# Development Of a Waypoint Navigation System For Rvm Robots Using Path Planning Algorithms

Selamat Muslimin[1], Ekawati Prihatini[2], Nyayu Latifah Husni[3], Satria Pebrian[4]

Sarjana Terapan Teknik Elektro, Jurusan Teknik Elektro – Politeknik Negeri Sriwijaya
selamet_muslimin@polsri.ac.id

**Abstract.** This research aims to develop and test a RVM navigation system using waypoint-based path planning method. In this study, YOLOv7 is used to detect the robot with the help of a webcam mounted on the ceiling of the room. The robot is equipped with red and green dots on it, where the red dot signifies the front and the green dot the back, thus allowing determination of the robot's orientation. The system allows the user to specify the waypoint that the robot should reach. The motor movement data is sent to Firebase once the coordinates of the robot and the waypoint destination are determined. NodeMCU then retrieves the data and sends it to Arduino which controls the motor driver to move the robot towards the waypoint. The test results show that the navigation system successfully directs the robot to the waypoint appropriately. The conclusion of this research shows that the designed navigation system is successful and stable in directing the robot to the predetermined waypoint.

**Keywords:** Robot RVM; Navigationi Waypoint; Image Processing; YOLOv7

## 1 Introduction

Modern technological advances have provided innovative solutions to make various daily needs easier. Nowadays, busy lifestyles make time a valuable asset. Individuals often face challenges in balancing work, outdoor activities, and rest time (Perez & Fuentes, 2022). One of the increasingly important needs is easy and fast access to electronic device charging. Robot vending machines as charging stations can be an efficient solution, improving user comfort in public locations such as shopping malls, airports, campuses, and offices (Zhao, Chen, & Li, 2021; Tan & Qiu, 2023).

The importance of this research lies in the need for intelligent and efficient autonomous technology. In this case, the charging robot vending machine not only simplifies charging access, but also opens up opportunities for the development of intelligent navigation and sensing systems that are able to adapt to the surrounding environment (Huang, Zhao, & Lu, 2019). With sensing methods such as cameras or sensors as well as effective path planning, robots can move autonomously, meeting user needs while supporting adaptive and interactive smart city concepts (Kumar & Bhattacharya, 2020).

In addition, this charging vending machine robot is a model for the implementation of automation technology in daily life. With the integration of YOLO (You Only Look


N. L. Husni et al. (eds.), *Proceedings of the 8th FIRST 2024 International Conference on Global Innovations (FIRST-ESCSI 2024)*, Advances in Engineering Research 261,
https://doi.org/10.2991/978-94-6463-678-9_41

Once) and PID (Proportional, Integral, Derivative) visual identification methods for navigation stability, the robot can accurately recognize the environment and move efficiently according to the specified path (Tan & Qiu, 2023; Perez & Fuentes, 2022). This study is expected to be able to contribute to the development of more applicable robotics technology, improving the quality of life of the community by offering fast, accessible, and smart charging solutions.

This research offers advantages in the further development of way point navigation in robots by applying path planning algorithms. With this algorithm, the robot can determine the optimal path to each intended charging point, ensuring efficiency in moving, and adaptability to various environmental conditions. This approach is expected to be able to make an important contribution to robotics technology innovation, especially in faster, more accessible, and intelligent charging efficiency in various situations.

## 2 Waypoint-Based Path Planning Method

Waypoint-based Path Planning is a path planning method used to determine the route that robots must follow to reach a specific goal. In this method, a series of points or "waypoints" are set along the desired path. The robot then moves from one waypoint to the next until it reaches its final destination. The process begins by determining the coordinates of each point by considering the environment that the robot will travel through. Users can set these waypoints manually or create their own. The robot uses sensors and control algorithms such as PID (Proportional-Integral-Derivative) to set the route from one waypoint to the next with precision.

Each waypoint serves as a reference point that guides the robot in adjusting its direction and speed. The robot continuously monitors its position relative to the waypoint it is heading to and adjusts stay on the right track. Sensors, such as cameras, can detect the robot's position and ensure that the robot stays on the desired path.

