

CNN-Based Ball and Goal Detection for KRSBI Robot with Omnidirectional Camera

T Mohd Farhan^{1*}, Feri Candra²

^{1,2}Dept. of Informatics Engineering, Faculty of Engineering, Universitas Riau, Pekanbaru, Indonesia
¹t.mohd3753@student.unri.ac.id, ²feri@eng.unri.ac.id

*Corresponding author: t.mohd3753@student.unri.ac.id

Abstract—The Wheeled Soccer Robot Contest (KRSBI-Beroda) challenges robots to autonomously detect, dribble, and score using vision-based systems. Traditional object detection methods like HSV color filtering are widely used but perform poorly under varying lighting conditions. This study proposes a Convolutional Neural Network (CNN)-based object detection system using the YOLO (You Only Look Once) algorithm to enhance the accuracy and reliability of ball and goal detection in KRSBI robots equipped with omnidirectional cameras. A dataset of 1,125 images comprising diverse lighting and object positions was collected and split into 80% training and 20% validation sets. The YOLOv8 model was trained using Ultralytics on Google Colab with 100 epochs. The resulting model achieved a high detection performance, with an accuracy of 95.87%, precision of 1.00 at a confidence threshold of 0.921, recall of 0.99, and an F1-Score of 0.97. The results confirm that the YOLOv8-based CNN model provides a robust and efficient solution for real-time ball and goal detection in robotic soccer applications.

Keywords: KRSBI-Beroda, YOLO, HSV, ball detection, goalpost detection, omnidirectional camera



This work is licensed under a [CC BY-SA](https://creativecommons.org/licenses/by-nc/4.0/). Copyright ©2025 by Author. Published by Universitas Riau.

INTRODUCTION

Robotics combines electrical, mechanical, and computer science disciplines, with applications in competitions like the Indonesian Robot Contest (KRI) under the Wheeled Soccer Robot Contest (KRSBI Beroda) category. In this contest, robots must autonomously detect the ball and goal to dribble and score [1]. The Electrical Robotic Club (ERC) of Universitas Riau has participated in KRSBI Beroda since 2019, reaching the national level in 2021 and 2022. However, since 2023, changes in competition standards have required adjustments, particularly in improving ball and goal detection under varying lighting conditions. The robots competing in KRSBI Beroda are designed as autonomous soccer-playing machines. They have a cubic body equipped with an omnidirectional camera mounted on top for object detection. This camera captures images in all directions using a convex mirror directed toward the ground. The robots also feature a solenoid-based kicker mechanism for shooting the ball and small gripping wheels to maintain ball control while dribbling. To ensure flexibility in movement, each robot is equipped with three wheels,

including omni-wheels that allow movement in multiple directions [1]. Figure 1 below illustrates the KRSBI Beroda robot.

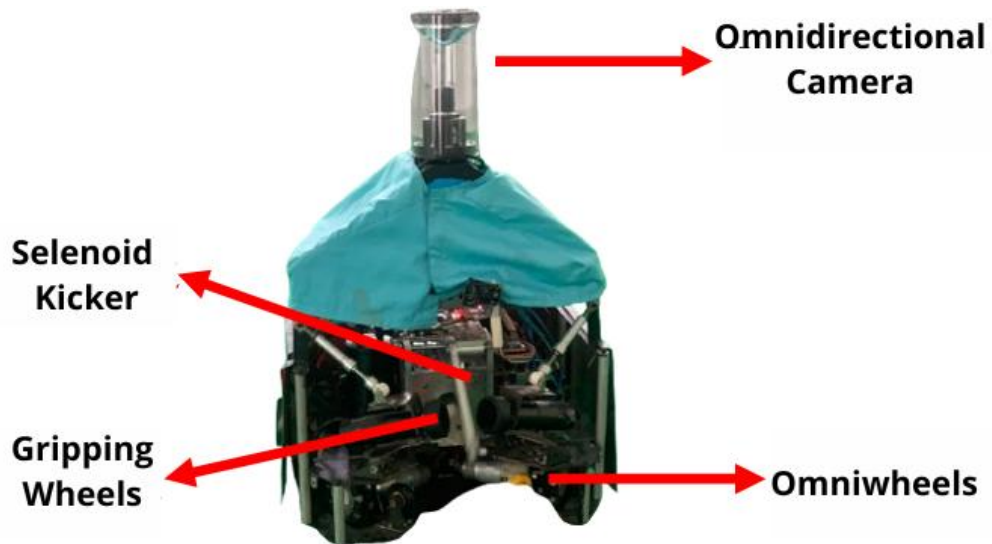


Figure 1. KRSBI-Beroda's Robot

In accordance with KRSBI Beroda regulations, the ball used in the competition is a standard futsal ball with an orange color. It has a circumference of 63-66 cm, a diameter of approximately 20-21 cm, and weighs around 400 grams. The distinctive color of the ball helps in object detection, but under certain lighting conditions, it may blend with the environment, posing a challenge for vision-based algorithms [1]. Figure 2 shows what the ball looks like.



