Sinkron : Jurnal dan Penelitian Teknik Informatika
Volume 10, Number 3, July 2026
DOI : https://doi.org/10.33395/sinkron.v10i3.16287
e-ISSN : 2541-2019
p-ISSN : 2541-044X

Evaluation of YOLOv8n Performance for Real-time Human Detection on Autonomous Mobile Robots

Alif Daffa Dziqy Riyansah[1)*], Febrian Hadiatna[2)], Ratna Susana[3)]
[1,2,3)] Electrical Engineering, National Institute of Technology, Bandung, Indonesia
[1)] alif.daffa@mhs.itenas.ac.id, [2)] Febrian_Hadiatna@yahoo.com, [3)] ratnassn@itenas.ac.id

Submitted :May 28, 2026 | **Accepted** : June 27, 2026 | **Published** : July 9, 2026

Abstract: This study presents the implementation and evaluation of the You Only Look Once version 8 nano (YOLOv8n) algorithm for real-time human detection on an autonomous mobile robot. The proposed system is designed as an edge-computing-based surveillance solution for monitoring restricted or difficult-to-access areas. The hardware platform integrates a Raspberry Pi 4B for visual processing and an Arduino Mega 2560 for navigation control through serial communication. Human detection is performed using a night-vision camera, while obstacle avoidance is supported by three ultrasonic sensors. A custom dataset was collected under various human postures, object distances ranging from 1 to 10 meters, and different lighting conditions. The YOLOv8n model was trained using 300 epochs with an image resolution of 640 × 640 pixels. Experimental results demonstrate that the proposed system achieves reliable real-time performance under varying environmental conditions. Under lighting variation tests, the model achieved 100% precision, 93.5% recall, 96.6% F1-score, and 93.55% accuracy with an average processing speed of 24.30 frames per second. Distance-based testing produced 100% precision, 92.42% recall, 96.06% F1-score, and 92.42% accuracy at 23.2 frames per second. Furthermore, autonomous navigation experiments confirmed that the robot was capable of simultaneously detecting humans and avoiding obstacles with response times ranging from 2.4 to 3.2 seconds. These findings indicate that You Only Look Once version 8 nano (YOLOv8n) provides an effective balance between detection accuracy, processing speed, and computational efficiency, making it suitable for deployment on edge-computing-based autonomous mobile robots.

Keywords: Edge Computing, Human Detection, Obstacle Avoidance, Robot Mobile, You Only Look Once version 8 nano (YOLOv8n).

INTRODUCTION

Surveillance of border areas and restricted areas is important in the security system because it is related to the prevention of the entry of foreigners and illegal activities. Currently, artificial intelligence technology is starting to be widely applied to surveillance systems because it is able to improve detection accuracy and speed up responses compared to manual surveillance by humans (Badhan et al., 2024). In its application, deep learning-based object detection is widely used to automatically detect humans in surveillance environments. One of the widely used methods is YOLO because it has a high detection speed so it is suitable for the real-time system implementation (Valarmathi et al., 2023).

Along with the development of intelligent surveillance technology, the use of mobile robots began to be developed to increase the flexibility of the monitoring process in certain areas. Mobile robots allow surveillance systems to monitor on the move so that large and hard-to-reach areas can be monitored more effectively. In addition, the integration between autonomous navigation and human detection allows robots to work independently without relying entirely on human operators.

In recent years, the development of YOLOv8 has shown improved performance in terms of accuracy and speed of detection. Several studies have shown that YOLOv8 is capable of detecting humans in mobile robots with a fairly high level of accuracy in various environmental conditions (Alnabulseih et al., 2025; Lwin et al., 2025). In addition, optimization in the YOLOv8 architecture is also able to increase FPS without significantly lowering

*Corresponding author

This is anCreative Commons License This work is licensed under a Creative Commons Attribution-NonCommercial 4.0 International License.

e-ISSN : 2541-2019
p-ISSN : 2541-044X

accuracy so that it is suitable for use on resource-constrained devices such as mobile robots and *edge computing* (Cao et al., 2025; Jin et al., 2025). The integration of YOLOv8 with ROS-based autonomous robots also shows that the system can perform simultaneous human navigation and detection in dynamic environments (Bhaskar et al., 2025).

In addition to the development of detection models, the use of the edge computing devices onsuch as Raspberry Pi is also a concern in the implementation of the system *real-time*. The device has a compact size and low power consumption, making it suitable for mobile robots. However, the limitation of computing resources on embedded devices is a challenge in running deep learning models optimally. Therefore, a detection model that is lightweight but still capable of maintaining detection performance under various environmental conditions is required. YOLOv8n is one of the variants that is quite suitable because it is designed with a lightweight and efficient architecture for fast processing needs on embedded devices.

Nonetheless, some studies have shown that YOLOv8n still has limitations in certain conditions, such as low lighting, small objects, and conditions *Occlusion* which can affect detection results (Magdy et al., 2025; Silvanus, 2025). In addition, most of the previous research was still conducted under controlled test conditions and has not tested the performance of the system directly on autonomous robots under real conditions (Oise et al., 2025). This condition shows that the evaluation of YOLOv8n performance in the implementation of mobile robots is based on *real-time* It still needs to be studied further, especially on the influence of variations in object distance and lighting conditions.

Based on these problems, this study was conducted to evaluate the performance of YOLOv8n in detecting humans in real-time on an autonomous mobile robot based on edge computing. The test was carried out using variations in object distance and lighting conditions to determine the stability of the system against environmental changes. In addition, the robot is designed to be able to move autonomously using the obstacle avoidance method so that the human navigation and detection processes can run simultaneously on the restricted area surveillance system. This research also contributes to the implementation and evaluation of Raspberry Pi 4B-based YOLOv8n with the integration of human detection systems and obstacle avoidance on edge computing platforms that have limited computing resources.

