Enhancing Humanoid Robot Soccer Ball Tracking, Goal Alignment, and Robot Avoidance Using YOLO-NAS

Handaru Jati [1*], Nur Alif Ilyasa [2], Dhanapal Durai Dominic [3]
[1, 2] Department of Electronics and Informatics Engineering Education, Universitas Negeri Yogyakarta, Yogyakarta, Indonesia
[3] Department of Computer and Information Sciences, Universiti Teknologi Petronas, Malaysia
Email: [1] handaru@uny.ac.id, [2] parasyst@gmail.com, [3] dhanapal_d@utp.edu.my
*Corresponding Author

Abstract—**This research aims to enhance humanoid robot soccer Ball Tracking, Goal Alignment, and Robot avoidance tasks using YOLO-NAS. The study followed a three-stage approach involving model engineering, which involves model training, code integration, and testing by comparing it with YOLO-v8 and YOLOv7. We measured the mAP (Mean Avegara Precision) and the speed of the detection of each model. Descriptive and Friedman techniques were employed to interpret testing results. In the ball tracking task, YOLO-NAS achieved a success rate of 53.3% compared to YOLOv7 with 68.3%. In the goal alignment task, YOLO-NAS achieved the highest success rate of 91.7%. In the Robot Avoidance task, YOLO-NAS, the same as YOLOv8, 100% nailed the test. These findings suggest that YOLO-NAS performs exceptionally well in the goal-alignment task but does not excel in two other tasks related to humanoid robot soccer.**

Keywords—Humanoid Robot Soccer; YOLO; Object Detection; ML Deployment; Ball Tracking.

I. INTRODUCTION

A. Background

The field of humanoid robot soccer, or RoboCup humanoid league, is a challenging in in robotics [1], [2], [3]. The primary goal in this field is to develop autonomous humanoid robots that are proficient in soccer gameplay and demonstrate advanced functionalities like walking, running, dribbling, passing, shooting, and defending without external human control [4], [5], [6], [7], [8]. The robot must be able to discern the nature of objects within its visual scope, before determining its course of action, to mimic human ability to identify objects accurately. Achieving the level of accuracy demonstrated in human, poses a significant challenge for computers [9], [10], [11].

These robots rely on camera vision to perceive the environment [12], [13], [14], make real-time decisions, and engage with the ball and other robots through their mechanical bodies. This domain serves as a testbed for state-of-the-art artificial intelligence (AI), computer vision, motion planning, and control algorithms and contributes to the progress of robotics and AI [15], [16], [17]. It provides an entertaining platform to showcase the capabilities of autonomous humanoid machines [18].

In addressing the challenges in humanoid robot soccer, a key focus lies in refining the object detection system [19], [20], [21]. This entails seamlessly integrating the object detection system into the broader humanoid robot soccer software framework. Our research centers on three interconnected tasks pivotal to this problem: (1) ball tracking, (2) goal alignment, and (3) robot avoidance system. The effectiveness of these fundamental tasks within humanoid robot soccer is essential in the object detection system.

B. Related Research

The humanoid robot soccer for object detection system has transformed towards deep learning, particularly leveraging Convolutional Neural Network (CNN) based techniques [19], [22], [23], [24], [25]. The recent successes in the field of object detection are mostly due to the use of deep neural networks, more specifically convolutional neural networks (CNN) [26], [27], [28], [29]. Noteworthy models such as SSD (Single Shot Detection), MobileNet, and YOLO (You Look Only Once) have gained popularity across various domains, including humanoid robot soccer. Melo and Baros successfully utilized MobileNet V2, achieving a commendable 30 FPS detection speed on the NVIDIA Jetson Nano platform [30]. Yinka-Banjo et al utilized YOLOv2 to detect ball and goal and achieved 0.85 mAP at 50% IoU (Intersection over Union) [31].

Additionally, Nugraha et al. demonstrated a 95% accuracy for humanoid robot soccer for ball detection with YOLOv3 [32]. Narayanaswami Sai Kiran et al. pursued a similar approach, attaining an impressive 98.7 mAP (mean average precision) at a swift 30 ms detection speed [33]. Barry et al. utilized xYOLO, achieving a 93.6 mean Average Precision (mAP), encompassing the successful detection of goalposts in their study [34]. Despite widespread reports on implementing cutting-edge object detection systems in humanoid robot soccer, a noticeable gap exists in research addressing unique tasks within this domain. The predominant focus remains on evaluating model performance in terms of mAP and inference speed, leaving specific humanoid robot soccer tasks largely unexplored.

C. Research Gap

The integration of the YOLO object detection system into the humanoid robot soccer has also not been explored. The YOLO series is a target detection algorithm based on deep learning and convolutional neural networks [35]. Its advantages include fast speed, high detection accuracy, and real-time detection speed. Many variants of YOLO have been developed [36], [37]. Among those variants, YOLO-NAS, YOLOv7, and YOLOv8 are claimed to have the best performance against older models. YOLOv7 is a real-time object detector with high accuracy [38]. It features a design of trainable bag-of-freebies methods to improve detection accuracy without increasing inference cost YOLOv7-E6 is faster and more accurate than other transformer-based and convolutional-based detectors [38]. YOLOv7 was used by the Barelang team in the Robocup 2023 Humanoid Robot Soccer Competition [39], and they achieved third place.

YOLOv8 was released in January 2023 by Ultralytics, the company that developed YOLOv5. YOLOv8 provided five scaled versions and implemented in many types of research: YOLOv8n (nano) [40], [41], [42], [43], YOLOv8s (small) [44], [45], [46], YOLOv8m (medium) [47], [48], [49], [49], YOLOv8l (large) [50], [51], [52], and YOLOv8x (extra-large) [53], [54], [55]. YOLOv8 supports multiple vision tasks such as object detection [56], segmentation [57], pose estimation [58], tracking [59], [60], [61], and classification [62], [63], [64], [65], [66]. YOLO-NAS was released in May 2023 by Deci, a company that develops production-grade models and tools to build, optimize, and deploy deep learning models. YOLO-NAS is designed to detect small objects [67], [68], improve localization accuracy [69], and enhance the performance-per-compute ratio, making it suitable for real-time edge-device applications [62], [63], [64], [65], [66], [67], [68], [69], [70].

Upon integrating YOLO variants into humanoid robot soccer, rigorous testing must be conducted to evaluate the robot's success in performing tasks such as ball tracking, goal aligning, and robot collision avoidance. Research, therefore, is required to facilitate a systematic exploration of the interaction between the YOLO object detection system and the humanoid robot soccer, providing valuable insights into the system's effectiveness in real-world scenarios. Therefore, this research aims to bridge the identified gap by concentrating on evaluating the performance of three crucial tasks in humanoid robot soccer, which include ball tracking, goal aligning, and robot collision avoidance by comparing the capabilities of three YOLO variants encompassing YOLO-NAS, YOLOv7, and YOLOv8. Unlike previous studies primarily assessing the object detection model, our approach involves measuring the achievements of specific humanoid robot soccer tasks enabled by these state-of-the-art YOLO-NAS object detection systems. This research is expected to provide a comprehensive understanding of how these models contribute to the success of distinct tasks of humanoid robot soccer.

II. Method

A. Methodology

The research was conducted in three stages, encompassing (1) model engineering, (2) code engineering, and (3) testing. The first stage (model engineering) focused on crafting an optimal model tailored for the experiment, ensuring its suitability for the envisioned objectives [71]. This involves model fine-tuning using a prepared dataset and optimizing the model. The second stage (code engineering) is conducted by integration into the broader robotic software system. The integration stage involves embedding the model using Pytorch and TensorRT. This integration extended to interfacing the model with the three specific humanoid robot soccer tasks (ball tracking, goal aligning, and robot collision avoidance), establishing a cohesive and interconnected system. The final stage (testing) scrutinized the performance of the integrated model within the context of the designated tasks, providing valuable insights into its real-world applicability and efficacy. The test was designed to evaluate the functionality of the newly integrated detection system when applied to each specific task. It aimed to ascertain the accuracy and effectiveness of the object detection system in facilitating the correct execution of tasks dependent on it. Controlled soccer game environments and scenarios are set up to conduct this test, systematically evaluating the utilization of the object detection system in real-world situations.

