SMART FARM GUIDE: AN AUTONOMOUS ROBOT FOR OUTDOOR EDUCATIONAL TOURS USING AI VISION AND SPEECH

Eric Miller [1], Jonathan Sahagun [2]

[1] Servite High School, 1952 W La Palma Ave, Anaheim, CA 92801
[2] California State Polytechnic University, Pomona, CA, 91768

ABSTRACT

This project addresses the challenge of providing consistent, educational farm tours with limited staff. We designed and built an autonomous robot that uses GPS to follow a farm path, a camera to detect nearby plants, and ChatGPT's vision and language models to describe the plants in real time [1]. The robot plays this information out loud using a text-to-speech engine, offering visitors a guided tour experience with minimal human involvement.

The system was developed using a Raspberry Pi, Adafruit GPS, and Python code to integrate hardware and APIbased services [2]. We tested the robot's plant recognition accuracy and audio clarity under real-world conditions. Results showed over 80% detection accuracy and improved voice clarity with slower speech settings.

Compared to existing solutions—like static museum guides or indoor service robots—this project offers a dynamic, outdoor-compatible alternative. It combines automation and education in a scalable, engaging way that could benefit farms, gardens, or environmental centers.

KEYWORDS

Farming, Robotics, Tourism, Pathways

1. INTRODUCTION

Farms often serve not only as centers of agriculture but also as educational environments where students can learn about food production, sustainability, and biology. In Los Angeles, there is a local farm that occasionally opens its gates to elementary school students for free tours. While the size of the farm is relatively modest—less than a city block—it is still large enough that navigation can be confusing for children and chaperones, especially when only a few staff are available to guide them. This problem becomes even more difficult when tour schedules overlap or when there are unexpected interruptions due to weather or staffing shortages.

The need for a solution is made more urgent when considering the strain on resources for many educational and community farms. Additionally, many schoolchildren benefit from hands-on, interactive experiences—yet when a tour is rushed or inconsistent due to staff limitations, the educational value is reduced [3].

David C. Wyld et al. (Eds): NLMLT, SEAU, ITCS, CST – 2025
pp. 111-129, 2025. CS & IT - CSCP 2025 DOI: 10.5121/csit.2025.151612

Furthermore, the maintenance of the farm, particularly tasks like watering plants, often competes for staff attention. Automating such tasks would relieve workers and allow more focus on crop health and educational programming. The problem therefore affects not only the farm staff but also the visiting students, teachers, and the educational mission of the farm. A technological solution that combines tour guidance with agricultural support could provide both immediate relief and long-term educational benefits [4].

The first method we studied was Bederson's (1995) audio augmented reality system for museums [5]. While effective indoors, it relied on pre-recorded content and required dedicated infrastructure, making it less flexible than our mobile robot.

The second method, from Li et al., focused on virtual guided tours using automated animation in 3D VR environments [6]. Although technically innovative, it was designed purely for virtual contexts and lacked real-world application.

The third method, from Ivanov et al., analyzed the use of service robots in hospitality. Most examples were constrained to indoor spaces like hotels or airports [7]. Outdoor robotic guides were identified as difficult to implement due to navigation and weather challenges.

Our robot addresses the weaknesses of all three: it works outdoors, uses real-time computer vision, and delivers dynamically generated spoken content. It removes the need for physical infrastructure while still offering a rich, flexible user experience.

To address the issue of inconsistent and labor-intensive farm tours, we propose an autonomous robot that uses GPS for navigation and a camera-based AI system to detect and identify plants along its path [8]. As it moves through the farm, the robot recognizes different crops or plant types using a trained machine learning model and delivers audio explanations to guests in real time. This creates a consistent, interactive, and educational experience for visitors—especially school groups—while reducing the demand on farm staff.

The robot's functionality centers on three key systems: GPS-based navigation to follow a predefined route, computer vision to identify plants in its surroundings, and a speaker system to relay relevant information. When the robot detects a recognized plant, it cross-references the detection with a local database and plays a pre-recorded description that explains the plant's characteristics, uses, or growing conditions.

