Real-Time Multimodal Object Detection for Robotic Perception: Integrating YOLOv8n with an Intel RealSense D455 Depth Camera

Ratna Dwi Lestari1*, Anugerah Wibisana1*

* Politeknik Negeri Batam
Robotics Engineering Study Program
Parkway Street, Batam Centre, Batam 29461, Indonesia
E-mail: ratnadwi.1505@gmail.com

Abstrak

Abstrak ini membahas pengembangan sistem deteksi objek untuk mendukung persepsi visual pada robot delivery yang beroperasi di lingkungan dengan variasi kondisi. Deteksi objek yang andal diperlukan agar robot dapat bernavigasi dan berinteraksi secara aman dengan lingkungan sekitarnya. Sistem yang diusulkan menggunakan kamera kedalaman Intel RealSense D455 yang terintegrasi dengan model YOLOv8n untuk memproses citra RGB dan depth colormap. Dataset terdiri dari skenario RGB Indoor, RGB Outdoor, dan Depth Colormap Outdoor dengan variasi jarak serta orientasi objek. Seluruh data dianotasi menggunakan Roboflow. Implementasi dilakukan pada Ubuntu 22.04 dengan Python 3.10 dan mencapai kinerja inferensi real-time minimal 30 FPS pada HP Victus 15 FA0116TX. Integrasi dengan ROS 2 Humble memungkinkan komunikasi yang efisien antara modul persepsi dan navigasi,, sehingga meningkatkan keandalan deteksi objek pada kondisi pencahayaan yang beragam.

Kata kunci: YOLOv8n, Deteksi Objek, Pembelajaran Mendalam, Persepsi Robot, Kamera RealSense

Abstract

This abstract discusses the development of an object detection system to support visual perception in delivery robots operating in environments with varying conditions. Reliable object detection is necessary for the robot to safely navigate and interact with the surrounding environment. The proposed system uses an Intel RealSense D455 depth camera integrated with the YOLOv8n model to process RGB imagery and depth colormap. The dataset consists of RGB Indoor, RGB Outdoor, and Depth Colormap Outdoor scenarios with variations in distance and object orientation. All data is annotated using Roboflow. The implementation was carried out on Ubuntu 22.04 with Python 3.10 and achieved real-time inference performance of at least 30 FPS on the HP Victus 15 FA0116TX. Integration with Humble's ROS 2 enables efficient communication between perception and navigation modules, thereby improving the reliability of object detection under diverse lighting conditions.

Keywords: YOLOv8n, Object Detection, Deep Learning, Robotic Perception, RealSense Camera

1. Introduction

Computer vision and robotics play an essential role in the development of autonomous systems [1]. To operate safely in human-centered environments, delivery robots must possess robust perception capabilities, particularly in object detection, which enables them to identify and respond to dynamic elements in real-time. Conventional RGB-based approaches often face difficulties in poor illumination, whereas depth cameras offer higher robustness under varying lighting conditions such as shadow and reflection [2]. Among available sensors, the Intel® RealSense™ D455 is widely recognized for its ability to capture both RGB and depth data, making it suitable for robotic perception and mapping tasks [3], [10]. Recent advances in deep learning, especially the YOLO (You Only Look Once) family of algorithms, have significantly altered real-time perceptions [4]. YOLOv8n, developed by Ultralytics, is a lightweight variant that balances accuracy and computing efficiency, making it suitable for resource-constrained platforms such as mobile robots [5]. Although depth cameras can provide reliable sensing in a variety of environments, the integration of depth-only data with

object detection models is still underexplored [2]. Most of the existing research focuses on RGB-based training, which limits endurance in low-light environments. This limitation is particularly relevant for delivery robots that must operate safely at different times of the day, where lighting conditions can vary drastically. To overcome this challenge, the study implemented an object detection pipeline that primarily leverages large-scale RGB datasets, complemented by depth colormap images from the Intel RealSense D455 for low-light scenarios. The datasets were collected systematically at various times and distances, annotated, and trained using YOLOv8n. The system was developed and tested in the ROS 2 Humble environment, with the aim of achieving reliable real-time object detection on simple computing hardware.

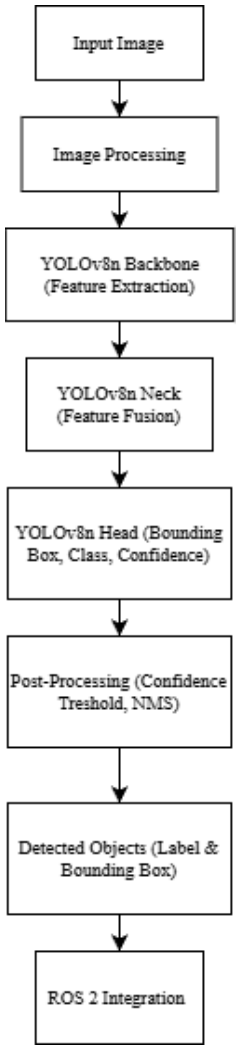
Several studies have investigated object detection in robotic perception. The original YOLO framework was introduced in 2016, enabling real-time object detection on GPU, followed by successive improvements in computing accuracy and efficiency [4], [6], [7]. YOLOv8 Ultralytics expands on this development with a modular design and optimized training channels [5]. In terms of sensing, Intel RealSense cameras have been widely adopted for tasks such as SLAM (Simultaneous Localization and Mapping), navigation, and motion recognition [9], [10]. However, research that specifically uses depth colormaps for object detection is still limited. Previous works have often blended RGB with depth or relied on LiDAR-based approaches, which increased the cost and complexity of the system [8].

