylvester, and Phokele Maponya. "Constraints and challenges facing the small scale farmers in Limpopo Province, South Africa." Journal of agricultural Science 6.4 (2014): 135.

[7] Rockler, Briana E., et al. "Something to eat: experiences of food insecurity on the farm." Agriculture and Human Values 40.4 (2023): 1419-1436.

[8] Touch, Van, et al. "Smallholder farmers' challenges and opportunities: Implications for agricultural production, environment and food security." Journal of Environmental Management 370 (2024): 122536.

[9] Shukla, Smriti, et al. "Challenges faced by farmers in crops production due to fungal pathogens and their effect on Indian economy." Fungal diversity, ecology and control management. Singapore: Springer Nature Singapore, 2022. 495-505.

[10] Tago, Damian, et al. "The impact of farmers' strategic behavior on the spread of animal infectious diseases." PLoS One 11.6 (2016): e0157450.

[11] Reilly, John, et al. "Global economic effects of changes in crops, pasture, and forests due to changing climate, carbon dioxide, and ozone." Energy Policy 35.11 (2007): 5370-5383.

[12] Vurro, Maurizio, Barbara Bonciani, and Giovanni Vannacci. "Emerging infectious diseases of crop plants in developing countries: impact on agriculture and socio-economic consequences." Food security 2.2 (2010): 113-132.

[13] Purcell, A., and S. Saunders. "Glassy-winged sharpshooters expected to increase plant disease." California Agriculture 53.2 (1999): 26-27.

[14] Sisterson, Mark S., et al. "Distribution and management of citrus in California: implications for management of glassy-winged sharpshooter." Journal of Economic Entomology 101.4 (2008): 1041-1050.

[15] Caffarra, Amelia, et al. "Modelling the impact of climate change on the interaction between grapevine and its pests and pathogens: European grapevine moth and powdery mildew." Agriculture, Ecosystems & Environment 148 (2012): 89-101.

[16] Gadoury, David M., et al. "Effects of powdery mildew on vine growth, yield, and quality of concord grapes." Plant disease 85.2 (2001): 137-140.

[17] Brazel, Skyler R., et al. "Overwatering may be as detrimental as underwatering in container-grown kale (Brassica oleracea L. acephala)." Scientia Horticulturae 316 (2023): 111961.

[18] Jack, B. Kelsey. "Market inefficiencies and the adoption of agricultural technologies in developing countries." (2013).

[19] Bain, Carmen, and Tamera Dandachi. "Governing GMOs: The (counter) movement for mandatory and voluntary non-GMO labels." Sustainability 6.12 (2014): 9456-9476.

[20] Trybała, Paweł, et al. "Under-canopy drone 3D surveys for wild fruit hotspot mapping." Drones 8.10 (2024): 577.

[21] Shafiekhani, Ali, et al. "Vinobot and vinoculer: Two robotic platforms for high-throughput field phenotyping." Sensors 17.1 (2017): 214.

[22] Sishodia, Rajendra P., Ram L. Ray, and Sudhir K. Singh. "Applications of remote sensing in precision agriculture: A review." Remote sensing 12.19 (2020): 3136.

[23] Wu, Bingfang, et al. "Challenges and opportunities in remote sensing-based crop monitoring: A review." National Science Review 10.4 (2023): nwac290.

©2026 By AIRCC Publishing Corporation. This article is published under the Creative Commons Attribution (CC BY) license.