Figure 2. competition-standard ball



Figure 3. competition-standard goalpost

Figure 3 shows the goal structure in KRSBI Beroda. It measures 2 meters in width, 50 cm in depth, and 1 meter in height. It is constructed using sturdy metal materials to withstand ball impacts during matches. The goal serves as a crucial reference point for the robot's navigation and scoring strategy, requiring precise detection for successful gameplay [1].

Currently, HSV color space is widely used for object detection in KRSBI Beroda robots due to its simplicity and efficiency. However, it is highly sensitive to lighting variations, leading to detection failures in different match conditions. Muldayani [2] highlights that RGB-to-HSV conversion is unstable under extreme lighting, making color-based detection methods less reliable when object colors resemble the surroundings. To overcome these limitations, this study proposes using Convolutional Neural Networks (CNN) for ball and goal detection. CNN is an object detection algorithm that applies object classification principles similar to neural networks, utilizing at least one convolution per layer to replace general matrix multiplication. CNN has the capability to classify image data with high accuracy as it can handle various image deformations such as translation, rotation, and scaling [3]. Several previous studies have demonstrated how CNN performs in detecting various objects. Cokorda et al. successfully detected 94.85% of 25 random images containing 87 ball and goal objects [4]. Dwi Novianto et al. achieved an accuracy of 0.97 and an F1-score of 0.97 when detecting balls and goals using YOLOv3 [5]. Jalu et al. developed a model trained with 202 images using YOLOv4, which was able to detect balls even when they were occluded by 50%, 60%, and 70% by other objects [6]. Farhan et al. utilized YOLOv3-Tiny for ball and goal detection, resulting in a model with an accuracy of 74.2% and an F1-score of 82.1% [7].

One of the CNN-based models is You Only Look Once (YOLO). YOLO introduces a novel approach to object detection, enabling real-time data processing. Image processing with YOLO is relatively simple, as it resizes the input image, applies a single convolutional network, and sets a threshold for detection results based on the model's confidence score [8]. Specifically, this study employs YOLOv8, one of the latest versions of YOLO, which identifies objects based on multiple visual features, making it more robust against lighting changes. Compared to its predecessors, YOLOv8 introduces improvements in model efficiency, feature extraction, and computational speed, making it better suited for real-time applications. The model benefits from anchor-free detection, dynamic label assignment, and a more streamlined neural network architecture, optimizing both precision and speed. However, implementing YOLOv8 on KRSBI Beroda robots presents challenges due to the complexity of building and training the model. The process involves extensive dataset preparation, hyperparameter tuning, and network optimization to ensure reliable object detection in dynamic competition environments. Hardware acceleration using GPUs or TPUs must be considered to ensure that training process is efficient. This study aims to develop a CNN-based YOLOv8 model detection system using an omnidirectional camera of KRSBI-Beroda to create an accurate ball and goal detection.

METHOD

This study focuses on designing, developing, and optimizing a system for ball and goal detection using You Only Look Once (YOLOv8). The research involves multiple stages, including data collection, model training, implementation, and performance evaluation. A detailed breakdown of each step in the process is illustrated in the following flowchart.