B. Research Instrument

Several instruments need to be mentioned for the experiment. These instruments include software and embedded scripts onto the rest of the robot system. The software consists of each model training and evaluation software (YOLOv7, Ultralitics, and Super Gradients). The researchers also utilized Netron to see the structure of the trained YOLO model. A video recorder script embedded into the rest of the robot software is utilized, and the result can be used to determine the tests' success accurately. A robot remote control software is utilized to perform the test experiment.

C. Data Analysis Techniques

The employed data analysis techniques include descriptive analysis. Raw testing results are subjected to this analysis. Friedman's Two-Way Analysis of Variance by Ranks, also known as Friedman's test, is also used in this research. It is a non-parametric statistical method that compares the means of three or more related groups. It involves ranking the data within each group, calculating a test statistic based on these ranks, and comparing it to a critical value from the chi-square distribution. If the computed statistic exceeds this critical value, the null hypothesis of no differences among the groups is rejected, indicating significant differences. In this research, we compared the test results of each model variant on three different tasks and determined if YOLO-NAS differs from YOLOv7 and YOLOv8.

III. Result and Discussion

A. Model Engineering

The model was trained using a custom-made dataset consisting of 2054 images, which were partitioned into training, validation, and test subsets using an 80:10:10% ratio. These images are a collection of our robot vision footage concatenated with multiple image datasets found publicly on the Roboflow platform. There are 3 kind of objects that are labeled, this includes ball, goal post and robot. The labels are represented in a bounding boxes which hold the information of the rectangullar coordinates as well as the object class identifier. Labelimg software was used to create a label of each image. The model enginering resulted in 1644 images for training, 205 for validation, and 205 for testing. The researchers utilised the mosaic technique to add variety to the model and avoid model overfitting. Fig. 1 shows an example of the training datasets.

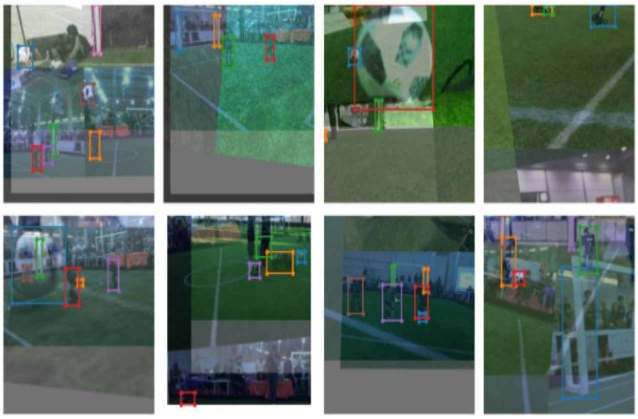

Fig. 1. Examples of training datasets

The initial training of our model was set for 100 epochs. The batch size was carefully chosen to fit our training hardware's memory capacity constraints. Three distinct model variants were trained using different frameworks: YOLOv7 was trained using Wongkinyiu's YOLOv7 framework, YOLO-v8 with Ultralytics, and YOLO-NAS with Supergradients. Python notebooks were utilized to streamline and organize the training process for each YOLO variant. The training was successful in all three models, identified by stabilized performance improvement in 100 training epochs as shown in Fig. 2.

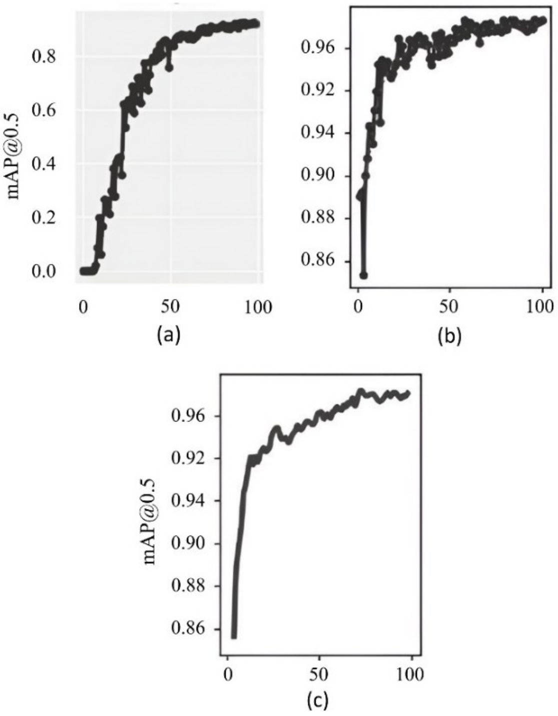

Fig. 2. Each YOLO model training performance

To reduce the complexity of our model, it was automatically simplified using ONNX (Open Neural Network Exchange). Furthermore, the model's precision is decreased from FP32 (Floating Point 32) to FP16 (Floating Potin 16), enhancing performance. Following these modifications, TensorRT was employed to compile the ONNX model. This compilation process incorporated techniques such as layer fusing and tensor fusing, all of which contribute to optimising the model's inference performance. Nevertheless, it is important to acknowledge that these optimization approaches may slightly compromise mAP metrics in exchange for enhanced inference speed.

The detection pipeline, implemented in Python, includes image preprocessing, inference, and post-processing stages. The preprocessing phase is minimal for YOLO-NAS, owing to the integration of an image normalization algorithm directly within the YOLO-NAS engine. Moreover, the requisite post-processing steps have been streamlined, as Non-Maximum Suppression (NMS) is executed within the TensorRT engine for all three models, contributing to an efficient and optimized detection process. Fig. 3 is an EfficientNMS Node as the end of model structure viewed using Netron.

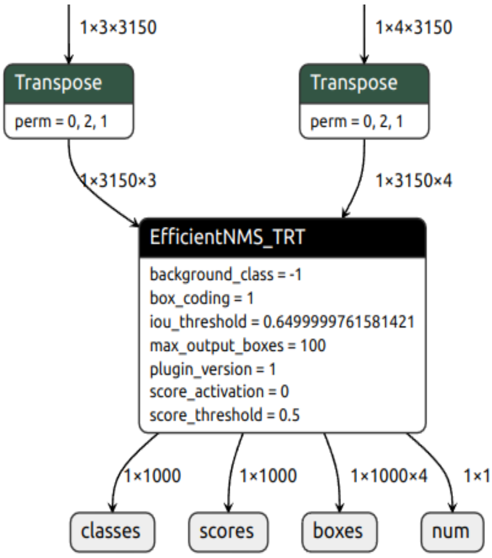

Fig. 3. The EfficientNMS node attached at the end of the YOLO model

The performance metrics of the model were assessed on both the training device and the target deployment device. The unseen test split dataset is used for this evaluation. The primary metrics was utilized to evaluate our trained YOLO models were mean average precision (mAP) and inference speed. Comparisons were also conducted between the raw trained model and the TensorRT compiled model to provide comprehensive insight into the performance enhancements achieved through model optimization.

Table I evaluates three models based on Mean Average Precision at 0.5 IoU. We also measure the mAP of each class. Table II shows the evaluation of speed metrics. The researchers measure the inference only speed and the whole pipeline from image acquisition, inference, and reading the model output speed.

From the table, we can gain insight into the best model in terms of mAP, which is YOLOv8 with 0.926 mAP@0.5 optimized model mAP for whole classes. Regarding speed, YOLOv7 has the best performance in this metric, with a total pipeline FPS of 28.9 on Jetson Nano. YOLO-NAS has the balance in terms of speed and mAP with 0.897 optimized model mAP@0.5 and 17.51 FPS of Pipeline speed on Jetson Nano.