This paper contributes to the development of an object detection system that utilizes large-scale RGB datasets, with colormap depth data from the Intel RealSense D455 depth camera combined as a complementary modality to improve detection resistance in low-light conditions. Contributions include: (i) a methodology for collecting structured datasets across multiple times and distances, (ii) the integration of YOLOv8n with ROS 2 Humble to support real-time inference in robotics applications, and (iii) evaluation of detection performance using RGB data and depth colormap to improve the reliability of robotic perception systems in dynamic environments. The objective of this research is to design, train, and evaluate a real-time human detection system using the Intel RealSense D455 depth camera and the YOLOv8n model within a ROS 2 environment, as a vision module prototype for future integration into a delivery robot platform.

2. Method

Our research applies a systematic pipeline for real-time object detection in robotic systems, designed to support the perception needs of delivery robots. This methodology includes the preparation of diverse image datasets, optimization of lightweight deep learning models, and integration of trained model models into robotic systems for hands-on evaluation. This approach aims to ensure robust, efficient, and applicable solutions in a real environment.

In this study, YOLOv8n (You Only Look Once version 8 – nano) was used as the core of the object detection system. YOLOv8n is a single-stage object detection method that is capable of detecting in a single inference, making it ideal for real-time applications. The RGB imagery and depth colormap obtained from the Intel RealSense D455 camera first go through the preprocessing and normalization stages before being processed by the model. Next, YOLOv8n extracts visual features using a lightweight CNN architecture optimized for compute speed, and then simultaneously predicts the bounding box, object class, and confidence score on each frame. The detection results are filtered using threshold confidence to reduce prediction errors, and the best model is used for real-time inference and is integrated into the ROS 2 Humble framework for use by the robot's navigation module. The overall the workflow of YOLOv8n is illustrated in Figure 1 (a) and workflow of this study is illustrated in Figure 1 (b).



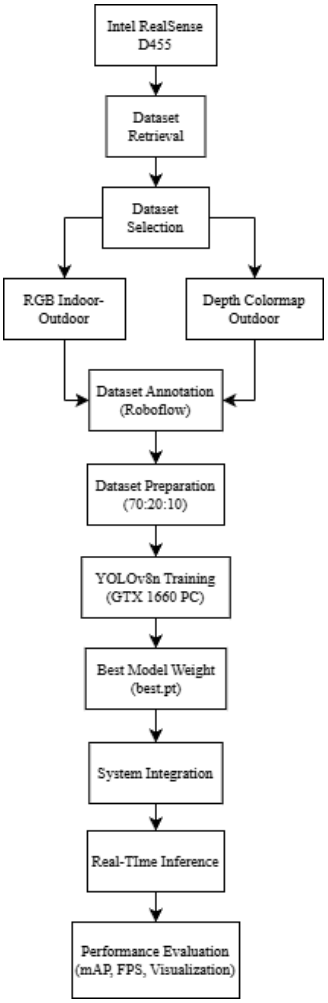


Figure 1: (a) YOLOv8n Diagram Block and (b) Diagram Block of Object Detection System

A. Dataset Preparation

The dataset was collected using the Intel RealSense D455 depth camera, which is capable of recording colormaps and RGB depth simultaneously. The collection of the datasets is carried out in both indoor and outdoor environments to ensure a diversity of lighting and scenery. RGB Indoor datasets are collected under bright and moderate artificial lighting that captures objects such as humans, tables, and chairs. RGB Outdoor datasets are collected at different times in the morning and afternoon that capture objects such as humans, cars, motorcycles, and trees. In addition, the Depth Colormap Outdoor datasets are collected at night to overcome the limitations of detection in low-light conditions.

Objects such as chairs, tables, and trees are still included in the dataset even though they are less directly relevant to the delivery robot. The selection of these objects is intended as distractor objects, which are environmental elements that the model must recognize so that they are not misclassified as humans. With these objects, the model can be trained more robustly in distinguishing humans from static objects around the robot's movement path.

To capture environmental variations, objects were recorded from three different distances: close (0.6-2 m), medium (2-4 m), and far (4-6 m). Each object is also recorded from three camera orientations: left, center, and right. This systematic data collection method ensures that the datasets represent a diverse range of conditions that delivery robots will face in real-world applications. A summary of the dataset composition is provided in Table 1.

