A Low-cost Autonomous Lawn Mower with AI-Based Obstacle Avoidance and GPS Guidance System

Thanapon Kosri[1], Tossawat Seekhamharn[1], Phasawut Phoonsrichaiyasit[1], Poowadon Khungpo[1], Phaophak Sirisuk[2] and Theerayod Wiangtong[1, *]
[1] Electrical Engineering Department, School of Engineering, King Mongkut's Institute of Technology Ladkrabang, Lat Krabang, Lat Krabang, Bangkok, 10520, Thailand
[2] Industrial Engineering Department, School of Engineering, King Mongkut's Institute of Technology Ladkrabang, Lat Krabang, Lat Krabang, Bangkok, 10520, Thailand
***Corresponding Author E-mail:** theerayod.wi@kmitl.ac.th
Received: Apr 21, 2025; **Revised:** Jun 23, 2025; **Accepted:** Jul 01, 2025

Abstract

This paper presents a cost-effective robotic system capable of manual control via RF remote and autonomous navigation using GPS-based information. The system employs artificial intelligence to dynamically classify and avoid non-grass obstacles, ensuring safe operation in real environments. The prototype integrates affordable hardware including Arduino board, sensors, actuators and Raspberry Pi with lightweight algorithms to balance performance and cost. Experimental validation confirms its ability to follow predefined paths with ±1.5 meters deviation in open area and 90% obstacle avoidance success rate. With a total hardware cost under $200, this prototype highlights feasibility for larger-scale implementation.

Keywords: Autonomous, Lawn mower, Image processing, YOLO, GPS navigation, Low-cost

1. Introduction

The rising of autonomous technologies has revolutionized various industries, from automotive to agriculture, by enhancing efficiency, reducing human labor, and more intelligence. Among technological improvements, autonomous lawn mowers have emerged as a promising innovation in real fields. While traditional lawn mowing methods require significant manual effort and time, autonomous solutions can offer convenience, consistency, and environmental benefits [1].

The motivation for autonomous lawn mowers is from the growing demand for smart home technologies. Using machines makes it easier to maintain green spaces such as lawns, gardens, and orchards. Autonomous lawn mowers can play a crucial role in maintaining these spaces efficiently, reducing the need for humans. Moreover, the development of autonomous lawn mowers presents a unique set of challenges and opportunities in the fields of robotics, artificial intelligence, and sensor technology. Key research areas include navigation systems that can handle complex and dynamic environments, obstacle detection and avoidance mechanisms [2–4].

Autonomous lawn mowers have the potential to enhance safety by reducing the risk of accidents associated with manual mowing [5]. They can also contribute to environmental sustainability by utilizing renewable energy sources, such as solar power [6]. Recently, the integration of machine learning algorithms can enable these devices to adapt to varying lawn conditions and user preferences, further enhancing their utility and performance.

The aim of this research is to develop a prototype capable of both manual and autonomous operation. Sensors such as GPS and ultrasonic devices are integrated with machine learning algorithms to enhance system ability. In autonomous mode, the system follows predefined waypoints using location data from GPS sensors and electronic compass. Ultrasonic sensors are used to measure the distance to objects along the path. Additionally, cameras equipped with AI algorithms are used to classify objects as either high grass or non-grass, enabling movement decisions. To develop a cost-effective autonomous version, off-the-shelf controller and processor boards, along with low-cost sensors are utilized. This approach has the potential to inspire further innovations in the broader field of specialized autonomous robotics, such as those used in agriculture.

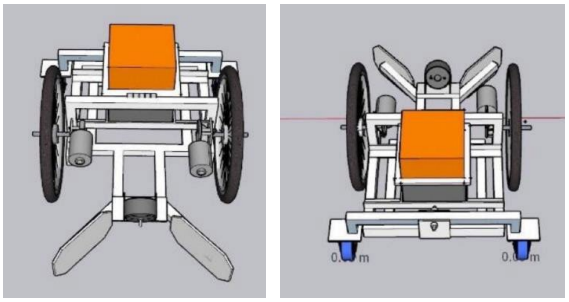
The next section is to explain the design methodology of both hardware and software, followed by experiments and results with discussions. The conclusion is presented in the last section.

2. Design Methodology

Traditional lawnmowers require labor intensity, environmental pollution, and safety risks. This work addresses these issues by proposing an electric, low-cost robotic lawnmower with dual-mode control, including RF remote for manual operation and GPS-guided for autonomous mode that follows predefine waypoints. Additionally in autonomous mode, it can properly make decisions using camera-based detection to classify and avoid non-grass obstacles.