TABLE I. Raw Model and Optimized Model Evaluation on Mean Average Precision

Model	Device	Engine	mAP@0.5			
			Ball	*Goalpost*	*Robot*	*Total*
YOLOv7	*RTX 3060 12GB*	*Pytorch*	0.986	0.966	0.896	0.949
		TensorRT	0.959	0.811	0.623	0.798
	Jetson Nano	*TensorRT*	0.959	0.811	0.623	0.798
YOLOv8	*RTX 3060 12GB*	*Pytorch*	0.995	0.972	0.935	0.967
		TensorRT	0.990	0.922	0.866	0.926
	Jetson Nano	*TensorRT*	0.990	0.922	0.866	0.926
YOLO-NAS	*RTX 3060 12GB*	*Pytorch*	0.995	0.972	0.935	0.965
		TensorRT	0.979	0.892	0.820	0.897
	Jetson Nano	*TensorRT*	0.979	0.892	0.820	0.897

TABLE II. Raw Model and Optimized Model Evaluation on Speed Metrics

Model	Device	Engine	Speed (Seconds)		Pipeline
			Inference	Pipeline	FPS
YOLOv7	*RTX 3060 12GB*	*Pytorch*	0.0018	0.0052	192.31
		TensorRT	0.0008	0.0017	588.24
	Jetson Nano	*TensorRT*	0.0248	0.0346	28.90
YOLOv8	*RTX 3060 12GB*	*Pytorch*	0.0036	0.0068	147.06
		TensorRT	0.0014	0.0027	370.37
	Jetson Nano	*TensorRT*	0.0611	0.0752	13.30
YOLO-NAS	*RTX 3060 12GB*	*Pytorch*	0.0032	0.005	200.00
		TensorRT	0.0012	0.002	500.00
	Jetson Nano	*TensorRT*	0.0482	0.0571	17.51

Fig. 4 shows the whole robot architecture. The researchers integrated the object detection system in Python. An abstraction layer, referred to as the YOLO interface, has been developed to streamline the integration of three different YOLO model versions into our object detection pipeline. This design choice simplifies transitioning between various versions of YOLO, ensuring a smoother and more efficient workflow to test the deployed models.

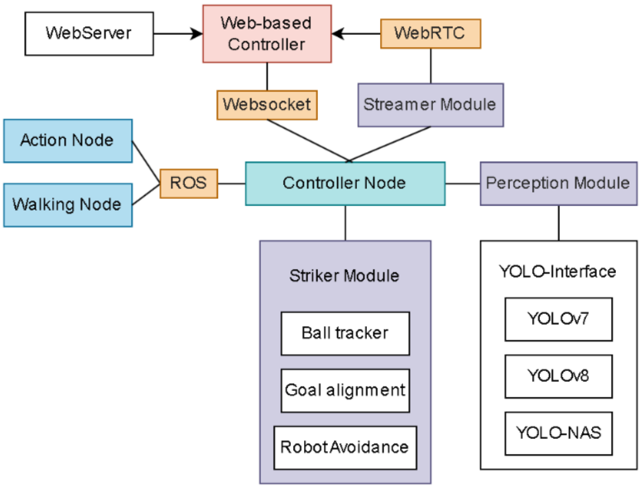

Fig. 4. Robot software structure

The controller, implemented in Python, functions as a ROS (Robot Operating System) node and serves as a central hub for communication among various nodes within the robot, including the walking and action nodes. Additionally, the controller houses a WebSocket server instance, facilitating communication with the web user interface. The node also consists of an instance of the striker module, which dictates the robot's behavior using a Finite State Machine (FSM) algorithm. Concurrently, the streamer manages vision-related tasks, including video capturing via OpenCV from GStreamer, vision recording, and hosting a live video streaming server using WebRTC.

B. Integration with the Ball Tracking System

A strategic approach is adopted to mitigate false positives in ball detection, particularly when the system identifies multiple balls. The system retains a memory of the ball’s previous position, serving as a vital reference point in subsequent iterations. This information plays a crucial role in accurately identifying and selecting the bounding box that corresponds to the nearest ball, ensuring consistency in tracking.

The ball-tracking mechanism is further enhanced through the implementation of a Proportional Integral Derivative (PID) algorithm. The ball tracking system will take the picked bounding box of the ball position to calculate its error. The PID takes an error difference with the target and gives an output of the head pan and tilt motion. The error is the ball detection relative to the center of the image. The feedback loop of this algorithm will always make the robot head align with the ball. If the ball is not at the center of the image, the system will force the head to perform pan and tilt motion so that the ball sits at the center of the camera frame. This motion is performed on the head so that the robot would not lose track of the ball. Fig. 5 shows the whole process of ball tracking in the robot.

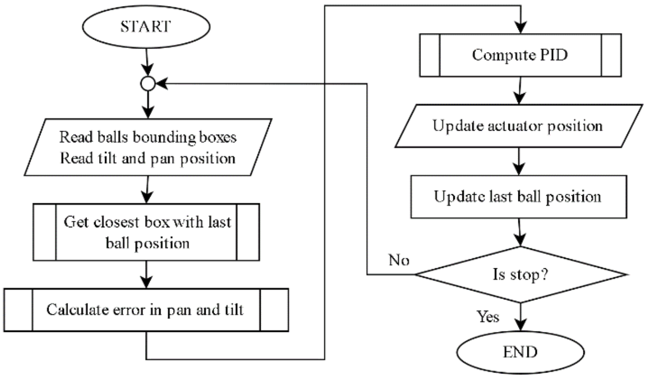
START
Read balls bounding boxes
Read tilt and pan position
Get closest box with last ball position
Calculate error in pan and tilt
Compute PID
Update actuator position
Update last ball position
Is stop?
No
Yes
END

Fig. 5. The ball tracking algorithm represented in a flowchart

C. Integrating with Goal Alignment

The alignment process commences when the robot positions itself directly in front of the ball, necessitating the integration of ball and goal detection. In this specific task, it is imperative to distinguish between aligning the robot with the goal and positioning it onto the goalpost, the latter being the detected object. Consequently, an additional post-processing algorithm is incorporated to achieve this precision in alignment.

First, the robot scans for the goalpost in a 180-degree span of the head pan. The goal yaw angle relative to the robot body orientation is calculated from the head yaw angle and x position of the detected goal boxes. This yaw position is collected. Then, the detection is clustered into 2 groups (left and right goalposts).

Denote S where each P represents the yaw value of each yaw position (1). S is separated into 2 clusters G_left and G_right. M can be calculated using the formula mentioned in (2). M is used to divide the yaw position as mentioned in (3) and (4).

$$S = \{P_1, P_2, \dots, P_n\} \quad (1)$$

$$M = \frac{P_{max} + P_{min}}{2} + P_{min} \quad (2)$$

$$G_{left} = \{P_i \in S | P_i < M\} \quad (3)$$

$$G_{right} = \{P_i \in S | P_i \geq M\} \quad (4)$$

To get usable middle yaw of the goal information, the mean is calculated for each group which results in positions of 2 goalposts. Centre of the goal is calculated by summing 2 goalposts together and dividing it by 2. The mathematical representation of this formula is mentioned in (5).

$$Mid\ goal = \frac{1}{2}\left(\frac{\sum_{i=1}^{|G_{left}|} P_i}{G_{left}} + \frac{\sum_{j=1}^{|G_{right}|} P_j}{G_{right}}\right) \quad (5)$$

After the middle of the goal is found, the robot performs a walking motion to align its body to the center of the goal. When the desired pose is achieved, the robot will stop its walking motion and perform a kicking motion. Fig. 6 is the flowchart representation of the code.