TABLE I

THE NUMBER OF OBJECT DATASETS COLLECTED BASED ON TIME, TYPE, AND DISTANCE OF OBJECT

Dataset Type	Time of Day	Objects Captured	Distance Variations	Total Images
RGB Indoor	Bright, Moderate Indoor Lighting (for Humans Only)	Humans, Tables, Chairs	Close, Medium, Far	1.576
RGB Outdoor	Morning, Afternoon	Humans, Cars, Motorcycles, Trees	Close, Medium, Far	5.748
Depth Colormap Outdoor	Night (low-light conditions)	Humans, Cars, Motorcycles, Trees	Close, Medium, Far	2.836
Total				10.160

Table 1 shows the distribution of the number of datasets used in this study. It can be seen that RGB Outdoor data has the highest number of datasets (5.748 datasets), followed by Depth Colormap Outdoor (2.836 datasets), while RGB Indoor has the fewest number of datasets (1.576 datasets). The imbalance in the number of datasets is influenced by data collection conditions: RGB Outdoor is more varied because it includes morning and afternoon times; depth colormap can only be taken at night in low light; while RGB Indoor is limited by space variation.

All images are annotated using the Roboflow platform, which provides an efficient interface for labeling bounding boxes around target objects. Annotations follow the YOLO format, ensuring compatibility with the YOLOv8n training flow. Each class of object (e.g., humans, cars, motorcycles, trees) is labeled separately for consistency across the entire RGB colormap dataset and depth. The preparation of this structured dataset ensures reliable training and fair evaluation of model performance across a wide range of input modalities.

This difference in the distribution of this dataset has an impact on model performance. Models tend to show higher accuracy on RGB Outdoor data because the

number of datasets is larger and the variety is more diverse. The Depth Colormap Outdoor dataset has the potential to result in slightly lower performance due to its relatively limited number, although it remains important as a complement to nighttime lighting conditions. Meanwhile, the RGB Indoor dataset has the least number and more homogeneous objects, so its performance is expected to be stable in controlled scenarios, but its contribution to the model is more limited compared to the other two dataset categories.

B. Model Training

Training Environment. Model training was conducted using PCs with NVIDIA GTX Geforce 1660-based GPUs that support CUDA [11], allowing the training process to run faster than just using CPUs. The Geforce GTX 1660 GPU specification includes a mid-range GPU that is enough to handle YOLOv8n training on large-scale datasets.

The hardware and software specifications used for training are as follows:

TABLE II

DEVICE SPECIFICATIONS USED FOR TRAINING DATASET

Hardware (Training PC)	Software (Training PC)	Training Setup
GPU Geforce GTX 1660 (CUDA Cores)	Ubuntu 22.04 LTS (64-bit) Operating System	Conda Virtual Environment (Environment Manager)
Memory Bus 192-bit, Memory Bandwidth 192 GB/s	Python 3.10 Program Language	Roboflow for Dataset Annotation Tool
Storage 512 GB NVMe SSD	Framework PyTorch 2.0 with CUDA Support	Ultralytics Built-in Training Logs (results.csv) and TensorBoard Visualization
Thermal Design Power 120 W	YOLO Framework Ultralytics YOLOv8	
Compute Capability 7.5	OpenCV, NumPy, Torchvision, Matplotlib, Pandas Supporting Libraries	

The model used in this study is YOLOv8n (nano), chosen because it is lightweight, has a relatively small parameter size, and is able to run in real-time on devices with limited specifications. This model was chosen based on the need for a delivery robot system that must remain responsive even when using only non-high-end GPU hardware. With GTX Geforce 1660 GPU support, the training process on large datasets can be completed faster, enabling more optimal parameter exploration and validation. The trained model is used in the ROS 2 Humble environment running on Ubuntu 22.04. RealSense D455 is integrated using RealSense's ROS 2 pipeline to stream synchronized RGB and depth data. ROS 2 Humble is implemented to perform real-time inferences using the YOLOv8n model. HP Victus 15 intel CORE i5 with NVIDIA Geforce GTX 1650 is used as an implementation of the dataset train results to the camera. The system achieves real-time inference of at least 30 FPS when processing RGB streams and slightly lower performance when processing depth colormap data due to pre-processing overhead.

To maintain fairness, each subset of the dataset is separated by a 70:20:10 ratio (exercise, validation, testing), and evaluations are performed on each modality separately. Boundary box annotations are done using Roboflow, with the output format YOLO TXT.

C. Evaluation Metrics

To evaluate the performance of the YOLOv8n model, we use several standard metrics that are widely used in object detection. All of these metrics are based on the calculation of the Unity Intersection (IoU) between the prediction bounding box and the basic truth as the following equation:

1. Intersection of Union (IoU)

$$IoU = \frac{Area\ (Bpred \cap Bgt)}{Area\ (Bpred \cup Bgt)} \quad (1)$$

In equation (1), *Bpred* represents the bounding box of the predicted outcome and *Bgt* represents the bounding box of the basic truth. A prediction is categorized as True Positive if the IoU value ≥ a specified threshold (e.g. 0.5).

