Ball Detection and Color Identification for a Mobile Robot using a 2D Camera

Khac Trung Chu
Testing and Assessment Centre, Hanoi University of Industry, Vietnam
chu.trung@haui.edu.vn

Minh Hieu Hoang
School of Mechanical and Automotive Engineering, Hanoi University of Industry, Vietnam
hieuchy2309@gmail.com

Quoc Bao Tran
School of Mechanical and Automotive Engineering, Hanoi University of Industry, Vietnam
baotq@haui.edu.vn (corresponding author)

Received: 3 December 2024 | Revised: 9 January 2025 and 22 January 2025 | Accepted: 24 January 2025

 DOI: https://doi.org/10.48084/etasr.9821

ABSTRACT

In this study, a novel method is developed to help the mobile robot system accurately detect and recognize the color of a ball in environments with light disturbances using deep learning. The YOLOv8 algorithm is applied to detect the ball and identify its color. The effectiveness of the algorithm is tested in various lighting conditions and when the balls are inside a silo and when they are outside. The developed algorithm identifies balls even when they are partially obscured by shadows.

Keywords-YOLOv8; ball detection; 2D camera; disturbance environments; mobile robot

I. INTRODUCTION

Object detection plays a critical role in computer vision, allowing systems to effectively analyze and interpret their surroundings. It is essential for a variety of applications, including robotics, security, facial recognition, medical diagnostics, and autonomous vehicles. The advancements driven by Industry 4.0 have highlighted the increasing significance of Artificial Intelligence (AI) in these fields. Object detection technology has made significant progress leveraging Deep Learning (DL) and Machine Learning (ML). However, challenges remain, particularly when applied to real-world conditions that involve occlusions, varying lighting, and cluttered environments [1-3].

Image processing is fundamental for enabling robots to assess and interact with their surroundings. This technology supports vital functions such as navigation, surveillance, object detection, and human-robot interaction. These advancements not only improve the efficiency of robotic systems but also broaden their applicability in sectors such as healthcare, education, and customer service, leading to innovative solutions for assistive technologies and automated services [4-6].

Despite these improvements, object detection systems still face challenges. Among the most widely used models, the YOLO (You Only Look Once) series is renowned for its balance between speed and accuracy. Early versions, like YOLOv3 struggled with small object detection and handling overlapping instances due to their reliance on high-level feature representations [7]. YOLOv4 introduced Cross Stage Partial Networks (CSPNet) improving small object detection and computational efficiency, but deploying it on resource-constrained devices remains a challenge [8]. YOLOv5 refined the architecture and increased inference speeds, though it still struggles in environments with significant noise or occlusion where object boundaries become less distinct [9-10].

Newer versions like YOLOv8 and YOLOv10 try to address these challenges. YOLOv8 introduced state-of-the-art innovations, including a CSPDarknet backbone, anchor-free detection, and enhanced feature fusion techniques making it highly effective for real-time applications requiring both speed and accuracy [11-12]. YOLOv10 offers dual assignment strategies that improve training efficiency and inference performance. A notable advancement in YOLOv10 is its elimination of Non-Maximum Suppression (NMS) simplifying the detection process and increasing efficiency in practical applications [13-15].

This study presents a DL approach based on YOLOv8 to detect spherical objects and identify their colors under challenging conditions. The proposed method demonstrates real-time processing capabilities achieving frame rates of 25–30 fps while effectively adapting to various environmental conditions. Experimental results confirm the method's robustness across different lighting scenarios including low light, standard illumination, and bright outdoor settings. These findings demonstrate the potential of this algorithm for deployment in dynamic resource-constrained environments [16-17].

II. REAL-TIME SYSTEM FOR DETECTING BALL COLORS

A. System Framework

The proposed algorithm is shown in Figure 1. Initially, the system begins with a 2D camera. The input image is resized to 640×640 pixels. After that, the ball is detected using the YOLOv8 model. If an object is identified as a ball, YOLOv8 provides the bounding boxes and labels corresponding to the ball's actual color. If not, the system continues to search for the ball.

