

Tracking Ball Using YOLOv8 Method on Wheeled Soccer Robot with Omnidirectional Camera

Refli Rezka Julianda, Riky Dwi Puriyanto

Department of Electrical Engineering, Universitas Ahmad Dahlan, Yogyakarta, Indonesia

ARTICLE INFORMATION

Article History:

Received 15 June 2024
Revised 26 June 2024
Published 07 September 2024

Keywords:

Deep Learning;
YOLO;
YOLOv8;
Tracking Ball;
Omnidirectional

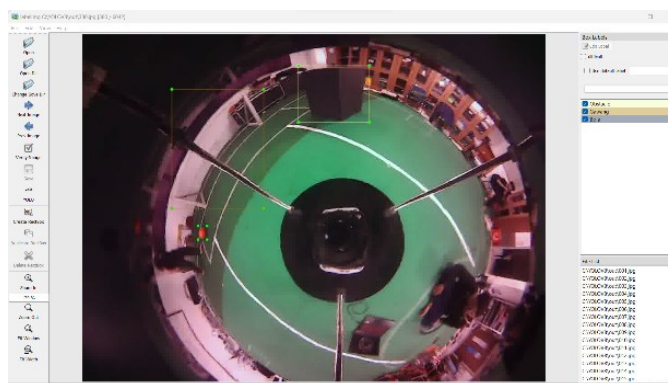
Corresponding Author:

Riky Dwi Puriyanto
Department of Electrical
Engineering, Universitas
Ahmad Dahlan, Yogyakarta,
Indonesia.
Email: rikypd@ee.uad.ac.id

This work is licensed under a [Creative Commons Attribution-Share Alike 4.0](https://creativecommons.org/licenses/by-sa/4.0/)



ABSTRACT



Object detection is very we often find in everyday life that facilitates every activities in the object recognition process, for example in the military field, intelligent transportation, face detection, robotics, and others. Detection target detection is one of the hotspots of research in the field of computer vision. The location and category of the target can be determined by using target detection. Currently, target detection has been applied in many fields, one of which includes image segmentation. You only look once (YOLO) is an algorithm that can perform object detection in realtime, YOLO itself always gets development and improvement from previous versions. YOLOv8 is a type of YOLO from the latest version. YOLOv8 is a new implementation of Deep Learning that connects the input (original image) with the output. This type of YOLOv8 algorithm uses A deep dive architecture, assisted by CNN and a new backbone which uses convolutional layers for pixels which when described will be shaped like a pyramid. YOLOv8 is a stable object detection processing method with 80% higher than the previous version of YOLO, which makes YOLOv8 a type of YOLO that is better at processing object data faster and more efficiently in Real-Time. The camera with omnidirectional system is able to detect spherical objects and other objects using the YOLOv8 model used. In performance testing with 320×320 and 416×416 frames, because it fits the grid structure of the YOLO architecture. YOLOv8 has a higher mAP value with a value of 95,5% compared to previous versions of YOLO. In the detection test, YOLOv8 has a better average object detection than the previous version of YOLO which is indicated by the number of objects detected more stable.

Document Citation:

R. R. Julianda and R. D. Puriyanto, "Tracking Ball Using YOLOv8 Method on Wheeled Soccer Robot with Omnidirectional," *Buletin Ilmiah Sarjana Teknik Elektro*, vol. 6, no. 2, pp. 203-213, 2024, DOI: [10.12928/biste.v6i2.10816](https://doi.org/10.12928/biste.v6i2.10816).

1. INTRODUCTION

Object detection is very important nowadays most of people using it everyday for their life, the application of object detection has been along known and is growing rapidly in several developed countries, in order to optimize technology in their countries [1]. Object detection capabilities have an important role in various daily activities, such as in military activities, education, traffic and others, object detection used to overcome problems in maximizing tasks that require optimal data details on activities carried out directly in real time [2]. Wheeled soccer robots are robots that have the ability to dribble, kick and pass the ball [3]. Wheeled soccer robots have various supporting components to perform and maximize their abilities, such as DC motors, arduino, solenoids, and cameras. The use of object detection is one of the things that is very important and necessary in wheeled soccer robots [4].

Wheeled soccer robots need the ability to see objects that will be used to dribble, kick, and pass the ball, therefore, robots need a sense of sight instead of a pair of eyes, to carry out their duties [5]. To overcome this, wheeled soccer robots use cameras as a sense of vision [6]. Omnidirectional cameras is a type of camera that uses a convex reflecting mirror to provide a reflection towards the camera so that it can see its surroundings 360° [7] are used on wheeled soccer robots as a sense of vision [8]. The omnidirectional camera is a tool that helps wheeled soccer robots detect objects [9].

Object detection has an important function in wheeled soccer robots [10] where object detection is the main benchmark in being able to see objects such as balls [11]. Object detection problems [12][13]. on wheeled soccer robots have many problems in object detection, such as errors in detecting objects, which are caused by the type of color, shape, and texture of objects [14], the percentage of object recognition [15], the number of objects detected, and various types of noise that cause errors and decrease fps in the object detection process [16].