LITERATURE REVIEW

Research on the implementation of YOLOv8n in autonomous mobile robots continues to evolve, particularly for surveillance purposes, robot mobile and real-time human detection on edge computing devices.

To provide a clear overview of the position of this study relative to previous research, a comparative summary of key literature is presented in Table 1. This table highlights the focus of each study regarding human detection, obstacle avoidance, and the evaluation of distance and lighting conditions.

Table 1. Previous Research

Researcher	Title	Human Detection	Obstacle Avoidance	Distance & Light Evaluation
Gholami et al. (2025)	Design and Implementation of a Near Real-Time Human Detection Robot Using YOLO Framework and IoT Technologies	✓	×	×
Rizk et al. (2025)	Bringing Intelligence to SAR Missions: A Comprehensive Dataset and Evaluation of YOLO for Human Detection in TIR Images	✓	×	×
Sodhro et al. (2025)	*Real-time Efficiency of YOLOv5 and YOLOv8 in Human Intrusion Detection Across Diverse Environments*	✓	×	✓
Majidi et al. (2025)	*Human Detection and Following by a Mobile Robot Using YOLO Structured Convolutional Neural Network*	✓	✓	×
This Study	Evaluation of YOLOv8n Performance for Real-time Human Detection on Autonomous Mobile Robots	✓	✓	✓

Based on previous studies, YOLOv8 is the latest generation of the You Only Look Once (YOLO) object detection family developed to improve both detection accuracy and computational efficiency. Unlike previous YOLO versions that rely on anchor-based detection, YOLOv8 adopts an anchor-free detection strategy, simplifying object localization while reducing computational complexity. Its architecture consists of three primary components: the Backbone, Neck, and Detection Head. The Backbone employs the Cross Stage Partial with Two Convolutions (C2f) module to enhance feature extraction efficiency, whereas the Spatial Pyramid Pooling Fast (SPPF) module enlarges the receptive field to improve multi-scale feature representation. Furthermore, the Neck

*Corresponding author
This is anCreative Commons License This work is licensed under a Creative Commons Attribution-NonCommercial 4.0 International License.

e-ISSN : 2541-2019
p-ISSN : 2541-044X

integrates feature maps from different scales to improve small-object detection, while the anchor-free Detection Head simultaneously performs object classification and localization. These architectural improvements enable YOLOv8 to provide an effective balance between inference speed, detection accuracy, and computational efficiency, making it highly suitable for deployment on embedded and edge-computing platforms (Terven et al., 2023; Varghese, 2024).

Edge computing is a distributed computing paradigm in which data processing is performed locally on embedded devices instead of transmitting all information to cloud servers. This approach significantly reduces communication latency, bandwidth utilization, and network dependency while enabling real-time decision making. In autonomous mobile robots, edge computing allows visual information captured by onboard cameras to be processed directly on embedded hardware such as the Raspberry Pi, thereby improving response speed and operational reliability. The combination of lightweight deep learning models and edge computing has become an effective solution for real-time robotic applications because it maintains high processing efficiency despite limited computational resources (Dinh, 2025; He, 2024).

Obstacle avoidance is one of the fundamental functions of autonomous mobile robots, enabling robots to navigate safely by detecting and avoiding surrounding obstacles while maintaining the desired trajectory. Various sensing technologies have been applied for obstacle detection, including ultrasonic sensors, LiDAR, cameras, radar, and sensor fusion techniques. Among these technologies, ultrasonic sensors remain widely adopted because they provide reliable distance measurements with low computational requirements and low implementation costs. Recent studies have highlighted that integrating obstacle detection sensors with intelligent navigation algorithms significantly improves navigation stability, environmental perception, and robot safety in dynamic environments (He, 2024; Katona et al., 2024).

METHOD

Chain-Wheeled Mobile Robot Platform

This study utilizes a tracked mobile robot for the implementation of real-time human detection. This platform was selected due to its superior movement stability across various surface conditions, making it highly suitable for maneuvering in dynamic environments (Majidi, 2025). 3D modeling was conducted to visualize the overall shape and proportions of the robot, as shown in Figure 1.



Figure 1. Autonomous Robot Design

The mechanical design was engineered to efficiently accommodate the layout of the hardware components. A Raspberry Pi 4B serves as both the YOLOv8n processing center and the primary navigation controller. The visual detection system is supported by a night-vision camera module, while the obstacle avoidance feature relies on three JSN-SR04 ultrasonic sensors (Bhaskar et al., 2025). The actuation of the tracked wheels is driven by DC motors connected through a motor driver. The system's power requirements are divided into two primary sources: two 2200 mAh LiPo batteries dedicated to supplying power to the actuators, and a 3000 mAh Lithium-Ion battery for the controller and processing components. The electrical schematic for this hardware integration is further illustrated in Figure 2.

*Corresponding author
This is anCreative Commons License This work is licensed under a Creative Commons Attribution-NonCommercial 4.0 International License.

