Volume 01, Issue 02, December 2025 : 30-37 Received: December 18, 2025. Revised: December 30, 2025.

International Journal of Electrical and Intelligent Engineering

Ball Detection in Wheeled Soccer Robot Using the YOLOv8 Model

Aqza Tri Ananda HAT[1]* Shoffin Nahwa Utama[2] M. Imamudin[3]

Yunifa Miftachul Arif[4] Ajib Hanani[5]

[1,2,3,4,5]*Universitas Islam Negeri Maulana Malik Ibrahim Malang, Indonesia*

* Corresponding author's Email: aqzatrianandahat@gmail.com

Abstract: This research designs and builds a wheeled soccer robot using YOLOv8 for real-time ball detection and distance estimation, aiming to improve efficiency in technology competitions. The system includes Arduino Uno R3, Raspberry Pi 3 model b, detection system, and navigation design. 691 ball image use as dataset that consist of 552 image as training dataset and 249 image as valid dataset. YOLOv8 demonstrated exceptional reliability in ball detection during testing, achieving an average accuracy of 100%, 100% precision, and 94% recall. Navigation testing toward the ball had an acceptable average error of 8.0466%. The results confirm that YOLOv8 is excellent for simplifying high-accuracy ball detection and distance estimation in wheeled soccer robots. Future work should consider a higher-spec Raspberry Pi, a high-resolution camera, additional sensors, and advanced systems to improve detection and obstacle avoidance (opponent robots, goal).

Keywords: Object Detection, Robot Navigation, Wheeled Soccer Robot, YOLOv8

1. Introduction

For years, robots have assisted humans in various fields because they have a greater capacity and are often better prepared to complete more complex tasks, as shown by a higher level of repeatability. Advances in the fields of batteries, sensors, artificial intelligence (AI), and machine learning (ML) have opened the door to new areas and new applications [1]. One interesting and challenging application in this field is the development of soccer robots, which combines artificial intelligence (AI), mechanics, and control engineering. Soccer robots offer an ideal platform for testing new algorithms, hardware, and strategies under complex and unpredictable conditions.

Object detection, especially the detection of the ball which is a key component in soccer robots. Without the ability to detect the ball accurately and quickly, the robot will not be able to interact effectively with its environment, thereby hindering its ability to compete in the game. In this context, image processing and machine learning technologies are very important to provide efficient and accurate solutions. Object detection can simplify the process of identifying object types from an image in a modern way, such as using a laptop camera or webcam. In the final project, the output of the detection ball will be used as data to detect the ball on the soccer robot [2].

The YOLO (You Only Look Once) model is widely known as one of the most effective and fastest object detection methods. The latest version of this model, YOLOv8, offers improvements in speed and accuracy, making it an ideal candidate for real-time applications such as soccer robots. YOLOv8 uses a more sophisticated architecture and optimization algorithms that enable fast and accurate object detection, even in poor lighting conditions or complex backgrounds.

This research focuses on the application of the YOLOv8 model to detect the ball on wheeled soccer robots. By utilizing the real-time detection capabilities of YOLOv8, it is expected that the robot can quickly and accurately obtain the necessary visual information to make decisions in the game. This implementation is expected to not only improve the robot's performance in detecting and following the ball but also provide an important contribution to the development of smarter and more responsive robotic technology.

The use of YOLOv8 in the context of soccer robots also opens up opportunities for further exploration in the fields of computer vision and AI. This includes the development of further algorithms to improve detection reliability under various environmental conditions, as well as integration with control systems that allow robots to respond with smarter and more adaptive actions.

International Journal of Electrical and Intelligent Engineering

Although previous studies have successfully applied earlier YOLO versions (primarily YOLOv3) for ball detection in soccer robots, they often rely on more powerful hardware such as NVIDIA GPUs or Jetson platforms and larger datasets. Few works demonstrate real-time performance using the latest YOLOv8 on low-cost, resource-constrained embedded systems like the Raspberry Pi 3. Moreover, integrated distance estimation and autonomous navigation using bounding box outputs remain underexplored on such hardware. This study addresses these gaps by implementing the lightweight YOLOv8n model for accurate real-time ball detection, pinhole-based distance estimation, and navigation on a low-cost wheeled soccer robot platform.