2.1 Hardware Development

This prototype lawn mower is designed using a steel frame as sketched in **Figure 1**, while the real one is shown in **Figure 2**. Its mobility is from two 12-inch front wheels powered by 12V DC motors (55 RPM), while 5-inch rear caster wheel aids gliding. The grass cutting mechanism is driven by a high-speed 12V brushless DC motor (3910 RPM). For navigation, it employs a GPS module (GY-NEO6M) to follow a predefined path, with an electronic compass (GY-271 QMC5883) assisting in maintaining correct heading and compensating for GPS drift. Ultrasonic sensors mounted around the mower detect obstacles within a 2-meter range, enabling it to avoid collisions and safely navigate through the mowing area.



(**a**) (**b**)

Figure 1 The designed lawn mower
(**a**) Front view (**b**)Rear View

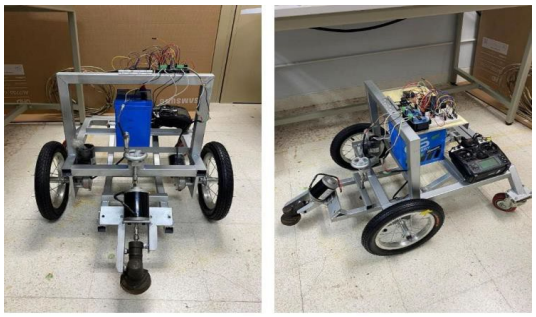


Figure 2 The real prototype

2.1.1 Motor Selection

The desired speed is approximately the same as a walking person, around 0.8 m/s. Wheel size is 12 inches (30.48 cm) diameter. The vehicle weight is approximately 20 kilograms. From the formula $V = \omega r$, by substitution $0.8 = \omega(\frac{30.48}{2\times100})$, then we can get $\omega = 5.25 rad/s$

To calculate the power of a motor given torque, wheel radius, and angular velocity (omega), we can use Eq. (1):

$$P = \tau \cdot \omega \quad (1)$$

Where:

- P is the power (in watts, W),
- τ is the torque (in Newton-meters, N·m),
- ω is the angular velocity (in radians per second, rad/s).

Driven by 2 wheels, so each motor is responsible for 10 kilograms (a half of its weight) or $10 \times 9.8 = 98$ N. As the torque which is calculated by $\tau = Fr$, so $\tau = 98 \times (\frac{30.48}{2\times100}) = 14.94$ N-m the power of the motor in this design is then at least $P = 14.94 \times 5.25 = 78.4W$. This design, however, selects the 120W motor that can provide 20N-m torque, accounting for friction and system losses.

The cutting speed referenced from the MAKITA ELM3720 GL300 lawn mower is 3400 RPM. Therefore, a 12V brushless DC motor with 3910 RPM 150W is used in this prototype.

2.1.2 Battery

The drive motor draws a maximum current of 10A for 12VDC battery. The cutting motor draws a maximum current of 15A. The total current for all three motors is around 35A. Hence, a 12 VDC 40Ah battery is selected to allow the lawnmower to operate for about 67.5 minutes (when running all three motors at full load). However, the actual current drawn during operation depends on the load, surface roughness, and grass thickness. In practice, the current is typically much lower, as seen in experiments where the current during no-load conditions is only around 1A, allowing for longer operation than initially expected.

2.1.3 Controller and Sensors

In this project, the Arduino Mega was employed because there are enough number of pins compared to Arduino UNO. The main 12V battery and a DC buck converter can be used to supply all components in this system, containing an emergency stop button for safety, various types of sensors and peripherals as shown in **Figure 3**.

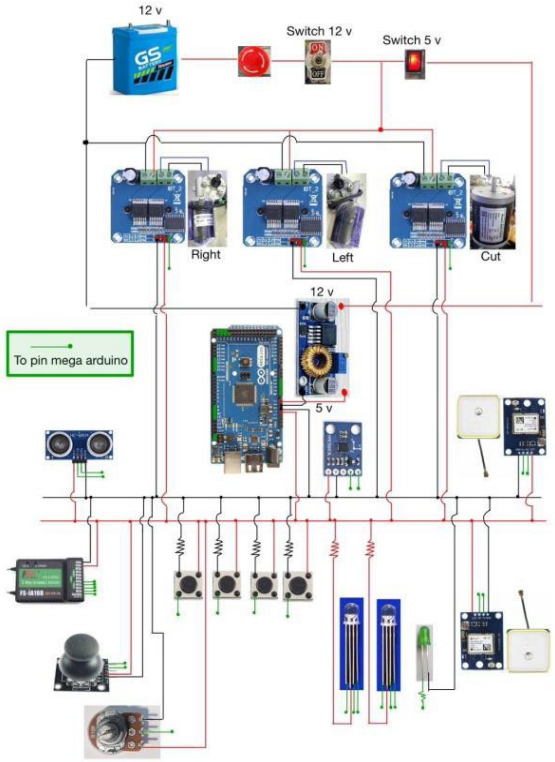


Figure 3 Power connectivity diagram

The guidance system in this project is generally based on coordination from GPS, and direction from electronic compass. The average coordinating value from two GY-NEO6M GPS modules is used to determine the destination point, while the GY-271 QMC5883 three-axis magnetic compass module is used to control moving directions.

