International Journal of Informatics and Computation (IJICOM)
Vol. 7, No. 2, 2025
ISSN: 2685-8711, E-ISSN: 2714-5263 DOI: 10.35842/ijicom

Object Recognition in Robosoccer on Wheeled Using YOLO and ROS

Agus Khumaidi[1], Muhammad Ainul Yaqin[2], Ryan Yudha Aditya[3], Sholahudin Muhammad Irsyad[4]*, Dhika Arya Pratama[5]*

Abstract

Object recognition is a critical capability for wheeled robosoccer robots operating in dynamic competition environments such as the Indonesian Wheeled Soccer Robot Contest (KRSBI-B). Limitations in real-time perception and system integration can reduce the effectiveness of autonomous navigation and opponent avoidance. This study proposes an object recognition system based on YOLOv5 integrated with the Robot Operating System (ROS) to enhance real-time perception and system responsiveness. The proposed approach employs YOLOv5 to detect opponent robots and utilizes ROS as a middleware to enable seamless communication between perception and navigation modules. Experimental results show that the system successfully detects robot objects in 11 out of 12 test scenarios, achieving an average detection confidence exceeding 0.90 within the optimal distance range of 50–350 cm. The best distance estimation performance is obtained at a distance of 350 cm, with a minimum error of 0.85%, while stable detection performance is maintained at distances up to 500 cm. These results demonstrate that the integration of YOLOv5 and ROS provides a reliable and effective solution for object recognition in wheeled robosoccer applications, supporting adaptive navigation and robust performance under dynamic operating conditions.

Keywords:
Stereo Vision, YOLOv5, Object Detection, ROS, Wheeled Soccer Robot

1. Introduction

Robosoccer on wheeled platforms represents a highly dynamic and competitive robotic application that integrates perception, decision-making, and control under strict real-time constraints. In the context of the Indonesia Robot Contest (KRI), wheeled soccer robots must autonomously identify balls, goals, field lines, teammates, and opponents while operating in an unstructured and fast-changing environment. The official KRI guidelines clearly define performance targets such as response speed, accuracy, and robustness, which place significant demands on the robot's vision system and software architecture. However, many teams still face limitations in perception reliability due to varying lighting conditions, occlusions, and rapid object motion on the field. These challenges highlight the need for an effective object recognition framework that complies with competition standards while remaining computationally efficient for embedded robotic platforms [1].

Object recognition plays a central role in enabling autonomous behavior in wheeled soccer robots, particularly for ball and goal detection. Traditional computer vision approaches often struggle with scale variation, background clutter, and motion blur, which

Corresponding Author: Agus Khumaidi
1. Agus Khumaidi, Politeknik Perkapalan Negeri Surabaya, aguskhumaidi@ppns.ac.id
2. Muhammad Ainul Yaqin, Politeknik Perkapalan Negeri Surabaya, muhammadainul21@student.ppns.ac.id
3. Ryan Yudha, Politeknik Perkapalan Negeri Surabaya, adityaryanyudhaadhitya@ppns.ac.id
4. Sholahudin Muhammad Irsyad, Politeknik Perkapalan Negeri Surabaya, muhammad.irsyad@ppns.ac.id
5. Dhika Arya Pratama, Politeknik Perkapalan Negeri Surabaya, dhikaarya12@student.ppns.ac.id

frequently occur in robosoccer scenarios. Deep learning–based object detection methods, especially those in the YOLO (You Only Look Once) family, address these issues by performing end-to-end detection with high speed and accuracy. Previous studies demonstrate that Darknet YOLO effectively detects balls and goals in soccer robots and significantly improves detection speed compared to classical methods. Nevertheless, detection performance still degrades when objects appear small, partially occluded, or under uneven illumination, indicating that object recognition remains an open research problem in real-world robotic competitions [2].

Accurate object recognition alone does not guarantee effective robot behavior; it must closely interact with navigation and path planning modules. Wheeled soccer robots rely on visual information to plan trajectories, avoid obstacles, and position themselves strategically on the field. Path planning algorithms such as A* require reliable and timely perception data to function correctly. Inaccurate or delayed object detection can lead to poor path decisions, collisions, or missed scoring opportunities. Studies on A* path planning for wheeled soccer robots emphasize that perception uncertainty directly affects navigation performance, reinforcing the importance of robust and real-time object recognition as a foundation for higher-level decision-making [3].

The adaptability of YOLO-based detection models across different robotic domains further motivates their use in robosoccer. Research on object detection for unmanned surface vehicles and other mobile platforms shows that YOLO can effectively identify color-based and shape-based targets in outdoor and dynamic environments. These findings suggest that YOLO's single-stage architecture provides strong generalization and speed advantages for mobile robots. However, transferring such models to robosoccer introduces unique challenges, including smaller object sizes, faster motion, and tighter real-time constraints. These factors require careful model selection, training strategies, and system integration to ensure consistent performance during matches [4].