The advantage of this method is its ability to provide clear and structured routes, which makes it easy to plan and control the robot's trip. In addition, the method is flexible and can be adapted for a variety of applications, such as navigation in both indoor and outdoor environments. However, this method requires initial mapping of the environment and precise waypoint determination to ensure the efficiency of navigation.

## 3 Hardware

In this process, there are various stages ranging from understanding system needs, component selection, electronic circuits and understanding of size and efficiency. In hardware design, there is an important step to create a design that is produced according to

functional needs. This process has stages that involve aspects of needs and safety. Hardware design involves the desired size, materials and types of materials, as well as the layout of components such as sensors, motors, and wheels that will be installed on the robot later. At this stage, size and accuracy are also the main concerns so that later it does not interfere with the robot's maneuvering or navigation process.

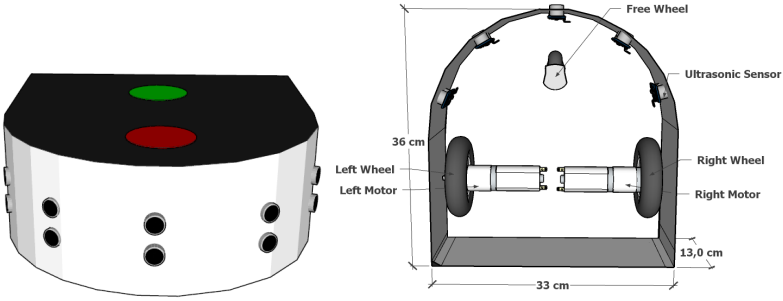



Fig. 1. Robot Design

## 4 Electronic Devices

Electronic design is the process of designing a system that involves components to create a process that will be used in robots. This process begins by selecting the right components, component types, component specifications and then combining the components into a series. Here is the schematic of the circuit on RVM

The RVM navigation uses path planning using an arduino mega 2560 as a microcontroller, an ultrasonic sensor HC-SR04, Monster Motoshield VNH2SP30 connected to 2 DC motors, a webcam that will be installed on the ceiling which aims to track the robot's position coordinates at the beginning. NodeMCU as a communication series for data recipients. The ultrasonic sensor HC-SR04 functions as an obstacle detection on the robot, the Monster Motoshield VNH2SP30 as the regulator of the two motors on the robot.