The problem of object detection in wheeled soccer robots occurs because it is difficult for robots to recognize objects such as balls, goals, obstacles, and other robots which are in the light contrast description, and the distance is quite far to be captured by ordinary cameras [17]. To overcome this, wheeled soccer robots use omnidirectional cameras to detect objects widely with a 360° viewing distance, which by using an omnidirectional camera will expand the view of the wheeled soccer robot to detect specified objects [18], which can overcome the problem of visibility in object detection [19].

2. METHODS

In previous research, there are many things that can be done to detect objects with shape, color, motion, and texture. With the advancement of time and civilization, object detection began to develop towards artificial nerves such as using machine learning. Machine learning has various developmental algorithms in it where many stages are brought by previous research into their research in object detection, namely deep learning.

Deep learning is an artificial nerve that is used in detecting objects and classifying objects in an image, methods that use developments from deep learning such as Convolutional Neural Networks (CNN), Region-Based Convolutional Neural Network (RCNN), fast Region-Based Convolutional Neural Network (fast R-CNN), and also You Only Look Once (YOLO), where these methods can detect objects in real-time [20].

In this research, we will test the wheeled soccer robot whether it is able to detect balls, goals or other objects on the field using the You Only Look Once (YOLO) method. The robot used in this research is the Ahmad Dahlan University wheeled soccer robot using an omnidirectional camera in taking images or images as a dataset, which is a technique that displays images with a range of views based on various directions, be it the front direction, the back direction, the right side, also the left side.

Choosing YOLO (You Only Look Once) as an object detection architecture compared to other architectures such as the conventional CNN (Convolutional Neural Network) or other detection methods is due to some unique advantages possessed by YOLO, such as speed and efficiency, high detection accuracy compared to other algorithms [21].

2.1. Design System

The design of the ball tracking system using the YOLOv8 method in this study is described in the system block diagram, image training process flow chart, and system flow chart. To get the best results and as expected, the design of this system refers to theories and datasheets that have been reviewed from various sources.

In general, the system consists of 3 component parts: input, process, and output. The input component of the system is an omnidirectional camera. In the process component, the microcontroller used as a data processor and sender is Arduino Due. Furthermore, the output component in this system consists of a laptop. The block diagram of the measuring instrument system in this study is shown in Figure 1.

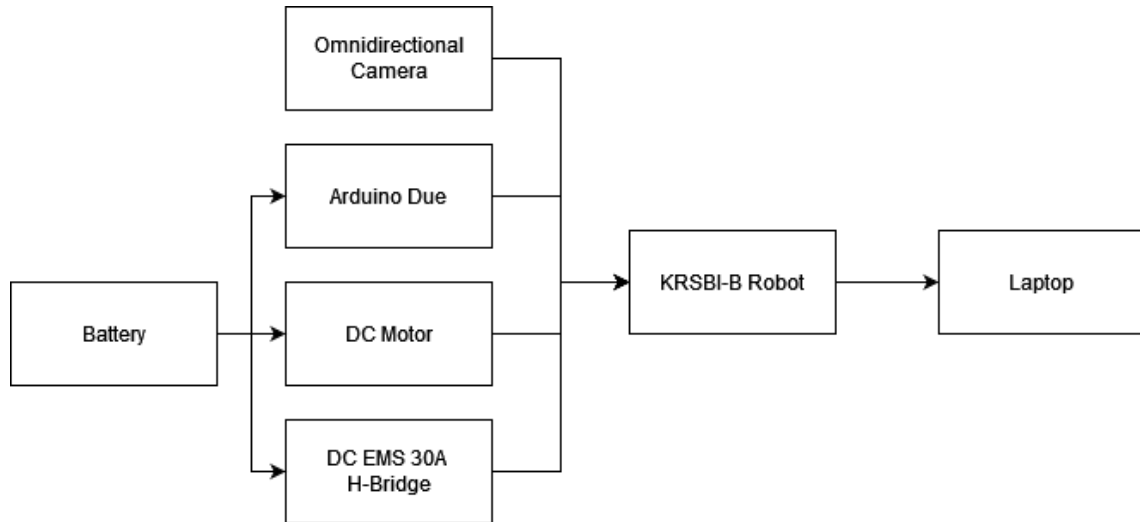


Figure 1. Block diagram design

2.2. Flowchart

In this study, initial testing will be carried out by detecting two different types of objects. Testing in this research this time the object detected is an orange ball and a white goal which will then be carried out the detection process on a green field. To be able to detect an object such as a ball and goal, this research uses a laptop connected to an omnidirectional camera that uses a Universal Serial Bus (USB) 2.0 port, then the results obtained will be processed using the YOLO library. In the process of implementing images in the software, this research must first conduct image data training using the library. The flowchart of the image training process can be seen in Figure 2 and the system flowchart can be seen in Figure 3.