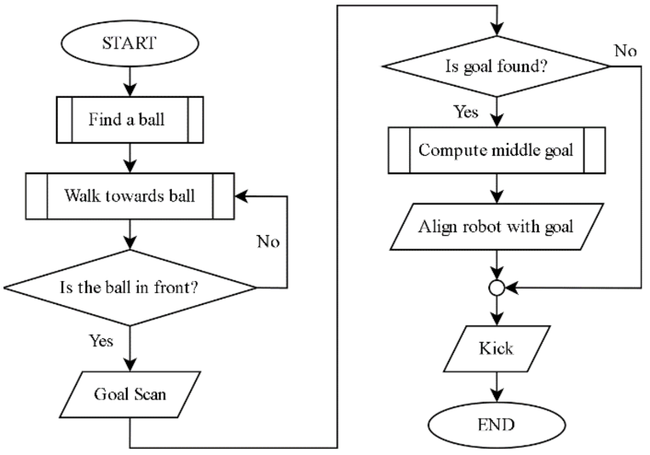

Fig. 6. The flowchart representation of YOLO integration with goal alignment task

Once the robot has secured the ball upfront, it is imperative to align it with the center of the goal to augment the likelihood of scoring. This task computes the discrepancy between the robot's heading poses and the center of the ball. A shifting or turning walking motion will be executed to align the robot with the goal's center. The objective is to enable the robot to align with the goal accurately from 7 meters. The robot should accomplish this task within a span of 120 degrees, with the robot's pose aligning with the goal.

D. Integrating with Robot Collision Avoidance

The integration begins by extracting robot bounding boxes from the model inference. To address false positive detection errors related to robots, the system records robot positions over a sequence of five consecutive iterations, a parameter subject to later adjustment. Five iterations prove both adequate and immediate when evaluating model performance in this context. The system initiates an avoidance maneuver only when a robot consistently appears across all iterations. While multiple robots may be within the robot's vision, the developed algorithm can only track a single robot to count appearance consistency. The closest appearance of the opposing robot is selected to initiate the evading maneuver, following a predefined path when the condition is triggered. Notably, this algorithm functions within a finite-state machine architecture. The procedure for robot collision avoidance is illustrated in Fig. 7 through a flowchart diagram.

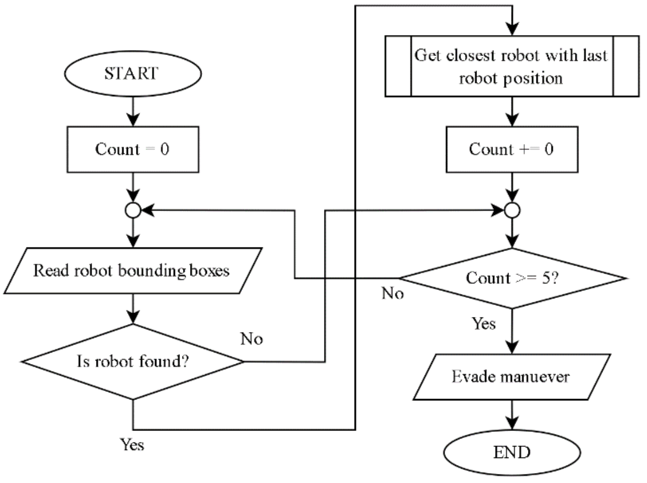

Fig. 7. Flowchart representation of goal aligning process

E. Testing the Ball Tracking

To ascertain the robot's ability to track the ball, a series of experiments were carried out to evaluate the tracking system's success rate. The ball moves at speeds ranging from 1 m/sec to 6 m/s. The number of the ball test speed is set at this variation of speed since this is the possible ball speed that are possible to occur in a humanoid robot soccer game. If the head of the robot can move its direction to the direction of the ball movement at a certain speed, the test is a success. The test would fail if the robot did not follow the ball's movement. To measure the consistency of success of the test, 10 performed tests are sampled. The distance between the robot and the ball are set to 1 meter in all iterations of the tests. The outcomes are summarized in Table III. Three models were tested on each 6 variations of speeds.

Fig. 8 shows the ongoing testing process of ball tracking. The ball is moved manually, and the speed will be measured later with the help of video recording. However, the measurement method utilized to gauge the object's speed is subject to potential inaccuracies, necessitating acknowledgement. The procedure unfolds as follows: the robot captures visual data, with a textual overlay at the image's corner, displaying a numerical value indicative of the temporal interval between successive frames. Fig. 9 shows the live video recording of the robot. Subsequent analysis of the recording entails determining the object's displacement, facilitated by identifying two white markers set at 0.5 meters on the field, in conjunction with the aforementioned time gap. The resultant data is subsequently categorized according to varying speed intervals.

TABLE III. THE TEST RESULT OF THE BALL-TRACKING TASK

Variant	Speed (m/sec)	Trials										Success Rate
		1	*2*	*3*	*4*	*5*	*6*	*7*	*8*	*9*	*10*	
NAS	***1***	1	1	1	1	1	1	1	1	1	1	32/60 (53.3%)
	2	1	1	1	1	1	1	1	0	1	0	
	3	0	0	1	0	0	1	1	1	1	1	
	4	1	1	0	0	1	1	0	0	1	1	
	5	0	0	0	0	0	1	0	0	0	1	
	6	0	0	0	0	0	0	0	0	0	0	
V7	***1***	1	1	1	1	1	1	1	1	1	1	41/60 (68.3%)
	2	1	1	1	1	1	1	1	1	1	1	
	3	1	1	1	1	1	1	1	1	1	1	
	4	1	1	0	0	1	1	0	0	1	1	
	5	0	0	1	0	1	0	0	0	0	1	
	6	1	0	0	0	0	0	0	0	0	1	
V8	***1***	1	1	1	1	1	0	1	1	1	1	26/60 (43.3%)
	2	1	1	1	1	1	1	1	0	1	0	
	3	0	1	1	0	1	1	0	0	1	0	
	4	0	1	0	0	1	1	0	0	1	0	
	5	0	0	0	0	0	0	0	0	0	0	
	6	0	0	0	0	0	0	0	0	0	0	

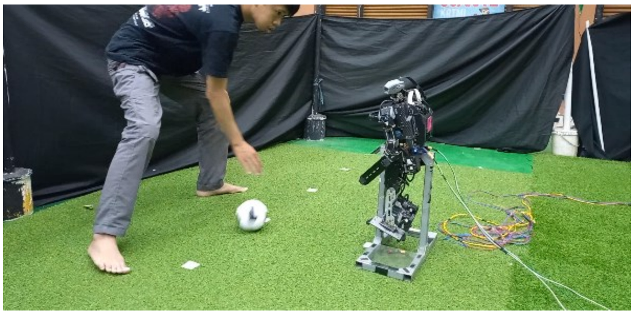

Fig. 8. Ball tracking testing

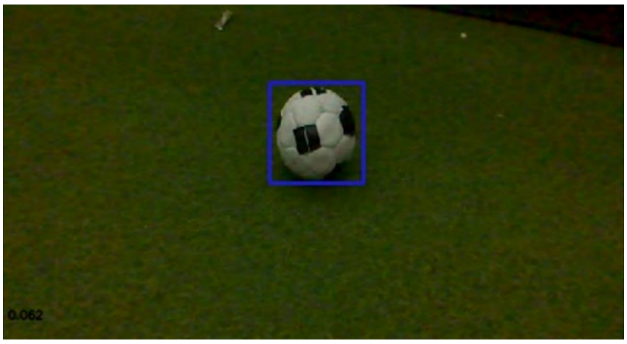

Fig. 9. Ball tracking testing robot vision recording

YOLOv7 has outperformed other models in this task. Ball Tracking using YOLOv7 maintained its reliability until the ball was moving at three m/sec before losing track of the ball. Ball tracking using YOLO-NAS and YOLOv8 has lower performance. YOLOv7 can still track the ball even at the speed of 6 m/sec. YOLO-NAS can track the ball at a speed of up to 5 m/sec. YOLOv8 can track the ball at a speed of up to 4 m/sec. Fig. 10 shows the complete comparison of 3 models' performance against the speed of the ball movement.

This result is affected by the computation speed of the model. YOLOv7 has the fastest computational speed at 24 ms, followed by YOLO-NAS at 48 ms and YOLOv8 at 61 ms on Jetson Nano. The fastest computation speed corresponds to a faster response against faster object movement. A low detection frame rate caused by slower computational speed is the limiting factor of the ball-tracking task.