2. Precision

Precision serves as measure of the reliability of the model's positive predictions. It quantifies the ratio of correct detections to the total number of detections made by the model. A high precision score indicates that when the model identifies the object, it is highly likely to be correct.

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

In equation (2), *TP (True Positives)* represents the count of instances where the model correctly identified an object, and *FP (False Positives)* represents the count of incorrect identifications, where the model detected an object that was not actually present.

Recall

Recall measures the model's ability to locate all

relevant objects within an image. It is calculated as the ratio of correctly identified objects to the total number of actual objects that should have been detected. A high recall score suggests the model misses very few objects.

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

In equation (3), *TP (True Positives)* are correctly detected objects, and *FN (False Negatives)* are actual objects that the model failed to detect.

F1-Score

The F1-Score provides a single, balanced metric that combines Precision and Recall. By calculating their harmonic mean, the F1-Score is particularly useful for evaluating a model when an uneven class distribution exists, ensuring the model performs well on both metrics.

$$F1 - Score = 2\ x\ \frac{Precision\ x\ Recall}{Precision+Recall} \quad (4)$$

3. Average Precision (AP)

For a specific object class, Average Precision (AP) provides a summarized performance score across all possible detection confidence thresholds. This value is derived from the area under the Precision-Recall curve, offering a single, robust measure of the model's accuracy for that class.

$$AP = \sum_{n=1}^{N}(R_n - R_{n-1}).P_n \quad (5)$$

In equation (5), P_n and R_n are the Precision Recall values at the *n*-th threshold.

4. mean Average Precision (mAP)

The mean Average Precision (mAP) is the final, comprehensive metric used to evaluate the overall performance of the model across all object classes. It is computed by averaging the AP values for each individual class, providing a single score that reflects the model's general performance across the entire dataset.

$$mAP = \frac{1}{N} + \sum_{i=1}^{N} APi \quad (6)$$

In equation (6), *N* is the total number of object classes and *APi* is the Average Precision for a specific class *i*.

5. mAP@0.5

mAP@0.5 is calculated with the IoU threshold = 0.5.

$$mAP@0.5 = \frac{1}{N}\sum_{i=1}^{N} APi\ (IoU \geq 0.5) \quad (7)$$

6. mAP@0.5:0.95

mAP@0.5:0.95 is the average mAP at the ten IoU thresholds (0.5 to 0.95 with a step of 0.5).

$$mAP@0.5{:}\,0.95 = \frac{1}{10N}\sum_{j=0}^{N} \quad \sum_{i=1}^{N} APi\ (IoU = 0.5 + 0.95j) \quad (8)$$

3. Results

The following are the results of the training dataset.

A. Training Process Visualization

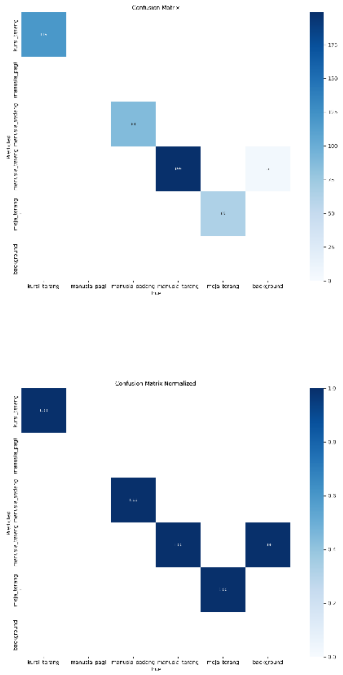


Figure 2: (a) Confusion Matrix Dataset RGB Indoor and (b) Confusion Matrix Normalized Dataset RGB Indoor

To provide a deeper understanding of the class-based classification accuracy of the trained YOLOv8n model, we analyzed the Confusion Matrix and the Normalized Confusion Matrix.

The Confusion Matrix in Figure 2 (a) shows that the YOLOv8n model is able to detect all classes of objects well in indoor scenarios. Class kursi_terang was detected perfectly with 119 correct predictions, while class meja_terang also obtained high results with 65 correct predictions. For human objects, the model managed to recognize manusia_sedang (88 correct predictions) and manusia_terang (199 correct predictions) without significant classification errors.

Meanwhile, in Figure 2 (b) shows that all classes achieved an accuracy of 1.00 (100%), indicating that there was no misclassification between classes in this scenario. These results show that stable indoor lighting and controlled background conditions contribute significantly to optimal detection performance.