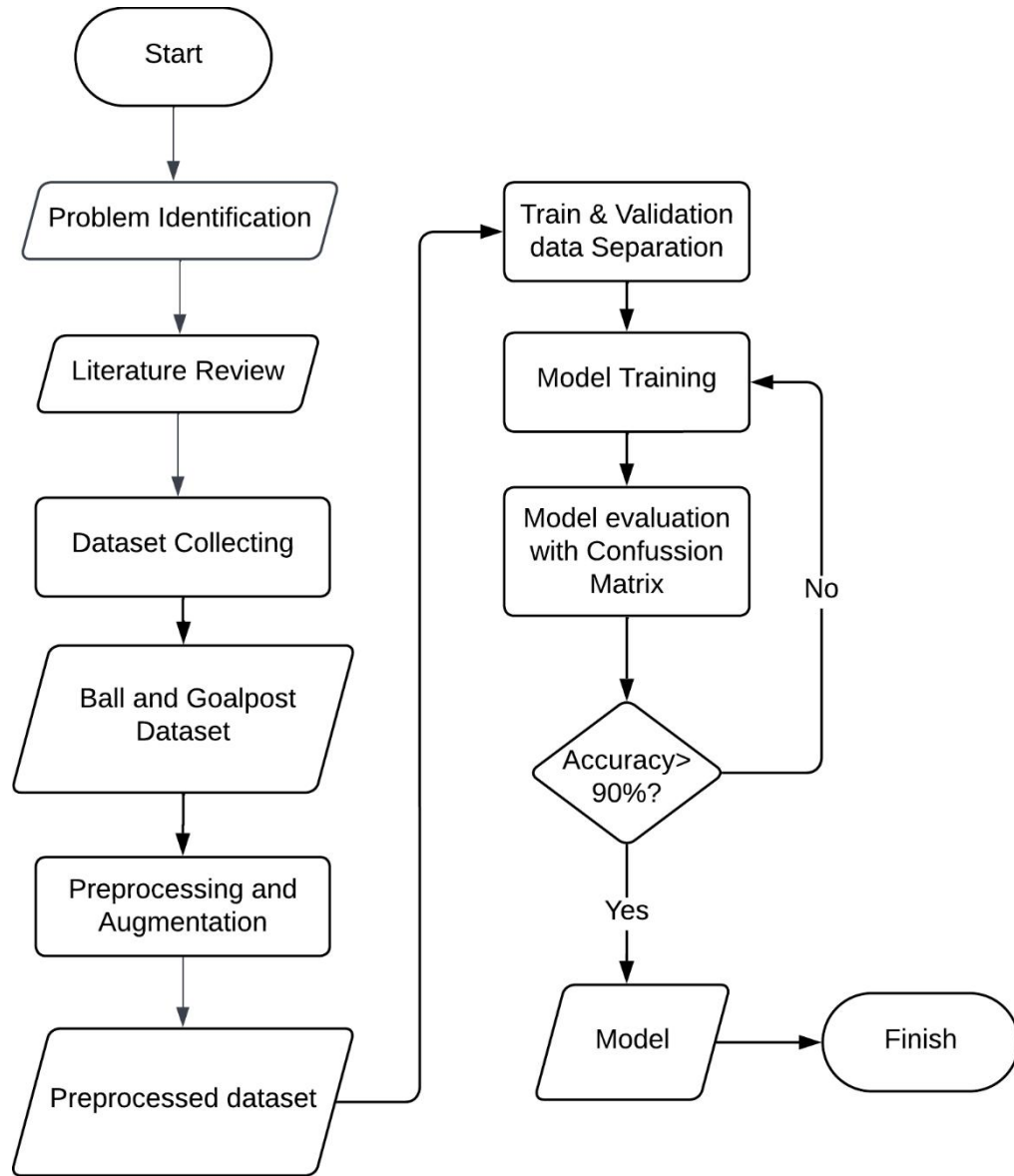


Figure 4. Research Methodology's Flowchart

Based on Figure 4, the research process begins with problem identification, specifically the limitations of HSV-based color detection methods in handling lighting variations on KRSBI-Beroda robots. After identifying the problem, a literature review is conducted to understand the YOLO method that will be used for ball and goal detection. Next, the implementation of YOLO is carried out. The research starts with collecting a dataset of balls and goals to train the model. The data collection for ball and goalpost detection was conducted by capturing images directly using the omnidirectional camera mounted on the KRSBI robot. This dataset undergoes preprocessing, including image annotation, to ensure optimal model training. Subsequently, the model is trained using the prepared dataset. Once trained, the model is tested in the evaluation phase, where the effectiveness of YOLO in detecting balls and goals under various lighting conditions is assessed. If accuracy exceeds 90%, the model is finalized; otherwise, retraining is performed.

RESULT AND DISCUSSION

A. Data Collecting

The dataset collection process is conducted to initiate model development. As a result, a total of 1,125 images consisting of ball and goal images were obtained. The images were captured directly using an omnidirectional camera. The ball and goalpost were placed in different positions in each image to ensure a diverse dataset. Some examples of the collected dataset can be seen in Figures 5 and 6.



Figure 5. Some of ball datasets



Figure 6. Some of goalpost datasets

All collected datasets were labeled. Labeling is the process of assigning labels to a set of pixels that form an object within an image. Different objects are given different labels to distinguish them. This process is crucial for shape analysis and pattern recognition in image processing, as it helps the model accurately identify and differentiate objects within a dataset [9]. This labeling process involves marking the ball and goal objects with bounding boxes. The coordinates of these boxes are then extracted and saved in a .txt file format. An example of a labeled dataset can be seen in Figure 7.