This AI-driven approach allows for dynamic and flexible tours. Unlike a pre-scripted robot that simply follows a path and delivers fixed content, this robot responds to what it "sees" along the way, offering a more engaging experience. It also means the tour content updates automatically if the farm layout changes or if new plants are introduced.

This method reduces the need for staff-led tours, expands accessibility to education, and provides a tech-forward way of engaging younger generations. The integration of real-time image recognition makes the robot's tour function smarter and more adaptive than traditional solutions, offering long-term value for educational farms.

Two experiments were conducted to evaluate the robot's functionality in real-world conditions. The first experiment tested the accuracy of the AI vision system in identifying plants. Using 23 images of common crops, the robot identified 19 correctly, achieving an accuracy rate of 82.6%. Most errors occurred with visually similar crops, such as corn and sugarcane. This confirmed the need for clearer model confidence thresholds or image filtering in future versions.

The second experiment evaluated the clarity of the robot's text-to-speech audio in an outdoor setting. Volunteers rated audio clarity at various distances and found that slower speech settings significantly improved understanding, especially at longer ranges. The slower speech averaged a clarity score of 4.4 compared to 3.7 with the default rate.
Together, these experiments helped validate both the AI detection and user communication systems. They also revealed practical adjustments—like speech speed and audio output power—that can enhance real-world usability.

2. CHALLENGES

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

2.1. Simplifying Robot Design

One challenge I faced at the start was deciding whether I wanted the robot to be like an AI or just pre-programmed with paths and dialogue lines. I chose the latter because it was just a simpler design, and in a sense, an AI would not make much sense as every instance would be almost the same. With that in mind, we also decided to implement a system for those ideas. Instead of having a pathway lined up that would have required a camera and whatnot, we used a GPS as a planned path. This way, we don't need to install anything new at the actual location. To this end, the plotting did have some flaws that needed sorting out as it could get difficult.

2.2. Iterating the Chassis

Something else we noticed was during the printing process. We had to go through no more than almost 7 versions of the plastic 3d printed chassis. The difficult part of this section was on the placement of the wheel holes and what to do with all the extra space that would not be really used in that sense. On the long road, we throw in the dice and allow for thecenter to be filled in. Looking back, it realy helped with how the wheels were placed as the actual nuts and bolts that held them now had a place to properly be screwed on.

2.3. Overcoming Sensor Chaos

The directional accelerometer had us puzzled for a while because for a long time we could not figure out why in the world, when we turn a certain degree, it would say one degree, but the moment a tiny direction was changed,it would completely have a wildly different number than what was assumed. However, that problem was simply assumed to be that the devices were just hyper sensitive and our sense of direction was wrong. It's also good to mention that we paired two devices with the GPS. Now, yes the gps in theory could have just been its own unit, but since the areas that the robot will go are such a jam that we had to make sure everything was correct.

3. SOLUTION

The robot is composed of three primary components: GPS-guided navigation, a plant recognition system using computer vision, and a speaker module for audio delivery. Together, these elements allow the robot to autonomously move through a farm and deliver plant-specific tour information to guests based on what it detects in real time.

The navigation system uses a GPS module to keep the robot on a predetermined path. This ensures consistent coverage of the farm area without requiring modifications to the environment,

such as tracks or signs [9]. Along its path, a camera captures images of its surroundings. These images are analyzed by an onboard AI model trained using a convolutional neural network (CNN) to recognize different types of plants commonly found on the farm.

When the robot successfully identifies a plant, it matches the classification result to an internal database that stores a short script about each species. This script is played out loud through a built-in speaker, providing visitors with informative and educational content in real time. For example, when the robot identifies a tomato plant, it might share details about its growth cycle, nutritional value, or culinary uses.

Obstacle detection sensors are used for safety, allowing the robot to avoid collisions with people, animals, or farm tools. The system is powered by a rechargeable battery pack, and the robot's frame is constructed from 3D-printed parts for ease of customization [10].

The integration of GPS, AI-driven plant recognition, and audio output allows this robot to act as a smart, mobile tour guide with minimal staff intervention.