2. Related Works

In 2019, Soebhakti et al. [3] conducted a study on a soccer robot capable of detecting objects like the ball, goal, and center circle using a camera. They utilized YOLOv3 (You Only Look Once Version 3), a method based on CNN [4], for fast and highly accurate detection. The software specifications included CMAKE 3.8 for modern CUDA, CUDA 10.0, OpenCV 2.4, cuDNN 7.0, running on Linux 16.04. With an Octa Core Intel Core i7-7700HQ processor, 16 GB RAM, and a 3GB NVIDIA GeForce GTX 1060 graphics card, they achieved a detection speed of 28.3 FPS, IOU of 71.76%, recall of 0.92, precision of 0.92, and mAP of 87.07% using 52,000 data samples. The study also proved YOLOv3's ability to detect objects under three different lighting conditions, with a maximum range of 3 meters for the ball and 8 meters for the goal.

YOLOv3 was also used in a 2020 study by Susanto et al. [5], who implemented XNOR-Net [6] on a humanoid soccer robot [7]. The addition of XNOR-Net, which uses a binary Convolutional Neural Network for image classification, significantly lightened the YOLO [8] operation, making it 58 times faster and using 32 times less memory than other methods. With XNOR-YOLO on a Logitech C92 1080p webcam, an Intel NUC6i5sYH Core i5 miniPC, and an NVIDIA Jetson TX1 GPU, they achieved a detection rate of 30 FPS for both the ball and the goal.

Another study utilizing YOLOv3 on a humanoid soccer robot was conducted by Nugraha et al. [9] in 2021. They built a robot capable of detecting the ball, goal, field boundaries, and other robots (teammates or opponents). This research tested YOLOv3's detection accuracy in various situations: full and partial visual ball detection, slow and fast ball movement, ball distance, and detection time. The results showed good detection for full and partial visual balls, but YOLOv3 struggled with moving objects, requiring the robot to approach the object for effective detection. The optimal ball detection range was found to be 50-900 cm, with an average detection time of 0.033 seconds, based on 3000 trained image samples.

In 2022, Sanubari & Puriyanto [10] utilized YOLO (YOLOv3 and YOLOv3-Tiny) on a KRSBI-B robot to detect the ball and goal using an Omnidirectional camera. Using 8000 datasets (7000 training and 1000 validation) with frame sizes 320x320 and 416x416, they achieved accuracies of 81.8% (YOLOv3) and 74.2% (YOLOv3-Tiny) for the former, and 93.2% (YOLOv3) and 81% (YOLOv3-Tiny) for the latter. These results confirmed that YOLOv3's mAP was consistently higher than YOLOv3-Tiny's, indicating both models could detect the ball and goal effectively. They suggested incorporating other deep learning methods for robot movement strategies.

Most recently, Jati et al. (2024) [11] used YOLO-NAS (Neural Architecture Search) in their study, "Enhancing Humanoid Robot Soccer Ball Tracking, Goal Alignment, and Robot Avoidance Using YOLO-NAS," for ball detection, goal alignment, and obstacle avoidance maneuvers. This research also use YOLOv8 [12] as a comparison for the model. The research achieved an average success rate of 53.3% for ball detection (from 60 samples), 91.7% for goal alignment, and 100% for opponent avoidance maneuvers (from 10 samples). They recommended using more data to remove research limitations.

Previous studies predominantly employed YOLOv3 and its variants, achieving good detection performance but typically on higher-end hardware (e.g., NVIDIA GTX 1060 [3], Jetson TX1 [5]) that exceeds the computational constraints of low-cost platforms like the Raspberry Pi 3. These approaches also required larger datasets (thousands to tens of thousands of images) and did not integrate simple yet effective distance estimation for autonomous navigation on constrained hardware. In contrast, the present work utilizes the more efficient YOLOv8n architecture with a significantly smaller dataset (691 images) to achieve comparable or superior precision while enabling real-time operation and navigation on

Volume 01, Issue 02, December 2025 : 30-37 Received: December 18, 2025. Revised: December 30, 2025.