Sinkron : Jurnal dan Penelitian Teknik Informatika
Volume 10, Number 3, July 2026
DOI : https://doi.org/10.33395/sinkron.v10i3.16287
e-ISSN : 2541-2019
p-ISSN : 2541-044X

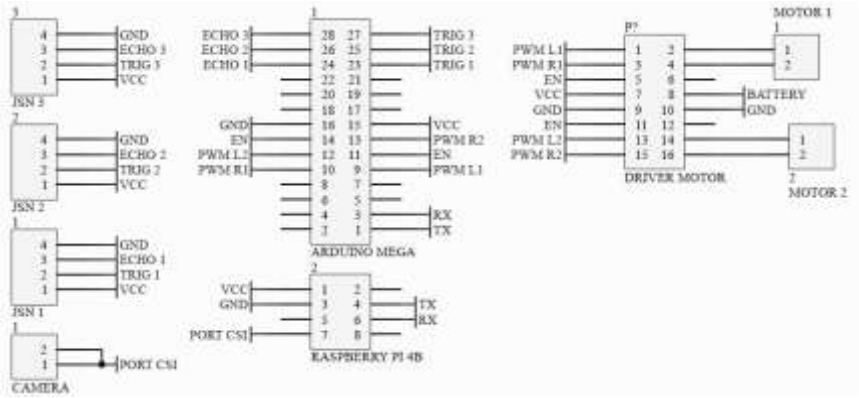


Figure 2. Network Schematics

The robot operates utilizing an obstacle avoidance method, assisted by ultrasonic sensors to read the distance of obstacles around it. When an obstacle is detected, the robot automatically adjusts its direction of movement. Simultaneously, the camera processes images in real-time using YOLOv8n to detect humans within the surveillance area. The integration of navigation and human detection enhances the effectiveness of the autonomous robot system in real-time (Alnabulseih et al., 2025)

Human Detection System Architecture

The system designed in this study is a computer vision-based mobile robot that integrates the YOLOv8n deep learning model for real-time human detection. The system is designed using an edge computing approach, so that the entire process from image acquisition, model processing, to decision-making is carried out directly on the robotic device. This approach allows the system to respond to human presence directly, although its performance is affected by environmental factors such as distance and lighting (Alnabulseih et al., 2025; Magdy et al., 2025). YOLO as a one-stage detector architecture has proven to be superior in real-time applications due to its computing efficiency and processing speed (Valarmathi et al., 2023), YOLO's integration in autonomous robot systems also shows increased simultaneous detection and navigation capabilities (Bhaskar et al., 2025). Based on this concept, the system architecture consists of a main flow that includes camera input, preprocessing, detection using YOLOv8n, decision-making, and autonomous control of the mobile robot actuator, as shown in Figure 3.

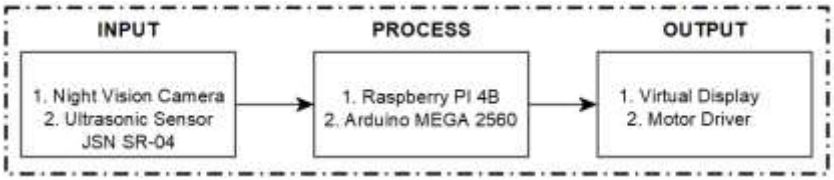


Figure 3. Human Detection System Architecture

Figure 3 shows the architecture of the human detection system on a mobile robot consisting of input, process, and output stages that are integrated with each other in real-time. At the input stage, the system uses a night vision camera to capture environmental imagery as a visual data source in the human detection process, including in low-light conditions, as well as the JSN-SR04 ultrasonic sensor to read the distance to obstacles around the robot to support the obstacle avoidance system. Data from the two sensors is then processed using two controllers, namely the Raspberry Pi 4B and the Arduino Mega 2560. The Raspberry Pi 4B is used as the main processing center to run the YOLOv8n model in real-time human detection, while the Arduino Mega 2560 is focused on controlling the robot's sensors and actuators so that the navigation process can run more responsive and stable. These two controllers are connected via serial communication (UART) utilizing RX and TX pins to enable real-time data exchange. In this scheme, the Raspberry Pi sends an instruction signal based on the results of vision detection to the Arduino, which then translates it into a physical movement command on the driver's motor. The use of two controllers is carried out so that the deep learning computing process and navigation control can run more optimally without overloading the system. The processing results are then forwarded to the output stage in the form of a display of detection results on a virtual display and a control signal to the motor driver to automatically regulate the robot's movement. With this integration, the system is able to perform human detection and robot navigation simultaneously in a surveillance environment. Furthermore, the stages of the human detection process carried out by the system are shown in Figure 4 which includes the process of data acquisition, preprocessing, YOLOv8n inference, decision making system, and robot actuator control.

*Corresponding author

This is anCreative Commons License This work is licensed under a Creative Commons Attribution-NonCommercial 4.0 International License.

Sinkron : Jurnal dan Penelitian Teknik Informatika
Volume 10, Number 3, July 2026
DOI : https://doi.org/10.33395/sinkron.v10i3.16287
e-ISSN : 2541-2019
p-ISSN : 2541-044X

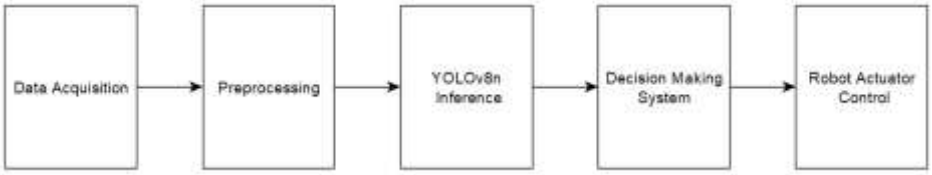


Figure 4. Stages of Human Detection System

Figure 4 shows the stages of the human detection process that the system runs in real-time, from data acquisition to robot actuator control. The process begins with taking images using a night vision camera to capture environmental conditions continuously. The images obtained then go through the preprocessing stage in the form of image size adjustment, normalization, and input format before being processed using the YOLOv8n model. In this study, YOLOv8n was chosen because it has a lightweight and efficient architecture so that it is suitable to run on Raspberry Pi 4B based on edge computing (Dinh, 2024). In addition, YOLOv8 has fast and stable detection capabilities through an anchor-free and decoupled head approach (Terven, 2023). The model was trained using a self-collected dataset with variations in object position and lighting conditions. Prior to the training process, the entire image was converted to a resolution of 640×640 pixels and labeled using the YOLO annotation format. The training process was carried out using 300 epochs, batch size 8, learning rate 0.01, confidence threshold 0.25, and IoU threshold 0.5 to maintain a balance between detection accuracy and computational efficiency. At the inference stage, YOLOv8n performs human detection by generating bounding boxes and confidence scores in real-time, where the one-stage detector architecture allows the detection process to be carried out with high computational efficiency (Valarmathi et al., 2023). The detection results are then processed in the decision making system to determine the robot's actions based on the environmental conditions and human presence detected. The decision is passed on to the actuator control system to automatically regulate the robot's movement. The integration between these stages allows the robot to simultaneously navigate and detect humans so that the system is able to work adaptively in a supervisory environment.