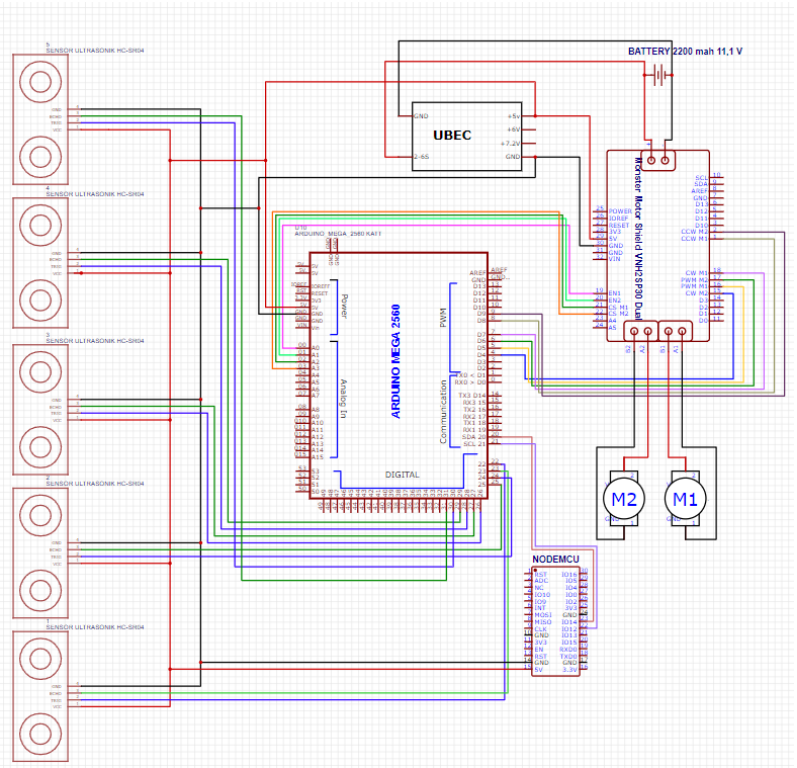



**Fig. 2**. Wiring Diagram

### 4.1.1. Block Diagram

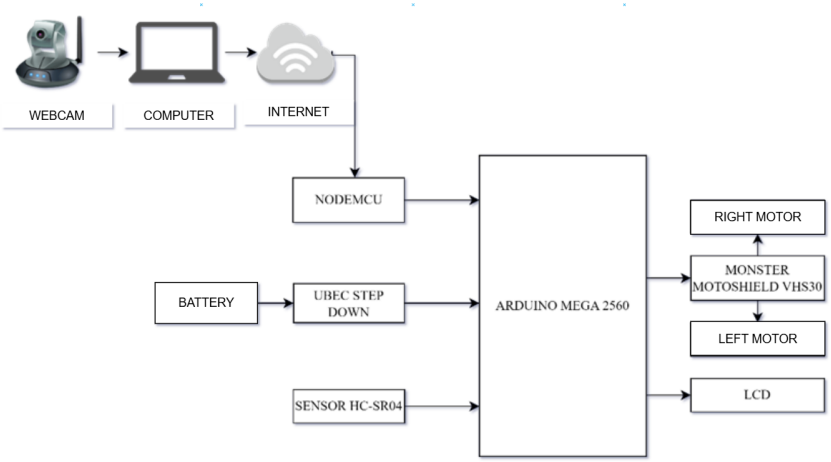



**Fig. 3.** Block Diagram

In the block diagram above, the webcam is used to detect the position of the robot in the room by sending an image to a laptop. The laptop processes the images using YOLOv7 to determine the robot's coordinates and orientation, and then sends the waypoint data to Firebase over the internet. NodeMCU, a WiFi-capable microcontroller module, takes data from Firebase and sends it to the Arduino Mega 2560. The Arduino Mega 2560 serves as a control center, receiving data from the NodeMCU to drive the motor according to the specified waypoint. This system is also equipped with UBEC step down which lowers the voltage from the battery so that it is safe to use by other components. The battery provides power for the entire system, including the motor, Arduino Mega 2560, and other components. The ultrasonic sensor HC-SR04 is used to detect objects around the robot and avoid collisions by transmitting distance data to the Arduino Mega 2560. The Monster MotoShield motor driver VNH2SP30 controls the robot's left and right motors based on signals from the Arduino Mega 2560 to regulate the speed and direction of the motor.

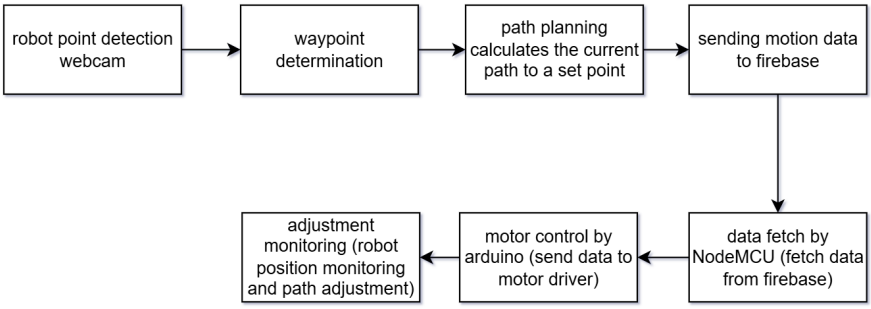



**Fig. 4.** Block diagram of the entire process of navigation

Based on the block diagram above, the webcam detects the dots on the robot to determine its position and orientation. After that, the user specifies the desired waypoint or destination for the robot. The path planning process then calculates the best route from the robot's current position to the waypoint. The motion data generated from this path planning is sent to Firebase. The NodeMCU then takes the motion data from Firebase and sends it to the Arduino. The Arduino controls the motor by sending signals to the motor driver to move the robot according to the planned route. During the trip, the system continuously monitors the robot's position and adjusts the path if necessary, ensuring the robot stays on the correct path to the waypoint.