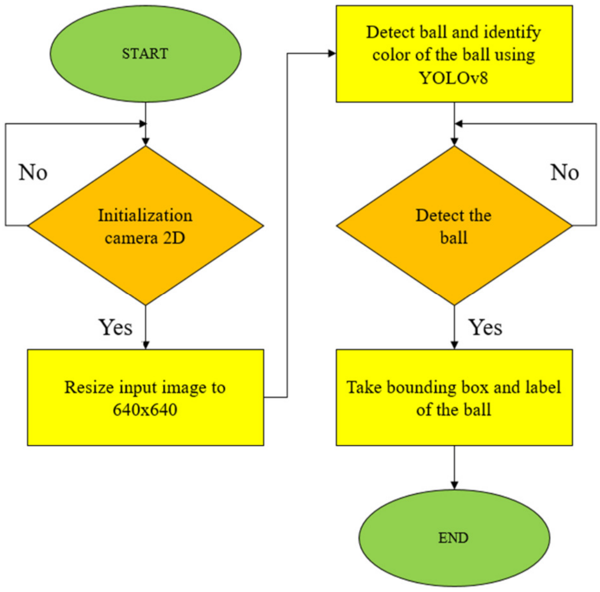

Fig. 1. Algorithm flowchart.

B. YOLOv8 Algorithm

YOLOv8 provides users with 5 AI neural network models used to detect objects: YOLOv8n (nano), YOLOv8s (small), YOLOv8m (medium), YOLOv8l (large), and YOLOv8x (extra large). In addition, Ultralytics also provides ML models to solve problems such as segmentation, object detection, classification, pose estimation [18-22].

Table I provides a detailed comparison of the YOLOv8 model variants in terms of size, number of parameters, and computational complexity (measured in FLOPs). It is used to analyze the trade-offs between model efficiency and performance enabling users to choose the appropriate version based on the specific requirements of their application such as speed, accuracy or hardware limitations.

TABLE I. COMPARISON OF YOLOv8 MODELS

Model	Size (pixels)	Model's size	Params (M)	FLOPs (B)
YOLOV8n	640x640	6.4 MB	3.2	8.7
YOLOv8s	640x640	22.0 MB	11.2	28.6
YOLOv8m	640x640	50.9 MB	25.9	78.9
YOLOv8l	640x640	85.7 MB	43.7	165.2
YOLOv8x	640x640	133.7 MB	68.2	257.7

The YOLOv8 series demonstrates a clear trade-off between computational complexity and model performance. Smaller models like YOLOv8n are compact (6.4 MB in size) and have fewer parameters (3.2 million) and FLOPs (8.7 billion) making them well-suited for applications requiring faster processing and lower computational costs. On the other hand, larger models such as YOLOv8x provide significantly higher accuracy but come with a substantial increase in size (133.7 MB), parameters (68.2 million), and FLOPs (257.7 billion), resulting in slower inference times. This trade-off allows users to select the most suitable model based on specific requirements such as prioritizing real-time performance or achieving maximum detection accuracy.

YOLOv8 achieves superior performance through a carefully designed architecture consisting of three main components, namely backbone, neck, and head. The backbone uses an enhanced CSP-Darknet53, which improves feature extraction by efficiently processing gradient flows and reducing computational complexity. The neck refines the extracted features enhancing their representational power for robust object detection. Finally, the head processes these features to classify objects and determine their bounding boxes with high accuracy. Advanced techniques such as CIF, an efficient data structure, streamline intermediate feature storage and processing, while OAM a sophisticated optimization algorithm, boosts the overall performance of the network.

1) Architectural Enhancements

YOLOv8 uses a refined version of the CSP-Darknet53 backbone. This improves feature extraction by better handling gradient flow and reducing computational complexity.