The YOLOv8n architecture in this study consists of three main parts, namely Backbone, Neck, and Detection Head. The Backbone section functions to extract important features from the input imagery using multiple layers of convolution and a C2f module so that the model is able to recognize the patterns of human objects more effectively. In addition, the SPPF (Spatial Pyramid Pooling Fast) module is used to expand the scope of feature extraction without significantly increasing computational complexity.

Furthermore, the Neck section functions to combine features from various scales using the upsample and concat processes. This stage helps the model retain information on small, medium, and large objects so that the detection ability remains stable at various positions and distances of objects. The incorporation of these multi-scale features also enhances the model's ability to detect objects in dynamic and complex environments.

Meanwhile, the Detection Head section is used to produce final predictions in the form of object classification, bounding box coordinates, and confidence scores in real-time. YOLOv8n performs detection on multiple scales of objects so that the system is able to better detect humans under various surveillance conditions. The combination of these structures makes YOLOv8n able to maintain a balance between detection accuracy and inference speed so that it is suitable for application to edge computing-based mobile robot systems.

Dataset and Pre-Processing

The dataset in this study was collected independently using mobile phone cameras in indoor environments with human objects as detection targets. Data collection was carried out by varying the distance of objects and light and dark lighting conditions to match the conditions of use of the system in the field. Distance variation is used to see the model's ability to detect objects at different sizes, while exposure variation is used to test the stability of detection under changing light conditions. The use of datasets with real conditions also helps the model to be more in line with actual operational conditions. An example of the dataset used in this study is shown in Figure 5.

*Corresponding author
This is anCreative Commons License This work is licensed under a Creative Commons Attribution-NonCommercial 4.0 International License.

Sinkron : Jurnal dan Penelitian Teknik Informatika
Volume 10, Number 3, July 2026
DOI : https://doi.org/10.33395/sinkron.v10i3.16287
e-ISSN : 2541-2019
p-ISSN : 2541-044X



Figure 5. Example of Human Detection Dataset

All data then goes through the pre-processing stage before being used in the model training process. This stage includes resizing the image according to YOLOv8n input, pixel normalization, and annotation conversion to YOLO format. The labeling process is carried out manually to produce a bounding box on human objects as ground truth. This pre-processing stage helps to improve the stability of the model in detecting objects under various environmental conditions and image quality. In addition, the use of datasets with variations in distance and lighting allows the evaluation of YOLOv8n performance to be carried out more representative on the implementation of real-time-based mobile robots.

Scenario Experiments

An experimental scenario was carried out to evaluate the performance of YOLOv8n in detecting humans in real-time on a mobile robot based on obstacle avoidance. The test was carried out at a distance of 1–10 meters as well as light and dark lighting conditions using a night vision camera to support detection in low-light conditions. In addition, the robot was tested to move autonomously using three JSN-SR04 sensors to avoid obstacles during the detection process. Performance evaluation was performed using precision, recall, F1-score, and Frames Per Second (FPS) metrics to measure the accuracy and processing speed of the model under various test conditions (Gholami et al., 2025; Majidi et al., 2025). The following is the formula for performance evaluation metrics used in this study.

Precision is used to measure the accuracy of the system in detecting humans based on the comparison between true positive detection and all objects detected as humans. A high precision value indicates that false positives are getting lower.

$$Precision = \frac{TP}{TP+FP} \quad (1)$$

In the equation, TP shows the number of human objects that have been detected correctly, while FP is the number of non-human objects that have been detected as humans.

Recall is used to measure the system's ability to detect all human objects in the test image. A high recall value indicates that the number of undetected objects (false negative) is getting lower so that the detection capabilities of the system are better in the real-time monitoring process (Gholami et al., 2025; Majidi et al., 2025).

$$Recall = \frac{TP}{TP+FN} \quad (2)$$

In the equation, FN shows the number of human objects that the system fails to detect.

F1-score is used to measure the balance between precision and recall so that it can provide a comprehensive picture of the performance of the detection system. This metric is widely used in YOLO-based object detection evaluations to represent the balance between the accuracy and completeness of human object detection (Majidi et al., 2025; Rizk et al., 2025).

$$F1 - Score = 2 \times \frac{Precision \times Recall}{Precision + Recall} \quad (3)$$

In the equation, the F1-score value is obtained from a combination of precision and recall to indicate the overall level of system detection performance.

Frames Per Second (FPS) is used to measure the speed at which a model processes images per second in a real-time detection process. The high FPS value indicates that the model is capable of processing quickly so that it is suitable to be applied to mobile robot-based robots edge computing (Sodhro et al., 2025; Zhang et al., 2025).

*Corresponding author

This is anCreative Commons License This work is licensed under a Creative Commons Attribution-NonCommercial 4.0 International License.

Sinkron : Jurnal dan Penelitian Teknik Informatika
Volume 10, Number 3, July 2026
DOI : https://doi.org/10.33395/sinkron.v10i3.16287
e-ISSN : 2541-2019
p-ISSN : 2541-044X

$$Average\ FPS = \frac{\sum FPS}{n} \quad (4)$$

In the equation, the Average FPS is obtained from the average of the entire FPS value during the testing process to show the overall processing speed of the system.