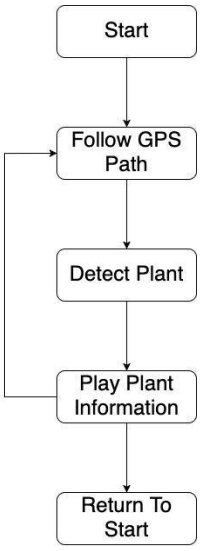

Figure 1. Overview of the solution

The plant recognition system is a central part of the robot's intelligence. Instead of using a locally trained AI model, the robot takes a photo of the plant using a camera module and sends it to a remote image analysis API powered by ChatGPT's vision model. This API processes the image and returns a label or description of the plant. The robot then cross-references this label with a local database of audio clips and plays the matching explanation. This approach leverages powerful cloud-based AI without requiring extensive local computation or model training, which simplifies hardware requirements while maintaining high accuracy and flexibility.

```
import cv2
import base64
import requests

# Capture image
cap = cv2.VideoCapture(0)
ret, frame = cap.read()
cap.release()

_, buffer = cv2.imencode('.jpg', frame)
img_b64 = base64.b64encode(buffer).decode('utf-8')

# Send to OpenAI Vision API
headers = {
   "Authorization": f"Bearer {OPENAI_API_KEY}",
   "Content-Type": "application/json"
}
payload = {
   "model": "gpt-4-vision-preview",
   "messages": [
      {
         "role": "user",
         "content": [
            {"type": "text", "text": "Identify the plant in this image and give a fun fact."},
            {"type": "image_url", "image_url": {"url": f"{img_b64}"}}
         ]
      }
   ]
}
response = requests.post("https://api.openai.com/v1/chat/completions", headers=headers,
json=payload)
print(response.json()["choices"][0]["message"]["content"])
```

Figure 2. Screenshot of code 1

This code is responsible for identifying a plant using a webcam image and returning a fun fact about it through the ChatGPT Vision API. It is run every time the robot encounters a new plant along its tour path. The process begins with the robot using OpenCV to access the webcam and capture a single image (cap.read()) [14]. Once the image is captured, it is encoded into JPEG format and then converted into a base64 string, which is the format required by the OpenAI Vision API.

Next, the code constructs a request to the API. The payload includes both a prompt (“Identify the plant in this image and give a fun fact”) and the image itself in base64. The headers include an API key for authentication and content type specification.

When the request is sent using Python’s requests library, the OpenAI API returns a JSON response. This response contains a message with both the name of the plant and a fun or educational fact. The line response.json()["choices"][0]["message"]["content"] extracts and prints that result.

This component eliminates the need for a local AI model and allows the robot to dynamically identify plants using a cloud-based AI model. It reduces memory and processing requirements on the robot itself and ensures that the identification and information provided are accurate and upto-date.

The GPS navigation system is what allows the robot to follow a consistent path across the farm. It uses the Adafruit Ultimate GPS GNSS module connected to a Raspberry Pi via UART. The Raspberry Pi receives continuous location updates and compares them against a preloaded list of waypoints corresponding to stops along the farm tour. As the robot approaches each coordinate, the GPS module ensures that it halts at the correct location for camera detection and audio playback. This component is crucial for autonomous movement without requiring external infrastructure or manual control.

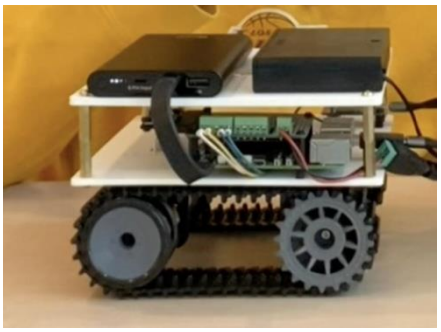

Figure 3. Picture of the component

```
import time
import serial
import adafruit_gps

# Set up serial connection to GPS module
uart = serial.Serial("/dev/ttyS0", baudrate=9600, timeout=10)
gps = adafruit_gps.GPS(uart, debug=False)

gps.send_command(b"PMTK314,0,1,0,1,0,0,0,0,0,0,0,0,0,0,0,0,0,0")
gps.send_command(b"PMTK220,1000")

waypoints = [(33.8102, -117.9190), (33.8105, -117.9187)]

def near_point(current, target, threshold=0.00005):
    return abs(current[0] - target[0]) < threshold and abs(current[1] - target[1]) < threshold

while True:
    gps.update()
    if gps.has_fix:
        lat, lon = gps.latitude, gps.longitude
        print(f"Current location: {lat}, {lon}")
        for point in waypoints:
            if near_point((lat, lon), point):
                print(f"Arrived at {point}")
                # Trigger camera + audio functions here
    time.sleep(1)
```