International Journal of Electrical and Intelligent Engineering

a Raspberry Pi 3, highlighting improved suitability for budget-constrained robotic applications.

Compared to earlier YOLO versions used in soccer robots (primarily YOLOv3 and YOLOv3-Tiny), YOLOv8n offers architectural improvements including anchor-free detection and advanced training strategies, resulting in better speed-accuracy trade-offs on resource-limited hardware. Ultralytics benchmarks indicate YOLOv8n outperforms YOLOv5n and YOLOv3-Tiny in mAP while maintaining similar or faster inference times on CPU-only systems, making it more suitable for deployment on the Raspberry Pi 3 without requiring GPU acceleration.

3. Research Method

This study develops a ball-detection system for a wheeled soccer robot using the YOLOv8 object-detection model. The proposed system enables the robot to detect a yellow ball in real time, estimate its distance, and navigate toward it autonomously. The research methodology encompasses requirement analysis, system design, YOLOv8 model implementation, system testing and comprehensive performance evaluation (Fig. 1).

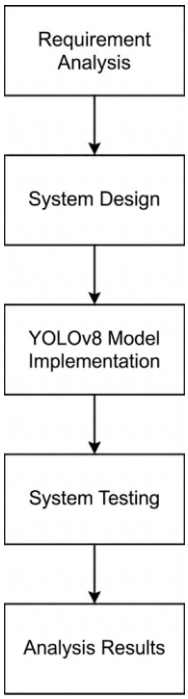

Figure 1. Research Procedure

The hardware platform consists of a Raspberry Pi [13] 3 Model B as the main processing unit, a compact CSI camera module for image acquisition, an Arduino Uno R3 [14] for low-level motor control, an L298N dual H-bridge motor driver, and four DC motors mounted on a custom wheeled soccer robot chassis. Serial communication via USB is employed to exchange detection results and distance commands between the Raspberry Pi (running Python) and the Arduino (programmed in C++ using Arduino IDE). The software environment includes Raspberry Pi OS 64-bit, Thonny IDE 4.0.2, Arduino IDE, and the Ultralytics YOLOv8 library.

Data collection involved two sources: (1) a custom dataset comprising photographs of the target yellow ball captured using a smartphone and the robot's own camera under varying lighting and angle conditions, supplemented by relevant images obtained from the internet; and (2) real-time video streams acquired directly from the mounted camera during testing. All collected images underwent manual annotation using bounding boxes and the class label "ball". Data augmentation techniques were applied to increase dataset diversity and robustness. The final dataset was split into training (80%) and validation/test (80%) sets to prevent overfitting and enable reliable performance evaluation (Fig. 2).

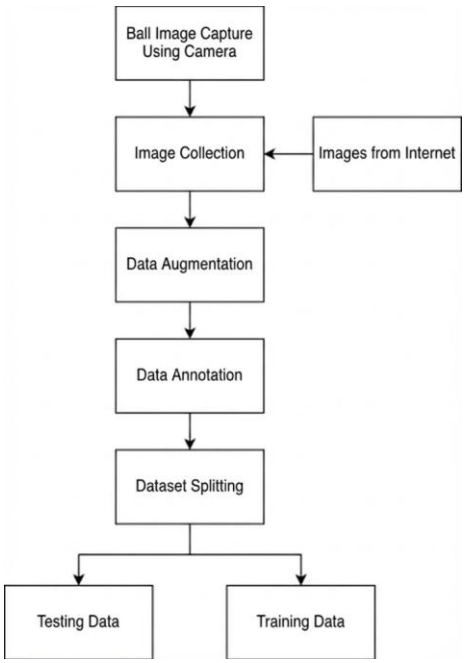

Figure 2. Data Design

The YOLOv8n (nano) pre-trained model was selected for its favorable trade-off between accuracy and inference speed on resource-constrained hardware. Custom training was performed by fine-tuning the model on the prepared dataset.

Volume 01, Issue 02, December 2025 : 30-37 Received: December 18, 2025. Revised: December 30, 2025.