Recent advancements in lightweight and optimized YOLO architectures address some of these challenges by improving small object detection and reducing computational complexity. Models such as YOLO-TLA demonstrate that architectural refinements can enhance detection accuracy while maintaining real-time performance on limited hardware. These developments are highly relevant for wheeled soccer robots, which typically use embedded processors with constrained resources. Despite these improvements, many optimized models remain underexplored in the specific context of robosoccer, particularly in combination with robotic middleware such as ROS. This gap motivates further investigation into selecting and deploying efficient YOLO variants for competitive robotic soccer systems [5].

In addition to RGB-based vision, depth sensing has gained attention for enhancing perception robustness in robotics. Depth cameras provide valuable spatial information that can support object localization and distance estimation, which are critical for ball tracking and goal positioning. Research using depth images for fall detection and motion analysis demonstrates that depth data improves perception reliability under challenging visual conditions. In Robosoccer, depth-assisted perception can mitigate issues caused by lighting variations and object overlap. However, integrating depth sensing with deep learning–based object detection increases system complexity and computational load, raising questions about feasibility in real-time competitive settings [6].

The Robot Operating System (ROS) serves as a standard middleware for integrating perception, control, and communication modules in mobile robots. ROS provides flexible tools for sensor handling, data synchronization, and algorithm deployment, making it widely adopted in robosoccer research. Tutorials and community resources show that ROS simplifies the integration of YOLO-based object detection pipelines and supports modular system development. Nonetheless, real-time performance in ROS-based systems depends heavily on node architecture, message latency, and hardware acceleration. Many existing implementations still struggle to achieve stable, low-latency perception during fast-paced soccer matches, indicating the need for optimized ROS–YOLO integration strategies [7].

Despite extensive research on object detection, path planning, and decision-making for wheeled soccer robots, a clear gap remains in holistic system-level evaluation. Many studies focus on individual components, such as detection accuracy or navigation algorithms, without fully addressing their interaction in real match conditions. Systematic reviews of deep learning–based object detection for mobile robots confirm that real-time performance, robustness, and scalability remain unresolved challenges. Therefore, this study focuses on object recognition in wheeled robosoccer using YOLO integrated with ROS, aiming to address practical issues of accuracy, speed, and system integration under competition constraints. By building upon relevant prior work, this research seeks to contribute a more reliable and deployable perception framework for autonomous wheeled soccer robots [22].

2. Related Works

Several studies have focused on applying YOLO-based object detection specifically to robosoccer environments, emphasizing ball and goal recognition as primary perception tasks. Nugroho and Anifah design a ball and goal detection system using Darknet YOLO and demonstrate that single-stage detectors significantly outperform traditional vision methods in detection speed and accuracy under competition-like conditions. Their results confirm that YOLO effectively supports real-time perception for wheeled soccer robots. However, their system primarily evaluates detection in controlled scenarios and does not deeply analyze robustness against lighting variations, occlusions, or integration with higher-level robotic frameworks such as ROS, which limits its applicability in full-match conditions [2].

Research on robotic soccer also highlights the close relationship between object recognition and navigation. Safitri et al. implement the A* path planning algorithm for wheeled soccer robot base stations and show that accurate environmental perception directly influences navigation efficiency and collision avoidance. Their work demonstrates strong performance in structured field representations but assumes reliable detection outputs. This assumption exposes a limitation, as perception errors from vision systems can propagate into navigation failures. The study indirectly emphasizes the need for robust and low-latency object recognition to support path planning, yet it does not address perception uncertainty or real-time vision integration [3].

Beyond soccer-specific applications, YOLO has been successfully applied to other mobile robotic platforms, providing insights relevant to robosoccer. Romadloni et al. use YOLO to identify buoy colors on an unmanned surface vehicle and report high detection accuracy in dynamic outdoor environments. This work highlights YOLO's adaptability and robustness in mobile contexts with environmental disturbances. Nevertheless, the detection targets in their study are larger and slower-moving than a soccer ball, and the system does not face the same strict latency constraints as robosoccer, limiting direct transferability of the results [4].

Recent developments in YOLO architecture aim to address small object detection and computational efficiency, which are critical challenges in robosoccer. Ji et al. propose YOLO-TLA, a lightweight model optimized for detecting small objects while maintaining real-time performance. Their experiments show improved accuracy compared to standard YOLOv5 variants on limited hardware. This contribution is highly relevant to wheeled soccer robots, where the ball often occupies a small image region. However, the study evaluates performance mainly on general datasets and does not validate the model in robotic or ROS-based environments, leaving practical deployment questions unanswered [5].