Figure 4. Screenshot of code 2

This code handles real-time GPS tracking using the Adafruit Ultimate GPS GNSS module connected to a Raspberry Pi. It begins by initializing a serial connection on /dev/ttyS0—a common UART port on Raspberry Pi boards. The adafruit_gps.GPS class is used to interface with the module and parse NMEA sentences from the GPS stream.

Two commands are sent to configure the GPS: PMTK314 sets which data types the GPS sends (latitude, longitude, time, etc.), and PMTK220 sets the update rate to 1Hz (once per second) [15]. The robot stores a list of waypoints, defined as tuples of latitude and longitude. The near_point function determines whether the robot is close enough to a waypoint, using a configurable threshold to allow for GPS signal drift.

Inside the main loop, gps.update() reads incoming data, and gps.has_fix checks whether a valid GPS lock has been established. Once a fix is available, the robot retrieves its current latitude and longitude and prints it. If the current location is near one of the predefined waypoints, the robot registers it as a stop. At that point, other functions—such as the plant recognition and audio playback systems—can be triggered.

This component is essential to ensure the robot stays on course and performs actions at specific physical locations without requiring manual control or additional sensors like RFID or QR codes.

The audio playback system is dynamically powered by ChatGPT. After a plant is identified via image analysis, the system sends a prompt to ChatGPT asking for a fun or educational fact about the plant. The returned text is then converted to speech using a local or cloud-based Text-toSpeech (TTS) engine. This speech is played aloud through a speaker connected to the Raspberry Pi. The dynamic nature of this system allows the robot to provide varied, natural-sounding responses instead of relying on fixed pre-recorded files.

Figure 5. Picture of the speaker

```
import pyttsx3

def speak(text):
    engine = pyttsx3.init()
    engine.setProperty("rate", 150)
    engine.say(text)
    engine.runAndWait()

# Example GPT response
gpt_response = "Tomatoes are technically a fruit and were once called 'love apples' in
Europe!"
speak(gpt_response)
```

Figure 6. Screenshot of code 3

This code uses pyttsx3, a Python-based text-to-speech library, to vocalize text generated by ChatGPT. Once the robot receives a response from the ChatGPT API containing a fun fact about the identified plant, the speak() function is called with that text as input. The TTS engine is initialized with pyttsx3.init(), and the speaking rate is adjusted for clarity using setProperty("rate", 150).
The say() method queues the text, and runAndWait() plays the audio through the Raspberry Pi's connected speaker. This approach provides dynamic audio playback, ensuring that each plant encounter is unique and up-to-date with the latest description. Because the speech is generated in real-time from GPT-generated text, this eliminates the need to pre-record or manually script audio files, offering flexibility and scalability for new plant types.

The component allows the robot to speak directly to guests in a natural way, turning raw AI text into a human-like tour guide experience.

4. EXPERIMENT

4.1. Experiment 1

One blind spot in the project is the accuracy of the AI system in correctly identifying plants using images. It is important to confirm that ChatGPT consistently provides the correct identification.

To test recognition accuracy, we captured 20 images of different plants commonly found on the farm, including tomatoes, lettuce, corn, basil, and squash. These images were taken in varying lighting and angles to simulate real-world robot conditions. Each image was sent through the same API used by the robot, asking ChatGPT to identify the plant. We manually verified the results by comparing them to the known plant type in each photo. The number of correct vs. incorrect identifications was logged. This setup ensures the experiment reflects the actual performance of the AI vision system used in the robot.