Accuracy is used to measure the level of accuracy of the entire system in detecting human objects based on the number of correct predictions of all test data. The high accuracy value indicates that the system is capable of detecting most of the test data well (Terven et al., 2023; Safaldin et al., 2024).

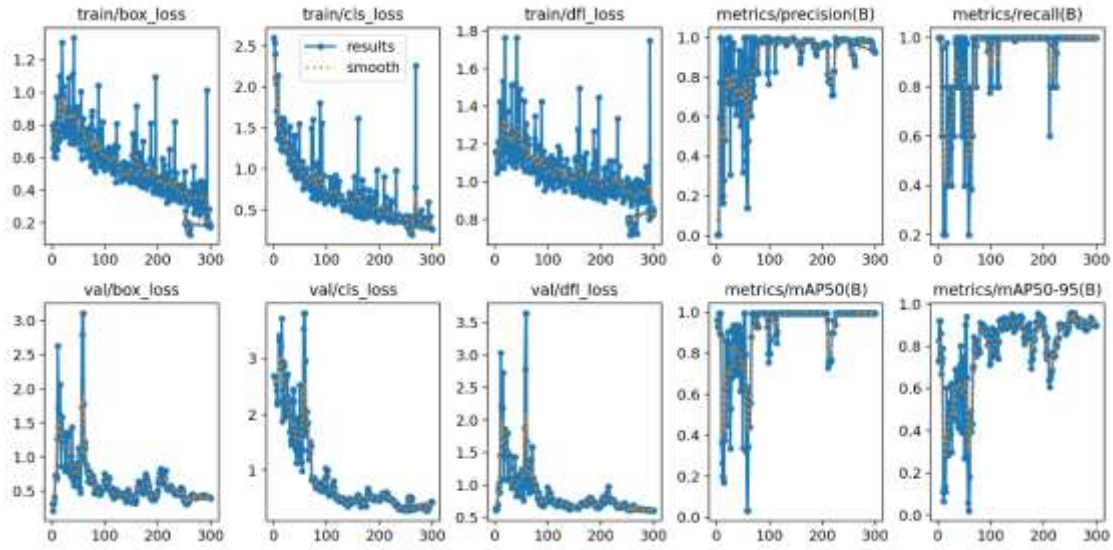
$$Accuracy = \frac{TP+TN}{TP+TN+FP+FN} \quad (5)$$

where TP is true positive, TN is true negative, FP is false positive, and FN is false negative.

RESULTS

YOLOv8n Model Training Results

For implementation on the autonomous mobile robot, the YOLOv8n model was first trained using a custom dataset that had undergone preprocessing and annotation. The training process was conducted for 300 epochs with an input image size of 640 × 640 pixels and a batch size of 8. This configuration was selected to achieve a balance between detection accuracy and computational efficiency, enabling deployment on a Raspberry Pi 4B edge-computing platform. The training performance of the YOLOv8n model is presented in Figure 6.



Gambar 6. Training Model YOLOv8n

Based on the training results in Figure 6, the YOLOv8n model shows a stable training process with a gradual decrease in train loss and validation loss until the end of the epoch. This condition shows that the model is able to learn the characteristics of human objects quite well during the training process. The fluctuations that occurred in the early epoch were still normal and were affected by variations in object position, object scale, and lighting conditions in the training dataset.

In addition, the high range of precision and recall values indicate that the model is capable of detecting human objects with a relatively low error rate. The high values of mAP50 and mAP50-95 indicate that the model is not only able to recognize human objects, but also to determine the location of objects fairly accurately at various IoU threshold conditions. Overall, the results of the training show that YOLOv8n has good enough performance and is stable enough to be applied to edge computing-based mobile robots with the need for real-time human detection.

Testing Based on Light Intensity

This test was carried out to analyze the effect of variations in light intensity on human detection performance. The test scenario was conducted under several different lighting conditions. The test results are presented in Table 2.

*Corresponding author

This is anCreative Commons License This work is licensed under a Creative Commons Attribution-NonCommercial 4.0 International License.

Sinkron : Jurnal dan Penelitian Teknik Informatika
Volume 10, Number 3, July 2026
DOI : https://doi.org/10.33395/sinkron.v10i3.16287
e-ISSN : 2541-2019
p-ISSN : 2541-044X

Table 2. Light-Based Test Results

No.	Light Conditions	Number of Frames	True Positive (TP)	False Negative (FN)	False Positive (FP)
1	Bright	31	29	2	0
2	Dark	31	29	2	0

Based on Table 2, the total data tested in this scenario is 62 test frames. From this data, the number of True Positive (TP), False Positive (FP), and False Negative (FN) was obtained, which was then used to calculate precision, recall, F1-score, accuracy, and frames per second (FPS) using Equations (1)–(5). The results of these calculations are used to evaluate the level of detection accuracy and system performance in carrying out the human detection process in real-time.

Precision:

$$Precision = \frac{58}{58+0} = 1 = 100\% \quad (1)$$

Recall:

$$Recall = \frac{58}{58+4} = 0.935 = 93.5\% \quad (2)$$

F1-Score:

$$F1 = 2 \times \frac{1 \times 0.935}{1+0.935} = 0.966 = 96.6\% \quad (3)$$

Average Frames Per Second (FPS):

$$Average\ FPS = \frac{972.2}{40} = 24.30\ FPS \quad (4)$$

Accuracy :

$$Accuracy = \frac{58}{62} \times 100\% = 93.55\% \quad (5)$$

Based on the calculation results, the YOLOv8n model obtained a precision value of 100%, recall 93.5%, F1-score of 96.6%, and accuracy of 93.55% with an average processing speed of 24.30 FPS. These results show that the system is able to perform real-time human detection with stable performance in light and dark conditions. Although some false negatives are still found in low-light conditions, the use of night vision cameras helps maintain the quality of system detection.