F. *Testing the Goal Alignment*

During the test, a successful detection prompts the robot to make a slight left or right turn, followed by a kicking motion. The test is deemed successful if the robot turns in the correct direction after scanning for goalposts. The results of the test are detailed in Table IV.

TABLE IV. RESULT OF GOAL ALIGNING TEST

Models	Distance (m)	Trial			Success Rate
		1	*2*	*3*	
NAS	***1***	1	1	1	11/12 (91.7%)
	3	1	1	1	
	5	1	1	1	
	7	1	1	0	
V7	***1***	1	1	1	10/12 (83.3%)
	3	1	1	1	
	5	1	1	1	
	7	0	0	0	
V8	***1***	1	1	1	9/12 (75.0%)
	3	1	1	1	
	5	1	1	1	
	7	0	0	0	

The goal alignment measurement is limited to only 3 for each distance sample since it requires enabling motion that costs servo age. Using only 3 samples we accumulated the result and YOLO-NAS has the best performance against another model. Only YOLO-NAS that able to successfully perform the goal-aligning task even at 7 meters of distance. It only has a single error at 7 meters of distance test. YOLO-v8 and YOLOv7 successfully strike all tests with distances up to 5 meters. Fig. 11 shows the robot performing the kick that has already successfully detected and aligned itself facing the goal.

Fig. 10. Robot successfully performing ball aligning task

G. *Testing the Robot Collision Avoidance*

To ensure that our robot can adeptly avoid collisions with other robots on the field, thereby testing the efficiency of the deployed detection system. The robot avoidance mechanism discerns the positions of different robots concerning our robot's current position. A walking path is then generated to circumvent potential collisions with incoming robots. The successful execution of this task signifies that the robot detection system is operational and ready for utilization. Within this task, the robot must demonstrate the ability to evade collisions with other robots.

The robot collision avoidance task is rooted in robot detection. The test commences as the robot prepares to move forward. When another robot is detected in its path, our robot will adjust its course to the left or right, contingent on the position of the opposing robot. The result of this test is in Table V.

TABLE V. ROBOT COLLISION AVOIDANCE TEST RESULT

		Variants		
		NAS	***V7***	***V8***
Trials	***1***	1	0	1
	2	1	0	1
	3	1	1	1
	4	1	0	1
	5	1	1	1
	6	1	1	1
	7	1	1	1
	8	1	0	1
	9	1	0	1
	10	1	1	1

YOLO-NAS and YOLOv8 successfully finished all 12 trials of the robot avoidance task. YOLO-v7, which has low detection metrics, could not finish all the tests with six errors due to low detection performance on robot datasets.

H. Friedman's Two-Way Analysis of Variance by Ranks

Using Related-Samples Friedman's Two-Way analisys of variance by ranks, as shown in Table VI, the researchers measured if there were significant differences between the 3 models' results or not. The result shows that YOLOv7 has the best success rate compared to YOLO-NAS in the Ball tracking task. Against goal alignment task, our Friedman's analisys found that YOLO-NAS has superior performance, as shown in Table VII. Against the Robot avoidance task, both YOLO-NAS and YOLOv8 nailed the task. Only YOLOv7 that has 50% valid result as shown in Table VIII.

The "Sig." value (significance level) reported as "<.001" indicates that the p-value associated with the test statistic is less than 0.001. This means that the probability of observing the data, assuming that the null hypothesis is true and that the distributions are the same, is less than 0.001. We reject the null hypothesis since the significance level is set at 0.05, and the p-value is much smaller than 0.05. Table IX of the Hypothesis Test Summary shows sufficient evidence to conclude that the distributions of NAS, V7, and V8 are different. It is also important to mention that the p-value provided is an asymptotic approximation, which means it's valid for large sample sizes. However, exact significance may differ for smaller sample sizes.

TABLE VI. BALL TRACKING TEST ANALISYS

Model	Valid	Frequency	Valid Percent	Cumulative Percent
NAS	0	28	46.7	46.7
	1	32	53.3	100.0
V7	0	19	31.7	31.7
	1	41	68.3	100.0
V8	0	34	56.7	56.7
	1	26	43.3	100.0

TABLE VII. GOAL ALIGNMENT TEST ANALISYS

Model	Valid	Frequency	Valid Percent	Cumulative Percent
NAS	0	1	8.3	8.3
	1	11	91.7	100.0
V7	0	2	16.7	16.7
	1	10	83.3	100.0
V8	0	3	25.0	25.0
	1	9	75.0	100.0

TABLE VIII. ROBOT AVOIDANCE TEST ANALISYS

Model	Valid	Frequency	Valid Percent	Cumulative Percent
NAS	0	0	0	0
	1	10	100.0	100.0
V7	0	5	50.0	50.0
	1	5	50.0	100.0
V8	0	0	0	0
	1	10	100.0	100.0

TABLE IX. HYPOTHESYS TEST RESULTS

Null Hypothesis	Test	Sig.^{a,b}	Decision
The distributions of NAS, V7 and V8 are the same.	Ball Tracking	.001	Reject the null hypothesis.
	Goal Alignment	.223	Retain the null hypothesis.
	Robot Avoidance	.007	Reject the null hypothesis.

Using Pairwise Comparison in Table IX, using between two samples shows the comparison of each test result. The test statistic is a numerical value calculated from the data that quantifies the sample relationship. The standard error indicates how much the sample statistic is expected to vary from sample to sample. The significance value (often denoted as the p-value) indicates the probability of observing the observed test statistic if the null hypothesis were true. A low p-value suggests that the observed difference between samples or groups is unlikely to have occurred by random chance alone, leading to the rejection of the null hypothesis. Adjusted Significance (Adj. Sig.) refers to a corrected or adjusted significance value, possibly accounting for multiple comparisons or other factors by analyzing these fields in the pairwise comparison table, we can assess the differences or relationships between the samples or groups being compared and determine whether these differences are statistically significant or not.

From the Table X We can conclude that statistical data related to different tests, including "Ball Tracking," "Goal Alignment," and "Robot Avoidance," with comparisons between various sample pairs such as V8-NAS, V8-V7, and NAS-V7. The key parameters analysed are the test statistic, standard error, standard test statistic, significance (Sig.), and adjusted significance (Adj. Sig.). For instance, in the "Ball Tracking" test, the comparison between samples V8-V7 yielded a test statistic of 0.375, a standard error of 0.183, a standard test statistic of 2.054, a significance of 0.04, and an adjusted significance of 0.12. This data suggests a significant difference between these samples in the context of ball tracking.

In the "Goal Alignment" test, the comparison between samples V8-NAS produced a test statistic of 0.25, a standard error of 0.408, a standard test statistic of 0.612, a significance of 0.54, and an adjusted significance of 1. This indicates that there is no significant difference between these samples in terms of goal alignment. Similarly, the "Robot Avoidance" test shows varying results for different sample comparisons, such as V7-NAS, V7-V8, and NAS-V8, with corresponding test statistics, standard errors, standard test statistics, significances, and adjusted significances. These findings provide valuable insights into the performance of the samples across different tests, allowing for informed decision-making in the context of the respective applications.

TABLE X. PAIRWISE COMPARISON

Test	Sample 1-Sample 2	Test Statistic	Std. Error	Std. Test Statistic	Sig.	Adj. Sig.[a]
Ball Tracking	V8-NAS	.150	.183	.822	.411	1.000
	V8-V7	.375	.183	2.054	.040	.120
	NAS-V7	-.225	.183	-1.232	.218	.653
Goal Alignment	V8-V7	.125	.408	.306	.759	1.000
	V8-NAS	.250	.408	.612	.540	1.000
	V7-NAS	.125	.408	.306	.759	1.000
Robot Avoidance	V7-NAS	.750	.447	1.677	.094	.281
	V7-V8	-.750	.447	-1.677	.094	.281
	NAS-V8	.000	.447	.000	1.000	1.000

IV. CONCLUSION

Based on our test results, YOLO-NAS was able to be used for three tasks of humanoid robot soccer. In the Ball tracking task, YOLO-NAS achieved 53.3% success rate from 60 samples. In the goal alignment task, YOLO-NAS achieved 91.7% success rate. Finally, in the robot avoidance task, YOLO-NAS gained a success rate of 100% in 10 samples.