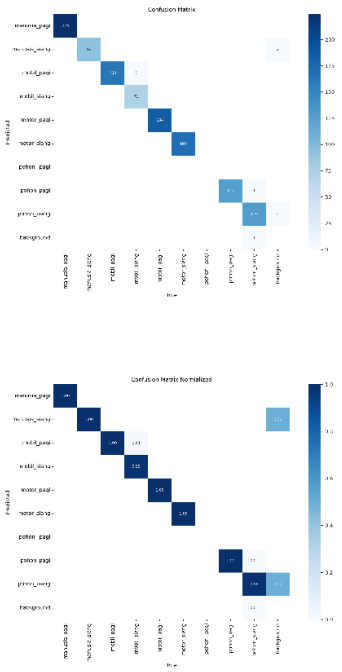
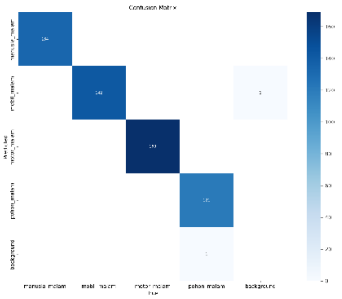


Figure 3: (a) Confusion Matrix Dataset RGB Outdoor and (b) Confusion Matrix Normalized Dataset RGB Outdoor

Figure 3 (a) shows that the model is able to recognize most classes with high accuracy. Classes manusia_pagi, manusia_siang, motor_pagi, and motor_siang were perfectly detected with precision and recall values close to 1.00. Classes mobil_pagi and mobil_siang are also well classified, although there is a slight misclassification to other classes (≤1%). For the vegetation class, pohon_pagi showed almost perfect performance with an accuracy of ≈ 0.99, while in pohon_siang there was still confusion with the background (around 2% - 50% relative error). Overall, in Figure 3 (b) shows that the model maintained stable performance in almost all classes, with a high average accuracy (>0.95).



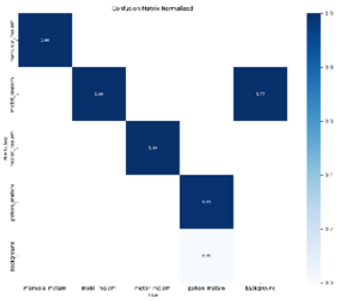
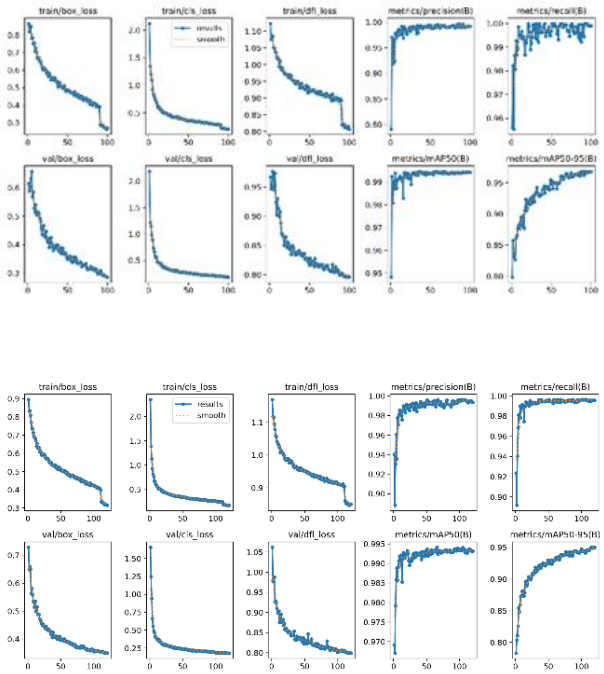


Figure 4: (a) Confusion Matrix Dataset RGB Depth Colormap Outdoor and (b) Confusion Matrix Dataset RGB Depth Colormap Outdoor

Figure 4 (a) shows that the model managed to recognize classes manusia_malam (134 samples), mobil_malam (142 samples), motor_malam (169 samples), and pohon_malam (121 samples) with an error rate of only one which was the prediction in class mobil_malam and one class pohon_malam detected as the background. Figure 4 (b) shows the accuracy per class with all major classes having an accuracy of almost 1.00 (≥0.99), indicating that the model is capable of detecting objects at night with high precision and minimal misclassification.

Next, we display the results graph from the train data. This graph provides visual evidence of the success and stability of the model's training process.