Testing Based on Distance Variation

This test was carried out to analyze the effect of variations in the distance of objects to the camera on human detection performance. The test was carried out at several different distances to determine the operational limits of the system in detecting human objects optimally.

During the experiment, the human subject was positioned at predefined distance intervals from the mobile robot's camera. For each specific distance, 10 frames were sequentially captured and analyzed by the YOLOv8n model running on the edge computing device. The system's performance is evaluated based on a confusion matrix approach, recording True Positive (TP) for correctly identified human subjects, False Negative (FN) when the system fails to detect the present subject, and False Positive (FP) when background elements are incorrectly classified as human. By observing these metrics across varying distances, this study aims to identify the maximum effective range and the exact point where spatial resolution degradation begins to affect the detection accuracy. The test results are presented in Table 3.

Table 3. Test Results Based on Distance

No.	Distance	Number of Frames	True Positive (TP)	False Negative (FN)	False Positive (FP)
1	1 meter	10	10	0	0
2	2 meters	10	9	1	0
3	3 meters	10	10	0	0
4	4 meters	6	5	1	0

*Corresponding author

This is anCreative Commons License This work is licensed under a Creative Commons Attribution-NonCommercial 4.0 International License.

5	5 meters	5	4	1	0
6	6 meters	5	4	1	0
7	7 meters	5	5	0	0
8	8 meters	5	5	0	0
9	9 meters	5	4	1	0
10	10 meters	5	5	0	0

Based on Table 3, the total data tested in this scenario was 66 test frames. From this data, the number of True Positive (TP), False Positive (FP), and False Negative (FN) was obtained, which was then used to calculate precision, recall, F1-score, accuracy, and frames per second (FPS) using Equations (1)–(5). The results of these calculations are used to evaluate the level of detection accuracy and system performance in carrying out the human detection process in real-time.

Precision:

$$Precision = \frac{61}{61+0} = 1 = 100\% \quad (1)$$

Recall:

$$Recall = \frac{61}{61+5} = 0.9242 = 92{,}42\% \quad (2)$$

F1-Score :

$$F1 = 2 \times \frac{1 \times 0.9242}{1+0.9242} = 0.9606 = 96{,}06\% \quad (3)$$

Average Frames Per Second (FPS):

$$Average\ FPS = \frac{1531.1}{66} = 23.2\ FPS \quad (4)$$

Accuracy :

$$Accuracy = \frac{61}{66} \times 100\% = 92.42\% \quad (5)$$

Based on the calculation results, the YOLOv8n model obtained a precision score of 100%, recall of 92.42%, F1-score of 96.06%, and accuracy of 92.42% with an average processing speed of 23.2 FPS. The results show that the system is capable of real-time human detection with a low error rate. In addition, the robot is also able to carry out simultaneous human navigation and detection using an obstacle avoidance system, although in some conditions there is still an increase in response time and the limitation of the reading angle of the ultrasonic sensor.

Autonomous Robot System Testing

The system test was conducted to evaluate the mobile robot's ability to perform real-time human detection and autonomous navigation in a surveillance environment. The robot maneuvers using the obstacle avoidance method with the help of three JSN-SR04 ultrasonic sensors placed on the front, left, and right sides to map the distance of the surrounding obstacles. Simultaneously, the camera module captures visual frames that are then processed directly by the YOLOv8n model on edge computing devices to detect human presence during the navigation process. To simulate a realistic surveillance scenario, this test not only detects humans in ideal positions, but also varies the visibility level and posture of the object. A summary of the results of the integration test between the navigation system and computer vision is shown in Table 4.

Table 4. System Test Results

No.	Sensor Objects	Front Sensor (cm)	Left Sensor (Cm)	Right Sensor (cm)	Detected Status	Response Time (s)
1	Full Body	22	44	36	Detected	2.8
2	Half Body	91	115	66	Detected	2.4
3	Slightly Visible	0	114	27	Detected	2.5
4	Prone	38	49	42	Detected	3.0
5	Sit	54	0	59	Detected	3.1
6	Side Supine	30	40	66	Detected	3.2

*Corresponding author

This is anCreative Commons License This work is licensed under a Creative Commons Attribution-NonCommercial 4.0 International License.

e-ISSN : 2541-2019
p-ISSN : 2541-044X

7	Lying On Your Back	0	86	53	Detected	2.9
8	Sitting Half Body	22	59	70	Detected	2.7
9	Slightly Detectable	32	28	66	Detected	2.5
10	Full Body	103	43	20	Detected	2.9

Based on the test results in Table 4, the robot is able to detect humans in various positions of objects during the navigation process. System obstacle avoidance also runs well through the continuous reading of the JSN-SR04 sensor so that the robot can still move while conducting detection real-time. The system's response time is in the range of 2.4–3.2 seconds, although in some conditions such as partially visible objects or abnormal body positions there is an increase in response time. In addition, some zero-value sensor readings indicate that there is still a limitation of the reading angle of the ultrasonic sensor under certain conditions.

Overall, the experimental results demonstrate that the proposed YOLOv8n-based autonomous mobile robot is capable of performing real-time human detection and obstacle avoidance under the evaluated test scenarios. However, the evaluation was limited to a custom dataset with relatively few samples and environmental variations. Future work should include larger datasets and more diverse testing environments to further assess the robustness and generalization capability of the proposed approach.

DISCUSSIONS

Based on the results of training and testing, the YOLOv8n model shows fairly stable performance for real-time human detection on edge computing-based mobile robots. Consistent reductions in train loss and validation loss during training indicate that the model was able to learn human object characteristics well, especially under variations in body position, distance, and lighting conditions. High precision, recall, and mAP values also indicate that the model can detect human objects with a low error rate while accurately localizing objects in the image.