The prototype involves calibration and adjustment of coordinating systems using NMEA from the GPS and information from the compass. NMEA is a standard data format supported by all GPS manufacturers. It can output data in multiple formats. The transmitted data from the GPS sensors used in this project is as following example [7].

```
$GPRMC,225446,A,4916.45,N,12311.12,W,
000.5,054.7,190125,020.3,E*68
```

It is prefixed with $GPRMC and other identifiers such as UTC time (22:54:46), Navigation receiver status (A=OK, V=Warning), Latitude (49 deg. 16.45 min North), Longitude (123 deg. 11.12 min West), Speed over ground (000.5Knot), Track angle (054.7 deg.), Date (19/01/2025), Magnetic variation (20.3 deg. East), and checksum (68).

GY-271 electronic compass module is designed for low-field magnetic sensing. It converts any magnetic field to a differential voltage output on 3 axes through I2C interface in degrees referring to the magnetic North. This voltage shift is the raw digital output value, which can then be used to calculate headings or sense magnetic fields coming from different directions.

Since both the GPS and compass modules use different references, the GPS references true North (geographical North), whereas the compass references magnetic North. The difference today is about 500 kilometers apart. Additionally, the magnetic North is recently moving around 25 kilometers a year towards Siberia. This phenomenon is known as the Polar Shift Theory [8]. For controlling the direction of movement to each waypoint, this consideration is included in our path calculation to correct direction errors

2.1.4 Controller Program Coding

To maneuver the mower, the program flowchart is shown in **Figure 4**. The radio-controlled operation mode is initiated by default hence the system can be controlled manually via the remote control. To set the path, waypoints can be input manually using the remote-control stick, e.g. moving left to set a waypoint or right to reset the waypoint. Multiple waypoints can then be added into the system. After completion, autonomous mode can then proceed.

The autonomous operation is activated by pressing the start button, the GPS will compare the distance to the first waypoint. If the distance is not zero, the angle heading toward the destination will be periodically estimated using information from the current GPS location and the destination waypoint. While moving, the direction from the electronic compass is measured and compared with the estimated angle. If the measured angle differs from the estimated angle, the direction of movement will be adjusted until it is within some specific value. The mower will then continue moving for a few seconds and recheck the direction again till the distance is in the acceptable boundary. To complete the path, the process will repeat this moving process to other waypoints sequentially.

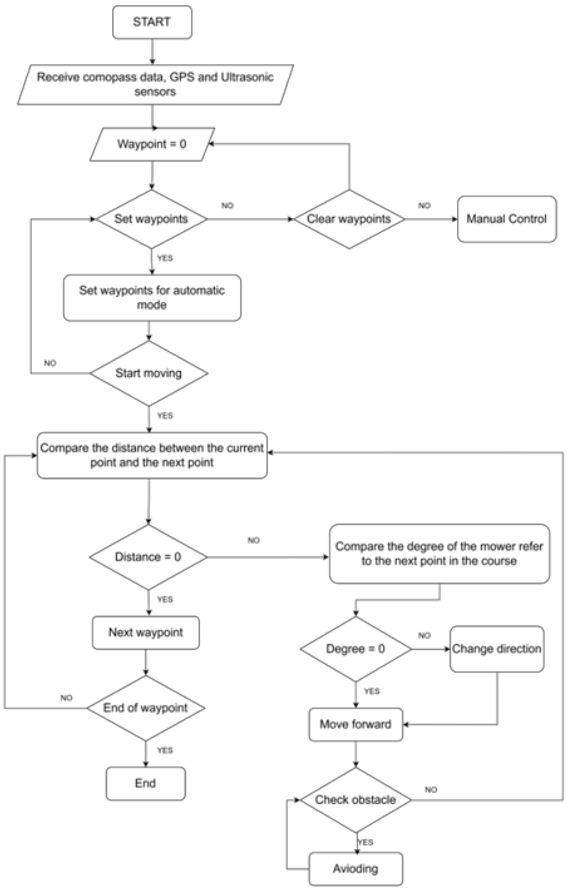


Figure 4 The mower control flowchart

2.2 Software Development: AI for Obstacle Avoidance

The software development includes an object detection module that identifies obstacles in real time, along with a depth estimation component that determines the distance of detected objects. These two modules work together to enable effective obstacle avoidance, preventing collisions and minimizing potential damage to the autonomous lawn mower. As a result, the mower can be operated autonomously through a GPS-based navigation system incorporated with AI-based obstacle avoidance system explained in this section.