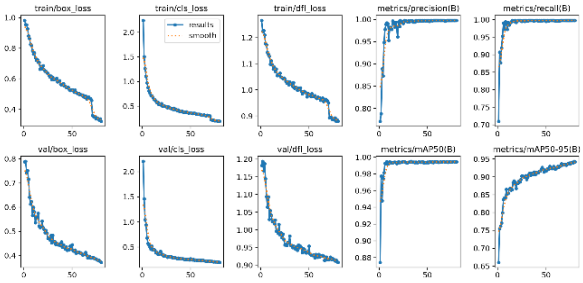


Figure 5: (a) Results of dataset RGB Indoor train graph, (b) Results of dataset RGB Outdoor train graph, and (c) Results of dataset Depth Colormap Outdoor train graph

A value of mAP@0.5 close to 1.0 on all datasets (> 0.99) is an indication of outstanding model performance. These results, although they look almost perfect, do not directly indicate the presence of overfitting. Instead, these values can be explained through an in-depth analysis of the training curve and dataset characteristics. First, from the training visualizations (see **Figure 5 (a)**, **Figure 5 (b)**, and **Figure 5 (c))** the validation loss curve (val/box_loss, val/cls_loss, val/dfl_loss) showing a consistent downward trend and parallel to the validation loss curve at the end of the epoch is strong evidence that the model does not simply memorize training data, but is able to generalize effectively on data that has never been seen before.

Second, the high value of this mAP is supported by the characteristics of a clean and controlled dataset. The dataset is collected with planned variations (distance, angle, and lighting) but with target objects that have clear visualizations and are easy to distinguish.

A. Model Performance Evaluation

TABLE III

SUMMARY OF THE AGGREGATE PERFORMANCE OF YOLOV8N MODEL ON VARIOUS DATASETS

Dataset Type	Metric	Precision	Recall	mAP@ 0.5	mAP@ 0.5:0.95
RGB Indoor	Overall	0.992	0.999	0.994	0.967
RGB Outdoor	Overall	0.998	0.995	0.993	0.954
Depth Colormap Outdoor	Overall	0.998	0.998	0.995	0.912

Table 3 presents a summary of the aggregate performance of the YOLOv8n model trained on three different datasets. In general, the model showed very strong performance in all scenarios, with consistently high Precision, Recall, and mAP values.

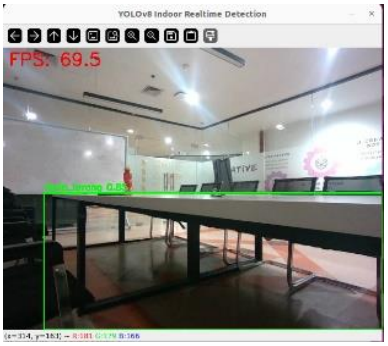
Models trained with the RGB Indoor dataset showed the best performance with mAP@0.5 of 0.994 and mAP@0.5:0.95 of 0.967. This high value is due to stable lighting conditions and a more uniform background in the indoor environment, which makes it easier for the model to learn and generalize.

The performance of the RGB Outdoor model is slightly lower on the stricter metric, which is mAP@0.5:0.95 by 0.954. This can be attributed to greater environmental variations outdoors, such as changes in natural lighting in the morning and during the day, as well as the visual diversity of objects.

The model trained on the Depth Colormap Outdoor dataset also showed excellent performance with a mAP@0.5 of 0.995, but with the most significant decrease at mAP@0.5:0.95 to 0.912. This decrease reflects the inherent challenges and processing of depth data, which, while effective in low-light conditions, has different visual characteristics from RGB images, such as a lack of texture and color detail.

B. Real-Time Inference analyzed the Range of Performance Demonstration

The following are the results of the real-time object detection.



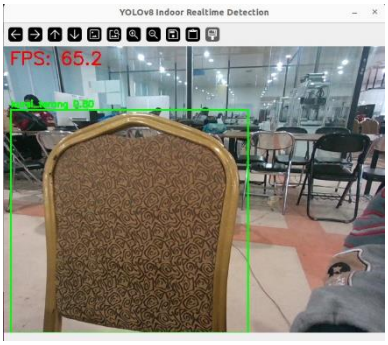


Figure 6: (a) RGB Indoor table object detection results, (b) RGB Indoor chair object detection results, and (c) RGB Indoor human object detection results

Figure 6 (a) depicts the results of detecting RGB Indoor table object using YOLOv8n at a frame rate of 69,5 FPS. The system correctly identified the table item with the meja_terang label at a confidence level of 0.83, as indicated by a green bounding box. Figure 6 (b) depicts the results of detecting RGB Indoor chair object at a rate of 65,2 FPS. The algorithm successfully recognizes the kursi_terang object with a confidence level of 0.80. Figure 6 (c) depicts the results of detecting RGB Indoor human object at a rate of 37.9 FPS. The algorithm correctly identified the manusia_terang object with a confidence level of 0.92.