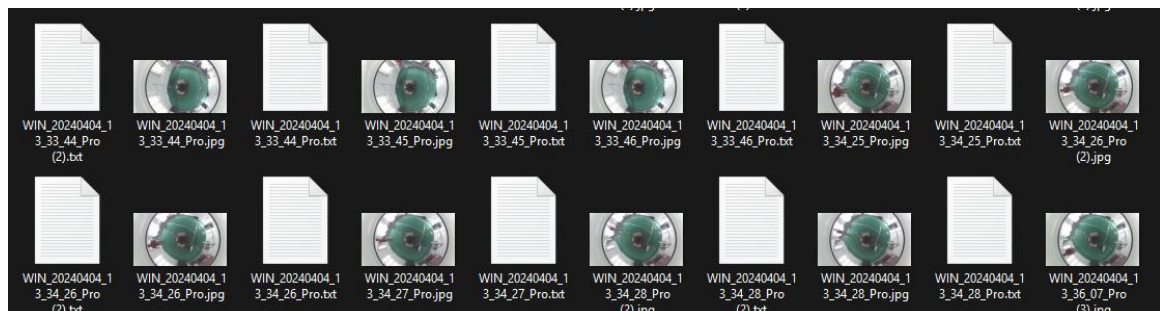


Figure 7. Some of labeled datasets

B. Data Preprocessing and Augmentation

After the labeling process, the dataset undergoes to preprocessing. Preprocessing is the stage of processing collected image data to make necessary adjustments [10]. The First step for preprocessing is resizing images to 600x400 resolution. Resizing images is essential to ensure that all images have consistent dimensions before being fed into the model, as YOLO requires fixed-size input images. This process helps improve processing efficiency and maintains consistency within the model.



Figure 8. Resizing image from 3840x2160 (left image) to 600x400 (right image)

Following this, image augmentation is applied. Image augmentation is an effective way to improve image classification models [11] [12]. This process artificially expand the dataset, enhancing the learning process and overall performance by inputting diverse data variations into the model [13]. The augmentation techniques applied include horizontal flipping, which mirrors the image to introduce variations in object patterns. Rotation is also used, with images being rotated 90 degrees clockwise, 90 degrees counterclockwise, 45 degrees clockwise, and 45 degrees counterclockwise. This allows the model to learn to recognize objects in different orientations. Additionally, brightness adjustment is applied by increasing the brightness by 20% and decreasing it by 20%, ensuring the model is more resilient to real-world lighting variations. All these augmentation techniques aim to make the model more robust in detecting objects under various conditions encountered during testing. The results of this augmentation are illustrated in the following table.

Table 1. Illustration of Augmentation Result

Augmentation Process	Original Image	After Augmentation
Horizontal Flip		
90° clockwise rotation		
90° counter-clockwise rotation		

45° clockwise rotation		
45° counter-clockwise rotation		
Brightness +20%		
Brightness -20%		

After preprocessing and augmentation, the dataset is divided into two groups: training data and validation data. The data is split with an 80% allocation for training and 20% for validation. This division is essential to ensure that the model learns effectively while also being evaluated on unseen data. The training set is used to adjust the model's parameters, helping it recognize patterns and features in the images. Meanwhile, the validation set is used to assess the model's performance and detect potential overfitting, where the model performs well on training data but struggles with new, unseen data. By maintaining this 80:20 ratio, the model can generalize better, improving its ability to detect balls and goals under various real-world conditions.

C. Model Training

The object detection model training process using the YOLOv8 algorithm is conducted with the Ultralytics library, which provides an efficient and user-friendly interface for training YOLO (You Only Look Once) models. This library is specifically designed to support the latest versions of YOLO, such as YOLOv8, and facilitates real-time implementation. This real-time capability is crucial for systems that require fast and accurate object detection, such as robotic vision applications. YOLOv8 includes built-in data augmentation techniques to improve model generalization. Mosaic augmentation combines four images into one for better object diversity, MixUp blends images to enhance robustness, Random Perspective adjusts object positions to simulate different viewpoints, and Color Jittering modifies brightness, contrast, and saturation to improve performance under varying lighting conditions. These augmentations help the model detect objects more accurately across different environments [14].

For learning rate, YOLOv8 employs an adaptive learning strategy, starting at 0.001 and dynamically adjusting using a cosine annealing scheduler. This prevents overshooting or slow convergence by gradually reducing the learning rate as training progresses, ensuring stable optimization without manual tuning [14]. The YOLOv8 model training is conducted using the Google Colab platform. This platform is chosen because it provides access to a T4 GPU, which offers excellent performance for model training, ensuring that the process does not take too long. The dataset is trained for 100 epochs with a batch size of 16, optimizing the model's ability to learn object features effectively.