Raspberry Pi 4B, a compact platform equipped with cameras, is adopted to implement AI-based algorithms for object classification and depth estimation. To avoid non-grass, the lightweight YOLO (You Only Look Once) classification algorithm is adopted in the Raspberry Pi. Stereo/monocular cameras are used for depth estimation that needs as

vital information to alter the course and avoid an object. Stereo cameras generated disparity maps using OpenCV's StereoSGBM algorithm [9], while monocular cameras employed bounding box scaling and zone-based depth calculation.

2.2.1 Object Detection and Classification

Currently, YOLO models have gained widespread popularity and have undergone continuous development [10], resulting in a total of eight versions. In this research, YOLOV5 and YOLOV8 were compared for properly implementation. YOLOV5 is widely used due to its stability and extensive adoption in various applications, while YOLOV8 represents the latest iteration of the YOLO model, offering improved performance and advanced features.

For custom object training, both online and offline platforms were utilized. Google Colab was chosen as the online platform for model training due to its cloud-based computational power, while Visual Studio Code was used for offline development and deployment. The trained models were initially built and optimized in Google Colab before being exported for further integration and testing in Visual Studio Code.

We first take pictures from real sites and identify objects normally found in fields. In this case, there are six categories include trees, humans, rocks, trash, poles, and branches from trees. The process of labeling images for object detection and enhancing them through the Roboflow platform as shown in **Figure 5**. After running through all processes, the outcome is the YOLO model for the next use to detect objects. YOLOv8 seems to be more promising in terms of accuracy, processing speed, and model size, more results are shown in the next section.

When a non-grass object is detected, the location of the object also with the depth are required for moving plan. The object's frame will be used to determine the center point and angle to identify the object's position in the image. The location and distance between the object and the mower are periodically updated and used as conditions to make avoidance.

Figure 5 Labelling objects possibly in fields

2.2.2 Depth Estimation

Depth estimation refers to the process of determining the distance of objects within an image from a camera. It is a crucial technique in the fields of computer vision and 3D perception. Depth information plays a significant role in various applications, such as robotics, autonomous vehicles, and other related technologies [11]. Several methods can be employed to perform depth estimation, including the use of disparity maps, monocular depth estimation, and LiDAR technology. This paper will focus on the first two methods using USB cameras (QCAM-M400) and software for analysis, instead of using the high-cost LiDAR sensor.

Disparity Map: it is a technique used to estimate depth by utilizing two cameras, commonly referred to as a stereo camera system. The principle behind stereo vision simulates the human visual system by creating two images from the left and right "eyes" (cameras). These two images are then analyzed to compute the disparity, which refers to the difference in position between the same corresponding points in the two images, allowing for depth estimation of the scene.

As shown in **Figure 6**, O and O' are left and right unique cameras with the focal length of f. B is the baseline distance between these two cameras. Z is a real distance between the camera plane and the object at point X. From trigonometric ratios, the disparity [12] can be calculated as follows Eq. (2):

$$disparity = x - \acute{x} = \frac{B_f}{Z} \tag{2}$$

where x and $\acute{x}$ are the distance between points in image plane corresponding to the scene point and their camera center. It simply says that the depth of a point or object Z in a scene is inversely proportional to the difference in distance of x - x'which is called disparity.

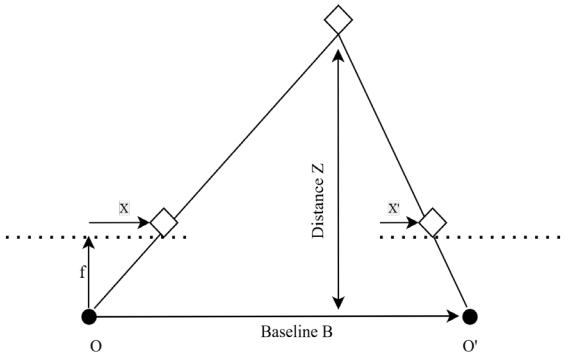


Figure 6 Disparity map diagram

Typically, a disparity map is visualized using a color scheme that combines red and blue. By coding in Python language to run OpenCV function StereoBM, in this representation, the red color indicates the highest disparity values, corresponding to objects that are closer to the camera, while the blue color represents the lowest disparity values, indicating objects that are further away.

Our preliminary results from our two USB cameras, as shown in **Figures 7–8**, can prove that the system could detect objects with high accuracy and estimate distances with relatively high precision, with only a slight margin of error.