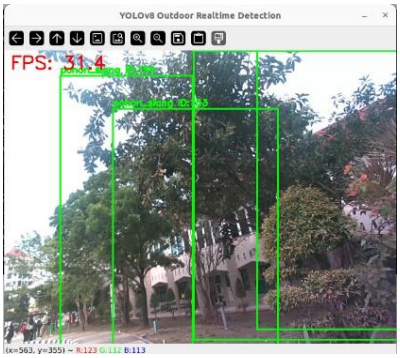
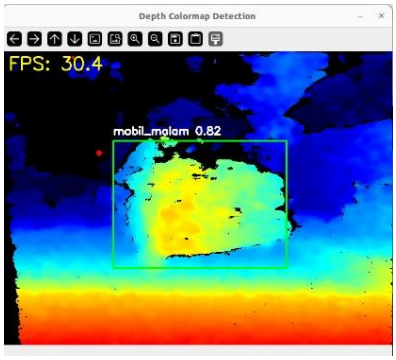
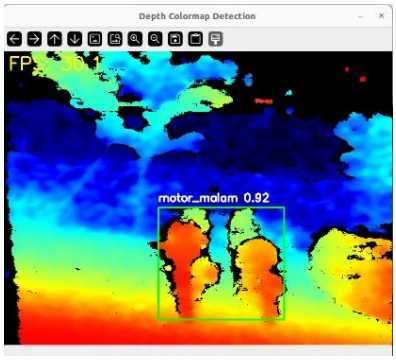


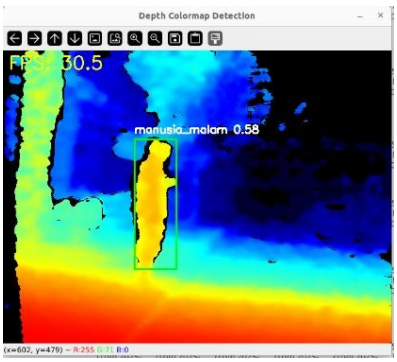
Figure 7: (a) RGB Outdoor tree object detection results, (b) RGB Outdoor motorcycle object detection results, (c) RGB Outdoor car object detection results, and (d) RGB Outdoor human object detection results

Figure 7 (a) depicts the results of detecting RGB Outdoor objects using YOLOv8n at a rate of 31.4 FPS. The system recognized many objects with pohon_siang labels indicated by a green bounding box and a confidence level greater than 0.80. Figure 7 (b) depicts the results of the detection of outdoor objects using YOLOv8 at a speed of 122.5 FPS. The system successfully recognized motor_siang object with a confidence level of 0.87, marked by a green bounding box that surrounds motorcyclists in groups. Figure 7 (c) depicts the results of detecting outdoor objects using

YOLOv8n at a very high speed of 227.0 FPS. The system successfully recognized mobil_pagi object with a confidence level of 0.97. Figure 7 (d) depicts the results of detecting outdoor human objects during the day using YOLOv8n at a speed of 151.8 FPS. The system successfully recognized manusia_siang object with a confidence level of 0.64, marked by a green bounding box surrounding an individual in a warepack uniform walking alone.







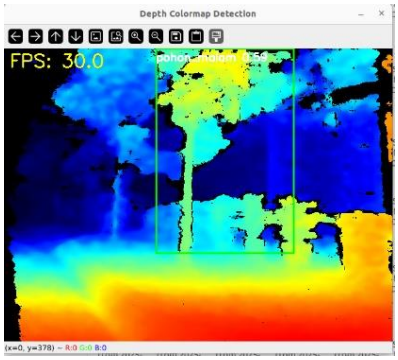


Figure 8: (a) Depth Outdoor car object detection results, (b) Depth Outdoor motorcycles object detection results, (c) Depth Outdoor human object detection results, and (d) Depth Outdoor tree object detection results

Figure 8 (a) shows the results of detecting car objects at night using the YOLOv8n-based Depth Colormap Detection method. The system managed to recognize mobil_malam object with a confidence level of 0.82 with an FPS of 30.4, marked by a green bounding box. The depth colormap visualization shows a variation in distance with color gradation, where red-yellow indicates a closer object (detected car) and dark blue indicates an area farther away from the detection range. Figure 8 (b) shows the results of the detection of motor objects at night. The system successfully recognizes motor_malam object with a confidence level of 0.92, with an FPS of 30.1. Figure (c) shows the results of the detection of human objects at night. The system successfully recognizes manusia_malam object with a confidence level of 0.58, with an FPS of 30.5. Figure 8 (d) shows the results of object detection at night. The system successfully recognizes pohon_malam object with a confidence level of 0.59, with an FPS of 30.0.

To provide a more comprehensive picture of computational efficiency, we analyzed the range of system performance from best-case scenarios to worst-case scenarios. Our system's peak performance, achieved on the Outdoor RGB dataset when detecting large objects such as cars under optimal morning lighting, reached 227 FPS (Figure 7 (c)). This value reflects the potential maximum speed of YOLOv8n under ideal conditions. The FPS value of object detection mobil_pagi reach well above the average FPS of other object detection due to the input resolution factor and the complexity of the features of the car object which are relatively simpler for the model to

recognize in a given test scenario. In addition, these results can appear due to GPU memory caching or testing on lighter batches, therefore, this figure does not necessarily represent the overall performance of the system, but rather the specific conditions during the test. In contrast, the lowest performance was recorded when the system processed the Outdoor Depth Colormap dataset to detect objects with complex shapes such as trees at night. In this scenario, the system operates at 30 FPS (Figure 8 (d)). While this is the lowest performance limit, it's important to emphasize that it still meets real-time standards for robotics applications.