2) Loss Function

YOLOv8 has three kinds of loss as bounding box regression loss, objectness loss, and class prediction loss. The overall loss is determined by their summation:

$$L = \lambda_{box}L_{box} + \lambda_{obj}L_{obj} + \lambda_{cls}L_{cls} \quad (1)$$

where λ_{box}, λ_{obj}, λ_{cls} are hyperparameters that balance the contributions of each component to the total loss. $L_{box} = \sum_{i} smooth\_L1(\hat{b}_i, b_i)$ for each predicted bounding box. It calculates the loss $\sum_{i} BCE(\hat{o}_i, o_i)$ between the predicted coordinates $(\hat{b})$ and the ground truth coordinates (b). $L_{obj} = \sum_{i} BCE(\hat{o}_i, o_i)$ for each anchor. It calculates the binary cross entropy loss between the predicted objectness score $(\hat{o})$ and the ground truth objectness (o). $L_{box} = \sum_{i} CE(\hat{c}_i, c_i)$ calculates the categorical cross entropy loss between the predicted class probabilities $(\hat{c})$ and the ground truth class label (c) for each positive anchor.

III. EXPERIMENTAL RESULTS

A. Training Model

To highlight the optimization and advancements of YOLOv8 over YOLOv7 [23], we conducted a performance comparison by training both models on the same dataset under identical conditions. The dataset consisted of 4,000 images of balls, including 500 sourced from Google and 3,500 captured using our camera. This ensured that both models had a consistent and equal foundation for training and evaluation. The training process for both YOLOv8 and YOLOv7 was executed on Google Colab, leveraging its computational resources to ensure efficiency and reproducibility. By maintaining identical training parameters and datasets, the comparison provided unbiased insights into the capabilities of each model.

The results clearly demonstrated YOLOv8's superior optimization. It consistently outperformed YOLOv7 across key metrics, including mean Average Precision (mAP), inference speed, and detection accuracy, particularly under challenging conditions such as poor lighting, varying object sizes, and occlusions [24]. These findings underscore the advancements in YOLOv8's architecture and loss function, which allow it to achieve higher accuracy and faster processing while maintaining efficiency. This makes YOLOv8 a more effective solution for real-time ball detection applications.

B. System Setup and Performance Evaluation

The hardware system, as shown in Figure 2, consists of a lightweight and modular frame designed to house the key components needed for object detection and manipulation tasks. A high-definition camera is positioned at an optimal height to capture environmental images, serving as the primary input for the object detection system. A laptop is integrated into the robot's structure, functions as the central processing unit handling real-time data processing and inference using pre-trained deep learning models.

The robot is equipped with multiple actuators and a gripping mechanism specifically designed for interacting with detected objects. Its locomotion system features omnidirectional wheels, allowing for smooth and precise movement across a variety of terrains. Additional sensors were incorporated to facilitate obstacle avoidance and navigation, enhancing the robot's adaptability in dynamic environments. The system is powered by a portable battery pack to ensure continuous operation during tasks. The proposed algorithm was executed on a system featuring 8 GB of RAM, an 11th Gen Intel(R) Core(TM) i5-11400H @ 2.70GHz processor, and an NVIDIA GeForce 3050 Laptop GPU. Image capture was performed with the Hikvision DS-U02 full HD 1080P webcam. The algorithm was implemented in Python 3.8 utilizing the OpenCV 4.0.7.68 and NumPy libraries for processing and computation. This hardware and software configuration is optimized for real-time object detection and task execution ensuring reliability and performance in controlled environments.