Based on the statistical data provided, we can conclude that in the "Ball Tracking" test, there is a significant difference between samples V8-V7, as indicated by the test statistic of 0.375 and a significance level of 0.04. However, there is no significant difference between samples V8-NAS and NAS-V7. No significant differences exist between the sample pairs for the "Goal Alignment" test, as all the test statistics have p-values above 0.05. The "Robot Avoidance" test shows a marginally significant difference between samples V7-NAS and V7-V8, with a test statistic of 0.75 and a significance level of 0.094. However, there is no significant difference between samples NAS-V8. In summary, the statistical analysis suggests that there are significant differences in the performance of the samples in the "Ball Tracking" and "Robot Avoidance" tests. In contrast, no significant differences were observed in the "Goal Alignment" test.

The limited data sample in this research stems from the constraints imposed by the actuators utilized in the humanoid robot soccer system. Due to these actuators' specific capabilities and limitations, the number of samples available for analysis was restricted. Consequently, the findings of this study may not fully capture the breadth of potential outcomes. It is recommended that future research endeavors expand the sample size to encompass a wider range of scenarios and conditions. By incorporating a more extensive dataset, researchers can achieve more robust and accurate results, thereby enhancing the reliability and validity of their findings in humanoid robot soccer.

Based on the insights and experiences garnered from our recent research, we have found some recommendations to guide and enhance future studies in this domain. As our existing hardware only supports FP-16 (Floating Point 16) quantization, future studies should consider employing newer versions of Jetson devices that can handle INT-8 (Integer 8) quantization, which could yield faster inference speeds. Our work only compares the performance of the models on training hardware (Nvidia RTX 3060) and deployment devices (Nvidia Jetson Nano 4 GB). Future research should use the newer hardware with extended computing capabilities for better detection performance. To improve the frame-per-second (FPS) rate, future work should explore the utilization of C++ for interfacing with the GPU, as opposed to the Python-based pipeline used in our research.

ACKNOWLEDGMENT

The completion of this research was made possible through the utilization of the Universitas Negeri Yogyakarta Humanoid Robot Soccer Team's robot.

REFERENCES

[1] D. Zhou, G. Chen, and F. Xu, "Application of Deep Learning Technology in Strength Training of Football Players and Field Line Detection of Football Robots," *Front. Neurorobot.*, vol. 16, p. 867028, Jun. 2022, doi: 10.3389/fnbot.2022.867028.

[2] P. Stone, "Designing Robots to Best World Cup Winners has Inspired Generations of Roboticists," *IEEE Spectrum*, vol. 60, no. 7, pp. 40–49, Jul. 2023, doi: 10.1109/MSPEC.2023.10177053.

[3] D. C. Melo, M. R. O. A. Maximo, and A. M. da Cunha, "Learning Push Recovery Behaviors for Humanoid Walking Using Deep Reinforcement Learning," *J. Intell Robot Syst.*, vol. 106, no. 1, p. 8, Aug. 2022, doi: 10.1007/s10846-022-01656-7.

[4] A. Risnumawan, "Parameter Adjustment of EROS Humanoid Robot Soccer using a Motion Visualization," *Complete*, vol. 3, no. 1, Jul. 2022, doi: 10.52435/complete.v2i1.203.

[5] R. Jánoš *et al.*, "Stability and Dynamic Walk Control of Humanoid Robot for Robot Soccer Player," *Machines*, vol. 10, no. 6, Jun. 2022, doi: 10.3390/machines10060463.

[6] C.-C. Liu, Y.-C. Lin, W.-F. Lai, and C.-C. Wong, "Kicking Motion Planning of Humanoid Robot Based on B-Spline Curves," *J. Appl. Sci. Eng.*, vol. 25, no. 4, pp. 723–731, 2021, doi: 10.6180/jase.202208_25(4).0007.

[7] A. Rezaeipanah, Z. Jamshidi, and S. Jafari, "A Shooting Strategy When Moving On Humanoid Robots Using Inverse Kinematics And Q-Learning," *International Journal of Robotics and Automation*, vol. 36, p. 2020, Nov. 2020, doi: 10.2316/J.2021.206-0393.

[8] A. F. V. Muzio, M. R. O. A. Maximo, and T. Yoneyama, "Deep Reinforcement Learning for Humanoid Robot Behaviors," *J. Intell Robot Syst.*, vol. 105, no. 1, p. 12, Apr. 2022, doi: 10.1007/s10846-022-01619-y.

[9] X. Ming, X. Nanfeng, Z. Mengjun, and Y. Qunyong, "Optimized Convolutional Neural Network-Based Object Recognition for Humanoid Robot," *J. Robotics Autom.*, vol. 4, no. 1, Feb. 2020, doi: 10.36959/673/363.

[10] Q. Bai, S. Li, J. Yang, Q. Song, Z. Li, and X. Zhang, "Object Detection Recognition and Robot Grasping Based on Machine Learning: A Survey," *IEEE Access*, vol. 8, pp. 181855–181879, 2020, doi: 10.1109/ACCESS.2020.3028740.

[11] E. Maiettini, G. Pasquale, L. Rosasco, and L. Natale, "On-line object detection: a robotics challenge," *Auton. Robot.*, vol. 44, no. 5, pp. 739–757, May 2020, doi: 10.1007/s10514-019-09894-9.

[12] A. Rizgi *et al.*, "Visual Perception System of EROS Humanoid Robot Soccer," *International Journal of Intelligent Information Technologies*, vol. 16, pp. 68–86, Oct. 2020, doi: 10.4018/IJIIT.2020100105.

[13] M. Abreu, T. Silva, H. Teixeira, L. P. Reis, and N. Lau, "6D localization and kicking for humanoid robotic soccer," *Journal of Intelligent & Robotic Systems*, vol. 102, no. 2, p. 30, 2021.

[14] I. Da Silva, D. Perico, T. Homem, and R. Bianchi, “Deep Reinforcement Learning for a Humanoid Robot Soccer Player,” *Journal of Intelligent & Robotic Systems*, vol. 102, p. 69, Jul. 2021, doi: 10.1007/s10846-021-01333-1.

[15] C. Hong, I. Jeong, L. F. Vecchietti, D. Har, and J.-H. Kim, “AI World Cup: Robot-Soccer-Based Competitions,” *IEEE Transactions on Games*, vol. 13, no. 4, pp. 330–341, Dec. 2021, doi: 10.1109/TG.2021.3065410.

[16] P.-H. Kuo, W.-C. Yang, P.-W. Hsu, and K.-L. Chen, “Intelligent proximal-policy-optimization-based decision-making system for humanoid robots,” *Advanced Engineering Informatics*, vol. 56, p. 102009, Apr. 2023, doi: 10.1016/j.aei.2023.102009.

[17] E. Antonioni, V. Suriani, F. Riccio, and D. Nardi, “Game Strategies for Physical Robot Soccer Players: A Survey,” *IEEE Trans. Games*, vol. 13, no. 4, pp. 342–357, Dec. 2021, doi: 10.1109/TG.2021.3075065.

[18] D. Steffi, S. Mehta, and V. Venkatesh, “Object detection on robosoccer environment using convolution neural network,” *IJEECS*, vol. 29, no. 1, p. 286, Jan. 2022, doi: 10.11591/ijeecs.v29.i1.pp286-294.

[19] E. R. Jamzuri, H. Mandala, and J. Baltes, “A Fast and Accurate Object Detection Algorithm on Humanoid Marathon Robot,” *Indonesian Journal of Electrical Engineering and Informatics (IJEEI)*, vol. 8, no. 1, Art. no. 1, Mar. 2020, doi: 10.52549/ijeei.v8i1.1960.

[20] N. Cruz, F. Leiva, and J. Ruiz-del-Solar, “Deep learning applied to humanoid soccer robotics: playing without using any color information,” *Auton. Robot.*, vol. 45, no. 3, pp. 335–350, Mar. 2021, doi: 10.1007/s10514-021-09966-9.

