The Indonesian Journal of Computer Science
www.ijcs.net
Volume 14, Issue 3, June 2025
https://doi.org/10.33022/ijcs.v14i3.4896

Development and Performance Analysis of a Human Detection Robot Using YOLOv8 and PWM-Based Speed Control

Ni Ni Htay Lwin[1], Maung Aye[2], Tin Tin Hla[3]
ninihtaylwin@gmail.com[1], kokoaye528@gmail.com[2], tintinhla99@gmail.com[3]
[1,2,3] Department of Electronic Engineering, Mandalay Technological University, Myanmar

Article Information

Received : 7 Jun 2025
Revised : 12 Jun 2025
Accepted : 16 Jun 2025

Keywords

Human Detection, YOLOv8,COCO dataset, Raspberry Pi,PWM

Abstract

This paper presents the design and performance evaluation of a human detection robot using the YOLOv8 model and the COCO dataset for object recognition. The robot is equipped with a Pi camera, Raspberry Pi, four GM25 13CPR motors, an L298 motor driver, and a buck converter, ensuring efficient operation in real-time environments. The human detection accuracy was evaluated at different distances, achieving 99% at 2 feet, 98% at 15 feet, and 96% at 25 feet, demonstrating the effectiveness of the YOLOv8 model in varying conditions.The robot's movement is controlled using a PWM-based speed control technique, where the DC motors operate at different duty cycles. Experimental results show variations in speed accuracy, with error percentages of 7.6% at 20% duty cycle, 5.8% at 40%, 5.1% at 60%, 4.8% at 80%, and 3.8% at 100% duty cycle. These results indicate that higher duty cycles lead to improved speed accuracy, minimizing the deviation from the desired speed. The study highlights the integration of YOLOv8 for object detection and PWM for precise motor control, making the system suitable for applications in autonomous navigation, surveillance, and security.

A. Introduction

In recent years, the development of intelligent robotic systems has gained significant attention for applications in surveillance, security, and autonomous navigation. One of the critical aspects of such robotic systems is their ability to detect and track humans efficiently in dynamic environments. Object detection models play a crucial role in enabling robots to recognize and interact with their surroundings. In this study, a human detection robot is designed and tested using the YOLOv8 model, a state-of-the-art deep learning algorithm known for its high-speed and high-accuracy performance. The COCO dataset, a widely used benchmark for object detection, is employed for training and testing purposes to ensure reliable detection outcomes.

M. Vajgl and colleagues [1] suggested a way to incorporate distance estimation into the YOLOv3 model. A distance prediction graph and a specific distance estimate loss function were added by the authors to YOLOv3. A comparable approach was also used by the authors in [2], however they concentrated on depth estimation. During their experiments, they added a depth estimate output channel branch to the YOLOv4 network. By mainly utilizing stereoscopic principles, the authors' approach in [3] decreased the requirement for datasets that were specially improved with distance information.

Based on YOLOv8, Z. J. Khow et al. developed YOLOv8 CAW, an enhanced detection model that can precisely determine the distances of objects of interest in addition to detecting them. The model-maintained inference speeds comparable to the baseline model in the PASCAL VOC dataset, while achieving increases in recall (0.4%), precision (2.2%), and mean average precision (mAP) (1.5%) within the 0.5 to 0.95 threshold range. This marked a significant improvement in performance metrics following the experiment. The results are very promising and encouraging, as the distance estimation obtained an approximate average accuracy of 90% [4].

This paper aims to explore the integration of deep learning-based human detection and PWM-based motor control to enhance the performance of autonomous robots. The proposed system demonstrates the potential for applications in security patrols, autonomous assistance, and human tracking systems. The remainder of the paper discusses the methodology, implementation, experimental results.

B. Research Methodology

CNN Architecture

CNN is also known as ConvNet that is the type of artificial neural networks designed specifically for structured grid data, such as images and videos. Convolutional neural network (CNN) is a class of deep, feed-forward artificial neural network that has been utilized to produce an accurate performance in computer vision tasks, such as image classification and detection [5]. CNNs are like traditional neural network, but with deeper layers. It has weights, biases and outputs through a nonlinear activation. The neurons of the CNN are arranged in a volumetric fashion such as, height, width and depth. Figure.1 shows the CNN architecture, it is composed of convolutional layer, pooling layer and fully connected layer. Convolutional layer and pooling layer are typically alternated and the depth of each filter increases from left to right while the output size (height

and width) are decreasing. The fully connected layer is the last stage which is similar to the last layer of the conventional neural networks.