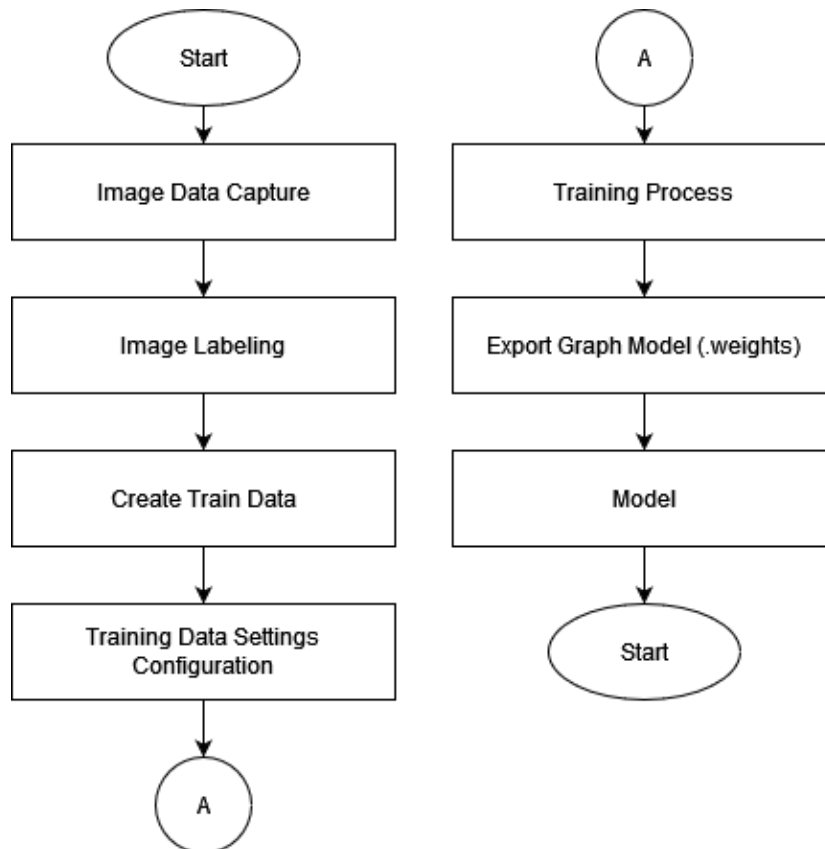


Figure 2. Flowchart of image training process

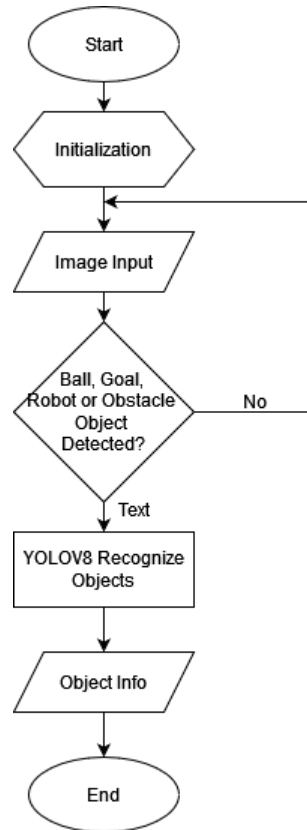


Figure 3. Flowchart system

2.3. Architecture YOLOv8

YOLOv8 algorithm uses a deep dive architecture, assisted by a CNN and a new backbone which uses convolutional layers for the pixels which when described will form a pyramid, can be seen in [Figure 4](#).

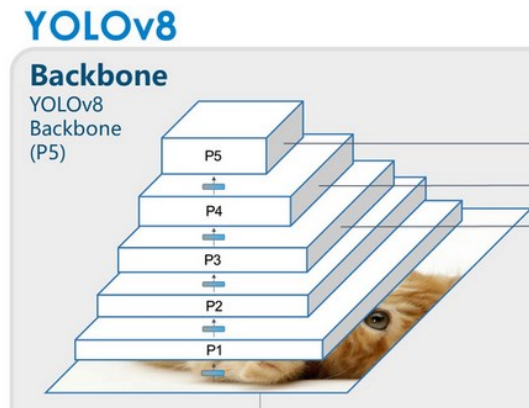


Figure 4. Architecture YOLOv8

Based on Figure 4 each layer has its own function, which is as follows:

- P1 is the first layer of the backbone, which functions to detect basic features of the input image such as edges, corners, and texture. It usually consists of several initial convolution layers followed by an activation function such as ReLU.
- P2 is the second layer of the backbone, which works to detect more complex features than P1, such as simple patterns and structures. Deeper than P1, it may include more convolution and pooling layers to reduce resolution while enhancing features.
- P3 is the third layer of the backbone, extracting features higher up in the abstraction hierarchy, such as object parts. Deeper convolution layers with more filters, as well as pooling operations to further reduce the resolution of the feature map.

- P4 is the fourth layer of the backbone, which serves to extract highly complex and abstract features that describe the objects in the image more thoroughly. It usually consists of several very deep and complex convolution blocks, designed to capture extensive spatial and contextual information.
- P5 is the fifth and final layer of the backbone, capturing the most abstract features that represent the image at the highest level. The convolution and pooling layers are the deepest and most complex, providing a feature map with very low resolution but with very rich and detailed feature information.

3. RESULT AND DISCUSSION

In this research, image capture, image labeling, training dataset results and model implementation were obtained.

3.1. Picture Taking

The images taken are images of orange balls, white goals, and black obstacles or robots with a green field background color that will be used as material for training. Taking pictures is done on the 3rd floor robotics lab room in the Ahmad Dahlan University lab building using an ELP USB-FHD08S camera that has an omnidirectional lens installed on it then the image is stored on a computer. The following are the results of images taken using an omnidirectional camera can be seen in [Figure 5](#).