Figure 7 The result of color scheme disparity map

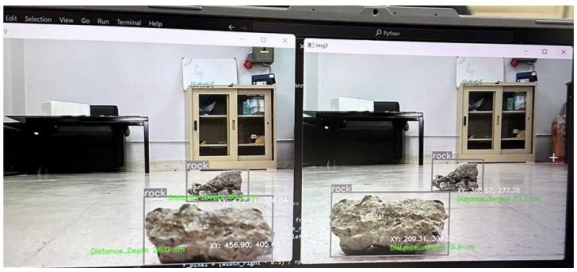


Figure 8 The boundary boxes of detected objects (rocks) and distances calculated from disparity

- *Monocular Depth Estimation*: This technique can estimate depth using a single camera to approximate the depth of objects [13]. In this paper, to predict the depth value from image input, the size of the boundary box refers to the real size of the known detected object and that information can be used to estimate the distance from some experiments. For example, the smaller size of the car in pixels refers to the longer distance from the camera. For the various sizes of the known objects such as rocks we might use the average size. The errors can be compensated by using the information from the camera updating the position while moving away from obstacles. This method allows for depth approximation with a single camera, making it applicable for low computational performance platform.

From our investigation, both depth estimation methods provide nearly the same level of accuracy in determining depth, but each has its advantages. Using a stereo camera, depth values are obtained through mathematical calculations based on the detected object and its center point. However, this method consumes more processing time. On the other hand, monocular depth estimation allows for adjustments based on the actual size of objects. This enables quicker depth estimation. Additionally, it requires a single camera, making it more practical in this prototype.

2.2.3 Obstacle Avoidance

To avoid obstacles or classified objects on the path within some distance getting from depth estimation, the algorithm in this paper is to divide an input image into 3 zones, including left zone, right zone and middle zone as shown in **Figure 9**. As the flow in **Figure 10,** the mower will perform an evasive turn for one second at a time. It will steer toward the wider side based on the object's position in the image. The system continuously checks input images. If the object persists, it will keep moving and repeat this process until no further detection (indicating that the obstacle has been successfully avoided).

Once the path is cleared, the system will execute the next function, returning to its original path. It will re-align with the predefined route using GPS to ensure proper navigation.

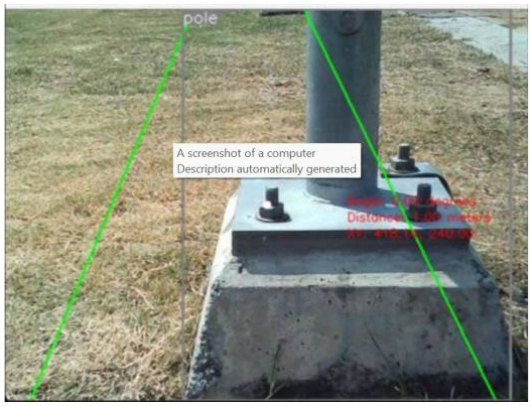


Figure 9 Zoning a picture area into left/middle/right using the green lines

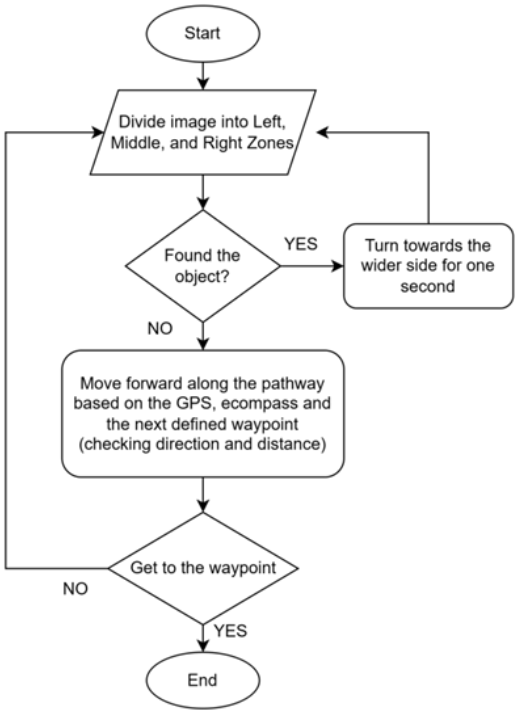


Figure 10 The avoidance flowchart

3. Experimental Results

3.1 AI and Video processing

From the results of training and testing the YOLO model in both versions, it was found that YOLOv8 provides higher accuracy and better performance compared to YOLOv5 when tested using the same source and test conditions.

From **Figure 11**, the graphs for *train/box_loss, train/clc_loss, val/box_loss* and *val/clc_loss* show a downward trend in the Y-axis, indicating a reduction in detection errors as training progresses. When compared to the X-axis (number of epochs), it is observed that after 50 epochs, the *loss/error* values become significantly low and slightly decrease.