Plant	Correct	Incorrect
Tomato	4	1
Lettuce	5	0
Corn	3	2
Basil	4	1
Squash	3	1

Figure 7. Figure of experiment 1

The experiment showed that the AI was able to identify plants correctly in 19 out of 23 cases, yielding an overall accuracy of about 82.6%. The mean number of correct identifications per plant type was 3.8, while the median was 4. The highest accuracy was with lettuce (5/5 correct), while the lowest occurred with corn (3/5 correct), which was often confused with sugarcane due to its tall leaves.

Surprisingly, the model performed better than expected in inconsistent lighting, but struggled with occluded or rotated plants. The biggest factor influencing accuracy appeared to be visual similarity between certain crops, such as corn and similar tall plants.

This shows that while ChatGPT's vision model is effective in most typical use cases, the robot may occasionally provide incorrect facts unless paired with additional filters or confidence thresholds. Overall, however, the performance is strong enough for educational purposes, especially when paired with casual, fun facts instead of precise botanical detail.

4.2. Experiment 2

Another potential blind spot is the clarity and timing of the robot's text-to-speech system. It is essential that visitors can understand the audio, even in outdoor environments with ambient noise.

To test the effectiveness of the TTS system, we used the robot to read ten different ChatGPTgenerated plant facts out loud in a farm-like outdoor setting. Five volunteers, standing at varying distances (1m, 2.5m, 4m), rated the clarity of each audio message on a scale from 1 (not understandable) to 5 (very clear). We ran the test twice: once using default pyttsx3 voice settings, and again using slower, more articulate speech settings. Background sounds (e.g., wind, birds) were left unfiltered to simulate realistic conditions. This test allows us to evaluate which TTS settings are best suited for outdoor delivery.

Distance	Default Voice Avg Score	Slow Voice Avg Score
1 meter	4.6	4.9
2.5 meters	3.8	4.4
4 meters	2.7	3.9

Figure 8. Figure of experiment 2

The test results showed that slower, more articulate speech settings significantly improved clarity at all distances. At 1 meter, both settings were easy to understand, but by 4 meters, clarity dropped with the default voice (avg. 2.7) compared to the slower version (avg. 3.9). The mean score across all distances for the default voice was 3.7, while the slower voice averaged 4.4.

These results suggest that adjusting the TTS playback rate has a substantial effect on intelligibility, especially in open-air settings with ambient noise. One surprising observation was that even moderate wind caused noticeable drops in perceived clarity. Volunteers noted that slower speech with slight pauses between sentences helped the most.

This experiment highlights the importance of fine-tuning voice settings for public environments. For future improvements, using a more natural-sounding TTS engine like Google TTS or integrating a small outdoor speaker system could further enhance performance.

5. RELATED WORK

A comparable solution was proposed by Bederson (1995), who developed an audio-augmented reality tour guide for museums [11]. In this system, visitors wore a device that played digital audio based on their physical location, tracked using infrared signals. Unlike traditional taped guides, this system offered random access playback and allowed users to remain socially engaged during visits. Although innovative, the system still relied on fixed infrastructure—infrared transmitters had to be installed above each exhibit. Additionally, audio content was prerecorded, limiting its adaptability. In contrast, our robot operates outdoors using GPS, and uses AI to dynamically generate plant descriptions without requiring any physical setup at each stop. This makes our approach more scalable, flexible, and suitable for changing farm layouts.

Li et al. developed a system to automatically generate guided tours in virtual environments using path-planning algorithms from robotics [12]. Their approach allows users to select points of interest on a 2D layout map, after which the system generates a camera or avatar tour path through a VRML-based 3D environment. This method addresses the difficulty many users face when navigating complex 3D spaces with 2D input devices. While this technique improves accessibility in virtual spaces, it remains confined to digital environments and lacks real-world interaction. Our robot builds upon the same principle of automated tours, but in a physical farm setting using GPS, AI, and text-to-speech. Instead of navigating a digital map, it autonomously traverses the real world, identifies objects with a camera, and speaks to human visitors—merging robotics with experiential learning in a tangible way.