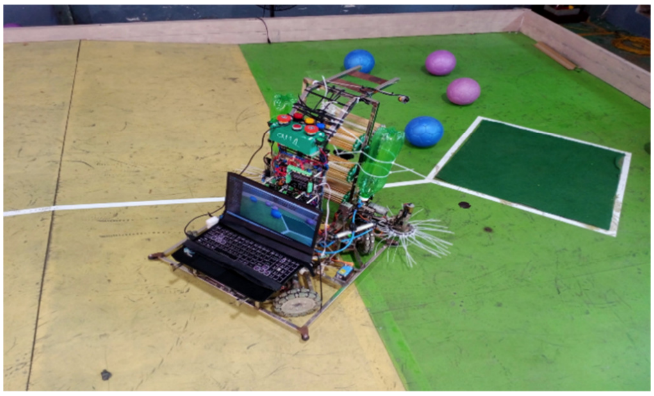

Fig. 2. The robot system integrates image processing and AI.

Fig. 3. Test results in six cases.

The performance of the algorithm was evaluated under varying lighting conditions, including low light, normal light, and bright light as well as in two distinct scenarios: when the ball was located inside and outside a silo. The test results demonstrate the algorithm's ability to detect and classify balls accurately across different environmental settings. Each case in Figure 3 illustrates specific scenarios highlighting the robustness of the detection system under diverse conditions.

Figure 4 shows the confusion matrix of Yolov8 for the testing dataset. Figure 5 shows the change in loss metrics and the performance of the YOLOv7 model over epochs. The plot includes box loss, objectness loss, and classification loss on both training and test sets. In addition, precision and recall plots on the test set are also presented, along with mean mAP at the 50% threshold and from 50% to 95%. The dotted lines on the graph are smooth averages, highlighting the general trend of these values during model training and evaluation. The losses (Boxes, Objects, Classifiers) are decreasing for both the training and validation sets, showing that the model is learning effectively.

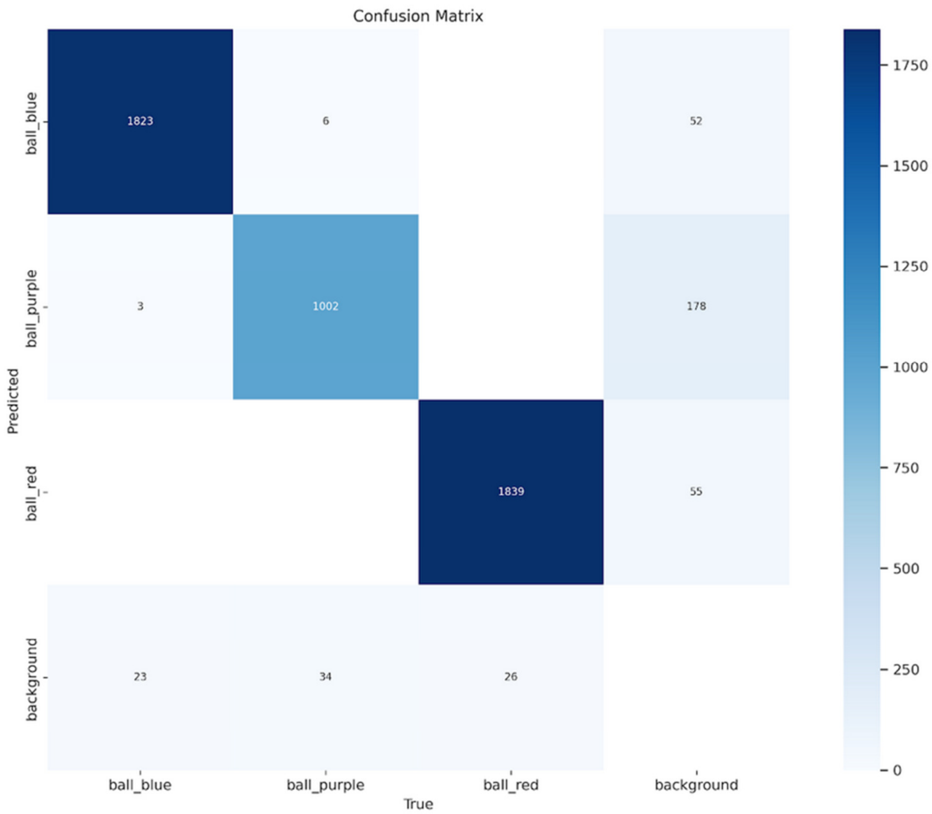

Fig. 4. Confusion matrix of YOLOv8.