Depth-based perception has also been explored as a complementary approach to RGB vision. Biswas et al. utilize depth images from an Orbbec Astra 3D Pro camera for automatic fall detection and demonstrate that depth information enhances robustness against lighting changes. Although their application domain differs from Robosoccer, the findings suggest that depth sensing can improve spatial understanding and object localization. The main limitation lies in the increased computational complexity and sensor cost, which may not be suitable for lightweight wheeled soccer robots competing under hardware constraints [6].

The Robot Operating System plays a central role in many robotic perception studies by enabling modular development and sensor fusion. Jalil provides a comprehensive guide to ROS, detailing how perception, control, and communication nodes interact in mobile robots. ROS-based tutorials and community resources further demonstrate the feasibility of deploying YOLO in ROS 2 environments. Despite these strengths, many ROS–YOLO implementations still experience latency issues due to message passing overhead and inefficient node configurations. These limitations become critical in robosoccer, where perception delays can directly affect decision-making and robot responsiveness [7].

Comparative and review studies provide broader insights into object detection methods for mobile and soccer robots. Szemenyei and Estivill-Castro introduce ROBO, a fully neural object detection framework designed for robot soccer, and show strong robustness compared to conventional pipelines. Similarly, comparative analyses in humanoid and wheeled robot soccer reveal that YOLO-based models often achieve the best balance between accuracy and speed. However, these studies also note that performance varies significantly depending on hardware, camera setup, and system integration, indicating that no single model universally solves all perception challenges [18], [19].

Systematic reviews further confirm that deep learning–based object detection dominates recent mobile robot perception research. Tjahyaningtijas et al. systematically analyze object detection methods for wheeled mobile robots and conclude that YOLO variants are the most widely adopted due to their real-time capability. Nonetheless, the review identifies persistent gaps in real-world validation, ROS integration, and end-to-end system evaluation. Many studies focus on detection accuracy alone without considering deployment constraints and interaction with navigation and decision-making modules. These limitations motivate the proposed study to focus on object recognition in wheeled robosoccer using YOLO tightly integrated with ROS, addressing both algorithmic performance and system-level practicality [22].

3. Proposed Method

The selection of algorithms and system architecture in competitive robotic environments such as the Wheeled Soccer Robot Contest (KRSBI-B) must consider real-time constraints, dynamic interactions, and limited computational resources. Therefore, this study adopts YOLOv5 for object detection and the Robot Operating System (ROS) as the system integration framework. YOLOv5 is selected due to its single-stage detection architecture, which enables simultaneous object localization and classification in a single forward pass. This characteristic significantly reduces inference latency compared to two-stage detectors, making YOLOv5 highly suitable for real-time robotic applications where rapid decision-making is required. In addition, YOLOv5 demonstrates a favorable balance between detection accuracy and computational efficiency, allowing deployment on embedded robotic platforms.

The Robot Operating System (ROS) is utilized to support modular and distributed system integration. ROS provides a node-based architecture that enables independent subsystems—such as vision processing, navigation, and control—to communicate through a publish–subscribe mechanism. This design enhances system scalability, fault isolation, and real-time data exchange, which are critical for autonomous robot operation in dynamic soccer environments.

3.1 Orbbec Astra Pro Plus Stereo Camera

The Orbbec Astra Pro Plus is a stereo camera used to collect data on wheeled soccer robots. This camera has a depth sensor that can detect and measure the distance of objects around the robot, such as balls, obstacles, or opposing robots. The robot uses this depth data to perform various tasks, such as tracking the position of the ball, avoiding obstacles, determining the direction of movement, and kicking the ball correctly. To ensure that the robot is able to adapt to dynamic game situations, various environmental conditions are tested [6].

Fig. 1. Orbbec Astra Pro Plus Stereo Camera

In order to use this camera, you need the OpenNI SDK and API to run all of its features properly. OpenNI acts as a bridge between data processing software and visual sensors, such as the Orbbec Astra Pro Plus camera. Its main purpose is to provide a standard data format so that developers don't have to worry about compatibility between sensors and software. This way, sensor manufacturers like Orbbec can focus solely on developing their hardware, while application developers can use the standard data generated to create cross-platform applications. Applications based on natural interactions, such as motion tracking or action analysis, are easier to develop with OpenNI, which allows developers to access 3D data directly [9].

3.2 Distances Estimation System on Stereo Vision Cameras

An intelligent perception system must be able to handle two main problems to interact effectively in a dynamic environment such as a robotic soccer field. This research uses a system that integrates two main technologies: object detection using You Only Look Once (YOLOv5) and distance estimation using stereo vision cameras. YOLOv5, a fast and accurate object detection algorithm, is used to initiate the object identification process. The YOLOv5 architecture consists of three main parts: a backbone used to extract visual features from images captured by the camera; a neck used to predict the location of objects in the form of bounding boxes and their classes. This architecture enables the robot to quickly recognize important objects such as balls or opposing robots.