International Journal of Electrical and Intelligent Engineering

Hyperparameters such as epoch count, batch size, and optimizer were adjusted empirically to achieve convergence. Upon completion of training, the best weights file (.pt) with the highest mAP@0.5 on the validation set was exported for deployment (Fig. 3).

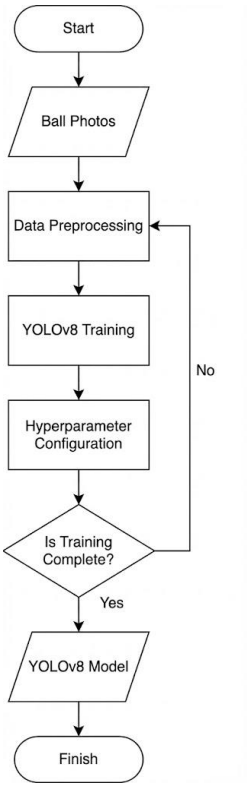

Figure 3. Training System Flowchart

The overall system workflow is as follows: the camera continuously captures frames, which are fed into the YOLOv8 model running on the Raspberry Pi. When confidence for the "ball" class exceeds 70%, the bounding-box center coordinates are obtained, and the estimated distance to the yellow ball is computed using the classical pinhole camera model. This model leverages the principle of similar triangles, where the real-world diameter of the ball D (in meters) and the camera's focal length f (in pixels, determined through prior calibration) are fixed known parameters, while the apparent projected width ω (in pixels) of the ball directly measured from the YOLOv8 bounding box varies inversely with distance; the range Z (depth along the optical axis) is thus calculated via the monotonic relationship $Z = \frac{f \times D}{\omega}$, assuming a predominantly fronto-parallel orientation of the ball to minimize perspective distortion and neglecting lens radial distortion (reasonable approximations for the low-cost CSI camera module operating within 1–3 meters). These data (bounding-box center and estimated distance) are transmitted via serial communication to the Arduino, which translates the distance into appropriate PWM signals for the DC motors, driving the robot forward toward the ball. If confidence falls below the threshold or no ball is detected, the robot resumes searching behaviour. (Fig. 4).

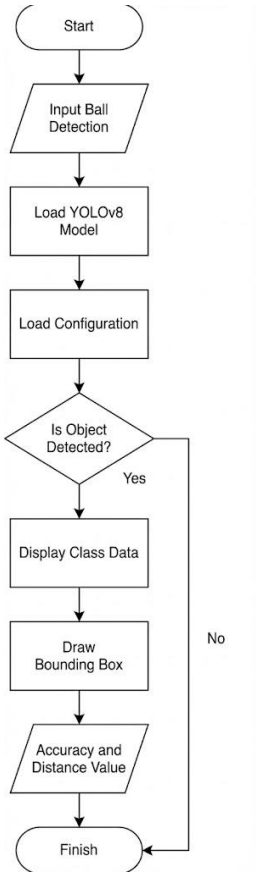

Figure 4. Object Detection Flowchart

A confidence threshold of 70% was selected for ball detection based on empirical evaluation during preliminary testing on the validation set. This value effectively balanced high precision (minimizing false positives in potential cluttered environments) with sufficient recall, aligning with common practices in real-time YOLO deployments for robotics.

System evaluation was conducted indoors under controlled yet varied lighting conditions. The yellow ball was placed at random positions and distances ranging from 1 m to 3 m from the robot. Performance metrics for object detection included True Positive (TP), False Positive (FP), False Negative (FN), and True Negative (TN), from which Accuracy,

Volume 01, Issue 02, December 2025 : 30-37 Received: December 18, 2025. Revised: December 30, 2025.

International Journal of Electrical and Intelligent Engineering

Precision, and Recall [15] were computed using the standard formulas:

$$Accuracy = \frac{TP + TN}{TP + TN + FP + FN} \times 100\% \quad (1)$$

$$Precision = \frac{TP}{TP + FP} \times 100\% \quad (2)$$

$$Recall = \frac{TP}{TP + FN} \times 100\% \quad (3)$$

Navigation performance was quantified by Navigation Error using Mean Absolute Percentage Error (MAPE) [16]:

$$MAPE\ (\%) = \frac{1}{n}\sum_{i=1}^{n}\left|\frac{Y_i - X_i}{Y_i}\right| \times 100\% \quad (4)$$

Where Yi is the ground-truth distance, and Xi is the robot's actual traveled distance. Precision-Recall curves and distance-dependent navigation error plots were generated to visualize model robustness across operating conditions.