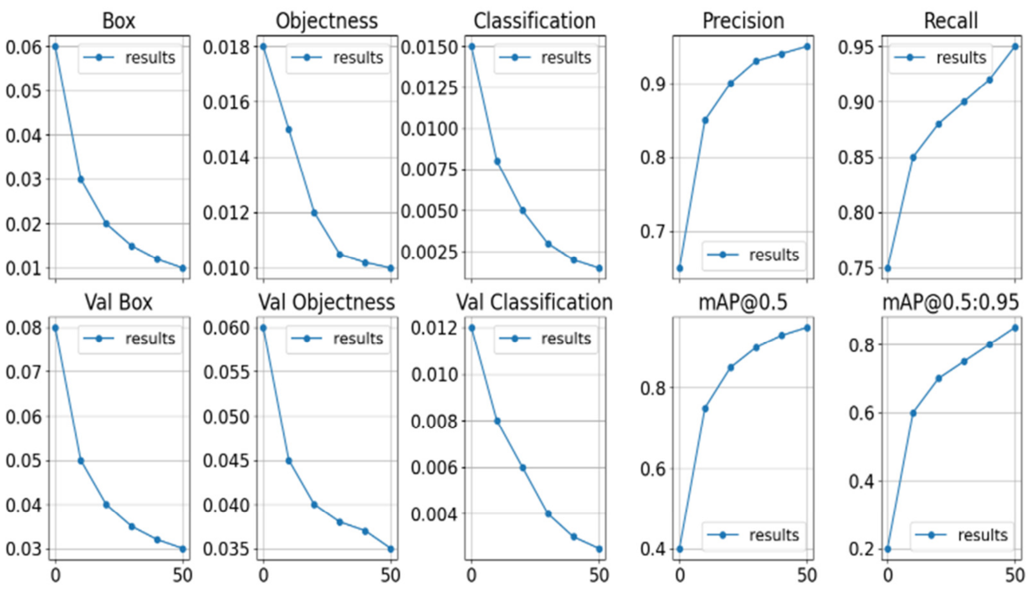

Fig. 5. Training results of the YOLOv7 model.

The evaluation indexes (Precision, Recall, mAP) have all improved proving that the model's performance in object recognition tasks is improving. There is some initial fluctuation in the validation set's object loss, which may indicate overfitting in the early stages, but then the loss decreases, suggesting the possibility of conceptual overfitting.

Figure 6 illustrates the training process and performance metrics of the YOLOv8n model. The image displays the loss and performance plots of the DL model over a series of training epochs. It includes metrics such as box loss, classification loss, and distance prediction loss alongside key evaluation measures like precision and recall rates on the test set. The dotted lines in the graph represent smooth averages, emphasizing the overall trends in these values throughout the training process. These results provide insight into the model's optimization and its ability to generalize effectively to unseen data. The losses (Box, Classification, DFL) are gradually decreasing, proving that the model is learning effectively and optimizing well. The evaluation indexes (Precision, Recall, mAP) all increase and remain stable, showing that the model is improving in predicting and classifying objects. There were no obvious signs of overfitting as the training and validation metrics improved uniformly.