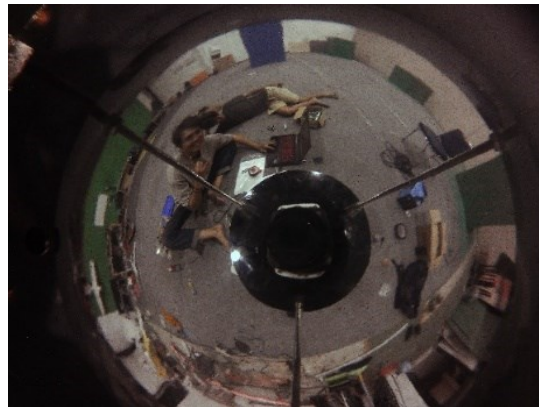


Figure 5. Image capture of an omnidirectional camera

3.2. Image Labeling

The data labeling process is a process where when the data that has been collected becomes one for labeling each object that will be learned by the YOLOv8 algorithm where each object at the time of labeling must adjust all possibilities as learning for blind spots or blur of an object. The image labeling process in this research uses the labeling method which is called through the anaconda prompt. The data labeling process can be seen in [Figure 6](#). Based on [Figure 6](#) is the process of labeling data using labellingm called through anaconda prompt, the drawback of using labellingm at anaconda prompt is that the computer/laptop used to do labeling must not run out of power or run out of battery.

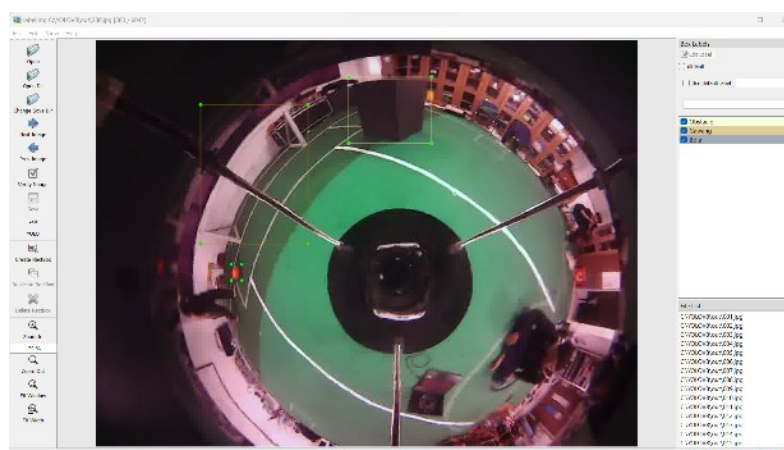


Figure 6. Image labelling process

3.3. Training Model

The model training process in YOLOv8 is the same as in the previous YOLO but the procession of index images, data classes, and model calls is more complex than the previous YOLO version. In this study, the training data model for YOLOv8 uses google collab where the number of epochs is 100 with image labeling totaling 2100 data labels and 2100 images, with a total background of 6042 false data and takes about 3 hours, can be seen in [Table 1](#).

Table 1. Validation dataset sharing

Object	Number of datasets
Ball	300
Goal	300
Robot/Obstacle	300
Ball and Goal	400
Ball and Robot/Obstacle	400
Goal and Robot/Obstacle	400
Total	2100

Based on [Table 1](#), the dataset above will be tested using the YOLOv8 model which produces experimental data, which will later be used for training data test, can be seen in [Table 2](#).

Table 2. Test results on the model

Confusion Matrix	YOLOv8		
	Ball	Goal	Robot/Obstacle
TP	830	708	530
FP	29	133	78
TN	125	125	125
FN	80	82	93

Based on [Table 2](#), True Positive (TP) value gets the final result YOLOv8 with a value of 830 for the ball, 708 for the goal, 530 for the robot/obstacle. On the False Positive (FP) value gets the final result YOLOv8 with a value of 29 for the ball, 133 for the goal, 78 for the robot/obstacle. While the True Negative (TN) value gets the final result YOLOv8 with a value of 125 for the ball, 125 for the goal, 125 for the robot / obstacle. Furthermore, the False Negative (FN) value gets the final result of YOLOv8 with a value of 80 for the ball, 82 for the goal, 93 for the robot/obstacle. And the final results of the YOLOv8 dataset training process can be seen in [Figure 7](#).