In the light intensity test, the system achieved a precision of 100%, recall of 93.5%, F1-score of 96.6%, and accuracy of 93.55% with an average processing speed of 24.30 FPS. These results indicate that YOLOv8n can maintain stable detection performance in both bright and dark conditions. The absence of false positives shows that the system can distinguish human and non-human objects effectively. However, several false negatives still appeared under low-light conditions due to reduced object details, although the night vision camera helped maintain image quality for detection.

Distance variation also affected system performance. In the 1–10 meter test, the system obtained a precision of 100%, recall of 92.42%, F1-score of 96.06%, and accuracy of 92.42% with an average speed of 23.2 FPS. The system was still able to detect humans consistently at various distances without producing false positives. However, at farther distances, several false negatives appeared because the object size became smaller, making human features harder for the model to recognize.

Further testing focused on the autonomous robot system to evaluate the integration between navigation and human detection simultaneously. The robot successfully detected humans in various positions while navigating using the JSN-SR04 ultrasonic sensor. Although YOLOv8n was capable of real-time detection at around 23–24 FPS, the overall system response time ranged from 2.4 to 3.2 seconds due to motor actuation and communication latency between devices. Response time also tended to increase when the object was partially visible or when body posture was unclear.

In addition, several ultrasonic sensor readings still produced zero values, indicating limited sensor reading angles and blind spots in certain areas. Future improvements could apply sensor fusion techniques, such as adding more ultrasonic sensors or integrating LiDAR to improve environmental perception. The real-time performance achieved in this study is consistent with the findings of Zhexebay et al. (2025), which showed that YOLOv8n can maintain stable real-time performance on embedded devices. Compared with the static surveillance system developed by Wang et al. (2025), the proposed mobile robot system provides better flexibility and environmental adaptability through obstacle avoidance integration.

However, this study still has several limitations. The proposed system employs the YOLOv8n model, which was selected for its lightweight architecture and efficient real-time performance on Raspberry Pi-based edge devices. Compared with larger variants such as YOLOv8s and YOLOv11, YOLOv8n requires lower computational resources but generally provides lower detection accuracy in more complex environments (Terven et al., 2023; Zhexebay et al., 2025). In addition, compared with YOLOv5, YOLOv8n adopts an anchor-free architecture that improves computational efficiency while maintaining competitive detection performance (Terven et al., 2023; Sodhro et al., 2025). Future work should compare different YOLO variants under identical testing conditions to further evaluate their trade-offs between accuracy and computational efficiency.

*Corresponding author
This is anCreative Commons License This work is licensed under a Creative Commons Attribution-NonCommercial 4.0 International License.

e-ISSN : 2541-2019
p-ISSN : 2541-044X

CONCLUSION

This research successfully implemented YOLOv8n on a mobile robot based on edge computing for real-time human detection in prohibited area surveillance. Based on the experimental results, the proposed system achieved satisfactory performance, obtaining a precision of 100% in the lighting test and an accuracy of 93.55%, while achieving an accuracy of 92.42% in the distance test with an average processing speed exceeding 23 FPS, thereby satisfying real-time operational requirements. Furthermore, the integration of human detection and obstacle avoidance enabled the robot to perform simultaneous navigation and detection with a response time ranging from 2.4 to 3.2 seconds. These results demonstrate that YOLOv8n provides a suitable balance between detection accuracy, inference speed, and computational efficiency for Raspberry Pi 4B-based autonomous mobile robots. However, this study was evaluated using a custom testing dataset with limited environmental variations. Therefore, future work should include larger testing datasets, more diverse indoor and outdoor environments, and comparative evaluations with other YOLO variants, such as YOLOv5, YOLOv8s, and YOLOv11, to further improve the robustness and generalization capability of the proposed system.

REFERENCES

Alnabulseih, K., Abd-EL-Rehim, A.-E., Badawi, M., & Elgohary, R. (2025). Enhancing disaster response efforts with YOLOv8-based human detection in mobile robotics. *International Integrated Intelligent Systems.* https://doi.org/10.21608/iiis.2025.292458.1036

Badhan, I. A., Hasnain, M. N., Rahman, M. H., Chowdhury, I., & Sayem, M. A. (2024). Strategic deployment of advance surveillance ecosystems: An analytical study on mitigating unauthorized U.S. border entry. *Inverge Journal of Social Sciences, 3*(4), 82–92.

Bhaskar, Y., Dolaskar, K., Jadhav, H., Shirke, M., & Kulkarni, K. (2025). Autonomous navigation and object detection using ROS 2 and YOLOv8. *International Journal of Scientific Research in Engineering and Management, 9*(3). https://doi.org/10.55041/IJSREM42857

Cao, Y., Luo, H., Wang, M., Wang, Y., & Yan, H. (2026). Enhanced YOLOv8 for accurate and efficient floating object detection on water surfaces. *Scientific Reports, 16*, 2907. https://doi.org/10.1038/s41598-025-32817-x

Dinh, H. T., & Kim, E.-T. (2025). A lightweight network based on YOLOv8 for improving detection performance and the speed of thermal image processing. *Electronics*, *14*(4), 783. https://doi.org/10.3390/electronics14040783

Gholami, R., Jahromi, H. D., & Sedaghat, S. (2025). Design and implementation of a near real-time human detection robot using YOLO framework and IoT technologies. *IEEE Access, 13*, 175960–175983. https://doi.org/10.1109/ACCESS.2025.3618302

He, A. J., & Soe, M. T. (2024). A review on sensor technologies and control methods for mobile robot with obstacle detection system. *International Journal on Robotics, Automation and Sciences, 6*(1), 78–85. https://doi.org/10.33093/ijoras.2024.6.1.11