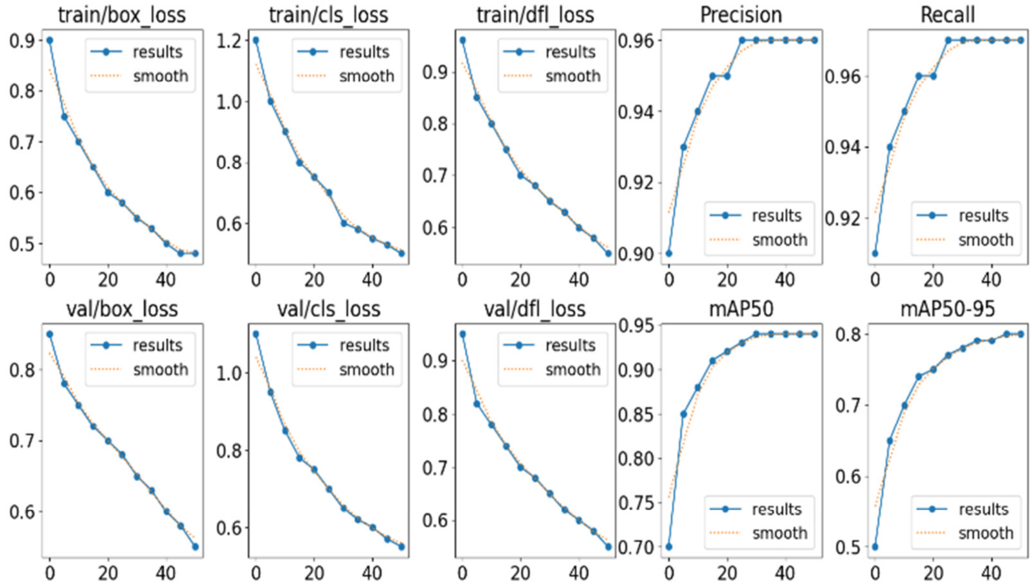

Fig. 6. Training results of the YOLOv8n model.

It can be concluded that YOLOv8 is more efficient and more suitable for real-time applications. The training images of the YOLOv8 model provide clearer and more detailed chart labels, additional smooth lines for better trend observation, and include important indicators such as DFL Loss. Furthermore, performance indicators are more clearly defined, resulting in a more stable and interpretable pattern. This makes it easier to monitor, analyze, and evaluate model performance compared to earlier stages.

IV. CONCLUSION

This study successfully implemented and assessed the YOLOv8 model for object detection tasks, particularly focusing on the detection and classification of balls in diverse environments. The training results and evaluation metrics highlight the model's ability to learn effectively and generalize well to unseen scenarios. The performance plots reveal consistent reductions in losses (Box, Classification, and Distance Prediction Loss) throughout the training process, signifying efficient optimization without evidence of overfitting. Additionally, evaluation metrics such as Precision, Recall, and mean Average Precision (mAP) showed steady improvement and stability, further demonstrating the model's accuracy and reliability in object detection.

Compared to its predecessor YOLOv7, the YOLOv8 model demonstrates significant advancements across multiple dimensions, making it a superior choice for real-time object detection tasks. First and foremost, YOLOv8 achieves more stable and consistent performance metrics, as evidenced by smoother and more predictable training and validation loss curves. These trends indicate better convergence during training and reduced fluctuations, which simplify analysis and monitoring of the training process. YOLOv8 also demonstrates substantial improvements in inference speed, resulting in an optimal balance between high detection accuracy and reduced computational complexity, making it particularly valuable for real-time applications. The model’s faster convergence and reduced overfitting compared to YOLOv7 ensure efficient training with minimal resource consumption.

In addition to these technical improvements, YOLOv8's ability to handle diverse, challenging environments further underscores its advancement. The model maintains high detection accuracy even in scenarios involving poor lighting, occlusions, and scale variations, where YOLOv7 struggles to maintain robustness. These enhancements in precision, efficiency, and overall robustness solidify YOLOv8 as a more reliable model for modern object detection tasks.

Finally, the findings from our comparative study validate YOLOv8's efficiency and robustness in tackling real-world object detection scenarios. The enhanced performance in both training and testing makes it an ideal solution for applications in industrial automation, robotics, environmental surveillance, and beyond. Moving forward, future research could build on these results by exploring multi-class object detection, optimizing computational efficiency, and incorporating advanced methodologies to further improve YOLOv8's capabilities in dynamic, complex environments. In conclusion, YOLOv8 not only offers significant advancements in detection accuracy, speed, and robustness but also ensures faster training, making it a more effective and reliable choice for real-time applications.