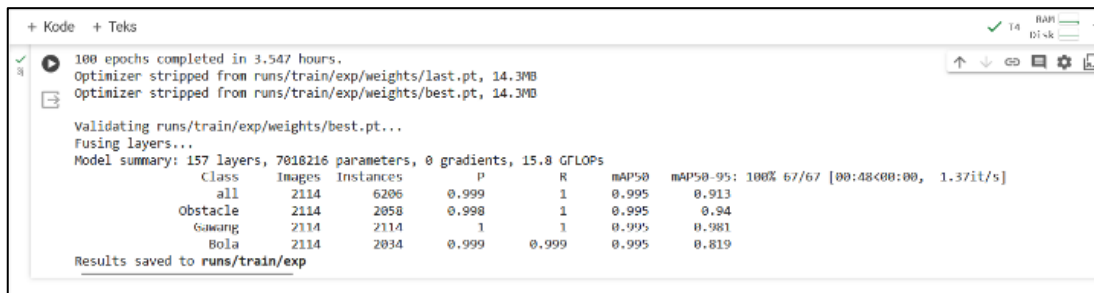


Figure 7. Final result of training dataset

3.4. Results

The results of the dataset training process can be seen in the runs folder, in the runs folder there are many arrays of data results that have been obtained after going through the dataset training process on googlecollab, such as the weights folder which contains last.pt and best.pt which are used as libraries to detect objects that have been trained, in other folders there are assessment curves from the final results of training datasets such as confusion_matrix, F1_curve, and Result_curve. The implementation process of modeling results with ball and obstacle objects can be seen in [Figure 8](#), [Figure 9](#), and [Figure 10](#). In [Figure 8](#) the omnidirectional camera successfully detects the ball object with a value of 0.92 to 1, without the interference of other objects in the area with the state of the orange ball used as object detection on a green carpet to avoid color contrast in the background when object detection takes place in real-time.

In [Figure 9](#) the omnidirectional camera successfully detects obstacle objects with a value of 0.84 to 1, without the interference of other objects in the area with black obstacles used as object detection on a green carpet to avoid color contrast in the background when object detection takes place in real-time.

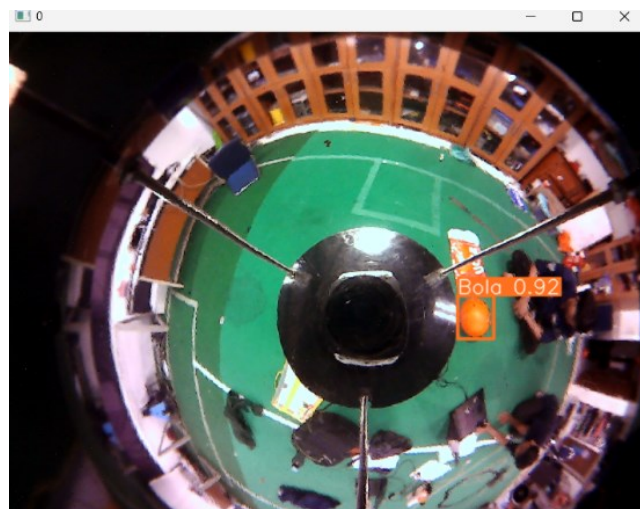


Figure 8. Implementation of modeling results with ball

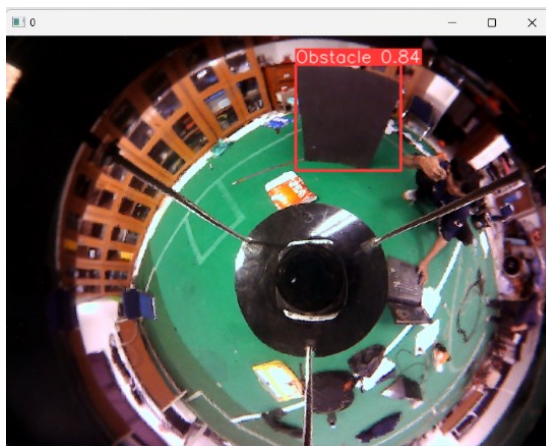


Figure 9. Implementation of modeling results with obstacle

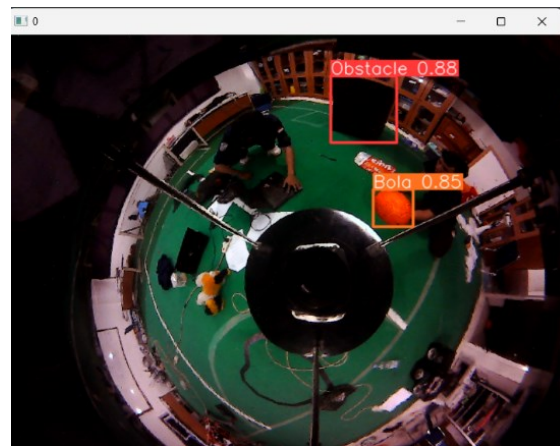


Figure 10. Implementation of modeling results with two different objects

In [Figure 10](#) the omnidirectional camera successfully detects 2 different objects, namely the ball and the obstacle at the same time in real-time with a ball value of 0.85 to 1 and an obstacle value of 0.88 to 1, without the interference of other objects in the area with the condition of 2 obstacles in one omnidirectional camera frame, namely the orange ball object and the black obstacle used as object detection, which are both on a green carpet to avoid color contrast in the background when object detection takes place in real-time. In this test there is a graph of the results of the dataset training process which can be seen in [Figure 11](#), [Figure 12](#), [Figure 13](#), and [Figure 14](#).

Based on [Figure 11](#) confusion matrix is a table used to evaluate the performance of classification models in machine learning. It compares the predicted results generated by the model with the actual results from the test data, thus helping in understanding how well the model works. The confusion matrix provides a detailed overview of the errors made by the model as well as the areas where the model needs to be improved. precision, recall and accuracy values on the confusion matrix in this study using YOLOv8 in object detection the average value is close to 1 to 1 for overall objects such as balls, goals, robots / obstacles.

Based on [Figure 12](#) F1 curve is a visualization of the F1 score produced by a classification model at various thresholds or parameters. F1 score is a matrix that combines precision and recall, thus providing a more thorough picture of the model's performance, especially when there is class imbalance.