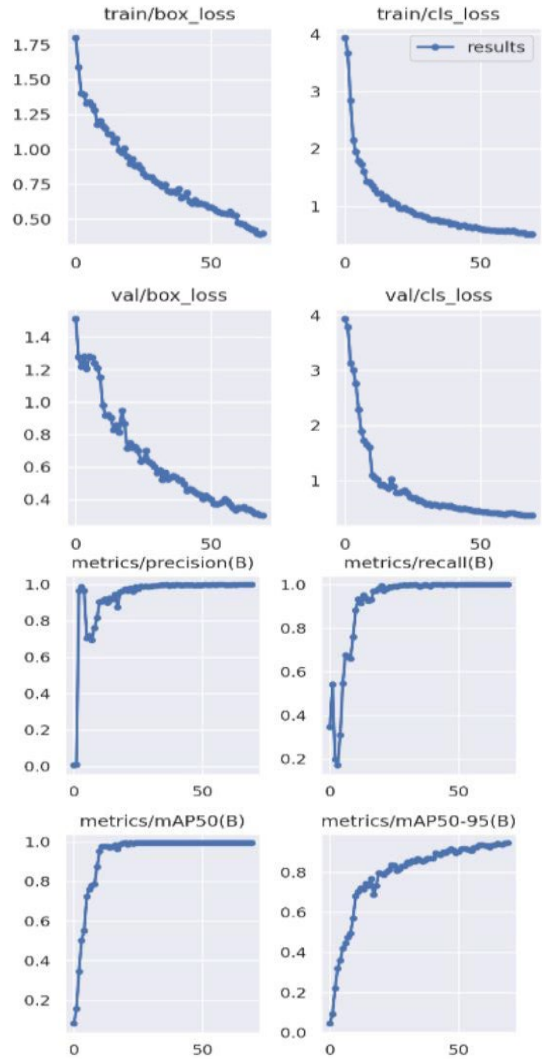


Figure 11 Training results from YOLOv8

Similarly, the graphs for *metrics/precision(B), metrics/recall(B), metrics/mAP50(B)* and *metrics/mAP50-95(B)* show an upward trend, indicating increased object detection accuracy. After 50 epochs, the accuracy values become relatively high and show minimal further improvement. This is the reason for selecting epoch = 70 for our training the model.

Figure 12 shows the confusion matrix of the trained model follows a structure larger than 2×2, with a high True Positive (TP) rate, approaching 100%, indicating strong detection accuracy for the six trained objects. The error values (TN, FP, FN) in the graph are minimal, close to zero, which demonstrates high detection performance and accuracy for objects similar to those in the trained dataset. Some results are shown in **Figure 13**.

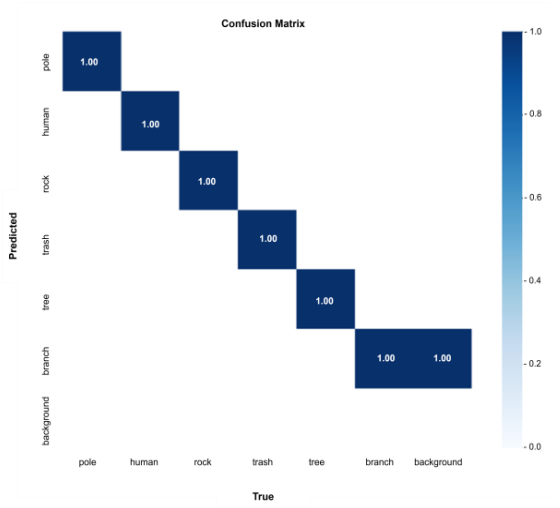


Figure 12 Confusion matrix of all test objects

Figure 13 Examples of object detection and classification from YOLO models

YOLOv8 demonstrated superior performance over YOLOv5, achieving 98% precision and 96% recall compared to YOLOv5's 92% precision and 89% recall. This improvement was attributed to Ultralytics' enhanced library support and a balanced dataset with diverse augmentation.

For the depth estimation, stereo cameras provided accurate measurements (±5 cm error) under controlled conditions but suffered from noise on grassy surfaces, while monocular depth estimation achieved ±8 cm error after calibrating scaling factors but required manual adjustments for irregular objects.

When using Raspberry Pi 4B (8GB RAM) with a single camera to detect objects on the grassy field using custom model, it was found that the processing delay for object detection with one camera was approximately 6-9 seconds for 640x480 pixels images. Raspberry Pi board experienced significant latency

and resource consumption making it unable to detect objects faster enough for practical use. To alleviate this problem, the image resolution was reduced to enable faster processing on the Raspberry Pi, improving the detection speed.