The integrated methodology, combining state-of-the-art deep learning-based detection with lightweight embedded implementation, provides a complete framework for real-time ball detection and autonomous approach in soccer robot soccer applications, with quantitative benchmarks facilitating objective assessment and future improvements.

4. Results and Discussion

The proposed ball-detection and navigation system for a wheeled soccer robot was successfully implemented and comprehensively evaluated. The final hardware realization of the robot closely followed the design outlined in Section 3, with overall dimensions of 25 cm × 25 cm × 15 cm. The mechanical structure was driven by four DC motors controlled through an L298N dual H-bridge driver, while low-level motion control was handled by an Arduino Uno R3. Real-time image processing and inference were performed on a Raspberry Pi 3 Model B equipped with a compact CSI webcam. Power was supplied separately by a 12 V (3-cell) LiPo battery pack for the motors and Arduino, and a power bank for the Raspberry Pi to ensure stable operation during extended tests (Fig. 5).

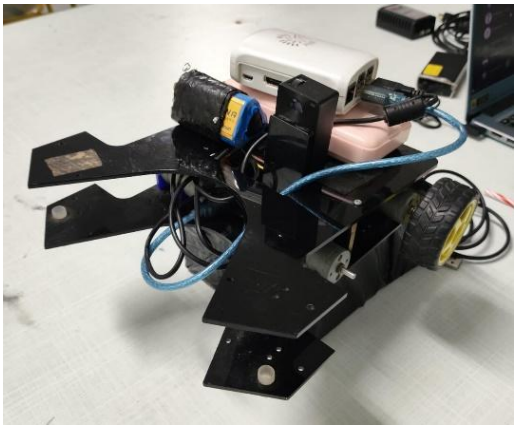

Figure 5. Soccer Robot Final Design

The YOLOv8n model was custom-trained on a dataset consisting of 691 annotated images of a yellow target ball (*kasti*/tennis ball) collected from both direct camera captures. After manual labelling and augmentation, the dataset was split into 552 training images and 139 validation images. Training was conducted for 30 epochs with a batch size of 16, AdamW optimizer, and 640×640 input resolution. The best-performing weights were deployed on the Raspberry Pi for real-time inference.

Detection logic was enhanced with a central blue region-of-interest (ROI) (Fig. 6) in the frame and predefined distance thresholds (1.0, 1.5, 2.0, 2.5, and 3.0 m). Distance estimation was performed using the known ball diameter and pinhole camera model. When a ball was detected inside the ROI with confidence ≥ 0.7 and within one of the predefined distance bands, the corresponding command (1-5) was sent via USB serial to the Arduino, which executed calibrated forward motion using PWM and timed delays. Upon reaching the target distance, both detection and motion loops terminated.

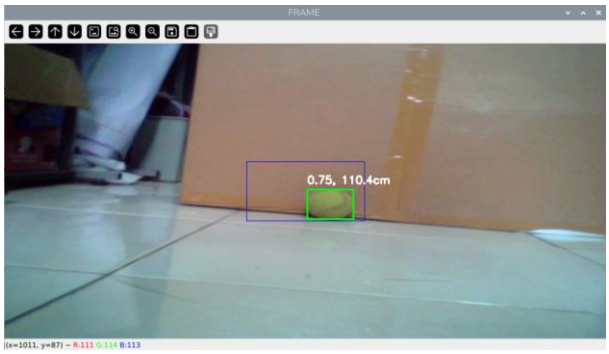

Figure 6. Detection Interface

Volume 01, Issue 02, December 2025 : 30-37 Received: December 18, 2025. Revised: December 30, 2025.

International Journal of Electrical and Intelligent Engineering

Detection performance was evaluated over 10 trials at each distance (1–3 m). The system achieved an average accuracy of 100%, precision of 100%, and recall of 94% (Table 1). Perfect precision across all distances indicates zero false positives — the model never misclassified a non-ball object as a ball. Recall remained 100% up to 2 m but dropped to 90% at 2.5 m and 80% at 3 m, primarily due to occasional processing delays on the resource-constrained Raspberry Pi 3 and minor degradation of bounding-box accuracy at longer ranges.