## 5. Image Processing Design with YOLOv7

Today, there are many methods developed in computer vision. This development results in more accurate results and faster processing, bringing it closer to real-time detection results. YOLOv7, also known as You Only Look Once version 7, is an object detection model that is a continuation of the YOLO family of algorithms, which is very

famous for its speed and efficiency in finding objects in real-time. Here are the stages of image processing using YOLOv7.

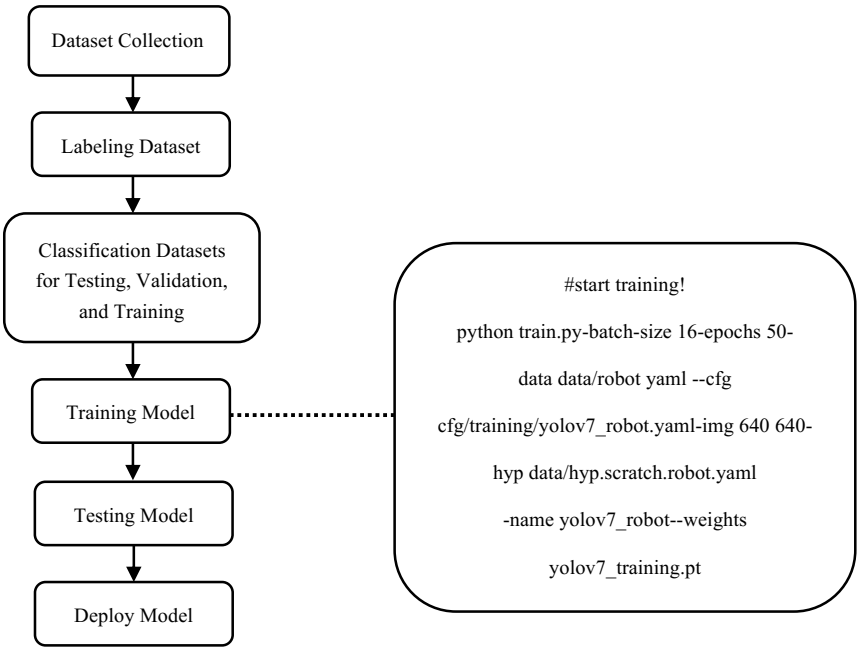



**Fig. 5.** Flow diagram of the stages of image processing

### 5.1.1. Dataset Collection

In the development of machine learning models, including object detection models such as YOLOv7, dataset collection or data acquisition is an important stage. It starts with defining the project objectives and project scope, i.e. the types of objects to be detected and the context in which the model is used. Data is collected from a variety of sources, such as photos taken from cameras, images found on the internet, or publicly available datasets. To support effective training, data must be representative, varied, and in sufficient quantities, because data quality is critical. In this study, the dataset used is in the form of a photo of the top of the robot with various conditions, directions, and different backgrounds. The process of collecting this dataset uses a camera from *a mobile phone* with an aspect ratio of 3:4. The photo dataset in this study totals 326 photos.

### 5.1.2. Labeling Dataset

The process of annotating or labeling a dataset that will be used to train a machine learning model is called dataset labeling. Labeling is used in object detection to mark each object in an image with a bounding box and the corresponding label class. This process is crucial because machine learning models need properly labeled data to learn

to recognize relevant patterns and features. Using annotation tools like LabelImg, human annotators can do the labeling manually. The quality of the labeling greatly affects the performance of the model, as accurate and consistent annotations ensure that the model can learn correctly and generate reliable predictions. To make the model more resistant to diverse real-world circumstances, labeling should also include a wide range of conditions and variations of objects. In the dataset research, 3 classes of labels were given, namely robot, green, red.

#### 5.1.3. Dataset Classification for Testing, Validation, and Training

One of the important steps in the process of developing a machine learning model is the division of the dataset into three parts, namely training, validation, and testing. To avoid overfitting and ensure that the trained model can be well generalized on data that has never been seen before, this division is the main goal. The distribution of datasets in this study is 80% for training, 20% for validation, 20% for testing. The overall dataset of 326 photos is distributed into 262 photos for training, 32 photos for validation, and 32 photos for testing.