After an object has been successfully identified in two dimensions, the next step is to determine its actual distance. For this task, a stereo vision camera system, such as the Orbbec Astra Pro Plus, is used, which mimics the way human vision works. This system uses two lenses to capture images from the left and right perspectives simultaneously. The main principle is disparity.

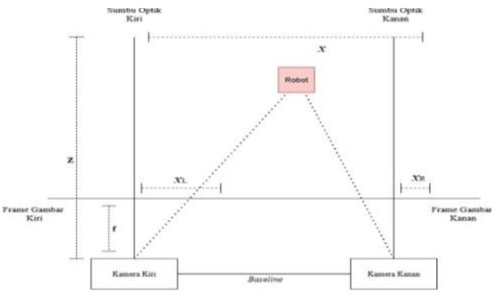

Fig. 2. Principle of Disparity

$$\frac{Z}{f} = \frac{X}{Xl} \quad (2.1)$$

$$\frac{Z}{f} = \frac{X - b}{X_R} \quad (2.2)$$

$$X_l = \frac{X}{Z} \, x \, f \quad (2.3)$$

$$X_R = \frac{X - b}{Z} \, x \, f \quad (2.4)$$

$$Disparity = X_L - X_R \quad (2.5)$$

Fig. 3. Equation of Disparity

In stereo vision cameras, the disparity system uses the difference in view between two cameras mounted at a certain distance, known as the baseline, to measure the depth or distance of objects. It uses the position of objects detected on both cameras (left and right) to calculate the distance of objects. While XL and XR are the positions of objects on the left and right camera images, X indicates the position of objects on the horizontal axis. The relationship in equation 2.1, where Z/f = X / XL, shows how the depth of an object (Z) is calculated based on the position of the object in the left image (XL). The XL value is related to the position of the object closer to the left camera, and the depth of the object can be calculated by comparing the position of the object (X) with the position in the left image (XL) [10].

The disparity system in stereo vision cameras measures the depth or distance of objects using the difference in view between two cameras mounted at a certain distance, called the baseline. It calculates the distance of objects using the detected positions on both cameras (left and right). While XL and XR indicate the position of objects in the left and right camera images, X indicates it on the horizontal axis. In equation 2.1, the relationship Z/f = X / XL shows that the depth of the object (Z) is based on the position of the object in the left image (XL). The XL value is related to the position of the object closer to the left camera, and the position of the object (X) can be compared to determine the depth of the object [10].

3.3 Object Recognition with YOLOv5

The You Only Look Once (YOLO) algorithm is a convolutional neural network (CNN)-based approach designed to perform object detection in real time. Unlike conventional detection methods that separate object localization and classification into multiple stages, YOLO integrates both processes into a single inference stage. This unified detection strategy enables YOLO to identify objects efficiently at high processing speeds, reaching up to 45 frames per second, which is suitable for real-time robotic applications [4].

In the YOLO detection mechanism, the input image is processed holistically, enabling the model to detect objects at various locations and scales simultaneously. The algorithm assigns confidence scores to candidate object regions, where regions with the highest scores are considered valid detections, allowing YOLO to prioritize relevant objects while maintaining computational efficiency [4]. As illustrated in Fig. 4, the YOLO network architecture is inspired by the GoogleNet model for image classification but adopts a simplified fully convolutional structure to enhance detection performance. The architecture consists of 24 interconnected convolutional layers, where the original GoogleNet inception modules are replaced by a 1×1 convolutional reduction layer followed by a 3×3 convolutional layer. This architectural modification improves feature extraction capability while reducing computational complexity, making YOLO effective for real-time object detection. The convolutional layers are initially trained using the ImageNet classification dataset at a resolution of 224 × 224 pixels and subsequently adapted to full-resolution images for object detection tasks [4].

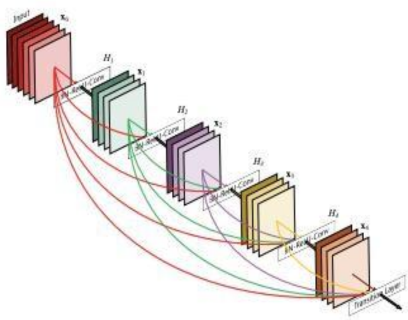

Fig. 4. YOLOv5 Network Architecture

In this study, the YOLOv5 version is employed due to its superior performance in detection speed and accuracy compared to earlier YOLO versions and other object detection methods. YOLOv5 supports deployment on both CPU and GPU platforms, providing flexibility for real-time implementation on robotic systems with limited computational resources. Furthermore, YOLOv5 utilizes the Sigmoid-weighted Linear Unit (SiLU) activation function, which enhances network performance by improving gradient flow during training and increasing detection efficiency during inference. These characteristics make YOLOv5 particularly suitable for applications that require fast, accurate, and responsive object recognition in dynamic environments [5].