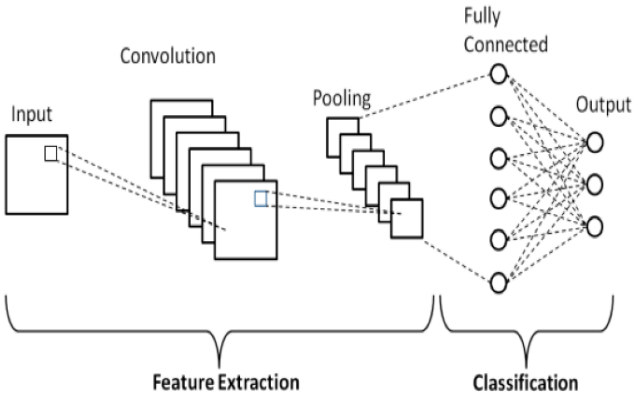

Figure 1. CNN Architecture

Convolutional layers, which apply a specified number of convolution filters to the image. For each sub region, the layer performs a set of mathematical operations to produce a single value in the output feature map. Convolutional layers then typically apply a Rectified Linear Unit (ReLU) activation function to the output to introduce nonlinearities into the model. The operation of convolutional layer is illustrated in Figure 2.

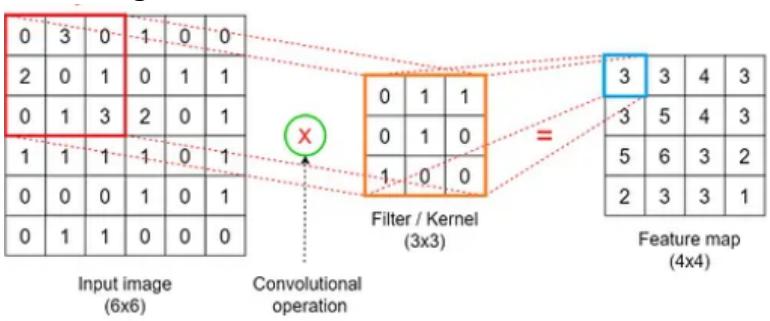

Figure 2. Convolutional layers

The input is an image that will hold pixel values. It has three dimensions such as width, height and depth (RGB channels) example is [50x50 x 3]. The convolutional layer will compute the output of neurons that are connected to local regions in the input. The layer's parameters are composed of a set of learnable filters (or kernels), which convolved across the width and height of the input volume extending through its depth, computing the dot product between the entries of the input and the filter. This produces a 2-dimensional activation map of that filter and as a result, the network learns filters that trigger when it detects some particular type of feature at some spatial position in the input. A convolutional layer applies a convolution operation to the input data, producing feature maps.

$$(I * K)(x, y) = \sum_{m=0}^{M-1} \sum_{n=0}^{N-1} I(x + m, y + n) \cdot K(m, n) \quad (1)$$

where: I is the input image or feature map.K is the convolutional kernel (filter). x, y are the coordinates of the output feature map.M,N are the dimensions of the kernel.

YOLOv8 Model

YOLOv8 (You Only Look Once, Version 8) is the latest version of the YOLO family of object detection models, developed by Ultralytics. It represents a significant evolution in the YOLO series, incorporating modern techniques and architecture improvements to achieve better performance, flexibility, and ease of use. YOLOv8 is designed for a variety of tasks, including object detection, instance segmentation, and image classification.

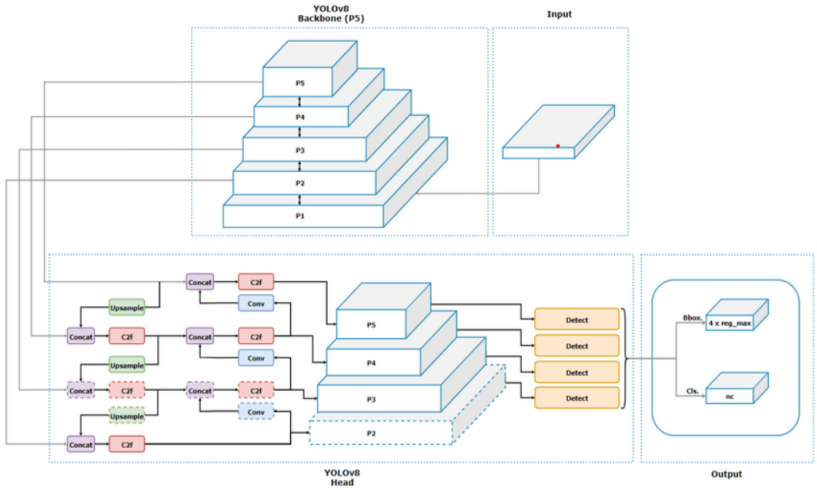

Figure 3. Convolutional layers

YOLOv8 introduces a new backbone and neck design, making it more efficient and accurate than previous versions. YOLOv8 uses a dynamic computation graph that enables task-specific optimization for better performance in detection, segmentation, and classification tasks. The architecture of the YOLOv8 is shown in Figure 3. The backbone is responsible for feature extraction from the input image. YOLOv8 uses a CSP (Cross-Stage Partial) architecture, but it includes enhancements like better convolutional blocks, efficient activation functions, and improved stride control. The neck aggregates and combines features from different levels of the backbone to ensure that both high-level (semantic) and low-level (spatial) features are passed to the head. The detection head predicts bounding boxes, class probabilities, and other outputs like masks (for instance segmentation tasks). YOLOv8 adopts a decoupled head design for separate classification and regression tasks, improving precision.