Table 1. Detection performance metrics across tested distances

Distance (meter)	Accuracy (%)	Precision (%)	Recall (%)
1	100	100	100
1.5	100	100	100
2	100	100	100
2.5	100	100	90
3	100	100	80
Average	**100**	**100**	**94**

The reported 100% accuracy and precision were obtained under controlled indoor conditions with stable lighting, a relatively small custom dataset of 691 images, and 10 trials per distance.

Navigation accuracy was assessed by measuring the actual stopping distance after each successful detection. Mean Absolute Percentage Error (MAPE) yielded an overall navigation error of 8.05% (Table 2). Error increased gradually with distance, from 5% at 1 m to approximately 10% at 2.5–3 m, which is attributed to cumulative wheel slippage, minor calibration drift in PWM-to-distance mapping, and small errors in distance estimation from YOLO bounding boxes.

Table 2. Navigation error results

Object Distance (cm)	Distance Reach (cm)	Navigation Error (%)
100	95	5
150	142	5.333
200	181	9.5
250	224	10.4
300	270	10
Average (%)		**8.0466**

The Precision-Recall curve (Fig. 7) confirms excellent precision maintained at 100% while recall decreases at longer ranges. The navigation error plot (Fig. 8) shows a near-linear increase, yet the maximum error remains below 11%, demonstrating robust and controllable navigation behavior suitable for indoor soccer robot applications.

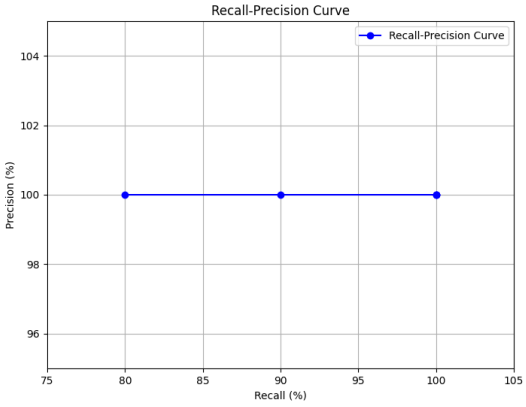

Figure 7. Precision-Recall Curve

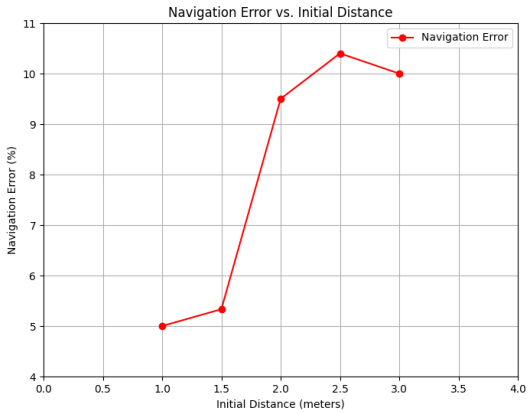

Figure 8. Navigation Error Graph

Overall, the integrated system successfully demonstrated reliable real-time ball detection and autonomous approach within 1–3 m under controlled indoor lighting. The perfect precision and high recall up to 2 m, combined with single-digit navigation error, validate the effectiveness of YOLOv8n on low-cost hardware for this task. Performance degradation beyond 2.5 m suggests future improvements in camera resolution, adoption of a more powerful single-board computer (e.g., Raspberry Pi 4/5 or NVIDIA Jetson Nano), refined distance estimation using stereo vision or depth sensors, advanced path-planning algorithms, and mechanical enhancements to reduce wheel slip. With these refinements, the proposed framework has strong potential for competitive robot soccer environments requiring fast and accurate ball acquisition.