Ivanov et al. (2017) examine how robots and service automation are being adopted across tourism and hospitality sectors—including hotels, theme parks, and museums [13]. While robotic concierges and restaurant servers are being tested in controlled indoor environments, the authors note that outdoor robotic tour guides present significantly greater challenges. These include unpredictable terrain, weather conditions, and diverse interaction requirements. Our robot builds on their framework by doing what the paper identifies as most difficult: navigating

an outdoor space and interacting with guests autonomously. It uses GPS, AI vision, and text-to-speech to perform real-world tour guidance, without requiring pre-installed infrastructure. This makes it a novel application that extends beyond the indoor-focused service robots currently seen in commercial settings.

6. CONCLUSIONS

While the robot performs its primary function reliably, several limitations remain. The most significant issue is the dependency on a strong and consistent GPS signal, which may be obstructed by nearby buildings, tree cover, or poor satellite alignment—causing brief navigation errors. Another limitation is the use of a remote AI model for image recognition. Although this removes the need for local training and large memory, it introduces potential issues such as latency or failure in environments without internet access. Additionally, while the text-to-speech engine is generally effective, it can struggle with correct pronunciation of plant names or technical terms. Improvements would include integrating a confidence score into plant recognition, allowing the robot to confirm identifications before speaking. A secondary offline fallback system could also be developed to ensure uninterrupted performance during connectivity drops. More robust outdoor speakers and protective casing would improve performance in real-world, unpredictable farm environments.

This project demonstrates how robotics and AI can work together to create an interactive, educational experience on a farm. By combining navigation, plant detection, and voice output, the robot bridges automation and education, offering a scalable solution for tour support that enhances both visitor engagement and farm accessibility.

REFERENCES

[1] Kaghat, Fatima Zahra, et al. "A new audio augmented reality interaction and adaptation model for museum visits." Computers & Electrical Engineering 84 (2020): 106606.

[2] Li, Tsai-Yen, et al. "Automatically generating virtual guided tours." Proceedings Computer Animation 1999. IEEE, 1999.

[3] Ivanov, Stanislav Hristov, Craig Webster, and Katerina Berezina. "Adoption of robots and service automation by tourism and hospitality companies." RevistaTurismo&Desenvolvimento 27.28 (2017): 1501-1517.

[4] Duckett, Tom, et al. "Agricultural robotics: the future of robotic agriculture." arXiv preprint arXiv:1806.06762 (2018).

[5] Mulla, David J. "Twenty five years of remote sensing in precision agriculture: Key advances and remaining knowledge gaps." Biosystems engineering 114.4 (2013): 358-371.

[6] LeCun, Yann, et al. "Gradient-based learning applied to document recognition." Proceedings of the IEEE 86.11 (2002): 2278-2324.

[7] Rimmer, Matthew. "Lady Ada: Limor Fried, Adafruit industries, intellectual property and open source hardware." Journal of Intellectual Property Law and Practice 16.10 (2021): 1047-1061.

[8] Johnston, Steven J., and Simon J. Cox. "The raspberry Pi: A technology disrupter, and the enabler of dreams." Electronics 6.3 (2017): 51.

[9] Goerzen, John, Tim Bower, and Brandon Rhodes. Foundations of Python Network Programming: The comprehensive guide to building network applications with Python. Apress, 2011.

[10] Wieser, Erhard, Philipp Mittendorfer, and Gordon Cheng. "Accelerometer based robotic joint orientation estimation." 2011 11th IEEE-RAS International Conference on Humanoid Robots. IEEE, 2011.

[11] Ravindra,PathareShailendra. "ADVANCING NATURALNESS AND INTELLIGIBILITY IN TEXT-TOSPEECH SYSTEMS USING ARTIFICIAL INTELLIGENCE." (2025).

[12] Tripathi, Padmesh, et al. "Applications of deep learning in agriculture." Artificial intelligence applications in agriculture and food quality improvement. IGI Global Scientific Publishing, 2022. 17-28.

[13] Doran, Derek, Kevin Morillo, and Swapna S. Gokhale. "A comparison of web robot and human requests." Proceedings of the 2013 IEEE/ACM international conference on advances in social networks analysis and mining. 2013.

[14] Ajami, Sima. "Use of speech-to-text technology for documentation by healthcare providers." The National medical journal of India 29.3 (2016): 148.

[15] Enocksson, Staffan. "Modeling in MathWorksSimscapeby building a model of an automatic gearbox."(2011).