In this study, the YOLOv5 version is employed due to its superior performance in detection speed and accuracy compared to earlier YOLO versions and other object detection methods. YOLOv5 supports deployment on both CPU and GPU platforms, providing flexibility for real-time implementation on robotic systems with limited computational resources. Furthermore, YOLOv5 utilizes the Sigmoid-weighted Linear Unit (SiLU) activation function, which enhances network performance by improving gradient flow during training and increasing detection efficiency during inference. These characteristics make YOLOv5 particularly suitable for applications that require fast, accurate, and responsive object recognition in dynamic environments.

To quantitatively evaluate the performance of the YOLOv5-based object recognition system, a confusion matrix approach is employed. The confusion matrix consists of four fundamental components: True Positive (TP), False Positive (FP), True Negative (TN), and False Negative (FN), which represent the outcomes of the detection and classification process. Based on the confusion matrix, the overall accuracy of the object recognition system is calculated as:

$$Accurasy = \frac{TP + TN}{TP + TN + FP + FN}$$

Precision and recall metrics are defined as :

$$Precision = \frac{TP}{TP + FP}$$

$$Recall = \frac{TP}{TP + FN}$$

These evaluation metrics provide a quantitative assessment of the reliability and effectiveness of the YOLOv5 model in recognizing objects, particularly in real-time robotic applications where detection accuracy directly influences navigation and decision-making performance [4].

3.4 Robot Operating System (ROS)

The Robot Operating System (ROS) is a Linux-based system designed specifically for mobile robots. With ROS, robot developers can design, develop, and manage complex robotic systems. The system consists of a number of nodes, each of which has a program that performs a specific function. The ROS Master acts as a message bus to facilitate communication between nodes. Thus, ROS enables developers to create integrated robotic systems consisting of many interconnected components [7]

Fig. 5. Robot Operating System (ROS) Logo

In addition, ROS provides simulation visuals for performing robotics simulations. Before using it directly on robots, users can perform visualizations such as environmental areas, robots, and so on. In this study, the “Noetic” version of ROS was used, an ROS distro with LTS (Long-Term Support) status, which is supported in the long term. ROS “Noetic” was chosen because it has many robot software libraries developed by the ROS community, which makes robotics system development easier and more effective [7].

3.5 Gazebo

Gazebo is open source simulation software created by the Open Source Robotics Foundation and intended to simulate hardware or dynamic systems, especially in robotics. Gazebo allows users to design, test, and simulate robots in various environments and conditions with highly accurate simulation capabilities.

Gazebo can simulate robots in highly complex environments, both indoors with various objects and obstacles and outdoors with various variables, such as weather and terrain conditions. The Gazebo application runs well on Linux and has excellent graphics that allow users to view robot simulations with realistic and interactive visuals. Gazebo's ability to work in real time, providing instant feedback on changes to the model or algorithm being tested, is one of its advantages [8].


Fig. 6. Gazebo Simulation

3.6 Hardware Implementation Design

With the help of stereo vision cameras, the mechanical design of the wheeled soccer robot hardware was designed with many factors in mind, such as component function efficiency and stability, which are certainly related to the Indonesian Robot Contest regulations. The design process was carried out using Autodesk Autocad and Fusion360 design programs, and aluminum was used to make the robot lightweight but strong. Each component of the robot is designed to be modular and has a different function to make it easier to assemble and maintain. The mechanical hardware system consists of many important parts.


Fig. 7. Overall Hardware Design

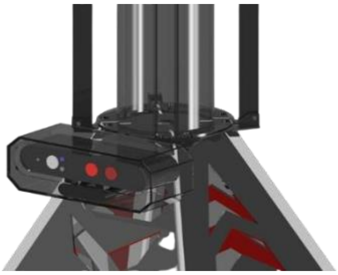
Fig. 8. Stereo Camera Placement Design

In this study, the Orbbec Astra Pro Plus camera serves as the primary visual sensor because its depth sensor can directly measure the distance of objects (balls, opponents, and obstacles). The robot requires distance data to make strategic decisions, such as navigation, obstacle avoidance, and executing accurate kicks. A series of complex scenarios simulating real-world conditions was used to test the reliability of this system. To maximize this, a stereo vision camera was placed at the top front of the robot. This placement allows the camera to see a wide area in front without interfering with the robot's sound.