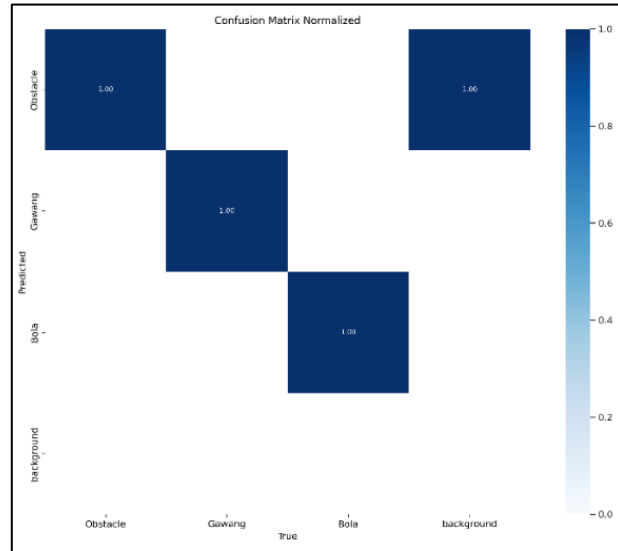


Figure 11. Confusion_matrix on the final result of the training dataset

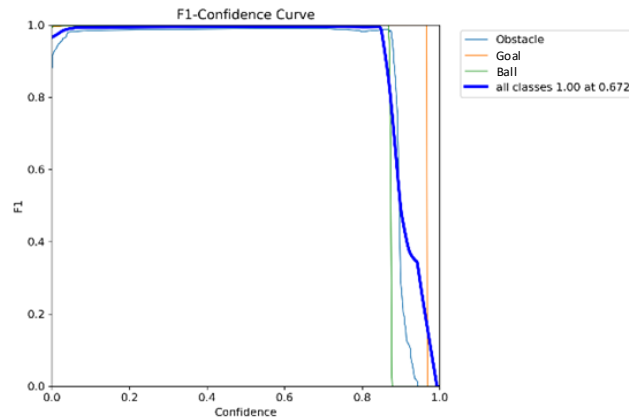


Figure 12. F1_curve on the final training dataset results

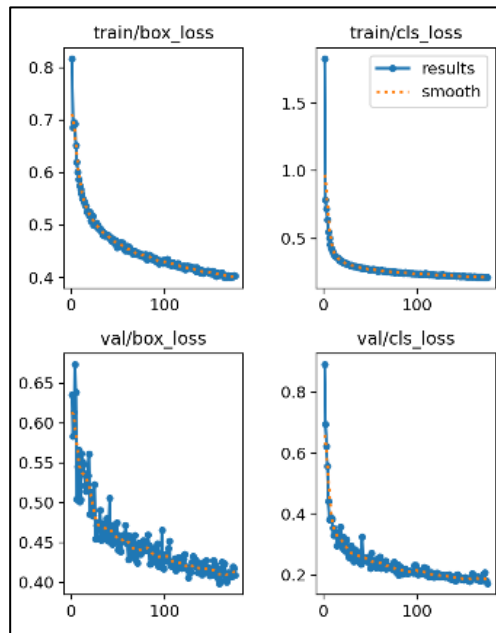


Figure 13. Result_curve(1) on the final training dataset results

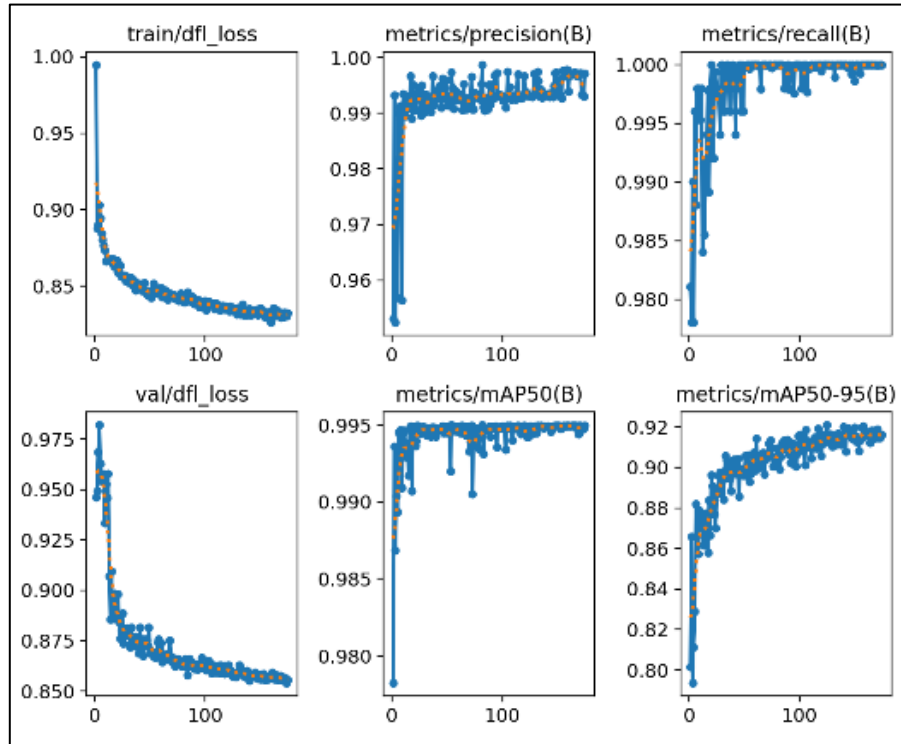


Figure 14. Result_curve(2) on the final training dataset results

Based on Figure 13 and Figure 14 the result curve, or often referred to as the precision-recall curve, is a visualization tool used to evaluate the performance of classification models, especially in situations with class imbalance. The precision-recall curve illustrates the relationship between precision and recall for various decision thresholds.