International Journal of Electrical and Intelligent Engineering

5. Conclusion

This study successfully demonstrates the effectiveness of the YOLOv8 object detection model for real-time ball detection and autonomous navigation in a low-cost wheeled soccer robot. Experimental results from ten trials per distance (1–3 m) confirm that the system achieves 100% accuracy, 100% precision, and 94% recall in ball detection, with zero false positives across all tested conditions. The navigation module, driven by distance estimates derived from YOLOv8 bounding boxes, yields an average navigation error of only 8.05% (MAPE), with errors remaining below 11% even at the maximum tested range of 3 meters.

The consistent high precision and acceptable recall up to 2 meters, combined with single-digit navigation error, validate the suitability of the lightweight YOLOv8n variant for deployment on resource-constrained platforms such as the Raspberry Pi 3. Although detection recall and navigation accuracy slightly degrade beyond 2.5 meters due to hardware limitations and cumulative mechanical factors, overall performance remains highly reliable for indoor robot soccer applications.

In conclusion, the proposed YOLOv8-based vision system provides an accurate, robust, and computationally efficient solution for ball detection and approach tasks in wheeled soccer robots, offering a strong foundation for further development in competitive robotic soccer environments.

These results demonstrate strong performance within the tested scenario but may not fully generalize to more dynamic environments involving moving balls, varying outdoor lighting, obstacles, or larger field sizes.

6. Limitation

Although the developed system demonstrated strong performance in indoor ball detection and navigation, several limitations were identified during testing that affect overall robustness and real-world applicability.

The primary constraint arises from the use of a Raspberry Pi 3 Model B, which exhibited occasional processing delays and limited RAM and storage, causing frame drops and reduced inference speed, particularly at longer distances. Recall degradation beyond 2.5 m and the observed increase in navigation error with distance are largely attributable to these hardware bottlenecks and the relatively low resolution of the CSI camera module. Additionally, the fixed forward-facing camera created a blind zone below approximately 20 cm, preventing reliable detection of very close objects, and the simple open-loop navigation strategy based on timed PWM commands was susceptible to wheel slippage and minor mechanical inconsistencies.

To address these shortcomings, future iterations of the system are recommended to incorporate the following enhancements:

1. Upgrade the computing platform to a Raspberry Pi 4/5 (≥4 GB RAM) or an NVIDIA Jetson series module, paired with a high-capacity microSD card (≥32 GB), to support higher inference rates and eliminate real-time processing bottlenecks.
2. Employ a higher-resolution camera (≥1080p) or an omnidirectional lens system to improve detection reliability at greater distances and widen the effective field of view.
3. Integrate supplementary proximity sensors such as ultrasonic or time-of-flight sensors to compensate for the camera's near-field blind spot and provide more accurate distance measurements when the ball is within 20–30 cm.
4. Extend the perception capabilities by training the model on additional classes (opponent robots, goalposts, field lines) and implement more advanced navigation algorithms (e.g., PID control, dynamic path planning, or obstacle avoidance) to enable competitive play scenarios, including dribbling past opponents and autonomous goal scoring.

Implementing these improvements is expected to significantly enhance detection recall at longer ranges, reduce navigation error to below 5%, and transform the prototype into a fully competitive soccer robot suitable for standardized RoboCup or similar tournaments.

Conflicts of Interest

The authors declare no conflict of interest.

Acknowledgments

The author expresses sincere gratitude to Shoffin Nahwa Utama, M.T., the primary supervisor, for his extraordinary dedication, continuous guidance, constructive feedback, and unwavering support throughout the entire research period. His approachable demeanor, willingness to provide

International Journal of Electrical and Intelligent Engineering

assistance at any time, and role as both mentor and fatherly figure greatly eased the completion of this work without undue pressure. The author also extends deep appreciation to Dr. M. Imamudin, Lc, MA, the co-supervisor, for his valuable insights, thoughtful suggestions, and consistent encouragement during the preparation and development of this study. Their combined expertise and kindness were instrumental in bringing this research to a successful conclusion.

References

[1] M. Kulshreshtha, S. S. Chandra, P. Randhawa, G. Tsaramirsis, A. Khadidos, and A. O. Khadidos, "Oatcr: Outdoor autonomous trash-collecting robot design using yolov4-tiny," *Electron.*, vol. 10, no. 18, 2021, doi: 10.3390/electronics10182292.