Fig. 9. Hardware Implementation

4. Experimental Setup

4.1 YOLO Model Training Dataset

The dataset used to train the YOLOv5 model focused specifically on one class of objects, namely “robots,” which were identified based on their distinctive visual characteristics: black color and upward-tapered shape. To build this dataset, images of robots were collected and labeled manually. Next, data augmentation processes were carried out, such as rotation, scaling, and lighting and contrast adjustments to enrich data variation and make the model more resilient to various conditions in the field. The reliability of the model was then tested in difficult scenarios, such as when the robot was obstructed, moving, or in unstable lighting, and validated with separate data to prevent overfitting. The ultimate goal of preparing this structured dataset was to create a fast and accurate robot detection system so that wheeled soccer robots could respond optimally during matches.

Table 1. Dataset Collection for Robot Objects

Class	Dataset	Total
Robot		2000

4.2 Dataset Pre-processing Stage

The YOLOv5 object detection model is evaluated after training on a dataset of 2,000 images representing a single class, namely robots, where each robot is manually annotated to enable accurate localization that supports field strategy execution. The dataset is systematically divided into training, validation, and testing subsets to ensure effective learning and objective evaluation: 70% of the images are used for training to allow the model to learn visual patterns and features of robot objects, 20% are allocated for validation to monitor performance during training, tune hyperparameters, and prevent overfitting, and the remaining 10% are reserved for testing using previously unseen images to assess real-world performance and generalization capability. This structured dataset partitioning ensures that the trained model achieves not only high accuracy during training but also robustness and adaptability when deployed in competitive robosoccer environments.

5. Result and Analysis

5.1 Robot Object Detection Test with YOLOv5 and Real Distance Estimation

Using the YOLOv5s-based detection model, robot detection tests were conducted to evaluate the system's ability to identify the presence of opposing robots. In this test, the main parameter used was the distance between the camera and the robot; observation scenarios were conducted at various distances, ranging from the closest to the furthest that was still possible for detection.

Fig. 10. Robot Object Detection with YOLOv5

The results of the detection test on robot objects using the YOLOv5-based detection system are shown in Figure 10. In this test, the robot objects were in a static position and approximately 2 meters from the camera. After the camera captured the image in real time, the system processed the image to identify the object using a pre-trained model. The image shows that the object can be recognized well, marked with a blue bounding box around it, and has a class label of “robot” and a confidence value of 0.91, which indicates that the model has a fairly high level of confidence in the prediction.

Table 2. Robot Object Detection Results with YOLOv5

Test Number	Light Intensity (Lux)	Detection Result	Confidence Value	Actual Distance (cm)	Camera Distance (cm)	Distance Error (%)
1	216	Detected	0.98	50	85	**41,18**
2	232	Detected	0.95	100	123	**18,70**
3	252	Detected	0.97	150	176	**14,77**
4	284	Detected	0.92	200	223	**10,31**
5	313	Detected	0.98	250	254	**1,57**
6	322	Detected	0.98	300	311	**3,54**
7	350	Detected	0.93	350	353	**0,85**
8	355	Detected	0.90	400	394	**1,52**
9	275	Detected	0.85	450	439	**2,51**
10	311	Detected	0.83	500	488	**2,46**
11	298	Detected	0.64	550	535	**2,80**
12	**276**	**Not Detected**	**0.00**	**600**	**0**	**0**

Overall, the experimental results indicate that the proposed system is capable of reliably detecting robot objects in most experimental scenarios, as indicated by the "Detected" status in 11 out of 12 experiments. The system was evaluated at distances up to 600 cm under varying light intensities, where reliable detection was achieved at distances below 600 cm. At the maximum tested distance of 600 cm with a light intensity of 276 lux, the object could no longer be detected due to insufficient visual information.

A comparison of detection performance across different distances shows that detection confidence remains very high, with an average value exceeding 0.90 within the optimal distance range of 50–350 cm. At closer distances, although the object is clearly visible, higher distance estimation errors occur due to perspective effects and camera calibration limitations. In contrast, medium distances provide a more balanced condition, resulting in higher confidence values and more accurate distance estimation. As the distance increases beyond this range, a gradual decrease in confidence is observed, which can be attributed to reduced object size and loss of visual detail in the input image.

Based on distance accuracy evaluation, the YOLOv5 model exhibits its best performance at medium distances, particularly at 350 cm, where the minimum distance estimation error of 0.85% is achieved. Although larger errors are observed at very close and far distances, the overall system performance remains stable, supported by consistent detection under varying light intensities. These results demonstrate that the proposed YOLOv5-based object recognition system provides optimal accuracy within typical operational ranges and is suitable for deployment in competitive robotic platforms such as the KRSBI-B wheeled soccer robot competition.