When training the YOLOv8 model using the Ultralytics library, the command used is as follows:

```
!yolo task=detect mode=train model=yolov8n.pt data=(data_path) epochs=100 batch=16
```

This command is used to train the object detection model with several important parameters. The task=detect parameter specifies that the task being performed is object detection. The mode used is train, indicating that the model will be trained using the prepared dataset. The base model selected is yolov8n.pt, a lightweight version of YOLOv8, which is suitable for resource-limited environments or applications that require high-speed inference. The data=(data_path) parameter defines the location of the dataset configuration file. This file contains essential information such as the number of object classes (ball and goal), the location of the images, and their corresponding labels. Proper configuration of this dataset ensures that the model correctly learns to detect the specified objects. The epochs=100 parameter determines the number of training iterations, which influences how well the model learns from the dataset. A higher number of epochs allows the model to refine its predictions through repeated training cycles. Additionally, the batch=16 parameter defines how many images are processed in one batch during training. A larger batch size can speed up the training process but requires more memory, making it essential to balance computational efficiency and performance.

D. Model Evaluation

After completing the model training process, a confusion matrix is obtained from the best-performing epoch during training. Confusion matrix is a commonly used accuracy measurement method in data mining. It is presented as a table that displays the number of correctly and incorrectly classified data points after the classification process[15].

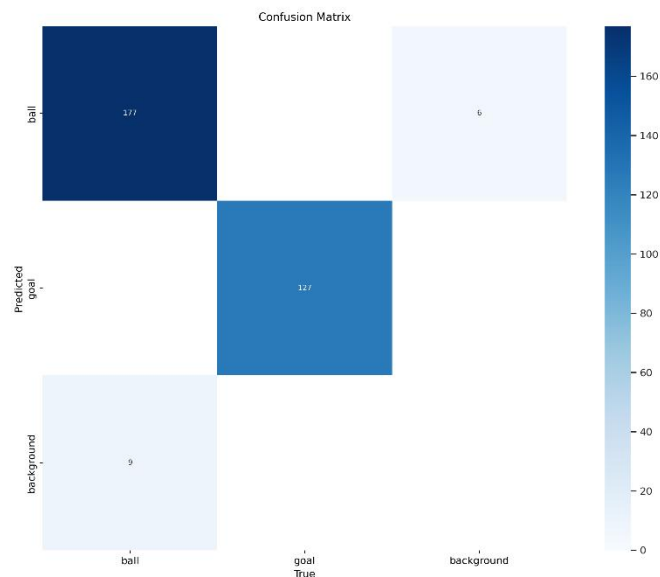


Figure 9. Confusion Matrix

Based on Figure 9, the results indicate that the trained YOLOv8 model performs well in classifying objects. Images containing a ball are correctly detected as a ball, and images containing a goal are accurately predicted as a goal, with very high values. This signifies that the model can classify objects with a high degree of accuracy. Additionally, the number of misclassified images is very small compared to the total predicted images, demonstrating that classification errors have been effectively minimized. By definition, Accuracy is the ratio

of correctly classified data to the total dataset [16]. To determine the accuracy, we first define the values in the confusion matrix:

- True Positives (TP) : 177 (correctly classified as ball) + 127 (correctly classified as goalpost)
- True Negatives (TN) : 0 (correctly classified as background)
- False Positives (FP) : 6 (misclassified background as ball)
- False Negatives (FN) : 9 (misclassified ball as background)

The exact accuracy value is calculated using the following formula:

$$\begin{aligned}
 \text{Accuracy} &= \frac{TP+TN}{TP+TN+FP+FN} \times 100\% \\
 &= \frac{175+127+0}{175+127+0+6+9} \times 100\% \\
 &= \frac{302}{317} \times 100\% \\
 &= 95,87\%
 \end{aligned}$$

Apart from the confusion matrix, the model training also produces several other curves, such as the precision curve, recall curve, and F1-score curve.