#### 5.1.4. Testing Model

The process of testing a model involves using a never-before-seen dataset to evaluate the performance of a machine learning model. The goal of testing a dataset is to ensure that the model can perform well on a different dataset than the dataset that has been used during training. During the trial, the model is used on a test dataset to make predictions, and these predictions are compared to the original labels to calculate performance metrics such as precision, recall, and mean mean accuracy (mAP), which helps us understand how well the model can identify and find the desired pattern or object.

#### 5.1.5. Deploy Model

To make a pre-trained machine learning model usable in a real application, deploying a model involves placing the model on a server or device to receive new data and provide prediction results. This process includes setting up the model, making it possible for other applications to access the model (such as through an API), and ensuring that the model works properly in a production environment. Models are continuously monitored and updated as needed once installed. The goal is to ensure that the model can make accurate and fast predictions in everyday use.

## 6. Results and Discussion

### 6.1. YOLOv7 Training Results

In this study, model training was carried out using Google Colab software with 50 epoch experiments, then the following results were obtained.

**Table 1.** YOLOv7 Training Results

| Class | P | R | mAP@.5 | mAP.5:.95 |
|---|---|---|---|---|
| all | 0.998 | 1 | 0.995 | 0.908 |
| Robot | 0.993 | 1 | 0.995 | 0.974 |
| Red | 1 | 1 | 0.995 | 0.89 |
| Green | 1 | 1 | 0.995 | 0.861 |

Note : P = *Precision*

R = *Recall*

mAP = *Mean Average Precision*

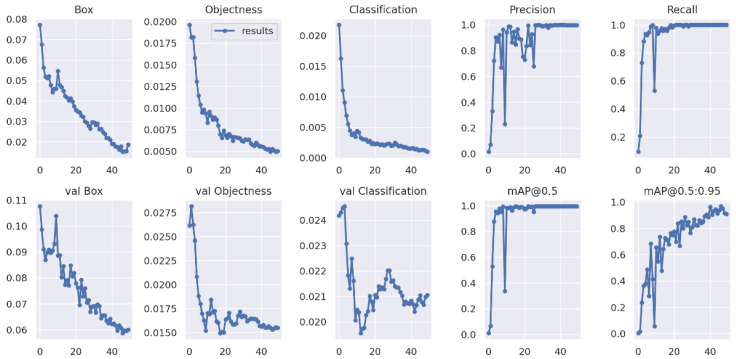



**Fig. 3**. Result metrics

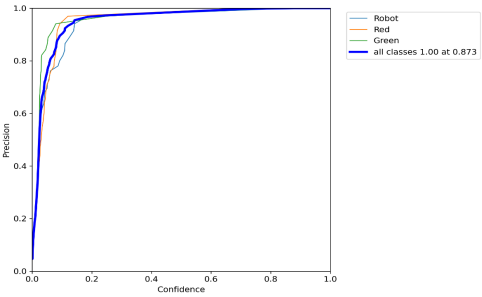



**Fig. 7.** P Curve

Based on figure 7, It shows that the YOLOv7 model is capable of providing highly accurate detection when the detection confidence level is high. The overall curve (dark blue) shows that the model performs very well across all object classes with a mAP of 1.00 at a confidence threshold of 0.873, indicating that the model is capable of detecting objects with very high precision at almost all confidence levels. Accuracy close to 1.0 for all class curves shows that the YOLOv7 model is very effective in distinguishing between different objects and giving few false positives.

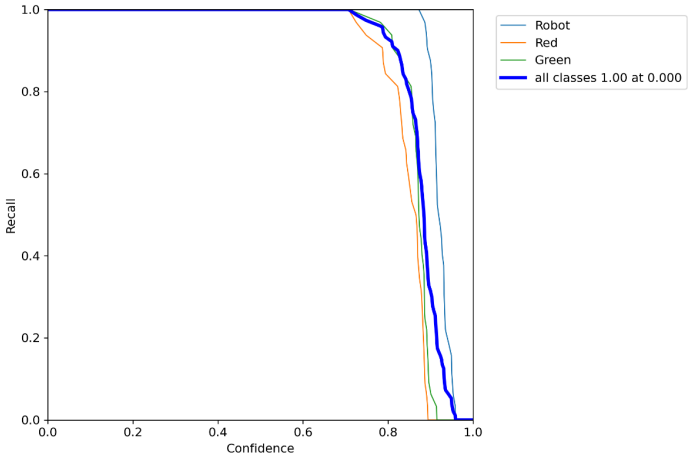



**Fig. 8.** R Curve

In Figure 8, it shows that the model has very good performance in all object classes with high recall at a confidence threshold of 0.000, indicating that this model is able to detect most objects well at low confidence levels. Overall, this curve shows that the trained YOLOv7 model has good recall performance at low confidence levels, but experiences a decrease in recall at high confidence levels.