Huang, M., Mi, W., & Wang, Y. (2024). EDGS-YOLOv8: An improved YOLOv8 lightweight UAV detection model. *Drones*, *8*(7), 337. https://doi.org/10.3390/drones8070337

Jin, M., Zhang, J., Peng, C., Chen, H., Chen, X., Lu, B., & Wang, X. (2025). Lightweight YOLOv8 for rapid object detection. *Journal of Computers, 36*(2), 315–326. https://doi.org/10.63367/19911599202504360202 1

Katona, K., Neamah, H. A., & Korondi, P. (2024). Obstacle avoidance and path planning methods for autonomous navigation of mobile robot. *Sensors, 24*(11), 3573. https://doi.org/10.3390/s24113573

Lwin, N. N. H., Aye, M., & Hla, T. T. (2025). Development and performance analysis of a human detection robot using YOLOv8 and PWM-based speed control. *The Indonesian Journal of Computer Science, 14*(3), 4153–4168. https://doi.org/10.33022/ijcs.v14i3.4896

Magdy, A., Moustafa, M. S., & Ebied, H. M. (2025). Evaluating YOLOv8 variants for object detection in satellite image. *International Journal of Intelligent Computing and Information Sciences, 25*(2), 18–31. https://doi.org/10.21608/ijicis.2025.374419.1387

Majidi, Y., & Hassanabadi, A. H. (2025). Human detection and following by a mobile robot using YOLO structured convolutional neural network. In *2025 33rd International Conference on Electrical Engineering (ICEE)* (pp. 1210–1216). IEEE. https://doi.org/10.1109/ICEE67339.2025.11213908.

Oise, G. P., Unuigbokhai, N. B., Onwuzo, C. J., Nwabuokei, O. C., Ejenarhome, P. O., Atake, O. M., & Bakare, S. K. (2025). YOLOv8-DeepSORT: A High-Performance Framework for Real-Time Multi-Object Tracking with Attention and Adaptive Optimization. *Journal of Science Research and Reviews*, *2*(2), 92–100. https://doi.org/10.70882/josrar.2025.v2i2.50

*Corresponding author

This is anCreative Commons License This work is licensed under a Creative Commons Attribution-NonCommercial 4.0 International License.

Rizk, M., & Bayad, I. (2025). Bringing intelligence to SAR missions: A comprehensive dataset and evaluation of YOLO for human detection in TIR images. *IEEE Access, 13*, 17208–17235. https://doi.org/10.1109/ACCESS.2025.3529484

Safaldin, M., Zaghden, N., & Mejdoub, M. (2024). An improved YOLOv8 to detect moving objects. *IEEE Access, 12*, 59782–59806. https://doi.org/10.1109/ACCESS.2024.3393835

Silvanus, T., Shankar, P. B., & Dhiya, A. (2025). Comparative analysis of YOLOv8, YOLOv9, and YOLOv10 for object detection: Performance metrics and real-world applications. *International Journal for Research in Applied Science & Engineering Technology, 13*(5), 5045–5052. https://doi.org/10.22214/ijraset.2025.71284

Sodhro, A. H., Kannam, S., & Jensen, M. (2025). Real-time efficiency of YOLOv5 and YOLOv8 in human intrusion detection across diverse environments and recommendation. *Internet of Things, 33*, Article 101707. https://doi.org/10.1016/j.iot.2025.101707

Terven, J., Córdova-Esparza, D.-M., & Romero-González, J.-A. (2023). Comprehensive review of YOLO architectures in computer vision: From YOLOv1 to YOLOv8 and YOLO-NAS. *Machine Learning and Knowledge Extraction, 5*(4), 1680–1716. https://doi.org/10.3390/make5040083

Valarmathi, B., Kshitij, J., Rajpurohit, D., Srinivasa Gupta, N., Harold Robinson, Y., Arulkumaran, G., & Mulu, T. (2023). Human detection and action recognition for search and rescue in disasters using YOLOv3 algorithm. *Journal of Electrical and Computer Engineering, 2023*, Article 5419384. https://doi.org/10.1155/2023/5419384

Varghese, R., & M., S. (2024). YOLOv8: A novel object detection algorithm with enhanced performance and robustness. In *2024 International Conference on Advances in Data Engineering and Intelligent Computing Systems (ADICS)* (pp. 1–6). IEEE. https://doi.org/10.1109/ADICS58448.2024.10533619

Wang, Q., Feng, G., & Li, Z. (2025). A lightweight person detector for surveillance footage based on YOLOv8n. *Sensors*, *25*(2), 436. https://doi.org/10.3390/s25020436

Muhamad Zamri, F. N., Gunawan, T. S., Yusoff, S. H., Alzahrani, A. A., Bramantoro, A., & Kartiwi, M. (2024). Enhanced small drone detection using optimized YOLOv8 with attention mechanisms. *IEEE Access, 12*, 90629–90643. https://doi.org/10.1109/ACCESS.2024.3420730

Zhang, J., Beliaeva, Z. V., & Huang, Y. (2025). Accuracy–Efficiency Trade-Off: Optimizing YOLOv8 for Structural Crack Detection. *Sensors*, *25*(13), 3873. https://doi.org/10.3390/s25133873

Zhexebay, D. M., Skabylov, A. A., Turlykozhayeva, D. A., Bolysbay, A., Turmaganbet, U. K., Namazbayev, T., & Tao, M. (2025). Real-time small object detection with YOLOv8n/8s and YOLOv11n/11s models in complex natural landscapes. *Physical Sciences and Technology*, 12(1–2), 142–153. https://doi.org/10.26577/phst202512114

*Corresponding author

This is anCreative Commons License This work is licensed under a Creative Commons Attribution-NonCommercial 4.0 International License.