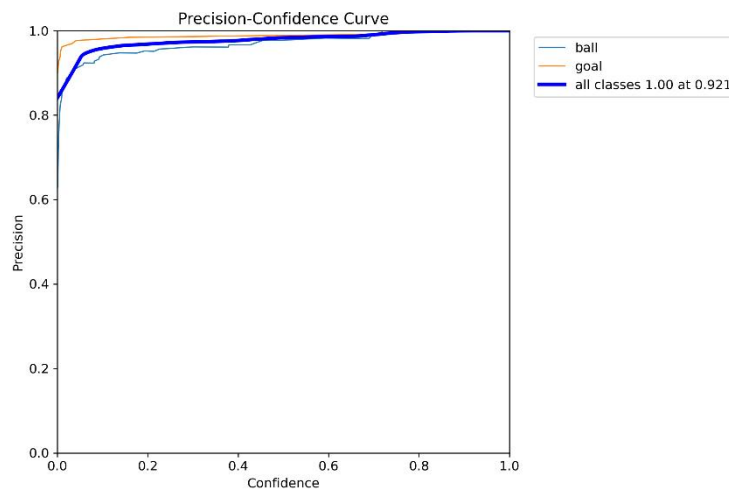


Figure 10. Precision curve

The curve shown in Figure 10 illustrates the relationship between precision and confidence threshold. Precision is a parameter that measures the accuracy of positive detections made by the YOLOv8 model [17], while confidence represents how certain the model is that an object exists [18]. The curve indicates that precision reaches a value of 1.00 at a confidence threshold of 0.921. This means that at this threshold, all detections are accurate, with no false positives. The curve also shows that precision remains consistently high at higher confidence thresholds, indicating that the model is highly reliable in avoiding incorrect detections. However, at lower confidence threshold, there are minor fluctuations, suggesting the presence of some false positives. This pattern highlights the trade-off between confidence and recall, where increasing the confidence threshold improves precision but may reduce the number of correctly detected objects. For practical implementation, setting a confidence threshold of 0.883 ensures the highest detection accuracy, but a slightly lower

threshold, such as 0.7 to 0.8, may be preferable to maintain a balance between precision and recall.

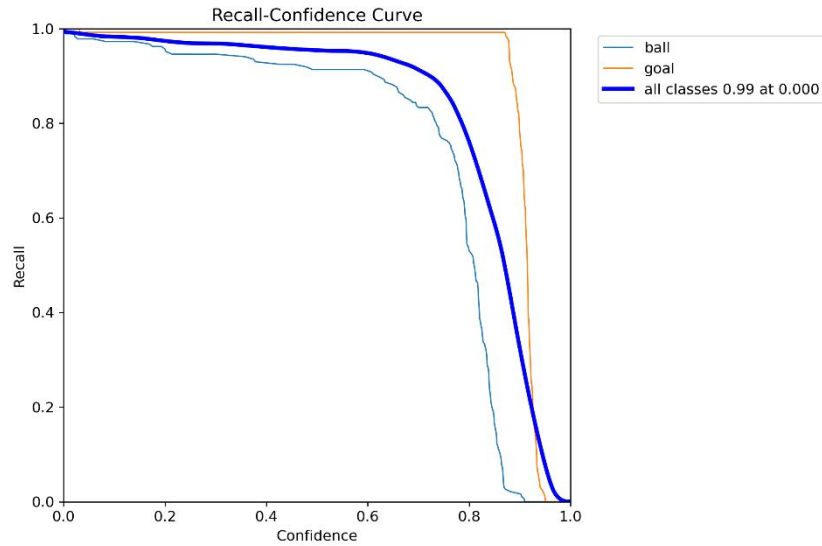


Figure 11. Recall curve

The curve shown in Figure 11 presents the recall value for each class concerning confidence thresholds. Recall represents the model's ability to detect objects that should be identified [19]. It is calculated as the number of correctly detected objects divided by the total number of objects present. At low confidence threshold, recall remains high for both classes, indicating that the model detects most objects but with varying certainty. However, as the confidence threshold increases, the recall for ball declines more rapidly compared to goalpost, which maintains a high recall until nearly 0.9 confidence. This suggests that goalpost is detected with higher certainty, while ball detections are more affected by the confidence threshold. The overall recall for all classes starts at 0.99 but decreases sharply after 0.8 confidence, highlighting a trade-off between detection completeness and certainty.