3.2 Autonomous Operation Mode

The lawnmower can be controlled to move automatically to the specified destination points. It was capable of moving to the defined waypoints and can continue operating sequentially through the defined waypoints 1, 2, 3, and so on, based on the accuracy of the electronic compass and GPS module.

3.2.1 GPS-guided system

From the code, the mower should stop within 1 meter of the destination point. Using a steel tape to measure distances as shown in **Figure 14**, although the coordination used in this system was averaged from two GPS sensors, the GPS accuracy could result in more deviations in different directions and distances in the range of 2.6 meters, as results shown in **Tables 1–2**. These experiments were conducted in the area under trees and between high buildings that might affect the accuracy. This could be improved in open areas as shown in the next results when the mower was to follow predefined waypoints.

The lawnmower operated effectively in all modes: remote control, manual joystick, and GPS-guided autonomy. It could run through rough terrain and complete a predefined pathway from A to D as shown in **Figure 15** where the distance between points was about 50 meters. Autonomous navigation exhibited positional deviations of ±1.5 meters due to signal strength and environmental interference. Calibration with the electronic compass could also reduce directional errors. The battery provided around an hour of runtime under full load (35A total current draw). Noise levels were significantly lower than combustion-engine equivalents.

3.2.2 Obstacle Avoidance in Autonomous Mode

An experiment was conducted to measure the deviation after the lawnmower turned to avoid obstacles. The deviation was measured using a camera positioned in the center of the mower to determine how far it was from its starting point after completely avoiding the object.

Figure 14 Experiment to measure deviation distance

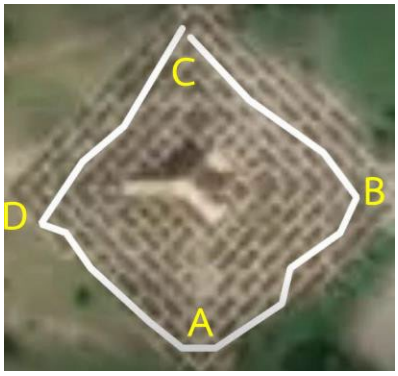


Figure 15 The comparison of real moving paths in Google Map

As shown in **Table 3**, the lawn mower trajectories diverged by 21.2 cm on concrete versus 28.6–30.8 cm on grass, attributed to motor power fluctuations and uneven terrain resistance. The controller successfully executed avoidance maneuvers though different environments. In terms of success rate, it achieved 22 times from 25 attempts or around 90%.

Table 1 GPS deviation by direction

Distance (m) from Starting Point	Direction	Average Deviation from Destination Point (m)
15	N	2.30
15	N/E	1.00
15	E	2.30
15	E/S	2.60
15	S	2.30
15	S/W	2.00
15	W	2.10
15	W/N	1.80

Table 2 GPS deviation by distance

Distance from Starting Point (m)	Average Deviation from Destination Point (m)
5	2.10
10	2.30
15	2.10
20	2.30
25	2.60
30	2.60

Table 3 Deviation when detecting obstacles

Trial	Deviation from Original Path (cm)		
	Pole (on grass)	People (on grass)	People (on concrete)
1	32.0	35.0	21.0
2	29.0	27.0	18.0
3	30.0	33.0	22.0
4	25.0	32.0	26.0
5	27.0	27.0	19.0
Average	28.6	30.81	21.2

Table 4 outlines how the proposed research compares with previous works as discussed earlier. It shows some differences in terms of the design aspects and results of this study.

Table 4 Comparison with previous mentioned works

Feature	This Work	Previous Works
Navigation System	GPS (NEO-6M) and Digital Compass (QMC5883)	Vision based [2], GPS-based [3],[4]
Control Modes	Manual (RF) and Autonomous	Autonomous or manual-only operation
Obstacle Avoidance	YOLOv8 and Depth Estimation	Machine vision [2], LIDAR [3], Sonar [4]
AI Algorithm	YOLOv8 is adopted after comparison with YOLOv5	Basic ML techniques (e.g., CNN, AI) used in navigation [2]
Depth Estimation	Stereo disparity map & monocular estimation	Depth info came from the RGB-D sensor [3]
Total Hardware Cost	< $200 USD	Unspecified in most, typically higher due to advanced sensors such as LIDAR
GPS Accuracy	±1.5 meters in open fields	Deviation commonly 2–3 m or more in semi-obstructed areas
Obstacle Avoidance Success Rate	90% (22 out of 25 trials)	92.7% for LIDAR-based [3]
Power source	Battery	Battery, Solar panel [6]
Processing Hardware	Raspberry Pi 4B (8GB)	PC or generic microcontroller

4. Conclusions

This paper demonstrates the successful implementation of a low-cost, dual-mode (manual RF and autonomous GPS-guided) system with object- avoidance capabilities. The prototype achieves reliable navigation and obstacle avoidance in controlled environments. Experimental validation confirms its ability to follow predefined paths and 90% obstacle avoidance success rate. The integration of YOLOv8 and monocular depth estimation successfully demonstrated obstacle avoidance, though path recovery mechanisms struggled on grass due to inconsistent traction. With a total hardware cost under $200, this work provides a foundational framework for larger-scale applications.