This analysis shows that although the system's performance varies depending on the complexity of the environment and the type of data, it still maintains the minimum speed required for reliable real-time operation. This shows that the model can function effectively in a variety of conditions, from the most ideal to the most challenging.

B. Conclusions

In this study, we successfully developed and implemented a real-time object detection system for delivery robot applications using an Intel RealSense D455 camera and YOLOv8n deep learning model. we focus on a multimodal approach by leveraging RGB data and depth colormap to address the main challenge faced by autonomous robots, namely the limitation of detection in varying lighting conditions, especially at night. RGB Indoor, RGB Outdoor, and Depth Colormap Outdoor with variations in distance and viewing angles. The YOLOv8n model is trained separately for each dataset and evaluated using standard metrics such as Precision, Recall, and mAP. The results of the evaluation showed that all three models achieved very high performance, with a value of mAP@0.5 above 0.99. The performance of the model trained on the RGB dataset shows the highest accuracy, supported by a stable and consistent environment. Qualitative comparative analysis validates our hypothesis, where models trained with Depth Colormap data show superior resistance in low-light conditions. Depth colormap-based models successfully detect objects accurately, showing that depth data is an important modality to ensure reliable perception 24/7. In addition, the system implemented on the ROS 2 Humble managed to achieve real-time performance at a speed of at least 30 FPS, proving the feasibility of its implementation on the robot platform. Overall, this study contributes by: (i) presenting a structured dataset collection methodology for robot perception, (ii) integrating the YOLOv8n model with ROS 2 for real-time robotic applications, and (iii) evaluating detection performance using RGB data and depth colormap to improve reliability in a campus environment. As a suggestion for future research, the system can be further developed by combining both modalities (RGB and depth colormap) into a single model for potential improvements in computational accuracy and efficiency.

Acknowledgment

The author would like to thank the supervisor and all parties who have provided support during the implementation of this research. Appreciation was also expressed to the Robotics Engineering Study Program for the facilities and academic environment that supported the implementation of this research. Thanks are also given to colleagues who have contributed through discussions, technical assistance, and constructive suggestions during the system development process. This research is part of academic activities.

References

[1] R. Szeliski, Computer Vision: Algorithms and Applications. London: Springer, 2010.

[2] K. Khoshelham and S. O. Elberink, “Accuracy and resolution of Kinect depth data for indoor mapping applications,” Sensors, vol. 12, no. 2, pp. 1437–1454, 2012.

[3] Intel Corporation, Intel® RealSense™ Depth Camera D455 Datasheet, 2020. [Online]. Available: https://www.intelrealsense.com/depth-camera-d455/

[4] J. Redmon, S. Divvala, R. Girshick, and A. Farhadi, “You only look once: Unified, real-time object detection,” in Proc. IEEE Conf. Computer Vision and Pattern Recognition (CVPR), 2016, pp. 779–788.

[5] G. Jocher, A. Chaurasia, and J. Qiu, “YOLO by Ultralytics,” GitHub repository, 2023. [Online]. Available: https://github.com/ultralytics/ultralytics

[6] E. Bochkovskiy, C. Y. Wang, and H. Y. M. Liao, “YOLOv4: Optimal speed and accuracy of object detection,” arXiv preprint arXiv:2004.10934, 2020.

[7] Y. Wu et al., “Detectron2,” Facebook AI Research,2019.[Online].Available: https://github.com/facebookresearch/detectron2

[8] Y. Zhu et al., “Deep learning-based human activity recognition: A survey,” ACM Computing Surveys, vol. 55, no. 1, pp. 1–34, 2022.

[9] Intel Corporation, “Intel RealSense SDK 2.0,” 2022.[Online].Available: https://github.com/IntelRealSense/librealsense

[10] D. Keselman, J. Iselin Woodfill, A. Grunnet-Jepsen, and L. B. Machula, “Intel RealSense stereoscopic depth cameras,” in Proc. IEEE Conf. Computer Vision and Pattern Recognition Workshops (CVPRW), 2017, pp. 1267–1276.

[11] A. Paszke, S. Gross, F. Massa, et al., “PyTorch: An imperative style, high-performance deep learning library,” Advances in Neural Information Processing Systems (NeurIPS), 2019.

[12] M. Abadi et al., TensorFlow: Large-Scale Machine Learning on Heterogeneous Distributed Systems, 2016. [TensorBoard tool included].

[13] G. Bradski, “The OpenCV Library,” Dr. Dobb’s Journal of Software Tools, 2000.

[14] Simanjuntak, W. P. S., & Wibisana, A. (2024). Depth camera-based human detection using YOLOv5. In Proceedings of the 7th International Conference on Applied Engineering (ICAE 2024) (pp. 150–162). Atlantis Press.