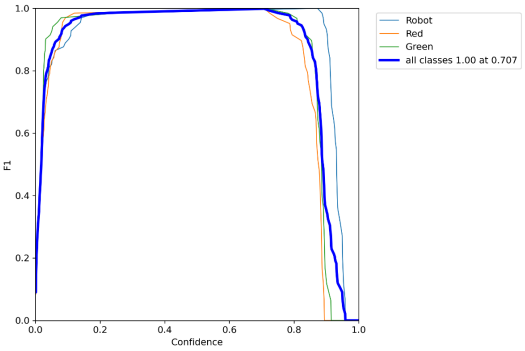



**Fig. 9.** F1 Curve

Based on Figure 9, the F1 curve indicates that the trained YOLOv7 model has excellent performance in detecting all three classes of objects (Robot, Red, and Green). The model shows a good balance between precision and recall at various confidence levels, with an F1 score close to 1.0 at a confidence level of around 0.7. This shows that the model can be relied upon in detecting and classifying these objects in navigation and path planning applications for RVM.

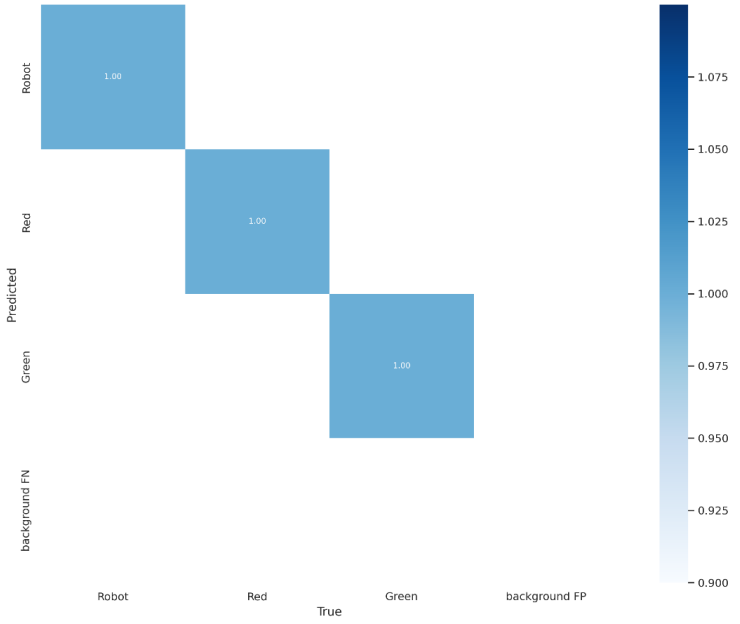



**Fig. 10.** Confusion Matrix

Based on figure 10, the model shows perfect performance in detecting and classifying red, green, and robotic dots as a whole. This can be seen from the value of 1.00 on

the main diagonal, which means that the model makes no mistakes in identifying the points. There is no misclassification between the "Robot," "Red," and "Green" classes, which indicates that the YOLOv7 model is very effective in distinguishing these three classes. There are no significant background errors (either false positives or false negatives), which indicates that the model does not incorrectly detect the background object as one of the three main classes and vice versa. Overall, this confusion matrix indicates that the trained YOLOv7 model has excellent detection and classification performance with perfect accuracy on the test dataset. This model is highly reliable in detecting and distinguishing red, green, and robotic dots in a variety of conditions, making it ideal for use in navigation and path planning applications on this RVM.

### 6.2. PID Control Testing

To determine the values of Kp, Ki, and Kd in testing the PID control system on robots, the trial-and-error method is used. The trial-and-error method is used to find a value until it finds a good result with the smallest error rate and closer to the supposed set point. In this test, the values of Kp, Ki, Kd were obtained, namely Kp = 0.3, Ki = 0.1 and Kd = 0.1.