5.2 Integration of YOLOv5 with ROS

This experiment evaluates the integration between the YOLOv5-based object recognition system and the ROS-based navigation framework using the A* path planning algorithm. The primary objective is to verify that object position information detected by YOLOv5 can be transmitted, interpreted, and utilized by ROS in real time to support adaptive robot navigation in dynamic environments.

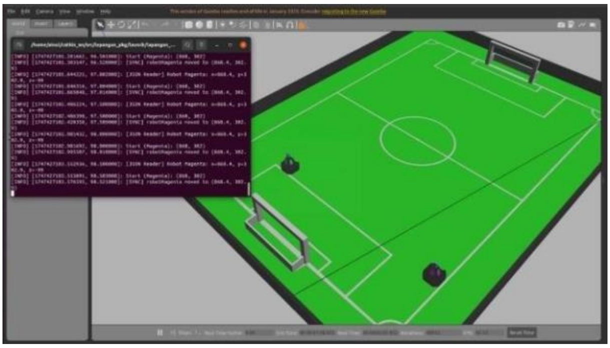

Fig. 11. Testing the Integration of YOLOv5 with ROS Using Simulation on Gazebo

As illustrated in Fig. 11, the YOLOv5 detection system outputs object position data in JSON format, containing the detected object's spatial coordinates. A ROS node is configured to periodically read and parse this data, allowing the navigation system to incorporate object position information—such as opposing robots—into the path planning process. The A* algorithm then generates a collision-free path by avoiding regions surrounding the detected objects.

The successful reception and utilization of object position data are confirmed through the ROS terminal output and the Gazebo simulation environment. The detected coordinates are applied to update the robot's position, resulting in observable changes in the robot's trajectory. These results demonstrate that the integration between YOLOv5 and ROS enables responsive and adaptive navigation, highlighting the effectiveness of combining object recognition and robotic middleware to enhance performance in dynamic Robosoccer scenarios.

6. Conclusion

Based on the experimental results, the proposed object recognition system using YOLOv5 and ROS demonstrates reliable performance for wheeled robosoccer applications. The YOLOv5 model successfully detects opposing robot objects in 11 out of 12 experimental scenarios, achieving high confidence values exceeding 0.90 within the optimal distance range of 50–350 cm. The best distance estimation performance is obtained at a medium distance of 350 cm, with a minimum error of 0.85%, indicating stable and accurate perception under varying distance and lighting conditions.

Furthermore, the integration of YOLOv5 with the ROS-based navigation system enables real-time utilization of object position data for adaptive path planning. The successful communication between the detection and navigation modules confirms that object recognition results can be effectively translated into navigation decisions in dynamic environments. Overall, these findings show that the combined use of YOLOv5 for object recognition and ROS for system integration enhances the responsiveness and robustness of wheeled robosoccer robots, making the proposed approach suitable for competitive robotic scenarios such as the KRSBI-B competition.

Acknowledgment

The author would like to thank all parties who have provided support and contribute in completing this research. Special thanks are extended to the supervising lecturer who has provided guidance and input throughout the research process. The author also expresses appreciation to the institution and laboratory that provided the necessary facilities and data. Not to forget, the author thanks colleagues who helped in conducting experiments and analyzing data. All forms of assistance, whether direct or indirect, were very meaningful in supporting the smooth running of this research.

References

[1] P. Ndra, I. M. K. Benyamin, and M. A. Kusumoputro, *Guidelines Book for the 2023 Indonesia Robot Contest (KRI)*. Jakarta, Indonesia: Balai Pengembangan Talenta Indonesia, Pusat Prestasi Nasional, Ministry of Education, Culture, Research, and Technology, 2023.

[2] D. N. Nugroho and L. Anifah, "Design of ball and goal object detection system for soccer robots using the Darknet YOLO method," *Journal of Information Engineering and Educational Technology*, vol. 7, no. 1, pp. 22–29, 2023, doi: 10.26740/jieet.v7n1.p22-29.

[3] L. I. Safitri, J. Sahertian, and D. W. Widodo, "Implementation of A* path planning algorithm on wheeled soccer robot base station," *Generation Journal*, vol. 7, no. 3, pp. 56–63, 2023, doi: 10.29407/gj.v7i3.20545.

[4] F. Romadloni, J. Endrasmono, Z. M. A. Putra, A. Khumaidi, I. Rachman, and R. Y. Adhitya, "Buoy color identification using the You Only Look Once method on an unmanned surface vehicle," *TRIAC Journal of Electrical and Computer Engineering*, vol. 10, no. 1, 2023, doi: 10.21107/triac.v10i1.19650.

[5] C.-L. Ji, T. Yu, P. Gao, F. Wang, and R.-Y. Yuan, "YOLO-TLA: An efficient and lightweight small object detection model based on YOLOv5," Feb. 2024, doi: 10.1007/s11554-024-01519-4.