In result_curve there are:

- Train/box_loss which refers to the loss associated with the bounding box prediction made by the model. In object detection, the model not only needs to classify the object in the image but also predict its location. “Box loss” measures how well the bounding box prediction made by the model matches the actual bounding box.
- Valbox_loss which refers to the stage where the model is evaluated on data that was not used during training to gauge the performance of the model and avoid overfitting.
- Train/cls_loss which refers to the classification loss calculated during the training phase.
- Val/cls_loss refers to the classification loss calculated during the validation phase. Just like “train/cls_loss,” “val/cls_loss” measures how well the model classifies objects, but this time on validation data that was not used during training.
- Train/df_l_loss refers to the loss calculated during the training phase related to Distribution Focal Loss (DFL). Distribution Focal Loss is a variant of Focal Loss designed to improve model performance on specific tasks, such as object detection or image segmentation.
- Val/df_l_loss refers to the Distribution Focal Loss (DFL) calculated during the validation phase of training an object detection or classification model.
- Metrics/precision(B) refers to the precision for the positive class (usually called class B).
- Metrics/mAP50(B) refers to “Mean Average Precision at 50 IoU Threshold (B)” in the context of object detection model evaluation, especially in the case of object detection in binary classification tasks where the positive class is labeled B.
- Metrics/recall(B) is a measure of recall (recall rate) obtained from the model's predictions on the positive class B in the context of binary classification or object detection where the positive class is labeled B. Recall (also known as sensitivity) measures how many of all positive instances are actually detected by the model.
- Metrics/mAP50-95(B) is the concatenation of Mean Average Precision (mAP) metrics over the IoU (Intersection over Union) threshold range from 0.5 to 0.95 for positive class B in object detection tasks.

Tests were conducted using the pre-trained YOLOv8 model to detect 3 classes of objects, namely, balls, goals and robots/obstacles in real-time using an omnidirectional camera. The output of the detection results can be seen in Table 3.

Table 3. Model performance

Load Model	YOLOv8
Accuracy	85.7%
Error Classification	20.3%
Precision	88.4%
Recall	89.7%
F1-Score	88.3%
mAP@0.5	95.5%

Based on Table 3 the final results of object detection testing using YOLOv8 get model performance values such as accuracy against detected objects such as balls, goals, and robots/obstacles of 85.7% of 100% with an error classification value against detected objects of 20.3% of 100%, a precision value of 88.4%, a recall value against objects of 89.7%, an f1 score of 88.3% with a final result of mAP@0.5 is 95.5%.

4. CONCLUSIONS

Based on the results of research that has been done on tracking the ball using the YOLOv8 method on a wheeled soccer robot with an omnidirectional camera, conclusions can be drawn. The omnidirectional camera is able to detect objects that have been trained by capturing these objects with a predetermined classification. By using YOLOv8 the results_curve obtained is more stable in measuring the mAP value. In performance testing with 320×320 and 416×416 frames YOLOv8 has a higher mAP value with a value of 95.5% and a more stable average fps rate during object detection compared to previous versions of YOLO, this shows a drastic improvement from the use of previous methods.

ACKNOWLEDGEMENT

Thank You Lots I say to the whole party that has supported study this, so study this can be utilized for robot development and so on outlook knowledge of the academic community, especially Ahmad Dahlan University, Don't forget I say to readers, criticism and suggestions for study this will I thanks.