### 6.3. Scenario Path Testing

At this stage is the testing stage that will be created with various scenarios. Here is the scenario path of the robot test

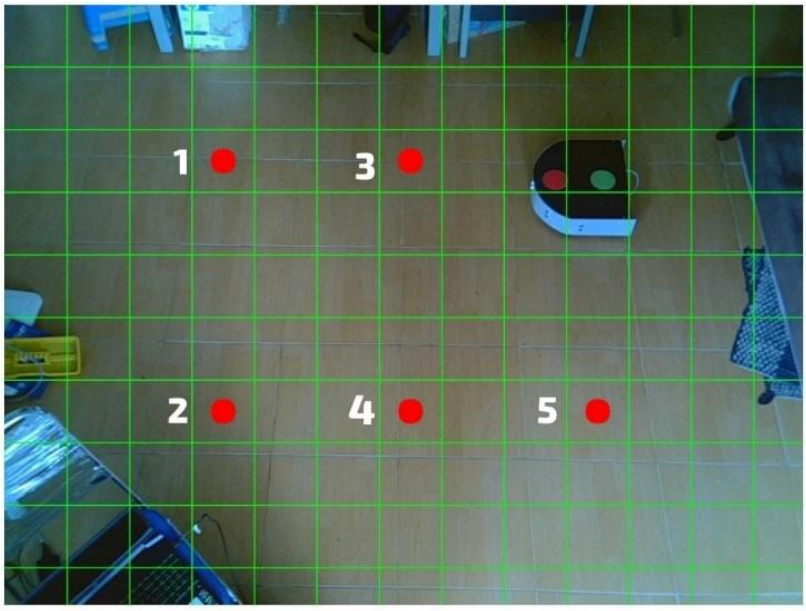



**Fig. 11.** Path test scenario 1

Based on Figure 11**,** this test will be carried out with 5 consecutive waypoints. In the waypoint-based Path Planning test, the robot will start from the starting point and move towards waypoints 1, 2, 3, 4, and 5. Here are the results of the data table obtained from the test scenario path.

**Table 2** Test Results Scenario 1

| Waypoint Start | Waypoint Goal | Coordinate X (start) | Coordinate Y (start) | Coordinate X (goal) | Coordinate Y (goal) | Distance (Px) |
|---|---|---|---|---|---|---|
| Start | Waypoint 1 | 441.5 | 139.5 | 158.4 | 122 | 283.5 |
| Waypoint 1 | Waypoint 2 | 158.4 | 122 | 174 | 323.6 | 193.2 |
| Waypoint 2 | Waypoint 3 | 174 | 323.6 | 328.8 | 125.2 | 243.4 |
| Waypoint 3 | Waypoint 4 | 328.8 | 125.2 | 334 | 327.8 | 194.1 |
| Waypoint 4 | Waypoint 5 | 334 | 327.8 | 484.6 | 322.6 | 140.9 |

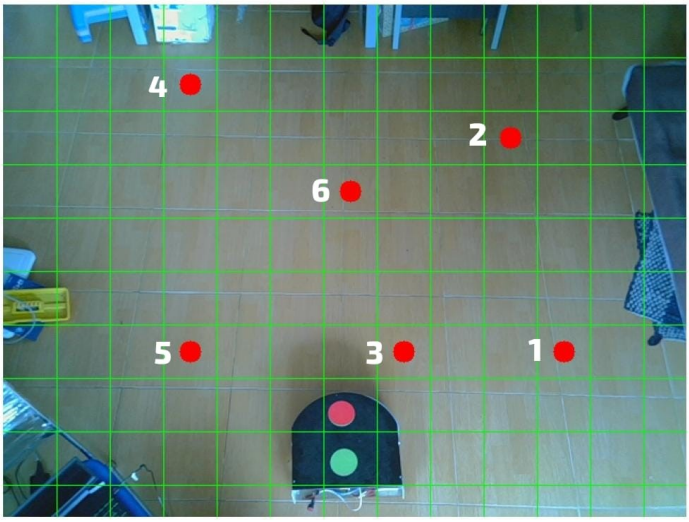
The next one is testing path with the second scenario.