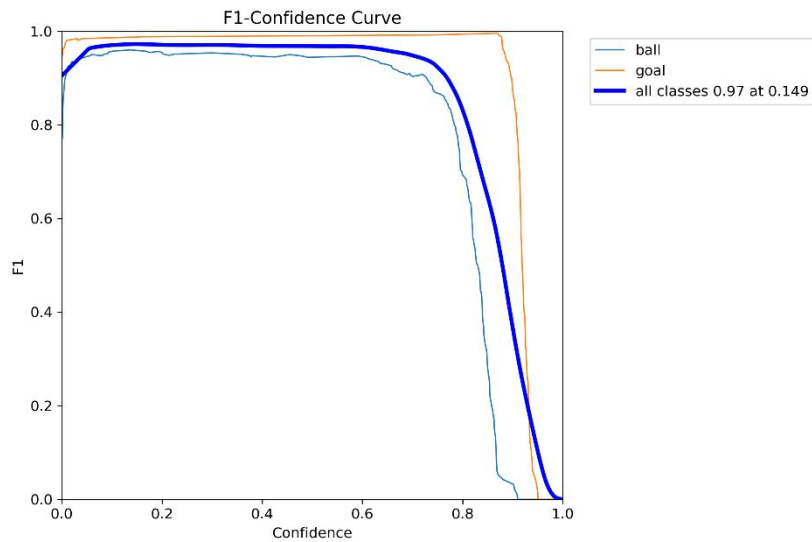


Figure 12. F1-Score curve

Figure 12 illustrates a curve graph representing the F1-Score values in relation to confidence thresholds. The F1-Score is a metric that combines precision and recall into a single value [20], providing a balanced measure of a model's performance. A high F1-Score indicates that the model has both strong precision (fewer false positives) and strong recall (fewer false negatives), making it a reliable indicator of classification performance [21]. From the graph, it is observed that the F1-Score reaches its highest value of 0.97 when the confidence threshold is at 0.149. However, it starts to decline sharply when confidence surpasses 0.8. This suggests that at lower confidence thresholds, the model maintains a good balance between precision and recall, meaning it can correctly detect a high number of objects. However, as the confidence threshold increases to 0.8, the model becomes more selective, discarding images that contain objects but are not distinctly clear. This results in a drop in recall, as some true positives are overlooked, affecting the overall F1-Score. A closer analysis of the class-specific curves (ball and goalpost) reveals that the goalpost class maintains a higher F1-Score across a broader confidence range compared to ball. This suggests that the model is more confident in detecting goalpost than ball, likely due to clearer object features or more consistent training data for goalpost. The ball class, on the other hand, experiences a steeper decline in F1-Score as confidence increases, indicating that it is more susceptible to misclassification at higher thresholds. This behavior may stem from lighting variations, occlusion, or similarities between the ball and background elements, which complicate detection.

CONCLUSION

Based on the conducted research, ball and goal detection using an omnidirectional camera on the KRSBI-Beroda robot can be effectively performed using the You Only Look Once (YOLO) method. After evaluating the model, the model's accuracy reaches 0.95. On the other hand, precision-confidence curve indicates that precision reaches 1.00 at a confidence threshold of 0.921, meaning that at this threshold, all detections made by the model are correct, with no false positives. Meanwhile, the recall-confidence curve shows that recall reaches 0.99 even at a low confidence threshold, demonstrating the model's ability to detect nearly all given objects. The F1-Score curve peaks at 0.97 when confidence is at 0.149 but begins to decline sharply after 0.8, indicating that the model performs optimally at moderate confidence thresholds before becoming overly selective.

ACKNOWLEDGMENT

I would like to express my heartfelt gratitude to everyone who supported the completion of this article. Although I am unable to mention each individual by name, their contributions, encouragement, and assistance are sincerely appreciated.

REFERENCES

- [1] B. Kusumoputro et al., "Pedoman Kontes Robot Indonesia (KRI) Pendidikan Tinggi Tahun 2024," Balai Pengembangan Talenta Indonesia, Pusat Prestasi Nasional, Kementerian Pendidikan, Kebudayaan, Riset dan Teknologi Republik Indonesia, 2023.
- [2] W. Muldayani, "Implementasi Sistem Object Tracking Untuk Mendeteksi Dua Objek Berbasis Deep Learning," *Simetris J. Tek. Mesin, Elektro dan Ilmu Komput.*, vol. 14, no. 1, pp. 1–14, 2023, doi: 10.24176/simet.v14i1.9236.