DATA AVAILABILITY

Training and testing data and results are available from the authors upon request.

REFERENCES

[1] P. Goswami, L. Aggarwal, A. Kumar, R. Kanwar, and U. Vasisht, "Real-time evaluation of object detection models across open world scenarios," *Applied Soft Computing*, vol. 163, Sep. 2024, Art. no. 111921, https://doi.org/10.1016/j.asoc.2024.111921.

[2] R. Gaddam, A. R. Bavanthula, M. K. Asoori, D. P. Chintakunta, and A. Volki, "A Comprehensive Object Detection Model Empowered with Deep Learning," in *IEEE International Conference on Computing*, Langkawi, Malaysia, Oct. 2023, pp. 380–385, https://doi.org/10.1109/ICOCO59262.2023.10397902.

[3] W. Tarimo, M. M.Sabra, and S. Hendre, "Real-Time Deep Learning-Based Object Detection Framework," in *IEEE Symposium Series on Computational Intelligence*, Canberra, ACT, Australia, Dec. 2020, pp. 1829–1836, https://doi.org/10.1109/SSCI47803.2020.9308493.

[4] A. Singh Rajawat *et al.*, "Reformist Framework for Improving Human Security for Mobile Robots in Industry 4.0," *Mobile Information Systems*, vol. 2021, no. 1, 2021, Art. no. 4744220, https://doi.org/10.1155/2021/4744220.

[5] N. Kalra, P. Verma, and S. Verma, "Advancements in AI based healthcare techniques with FOCUS ON diagnostic techniques," *Computers in Biology and Medicine*, vol. 179, Sep. 2024, Art. no. 108917, https://doi.org/10.1016/j.compbiomed.2024.108917.

[6] T. B. Sheridan, "Human–Robot Interaction: Status and Challenges," *Human Factors*, vol. 58, no. 4, pp. 525–532, Jun. 2016, https://doi.org/10.1177/0018720816644364.

[7] J. Redmon and A. Farhadi, "YOLOv3: An Incremental Improvement." arXiv, Apr. 08, 2018, https://doi.org/10.48550/arXiv.1804.02767.

[8] A. Afdhal, K. Saddami, S. Sugiarto, Z. Fuadi, and N. Nasaruddin, "Real-Time Object Detection Performance of YOLOv8 Models for Self-Driving Cars in a Mixed Traffic Environment," in *2nd International Conference on Computer System, Information Technology, and Electrical Engineering*, Banda Aceh, Indonesia, Aug. 2023, pp. 260–265, https://doi.org/10.1109/COSITE60233.2023.10249521.

[9] W. Hong, Z. Ma, B. Ye, G. Yu, T. Tang, and M. Zheng, "Detection of Green Asparagus in Complex Environments Based on the Improved YOLOv5 Algorithm," *Sensors*, vol. 23, no. 3, Jan. 2023, Art. no. 1562, https://doi.org/10.3390/s23031562.

[10] J. Wang *et al.*, "Research on Improved YOLOv5 for Low-Light Environment Object Detection," *Electronics*, vol. 12, no. 14, Jan. 2023, Art. no. 3089, https://doi.org/10.3390/electronics12143089.

[11] M. Talib, A. H. Y. Al-Noori, and J. Suad, "YOLOv8-CAB: Improved YOLOv8 for Real-time object detection," *Karbala International Journal of Modern Science*, vol. 10, no. 1, Jan. 2024, Art. no. 5, https://doi.org/10.33640/2405-609X.3339.

[12] G. Yang, J. Wang, Z. Nie, H. Yang, and S. Yu, "A Lightweight YOLOv8 Tomato Detection Algorithm Combining Feature Enhancement and Attention," *Agronomy*, vol. 13, no. 7, Jul. 2023, Art. no. 1824, https://doi.org/10.3390/agronomy13071824.