**Fig. 12** Path test scenario 2

Based on Figure 12, this test will be carried out with 5 waypoints in sequence. In the waypoint-based Path Planning test, the robot will start from the starting point and move towards waypoints 1, 2, 3, 4, and 5. Here are the results of the data table obtained from the test scenario path.

**Table 3** Test Results Scenario 2

| Way-point Start | Way-point Goal | Coordinate X (start) | Coordinate Y (start) | Coordinate X (goal) | Coordinate Y (goal) | Distance (Px) |
|---|---|---|---|---|---|---|
| Start | Way-point 1 | 458.5 | 381 | 373.5 | 275.8 | 135.1 |
| Way-point 1 | Way-point 2 | 373.5 | 275.8 | 478.4 | 160.5 | 157.6 |
| Way-point 2 | Way-point 3 | 478.4 | 160.5 | 221.8 | 159.5 | 242.3 |
| Way-point 3 | Way-point 4 | 221.8 | 159.5 | 218.7 | 379.7 | 224.3 |
| Way-point 4 | Way-point 5 | 218.7 | 379.7 | 120 | 117.9 | 266 |




The next one is testing path with the third scenario.

**Fig. 13** Path test scenario 3

Based on Figure 13, this test will be carried out with 6 waypoints in sequence. In the waypoint-based Path Planning test, the robot will start from the starting point and move towards waypoints 1, 2, 3, 4, 5 and 6. Here are the results of the data table obtained from the test scenario path.

**Table 4** Test Results Scenario 3

| Waypoint Start | Waypoint Goal | Coordinate X (start) | Coordinate Y (start) | Coordinate X (goal) | Coordinate Y (goal) | Distance (Px) |
|---|---|---|---|---|---|---|
| Start | Waypoint 1 | 317 | 383 | 529.3 | 318.4 | 221.9 |
| Waypoint 1 | Waypoint 2 | 529.3 | 318.4 | 473.2 | 124.1 | 196.5 |
| Waypoint 2 | Waypoint 3 | 473.2 | 124.1 | 376.6 | 322.6 | 212.6 |
| Waypoint 3 | Waypoint 4 | 376.6 | 322.6 | 160.5 | 72.2 | 319 |
| Waypoint 4 | Waypoint 5 | 160.5 | 72.2 | 165.7 | 321.5 | 235 |
| Waypoint 5 | Waypoint 6 | 165.7 | 321.5 | 326.7 | 166.7 | 220.1 |

Based on the test results, the developed navigation system shows excellent performance in directing the robot to various predetermined waypoints. The tests were conducted using x and y coordinates for each start and destination point in each scenario, as well as the distance traveled in pixels measured through the camera view on the laptop user interface (UI). The use of pixels as the unit of measurement was chosen because the waypoints are set directly through the camera visualization, which allows for more precise control and path determination in an indoor environment.

## 7. Conclusion

The designed navigation system successfully directs the robot vacuum to the waypoint specified by the user. The use of YOLOv7 for robot detection with the help of a webcam on the ceiling proved to be effective. The system is able to recognize and track the red and green dots on the robot well, thus allowing the determination of robot orientation and direction with high accuracy. The robot was able to reach each predetermined waypoint indicating that the navigation and control system worked well.

## References

1. Huang, X., Zhao, J., & Lu, J. (2019). *Smart Vending Machines and Mobile Robot Navigation in Urban Public Spaces*. IEEE Access, 7, 116372–116381.
2. Kumar, A., & Bhattacharya, S. (2020). *Machine Vision and Mobile Robot Navigation in Smart Cities Using YOLO and Deep Learning*. Journal of Robotics and Automation, 18(3), 301–315.
3. Perez, M., & Fuentes, C. (2022). *Robotic Vending Machines and Their Applications in Smart Cities*. Robotics, 11(2), 45-58.

4. Tan, R., & Qiu, X. (2023). *Energy Efficiency in Autonomous Robots for Public Charging Stations*. Journal of Intelligent & Robotic Systems, 96(2), 273–289.
5. Zhao, J., Chen, W., & Li, H. (2021). *Optimization of Battery Charging Stations Using Robotic Systems in Public Spaces*. Sustainable Cities and Society, 74, 103215.