[2] D. Diono, M. J. W. Wicaksono, A. Jefiza, and D. R. Prayudha, "Pendeteksian Objek Hasil Pengepresan Kaleng dan Botol dengan Metode You Only Look Once (YOLO) yang Diaplikasikan pada Mesin Sortir Pembelajaran PBL," *J. Integr.*, vol. 16, no. 1, pp. 1–10, 2024, doi: 10.30871/ji.v16i1.4598.

[3] H. Soebhakti, S. Prayoga, R. A. Fatekha, and M. B. Fashla, "The Real-Time Object Detection System on Mobile Soccer Robot using YOLO v3," *Proc. 2019 2nd Int. Conf. Appl. Eng. ICAE 2019*, 2019, doi: 10.1109/ICAE47758.2019.9221734.

[4] U. Aulia, I. Hasanuddin, M. Dirhamsyah, and N. Nasaruddin, "Heliyon A new CNN-BASED object detection system for autonomous mobile robots based on real-world vehicle datasets," *Heliyon*, vol. 10, no. 15, p. e35247, 2024, doi: 10.1016/j.heliyon.2024.e35247.

[5] S. Susanto, F. A. Putra, and R. Analia, "XNOR-YOLO: The high precision of the ball and goal detecting on the barelang-FC robot soccer," *Proc. ICAE 2020 - 3rd Int. Conf. Appl. Eng.*, 2020, doi: 10.1109/ICAE50557.2020.9350386.

[6] W. Lee, K. Kim, W. Ahn, J. Kim, and D. Jeon, "A Real-Time Object Detection Processor With XNOR -Based Variable-Precision Computing Unit," *IEEE Trans. Very Large Scale Integr. Syst.*, vol. 31, no. 6, pp. 749–761, 2023, doi: 10.1109/TVLSI.2023.3257198.

[7] T. Bräunl, *Robots and Controllers*. 2022. doi: 10.1007/978-981-16-0804-9_1.

[8] J. Redmon, S. Divvala, R. Girshick, and A. Farhadi, "You Only Look Once: Unified, Real-Time Object Detection".

[9] A. C. Nugraha, M. L. Hakim, S. Yatmono, and M. Khairudin, "Development of Ball Detection System with YOLOv3 in a Humanoid Soccer Robot," *J. Phys. Conf. Ser.*, vol. 2111, no. 1, pp. 1–17, 2021, doi: 10.1088/1742-6596/2111/1/012055.

[10] F. F. Sanubari and R. D. Puriyanto, "Deteksi Bola dan Gawang dengan Metode YOLO Menggunakan Kamera Omnidirectional pada Robot KRSBI-B," *Bul. Ilm. Sarj. Tek. Elektro*, vol. 4, no. 2, pp. 76–85, 2022, doi: 10.12928/biste.v4i2.6712.

[11] H. Jati, N. A. Ilyasa, and D. D. Dominic, "Enhancing Humanoid Robot Soccer Ball Tracking, Goal Alignment, and Robot Avoidance Using YOLO-NAS," *J. Robot. Control*, vol. 5, no. 3, pp. 829–838, 2024, doi: 10.18196/jrc.v5i3.21839.

[12] Д. Л. Я. Мобильного, Р. С. Использованием, and Y. И. Strong, "REAL-TIME OBJECT DETECTION AND TRACKING FOR MOBILE ROBOT USING YOLOV8 AND STRONG SORT," vol. 11, no. 116, 2023.

[13] E. Upton and G. Halfacree, *Raspberry Pi® User Guide*. 2016. doi: 10.1002/9781119415572.

[14] K. W. Humaidillah, "Modul Belajar Arduino Uno," p. 52, 2019.

[15] C. D. Manning, "Introduction to Information Retrieval," no. c, 2009, [Online]. Available: https://nlp.stanford.edu/IR-book/pdf/irbookonlinereading.pdf

[16] A. Kurniawan and A. Harumwidiah, "An evaluation of the artificial neural network based on the estimation of daily average global solar radiation in the city of Surabaya," vol. 22, no. 3, pp. 1245–1250, 2021, doi: 10.11591/ijeecs.v22.i3.pp1245-1250.