[21] S. A. Irfan and N. S. Widodo, “Application of Deep Learning Convolution Neural Network Method on KRSBI Humanoid R-SCUAD Robot,” *Bul. Il. Sar. TE.*, vol. 2, no. 1, p. 40, May 2020, doi: 10.12928/biste.v2i1.985.

[22] S. N. Aslan, A. Uçar, and C. Güzeliş, “New convolutional neural network models for efficient object recognition with humanoid robots,” *Journal of Information and Telecommunication*, vol. 6, no. 1, pp. 63–82, Jan. 2022, doi: 10.1080/24751839.2021.1983331.

[23] M. Bestmann *et al.*, “TORSO-21 Dataset: Typical Objects in RoboCup Soccer 2021,” in *RoboCup 2021: Robot World Cup XXIV*, vol. 13132, pp. 65–77, 2022, doi: 10.1007/978-3-030-98682-7_6.

[24] B. Tan, “Soccer-Assisted Training Robot Based on Image Recognition Omnidirectional Movement,” *Wireless Communications and Mobile Computing*, vol. 2021, pp. 1–10, Aug. 2021, doi: 10.1155/2021/5532210.

[25] R. Dikairono, S. Setiawardhana, D. Purwanto, and T. A. Sardjono, “CNN-Based Self Localization Using Visual Modelling of a Gyrocompass Line Mark and Omni-Vision Image for a Wheeled Soccer Robot Application,” *International Journal of Intelligent Engineering and Systems*, vol. 13, no. 6, pp. 442–453, 2020, doi: 10.22266/ijies2020.1231.39.

[26] D. D. R. Meneghetti, T. P. D. Homem, J. H. R. De Oliveira, I. J. D. Silva, D. H. Perico, and R. A. D. C. Bianchi, “Detecting Soccer Balls with Reduced Neural Networks: A Comparison of Multiple Architectures Under Constrained Hardware Scenarios,” *J. Intell Robot Syst.*, vol. 101, no. 3, p. 53, Mar. 2021, doi: 10.1007/s10846-021-01336-y.

[27] D. Lim, J. Kim, and H. Kim, “Efficient robot tracking system using single-image-based object detection and position estimation,” *ICT Express*, vol. 10, no. 1, pp. 125–131, Feb. 2024, doi: 10.1016/j.icte.2023.07.009.

[28] Q. Yan, S. Li, C. Liu, M. Liu, and Q. Chen, “RoboSeg: Real-Time Semantic Segmentation on Computationally Constrained Robots,” *IEEE Transactions on Systems, Man, and Cybernetics: Systems*, vol. 52, no. 3, pp. 1567–1577, Mar. 2022, doi: 10.1109/TSMC.2020.3032437.

[29] A. Yildiz, N. G. Adar, and A. Mert, “Convolutional Neural Network Based Hand Gesture Recognition in Sophisticated Background for Humanoid Robot Control,” *IAJIT*, vol. 20, no. 3, 2023, doi: 10.34028/iajit/20/3/9.

[30] J. G. Melo and E. Barros, “An Embedded Monocular Vision Approach for Ground-Aware Objects Detection and Position Estimation,” *Robot World Cup*, pp. 100-111, 2022.

[31] C. O. Yinka-Banjo, O. A. Ugot, and E. Ehiorobo, “Object detection for robot coordination in robotics soccer,” *Nig. J. Technol. Dev.*, vol. 19, no. 2, pp. 136–142, Aug. 2022, doi: 10.4314/njtd.v19i2.5.

[32] 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, p. 012055, Nov. 2021, doi: 10.1088/1742-6596/2111/1/012055.

[33] S. K. Narayanaswami *et al.*, “Towards a Real-Time, Low-Resource, End-to-End Object Detection Pipeline for Robot Soccer,” in *RoboCup 2022: Robot World Cup XXV*, pp. 62–74, Mar. 2023, doi: 10.1007/978-3-031-28469-4_6.

[34] D. Barry, M. Shah, M. Keijsers, H. Khan, and B. Hopman, "xYOLO: A Model For Real-Time Object Detection In Humanoid Soccer On Low-End Hardware," *2019 International Conference on Image and Vision Computing New Zealand (IVCNZ)*, pp. 1-6, 2019, doi: 10.1109/IVCNZ48456.2019.8960963.

[35] R. Nie, “Research of target detection method YOLO,” *ACE*, vol. 6, no. 1, pp. 620–632, Jun. 2023, doi: 10.54254/2755-2721/6/20230905.

[36] Y. Zhang, “YOLO Series Target Detection Technology and Application,” *Highlights in Science, Engineering and Technology*, vol. 39, pp. 841–847, Apr. 2023, doi: 10.54097/hset.v39i.6653.

[37] M. Hussain, “YOLO-v1 to YOLO-v8, the Rise of YOLO and Its Complementary Nature toward Digital Manufacturing and Industrial Defect Detection,” *Machines*, vol. 11, no. 7, p. 677, Jun. 2023, doi: 10.3390/machines11070677.

[38] O. M. Moradeyo, A. S. Olaniyan, A. O. Ojoawo, J. A. Olawale, and R. W. Bello, “YOLOv7 Applied to Livestock Image Detection and Segmentation Tasks in Cattle Grazing Behavior, Monitor and Intrusions,” *jasem*, vol. 27, no. 5, pp. 953–958, May 2023, doi: 10.4314/jasem.v27i5.10.

[39] E. R. Jamzuri *et al.*, “Barelang FC - Team Description Paper Humanoid Kid-Size League RoboCup 2023 France,” *robocuphumanoid*, 2023, doi: 10.13140/RG.2.2.30653.23527.

[40] J. Chen *et al.*, “Efficient and lightweight grape and picking point synchronous detection model based on key point detection,” *Computers and Electronics in Agriculture*, vol. 217, p. 108612, Feb. 2024, doi: 10.1016/j.compag.2024.108612.

[41] P. Yan, W. Wang, G. Li, Y. Zhao, J. Wang, and Z. Wen, “A lightweight coal gangue detection method based on multispectral imaging and enhanced YOLOv8n,” *Microchemical Journal*, vol. 199, p. 110142, Apr. 2024, doi: 10.1016/j.microc.2024.110142.

[42] A. Saadeldin, M. M. Rashid, A. A. Shafie, and T. F. Hasan, “Real-time vehicle counting using custom YOLOv8n and DeepSORT for resource-limited edge devices,” *TELKOMNIKA (Telecommunication Computing Electronics and Control)*, vol. 22, no. 1, Feb. 2024, doi: 10.12928/telkomnika.v22i1.25096.

[43] H. Wang, L. Fu, and L. Wang, “Detection algorithm of aircraft skin defects based on improved YOLOv8n,” *Signal, Image and Video Processing,* pp. 1-15, Feb. 2024, doi: 10.1007/s11760-024-03049-9.

[44] Y. Fan, L. Zhang, C. Zheng, Y. Zu, X. Wang, and J. Zhu, “Real-time and accurate meal detection for meal-assisting robots,” *Journal of Food Engineering*, vol. 371, p. 111996, Jun. 2024, doi: 10.1016/j.jfoodeng.2024.111996.

[45] K. Chen, B. Du, Y. Wang, G. Wang, and J. He, “The real-time detection method for coal gangue based on YOLOv8s-GSC,” *J. Real-Time Image Proc.*, vol. 21, no. 2, p. 37, Feb. 2024, doi: 10.1007/s11554-024-01425-9.

[46] M. Zhao *et al.*, “MED-YOLOv8s: a new real-time road crack, pothole, and patch detection model,” *J. Real-Time Image Proc.*, vol. 21, no. 2, p. 26, Jan. 2024, doi: 10.1007/s11554-023-01405-5.

[47] M. A. Soeleman, C. Supriyanto, and Purwanto, “Deep Learning Model for Unmanned Aerial Vehicle-based Object Detection on Thermal Images,” *RIA*, vol. 37, no. 6, pp. 1441–1447, Dec. 2023, doi: 10.18280/ria.370608.