- [3] L. A. Andika, H. Pratiwi, and S. S. Handajani, "Klasifikasi Penyakit Pneumonia Menggunakan Metode Convolutional Neural Network Dengan Optimasi Adaptive Momentum," pp. 331–340, 2019.
- [4] C. G. Wahyu Pramana, D. C. Khrisne, and N. P. Sastra, "Rancang Bangun Object Detection Pada Robot Soccer Menggunakan Metode Single Shot Multibox Detector (SSD MobileNetV2)," *J. SPEKTRUM*, vol. 8, no. 2, p. 28, 2021, doi: 10.24843/spektrum.2021.v08.i02.p4.
- [5] D. N. Nugroho and L. Anifah, "Perancangan Sistem Deteksi Objek Bola Dan Gawang Pada Robot Sepakbola Menggunakan Metode Darknet YOLO," *J. Inf. Eng. Educ. Technol.*, vol. 7, no. 1, pp. 22–29, 2023, doi: 10.26740/jieet.v7n1.p22-29.
- [6] J. Nuralim et al., "Perancangan Sistem Pendeteksian Obyek Bola Dengan Metode Framework YOLOv4," *J. Maest.*, vol. 5, no. 2, pp. 289–294, 2022. [Online]. Available: <https://jom.ft.budiluhur.ac.id/index.php/maestro/article/view/531>
- [7] 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.
- [8] J. Redmon, S. Divvala, R. Girshick, and A. Farhadi, "You only look once: Unified, real-time object detection," *Proc. IEEE Comput. Soc. Conf. Comput. Vis. Pattern Recognit.*, vol. 2016-Decem, pp. 779–788, 2016, doi: 10.1109/CVPR.2016.91.
- [9] E. Mozef, "Algoritma Labeling Citra Biner Dengan Performansi Optimal Processor-Time," pp. 67–77, 2003.
- [10] A. Anhar and R. A. Putra, "Perancangan dan Implementasi Self-Checkout System pada Toko Ritel menggunakan Convolutional Neural Network (CNN)," *ELKOMIKA J. Tek. Energi Elektr. Tek. Telekomun. Tek. Elektron.*, vol. 11, no. 2, p. 466, 2023, doi: 10.26760/elkomika.v11i2.466.
- [11] X. Chen et al., "Robust and accurate object detection via adversarial learning," *Proc. IEEE Comput. Soc. Conf. Comput. Vis. Pattern Recognit.*, pp. 16617–16626, 2021, doi: 10.1109/CVPR46437.2021.01635.
- [12] H. Zhang, M. Cisse, Y. N. Dauphin, and D. Lopez-Paz, "MixUp: Beyond empirical risk minimization," *6th Int. Conf. Learn. Represent. ICLR 2018 - Conf. Track Proc.*, pp. 1–13, 2018.
- [13] G. Moreira et al., "Benchmark of Deep Learning and a Proposed HSV Colour Space Models for the Detection and Classification of Greenhouse Tomato," *Agronomy*, vol. 12, no. 2, 2022, doi: 10.3390/agronomy12020356.
- [14] Ultralytics, "YOLOv8 Data Augmentation and Learning Rate Optimization," 2024. [Online]. Available: <https://docs.ultralytics.com>

- [15] M. F. Rahman et al., "Klasifikasi Untuk Diagnosa Diabetes Menggunakan Metode Bayesian Regularization Neural Network (RBNN)," *J. Inform.*, vol. 11, no. 1, p. 36, 2017, doi: 10.26555/jifo.v11i1.a5452.
- [16] B. P. Pratiwi, "Pengukuran Kinerja Sistem Kualitas Udara," *J. Inform. UPGRIS*, vol. 6, no. 2, pp. 66–75, 2020.
- [17] S. Clara et al., "Implementasi Seleksi Fitur Pada Algoritma Klasifikasi Machine Learning Untuk Prediksi Penghasilan Pada Adult Income Dataset," *Semin. Nas. Mhs. Ilmu Komput. dan Apl. Jakarta-Indonesia*, vol. 2, no. 1, pp. 741–747, 2021.
- [18] T. Le, T. Miller, R. Singh, and L. Sonenberg, "Explaining Model Confidence Using Counterfactuals," *Proc. 37th AAAI Conf. Artif. Intell. AAAI 2023*, vol. 37, no. 1, pp. 11856–11864, 2023, doi: 10.1609/aaai.v37i10.26399.
- [19] S. Sibuea and Y. B. Widodo, "Pengembangan Model Machine Learning untuk Rekomendasi Produk Berdasarkan Analisis Pola Pembelian," vol. 10, no. 2, pp. 567–583, 2024.
- [20] M. Fadli and R. A. Saputra, "Klasifikasi Dan Evaluasi Performa Model Random Forest Untuk Prediksi Stroke," *JT J. Tek.*, vol. 12, no. 02, pp. 72–80, 2023. [Online]. Available: <http://jurnal.umt.ac.id/index.php/jt/index>
- [21] R. R. Adhitya, W. Witanti, and R. Yuniarti, "Perbandingan Metode Cart Dan Naïve Bayes Untuk Klasifikasi Customer Churn," *INFOTECH J.*, vol. 9, no. 2, pp. 307–318, 2023, doi: 10.31949/infotech.v9i2.5641

BIOGRAPHIES OF AUTHORS



T MOHD FARHAN was born in Pekanbaru, Riau, Indonesia. He is a student in the Department of Informatics Engineering, Faculty of Engineering, Riau University since he joined in 2020. His studies focused on field of Intelligent Computing and Visualization (KCV).



FERI CANDRA is the Coordinator of the Informatics Engineering Program at Riau University with expertise in Machine Learning, Computer Vision, and Robotics. He holds a Ph.D. from Universiti Teknologi Malaysia, a master's degree from the Universitas Indonesia, and a bachelor's degree from Institut Sains dan Teknologi Nasional.