REFERENCES

- [1] P. Adarsh, P. Rathi, and M. Kumar, "YOLO v3-Tiny: Object Detection and Recognition using one stage improved model," *2020 6th Int. Conf. Adv. Comput. Commun. Syst. ICACCS 2020*, pp. 687–694, 2020, <https://doi.org/10.1109/ICACCS48705.2020.9074315>.
- [2] T. Ahmad *et al.*, "Object Detection through Modified YOLO Neural Network," *Sci. Program.*, vol. 2020, pp. 1–10, 2020, <https://doi.org/10.1155/2020/8403262>.
- [3] R. Bai, F. Shen, M. Wang, J. Lu, and Z. Zhang, "Improving Detection Capabilities of YOLOv8-n for Small Objects in Remote Sensing Imagery: Towards Better Precision with Simplified Model Complexity Improving Detection Capabilities of YOLOv8-n for Small Objects in Remote Sensing Imagery: Towards Better Pre.," *Res. Sq.*, pp. 0–9, 2023, <https://doi.org/10.21203/rs.3.rs-3085871/v1>.
- [4] T. Diwan, G. Anirudh, and J. V. Tembhurne, "Object detection using YOLO: challenges, architectural successors, datasets and applications," *Multimed. Tools Appl.*, vol. 82, no. 6, pp. 9243–9275, 2023, <https://doi.org/10.1007/s11042-022-13644-y>.
- [5] W. Fang, L. Wang, and P. Ren, "Tinier-YOLO: A Real-Time Object Detection Method for Constrained Environments," *IEEE Access*, vol. 8, pp. 1935–1944, 2020, <https://doi.org/10.1109/ACCESS.2019.2961959>.
- [6] 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, 2023, <https://doi.org/10.3390/machines11070677>.
- [7] P. Jiang, D. Ergu, F. Liu, Y. Cai, and B. Ma, "A Review of Yolo Algorithm Developments," *Procedia Comput. Sci.*, vol. 199, pp. 1066–1073, 2021, <https://doi.org/10.1016/j.procs.2022.01.135>.
- [8] M. Ju, H. Luo, Z. Wang, B. Hui, and Z. Chang, "The application of improved YOLO V3 in multi-scale target detection," *Appl. Sci.*, vol. 9, no. 18, 2019, <https://doi.org/10.3390/app9183775>.
- [9] S. Li, Y. Li, Y. Li, M. Li, and X. Xu, "YOLO-FIRI: Improved YOLOv5 for Infrared Image Object Detection," *IEEE Access*, vol. 9, pp. 141861–141875, 2021, <https://doi.org/10.1109/ACCESS.2021.3120870>.
- [10] Y. Li, Q. Fan, H. Huang, Z. Han, and Q. Gu, "A Modified YOLOv8 Detection Network for UAV Aerial Image Recognition," *Drones*, vol. 7, no. 5, 2023, <https://doi.org/10.3390/drones7050304>.
- [11] H. Lou *et al.*, "DC-YOLOv8: Small-Size Object Detection Algorithm Based on Camera Sensor," *Electron.*, vol. 12, no. 10, pp. 1–14, 2023, <https://doi.org/10.3390/electronics12102323>.
- [12] M. Maity, S. Banerjee, and S. Sinha Chaudhuri, "Faster R-CNN and YOLO based Vehicle detection: A Survey," *Proc. - 5th Int. Conf. Comput. Methodol. Commun. ICCMC 2021*, no. Iccmc, pp. 1442–1447, 2021, <https://doi.org/10.1109/ICCMC51019.2021.9418274>.

- [13] S. M. Nurin Miza Afifah Andrie Dazlee, Syamimi Abdul Khalil, Shuzlina Abdul-Rahman, "Object Detection for Autonomous Vehicles with Sensor-based Technology Using YOLO," *IJISAE*, pp. 129–134, 2022, <https://doi.org/10.18201/ijisae.2022.276>.
- [14] F. M. Talaat and H. ZainEldin, "An improved fire detection approach based on YOLO-v8 for smart cities," *Neural Comput. Appl.*, vol. 35, no. 28, pp. 20939–20954, 2023, <https://doi.org/10.1007/s00521-023-08809-1>.
- [15] Z. Trabelsi, F. Alnajjar, M. M. A. Parambil, M. Gochoo, and L. Ali, "Real-Time Attention Monitoring System for Classroom: A Deep Learning Approach for Student's Behavior Recognition," *Big Data Cogn. Comput.*, vol. 7, no. 1, pp. 1–17, 2023, <https://doi.org/10.3390/bdcc7010048>.
- [16] R. Yang, W. Li, X. Shang, D. Zhu, and X. Man, "KPE-YOLOv5: An Improved Small Target Detection Algorithm Based on YOLOv5," *Electron.*, vol. 12, no. 4, pp. 1–13, 2023, <https://doi.org/10.3390/electronics12040817>.
- [17] X. Yue, H. Li, M. Shimizu, S. Kawamura, and L. Meng, "YOLO-GD: A Deep Learning-Based Object Detection Algorithm for Empty-Dish Recycling Robots," *Machines*, vol. 10, no. 5, pp. 1–20, 2022, <https://doi.org/10.3390/machines10050294>.
- [18] Y. Zhang, Y. Sun, Z. Wang, and Y. Jiang, "YOLOv7-RAR for Urban Vehicle Detection," *Sensors*, vol. 23, no. 4, 2023, <https://doi.org/10.3390/s23041801>.
- [19] L. L. Zhao and M. L. Zhu, "MS-YOLOv7:YOLOv7 Based on Multi-Scale for Object Detection on UAV Aerial Photography," *Drones*, vol. 7, no. 3, 2023, <https://doi.org/10.3390/drones7030188>.
- [20] W. Xu, C. Cui, Y. Ji, X. Li, and S. Li, "YOLOv8-MPEB small target detection algorithm based on UAV images," *Heliyon*, vol. 10, no. 8, p. e29501, 2024, <https://doi.org/10.1016/j.heliyon.2024.e29501>.
- [21] L. Shen, B. Lang, and Z. Song, "DS-YOLOv8-Based Object Detection Method for Remote Sensing Images," *IEEE Access*, vol. 11, no. November, pp. 125122–125137, 2023, <https://doi.org/10.1109/ACCESS.2023.3330844>.

AUTHOR BIOGRAPHY



Refli Rezka Juliandan Student of the Electrical Engineering Study Program, Ahmad Dahlan University. Email : refli2000022045@webmail.uad.ac.id



Riky Dwi Puriyanto, Lecturer of Electrical Engineering Study Program, Ahmad Dahlan University. Email: rikydp@ee.uad.ac.id