[13] R. Kaur and S. Singh, "A comprehensive review of object detection with deep learning," *Digital Signal Processing*, vol. 132, Jan. 2023, Art. no. 103812, https://doi.org/10.1016/j.dsp.2022.103812.

[14] A. Wang *et al.*, "YOLOv10: Real-Time End-to-End Object Detection." arXiv, Oct. 30, 2024, https://doi.org/10.48550/arXiv.2405.14458.

[15] 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, Jul. 2023, Art. no. 677, https://doi.org/10.3390/machines11070677.

[16] R. Varghese and M. Sambath, "YOLOv8: A Novel Object Detection Algorithm with Enhanced Performance and Robustness," in *International Conference on Advances in Data Engineering and Intelligent Computing Systems*, Chennai, India, Apr. 2024, pp. 1–6, https://doi.org/10.1109/ADICS58448.2024.10533619.

[17] B. Ma *et al.*, "Using an improved lightweight YOLOv8 model for real-time detection of multi-stage apple fruit in complex orchard environments," *Artificial Intelligence in Agriculture*, vol. 11, pp. 70–82, Mar. 2024, https://doi.org/10.1016/j.aiia.2024.02.001.

[18] V.-T. Nguyen, C.-D. Do, T.-D. Hoang, D.-T. Nguyen, and N. T. Le, "Person Detection for Monitoring Individuals Accessing the Robot Working Zones Using YOLOv8," in *International Conference on Green Technology and Sustainable Development*, Ho Chi Minh, Vietnam, Jul. 2024, pp. 50–59, https://doi.org/10.1007/978-3-031-76197-3_5.

[19] V.-T. Nguyen, C.-D. Do, T.-V. Dang, T.-L. Bui, and P. X. Tan, "A comprehensive RGB-D dataset for 6D pose estimation for industrial robots pick and place: Creation and real-world validation," *Results in Engineering*, vol. 24, Dec. 2024, Art. no. 103459, https://doi.org/10.1016/j.rineng.2024.103459.

[20] V.-T. Nguyen and D.-T. Chu, "Implementation of Real-Time Human Tracking System Based on Deep Learning Using Kinect Camera," in *Intelligent Systems and Networks*, N. L. Anh, S.-J. Koh, T. D. L. Nguyen, J. Lloret, and T. T. Nguyen, Eds. New York, NY, USA: Springer, 2022, pp. 230–236.

[21] V.-T. Nguyen and D.-T. Chu, "Study on Tracking Real-Time Target Human Using Deep Learning for High Accuracy," *Journal of Robotics*, vol. 2023, no. 1, 2023, Art. no. 9446956, https://doi.org/10.1155/2023/9446956.

[22] V.-T. Nguyen, D.-T. Chu, D.-H. Phan, and N.-T. Tran, "An Improvement of the Camshift Human Tracking Algorithm Based on Deep Learning and the Kalman Filter," *Journal of Robotics*, vol. 2023, no. 1, 2023, Art. no. 5525744, https://doi.org/10.1155/2023/5525744.

[23] H.-R. Sun, B.-J. Shi, Y.-T. Zhou, J.-H. Chen, and Y.-L. Hu, "A Smoke Detection Algorithm Based on Improved YOLO v7 Lightweight Model for UAV Optical Sensors," *IEEE Sensors Journal*, vol. 24, no. 16, pp. 26136–26147, Dec. 2024, https://doi.org/10.1109/JSEN.2024.3422509.

[24] J. A. Vicente-Martinez, M. Marquez-Olivera, A. Garcia-Aliaga, and V. Hernandez-Herrera, "Adaptation of YOLOv7 and YOLOv7_tiny for Soccer-Ball Multi-Detection with DeepSORT for Tracking by Semi-Supervised System," *Sensors*, vol. 23, no. 21, Jan. 2023, Art. no. 8693, https://doi.org/10.3390/s23218693.