Recommendations for future work include adopting optimized YOLO architecture (e.g., Tiny-YOLO) for Raspberry Pi devices, upgrading to NVIDIA Jetson for enhanced computational capacity, and using inertial sensor or LiDAR for robust 3D mapping. Real-time kinematic positioning (RTK) could be used to enhance position accuracy with more complex maneuvering control methods for different layouts that could broaden its applicability in agricultural activities or field operations.

5. References

[1] R. P. Kizhakkeyil and N. Patel, "Autonomous Lawn Mower – A Comprehensive Review," *International Research Journal on Advanced Science Hub*, vol. 5, no. 12, pp. 420–428, 2023, doi: 10.47392/IRJASH.2023.079.

[2] K. Inoue, Y. Kaizu, S. Igarashi, K. Furuhashi and K. Imou, "Autonomous Navigation and Obstacle Avoidance in an Orchard Using Machine Vision Techniques for a Robotic Mower," *Engineering in Agriculture, Environment and Food*, vol. 15, no. 4, pp. 87–99, 2022, doi: 10.37221/eaef.15.4_87.

[3] P. Xie, H. Wang, Y. Huang, Q. Gao, Z. Bai, L. Zhang and Y. Ye, "LiDAR-Based Negative Obstacle Detection for Unmanned Ground Vehicles in Orchards," *sensors*, vol. 24, no. 24, 2024, Art. no. 7929, doi: 10.3390/s24247929.

[4] D. R. D. Wijewickrama, K. M. H. Karunanayaka, H. W. P. Senadheera and T. M. Godamulla, "Fabrication of an Autonomous Lawn Mower Prototype with Path Planning and Obstacle Avoiding Capabilities," *CINEC Academic Journal*, vol. 2, pp. 30–33, 2017, doi: 10.4038/caj.v2i0.51.

[5] J. C. Mayoral Baños, P. J. From and G. Cielniak, "Towards Safe Robotic Agricultural Applications: Safe Navigation System Design for a Robotic Grass-Mowing Application through the Risk Management Method," *robotics*, vol. 12, no. 3, 2023, Art. no. 63, doi: 10.3390/robotics12030063.

[6] T. Tahir, A. Khalid, J. Arshad, A. Haider, I. Rasheed, A. Rehman, S. Hussen, "Implementation of an IoT-Based Solar-Powered Smart Lawn Mower", *Wireless Communications and Mobile Computing*, vol. 2022, pp. 1–12, 2022, doi: 10.1155/2022/1971902

[7] *GPS - NMEA sentence information*, Glenn Baddeley, Jul. 20, 2001. [Online]. Available: https://aprs.gids.nl/nmea/

[8] P. W. Livermore, C. C. Finlay and M. Bayliff, "Recent north magnetic pole acceleration towards Siberia caused by flux lobe elongation," *nature geoscience*, vol. 13, pp. 387–391, 2020, doi: 10.1038/s41561-020-0570-9.

[9] H. Hirschmüller, "Stereo Processing by Semiglobal Matching and Mutual Information," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 30, no. 2, pp. 328–341, 2008, doi: 10.1109/TPAMI.2007.1166.

[10] J. Terven, D. -M Córdova-Esparza and J. -A. Romero-González, "A Comprehensive Review of YOLO Architectures in Computer Vision: From

YOLOv1 to YOLOv8 and YOLO-NAS," *machine learning & knowledge extraction*, vol. 5, no. 4, pp. 1680–1716, 2023, doi: 10.3390/make5040083.

[11] Y. Hu, W. Zhen and S. Scherer, "Deep-Learning Assisted High-Resolution Binocular Stereo Depth Reconstruction," in *2020 IEEE International Conference on Robotics and Automation (ICRA)*, Paris, France, May 2020, pp. 8637–8643, doi: 10.1109/ICRA40945.2020.9196655.

[12] *Depth map from stereo images*, OpenCV, Accessed: Mar 20, 2025. [Online]. Available: https://docs.opencv.org/3.4/dd/d53/tutorial_py_depthmap.html.

[13] C. Zhao, Q. Sun, C. Zhang, Y. Tang and F. Qian, "Monocular Depth Estimation Based On Deep Learning: An Overview," *Science China Technological Sciences*, vol. 63, no. 9, pp. 1612–1627, doi: 10.1007/s11431-020-1582-8.