[48] S. M. H. Rizvi, A. Naseer, S. U. Rehman, S. Akram, and V. Gruhn, “Revolutionizing Agriculture: Machine and Deep Learning Solutions for Enhanced Crop Quality and Weed Control,” *IEEE Access*, vol. 12, pp. 11865–11878, 2024, doi: 10.1109/ACCESS.2024.3355017.

[49] J. Farooq, M. Muaz, K. Khan Jadoon, N. Aafaq, and M. K. A. Khan, “An improved YOLOv8 for foreign object debris detection with optimized architecture for small objects,” *Multimed. Tools Appl.*, pp. 1-27, Dec. 2023, doi: 10.1007/s11042-023-17838-w.

[50] A. M. Qasim, N. Abbas, A. Ali, and B. A. A.-R. Al-Ghamdi, “Abandoned Object Detection and Classification Using Deep

Embedded Vision," *IEEE Access*, vol. 12, pp. 35539–35551, 2024, doi: 10.1109/ACCESS.2024.3369233.

[51] C. M. Badgujar, P. R. Armstrong, A. R. Gerken, L. O. Pordesimo, and J. F. Campbell, "Real-time stored product insect detection and identification using deep learning: System integration and extensibility to mobile platforms," *Journal of Stored Products Research*, vol. 104, p. 102196, Dec. 2023, doi: 10.1016/j.jspr.2023.102196.

[52] H. Duong-Trung and N. Duong-Trung, "Integrating YOLOv8-agri and DeepSORT for Advanced Motion Detection in Agriculture and Fisheries," *EAI Endorsed Transactions on Industrial Networks and Intelligent Systems*, vol. 11, no. 1, pp. e4–e4, Feb. 2024, doi: 10.4108/eetinis.v11i1.4618.

[53] X. Zhao, Y. He, H. Zhang, Z. Ding, C. Zhou, and K. Zhang, "A quality grade classification method for fresh tea leaves based on an improved YOLOv8x-SPPCSPC-CBAM model," *Sci. Rep.*, vol. 14, p. 4166, Feb. 2024, doi: 10.1038/s41598-024-54389-y.

[54] N. Hnoohom, P. Chotivatunyu, N. Maitrichit, C. Nilsumrit, and P. Iamtrakul, "The video-based safety methodology for pedestrian crosswalk safety measured: The case of Thammasat University, Thailand," *Transportation Research Interdisciplinary Perspectives*, vol. 24, p. 101036, Feb. 2024, doi: 10.1016/j.trip.2024.101036.

[55] N. Zou, Q. Xu, Y. Wu, X. Zhu, and Y. Su, "An Automated Method for Generating Prefabs of AR Map Point Symbols Based on Object Detection Model," *ISPRS International Journal of Geo-Information*, vol. 12, no. 11, Nov. 2023, doi: 10.3390/ijgi12110440.

[56] W. Yuan, "AriAplBud: An Aerial Multi-Growth Stage Apple Flower Bud Dataset for Agricultural Object Detection Benchmarking," *Data*, vol. 9, no. 2, p. 36, Feb. 2024, doi: 10.3390/data9020036.

[57] R. M. Sampurno, Z. Liu, R. M. R. D. Abeyrathna, and T. Ahamed, "Intrarow Uncut Weed Detection Using You-Only-Look-Once Instance Segmentation for Orchard Plantations," *Sensors*, vol. 24, no. 3, Jan. 2024, doi: 10.3390/s24030893.

[58] H. Boudlal, M. Serrhini, and A. Tahiri, "A novel approach for simultaneous human activity recognition and pose estimation via skeleton-based leveraging WiFi CSI with YOLOv8 and mediapipe frameworks," *Signal, Image and Video Processing*, pp. 1-17, Feb. 2024, doi: 10.1007/s11760-024-03031-5.

[59] K. Qi, W. Xu, W. Chen, X. Tao, and P. Chen, "Multiple object tracking with segmentation and interactive multiple model," *Journal of Visual Communication and Image Representation*, vol. 99, p. 104064, Mar. 2024, doi: 10.1016/j.jvcir.2024.104064.

[60] M. Sukkar, M. Shukla, D. Kumar, V. C. Gerogiannis, A. Kanavos, and B. Acharya, "Enhancing Pedestrian Tracking in Autonomous Vehicles by Using Advanced Deep Learning Techniques," *Information*, vol. 15, no. 2, Feb. 2024, doi: 10.3390/info15020104.

[61] W. Sheng, J. Shen, Q. Huang, Z. Liu, and Z. Ding, "Multi-objective pedestrian tracking method based on YOLOv8 and improved DeepSORT," *MBE*, vol. 21, no. 2, pp. 1791–1805, 2024, doi: 10.3934/mbe.2024077.

[62] J. Terven, D.-M. Córdova-Esparza, and J.-A. Romero-González, "A Comprehensive Review of YOLO Architectures in Computer Vision: From YOLOv1 to YOLOv8 and YOLO-NAS," *MAKE*, vol. 5, no. 4, pp. 1680–1716, Nov. 2023, doi: 10.3390/make5040083.

[63] T. Sharma *et al.*, "Deep Learning-Based Object Detection and Classification for Autonomous Vehicles in Different Weather Scenarios of Quebec, Canada," *IEEE Access*, vol. 12, pp. 13648–13662, 2024, doi: 10.1109/ACCESS.2024.3354076.

[64] Z. Gao, J. Huang, J. Chen, T. Shao, H. Ni, and H. Cai, "Deep transfer learning-based computer vision for real-time harvest period classification and impurity detection of Porphyra haitnensis," *Aquacult. Int.*, pp. 1-28, Feb. 2024, doi: 10.1007/s10499-024-01422-6.

[65] A. D. Dobrzycki, A. M. Bernardos, L. Bergesio, A. Pomirski, and D. Sáez-Trigueros, "Exploring the Use of Contrastive Language-Image Pre-Training for Human Posture Classification: Insights from Yoga Pose Analysis," *Mathematics*, vol. 12, no. 1, Jan. 2024, doi: 10.3390/math12010076.

[66] S. Chen, Y. Li, Y. Zhang, Y. Yang, and X. Zhang, "Soft X-ray image recognition and classification of maize seed cracks based on image enhancement and optimized YOLOv8 model," *Computers and Electronics in Agriculture*, vol. 216, p. 108475, Jan. 2024, doi: 10.1016/j.compag.2023.108475.

[67] S. Saluky, G. B. Nugraha, and S. H. Supangkat, "Enhancing Abandoned Object Detection with Dual Background Models and Yolo-NAS," *International Journal of Intelligent Systems and Applications in Engineering*, vol. 12, no. 2, 2024.

[68] H. Slimani, J. El Mhamdi, and A. Jilbab, "Advancing disease identification in fava bean crops: A novel deep learning solution integrating YOLO-NAS for precise rust," *Journal of Intelligent & Fuzzy Systems*, vol. 46, no. 2, pp. 3475–3489, Jan. 2024, doi: 10.3233/JIFS-236154.

[69] R. N. Anand, R. P. Singh, D. Gupta, and K. Palaniappan, "Ship Detection in Satellite Images using You Only Look Once-Neural Architecture Search," in *2023 9th International Conference on Signal Processing and Communication (ICSC)*, pp. 463–468, Dec. 2023, doi: 10.1109/ICSC60394.2023.10441207.

[70] K. Charoenjai, W. Kusakunniran, T. Thaipisutikul, N. Yodrabum, and I. Chaikangwan, "Automatic detection of nostril and key markers in images," *Intelligent Systems with Applications*, vol. 21, p. 200327, Mar. 2024, doi: 10.1016/j.iswa.2024.200327.

[71] F. Zhengxin *et al.*, "MLOps Spanning Whole Machine Learning Life Cycle: A Survey," *arXiv preprint arXiv:2304.07296*, 2023, doi: 10.48550/arXiv.2304.07296.