[6] Biswas, B. Dey, B. Poudyel, N. Sarkar, and T. Olariu, "Automatic fall detection using Orbbec Astra 3D Pro depth images," *Journal of Intelligent & Fuzzy Systems*, vol. 43, no. 2, pp. 1707–1715, 2022, doi: 10.3233/JIFS-219272.
[7] Jalil, *Complete Guide to Robot Operating System (ROS)*, 2023.
[8] D. Alhajir, Y. V. Via, and W. S. J. Saputra, "Hardware-based rice and foreign object detection system using Google Colab," *Journal of Informatics and Information Systems*, vol. 2, no. 3, 2021, doi: 10.33005/jifosi.v2i3.369.
[9] F. Li and P. Li, "Computer-aided teaching software of three-dimensional sports movement models based on Kinect depth data," *Computer-Aided Design and Applications*, vol. 18, no. S2, pp. 123–134, 2020, doi: 10.14733/cadaps.2021.S2.123-134.
[10] R. Ramadhan, A. Khumaidi, M. K. Mayangsari, M. Syai'in, I. Sutrisno, and A. R. Annisa, "Application of Harris corner detection and YOLOv5 on stereo vision cameras for distance estimation of KRSBI-B wheeled soccer robots," *Journal of Electronics and Industrial Automation*, vol. 12, no. 1, pp. 111–122, 2025, doi: 10.33795/elkolind.v12i1.7254.
[11] M. A. Yaqin *et al.*, "Implementation of A* path planning to avoid opponent robots using Robot Operating System in KRSBI-Beroda," vol. 12, 2025.
[12] W. Darmawan, M. B. Rahmat, A. Khumaidi, R. Y. Adhitya, and D. P. Riananda, "Design of decision-making strategies for wheeled soccer robots using the decision tree method," *Journal of Electronics and Industrial Automation*, vol. 10, no. 2, pp. 175–182, Jul. 2023, doi: 10.33795/elkolind.v10i2.3020.
[13] M. A. Adi, A. Khumaidi, M. B. Rahmat, D. P. Riananda, M. K. Hasin, and D. Sukoco, "Implementation of fire point detection system in graving dock areas using YOLOv5," *Journal of Electronics and Industrial Automation*, vol. 11, no. 2, pp. 473–482, Jul. 2024, doi: 10.33795/elkolind.v11i2.5233.
[14] Rizqyakbar and C. K. Dewa, "Application of object detection models for robots using SSD in ROS simulation environments," 2024. [Online]. Available: http://jurnal.mdp.ac.id
[15] H. Kusuma, D. Ahmad, S. Department, and R. D. Department, "Assisted depth imaging for scoring goals in wheeled soccer robots," 2024. [Online].
[16] W. Liu *et al.*, "Image recognition for garbage classification based on transfer learning and model fusion," *Mathematical Problems in Engineering*, vol. 2022, pp. 1–12, Aug. 2022, doi: 10.1155/2022/4793555.
[17] Y. Hu, G. Liu, Z. Chen, and J. Guo, "Object detection algorithm for wheeled mobile robots based on an improved YOLOv4," *Applied Sciences*, vol. 12, no. 9, p. 4769, May 2022, doi: 10.3390/app12094769.
[18] H. Jati, N. A. Ilyasa, Y. Indrihapsari, A. Chandra, and D. D. Dominic, "Enhancing object detection for humanoid robot soccer: Comparative analysis of three models," *TELKOMNIKA*, vol. 22, no. 4, p. 25906, 2024.
[19] M. Szemenyei and V. Estivill-Castro, "ROBO: Robust, fully neural object detection for robot soccer," *Neural Computing and Applications*, vol. 34, pp. 21419–21432, 2022.
[20] R. R. Julianda and R. D. Puriyanto, "Ball tracking using YOLOv8 on wheeled soccer robots with omnidirectional cameras," *Buletin Ilmiah Sarjana Teknik Elektro*, vol. 6, no. 2, pp. 203–213, 2024.
[21] H. Soebhakti *et al.*, "The real-time object detection system on mobile soccer robots using YOLOv3," in *Proc. 2nd Int. Conf. Applied Engineering (ICAE)*, 2019, doi: 10.1109/ICAE47758.2019.9221734.
[22] H. P. A. Tjahyaningtijas *et al.*, "Object detection for wheeled mobile robots based on deep learning: A systematic review," *Journal of Robotics and Control*, vol. 6, no. 3, pp. 1425–1433, 2025.
[23] ROS Community, "ROS 2 tutorial on visual object recognition," 2023.
[24] ROS Community, "YOLO deployment in ROS2 Humble," 2025.
[25] "Optimization of YOLO for real-time object detection in robotic applications," *Electronic Science, Technology and Application*, vol. 12, no. 1, 2025.