COCO Dataset

The COCO dataset (short for Common Objects in Context) is a large-scale, richly annotated dataset designed for training and evaluating computer vision models. It is widely used in tasks such as object detection, segmentation, key point detection, and image captioning. COCO provides extensive annotations, including

bounding boxes, object categories, segmentation masks. The dataset contains 80 object categories such as "person," "dog," "car," and "chair." Images are collected to reflect real-world scenarios with objects appearing in natural and cluttered contexts, challenging models to learn context-aware representations. The dataset consists of over 330,000 images;200,000 labelled images with detailed annotations,630,000 object instances annotated. The COCO dataset uses standard metrics for evaluating model performance.

PWM Technique For DC Motor Control

Pulse Width Modulation (PWM) is a technique used to control the speed of a DC motor by varying the average voltage supplied to the motor. Instead of changing the voltage directly, PWM rapidly switches the motor ON and OFF at a high frequency, adjusting the duty cycle to regulate speed.A PWM signal is a periodic waveform with two key parameters:Frequency (f): The number of times the signal completes an ON-OFF cycle per second. Duty Cycle (%): The percentage of time the signal is ON in one cycle.

$$\text{Duty Cycle} = (T_{on} / (T_{on} + T_{off})) \times 100\% \quad (2)$$

The speed of a DC motor (RPM) is proportional to the average voltage applied, which depends on the PWM duty cycle. The average voltage and motor speed is calculated by Equation (3) and (4).

$$V\_avg = V\_supplied\ ((\text{Duty Cycle})/100) \quad (3)$$

$$\text{Motor Speed(RPM)} = \text{Max RPM} \times ((\text{Duty Cycle})/100) \quad (4)$$

Where,V_avg is the average voltage applied to the motor. V_supplied is the full supply voltage. Max RPM is the speed of the motor at full voltage. Motor Speed (RPM) is the actual speed at a given duty cycle.To measure the actual motor speed RPM, an encoder attached to the motor shaft. The actual motor speed is calculated by Equation (5).

$$\text{Actual motor speed(RPM)} = ((\text{Pulse count} \times 60)/(\text{Pulses per Revolution})) \quad (5)$$

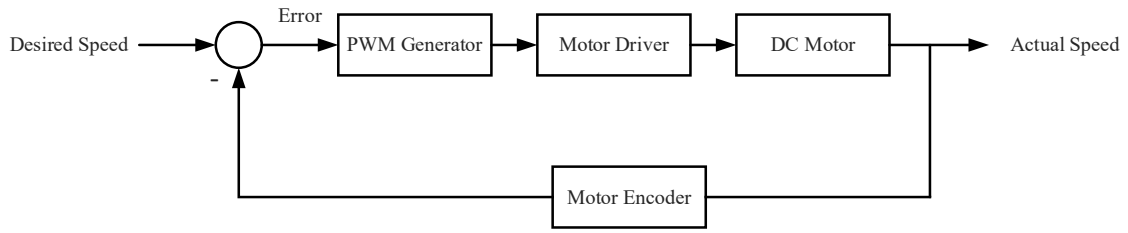

Figure 4. block diagram of the DC motor speed control

The block diagram of the DC motor speed control is shown in Figure 4. The system starts with a desired speed input. The actual speed of the motor (measured by the Motor Encoder) is subtracted from the desired speed. This generates an

error signal. The error is then sent to the PWM Generator. The PWM (Pulse Width Modulation) Generator adjusts the duty cycle of the signal. A higher duty cycle means more power is sent to the motor, increasing its speed. A lower duty cycle reduces power and slows the motor down. The motor driver (such as an L298 or similar H-bridge driver) receives the PWM signal. It amplifies the signal to provide enough current to the DC motor.This ensures that the motor operates at the correct speed. The DC motor rotates based on the power supplied by the motor driver. The actual speed of the motor depends on the applied PWM signal.The motor encoder continuously measures the actual speed of the DC motor.It sends this speed information back to the system for comparison with the desired speed. Since the system continuously adjusts the PWM signal based on feedback from the motor encoder, it forms a closed-loop control system. If there is a speed difference (error), the system automatically increases or decreases the PWM signal to correct the motor speed.

C. Hardware Requirement

The robot is equipped with a Pi camera, Raspberry Pi, four GM25 13CPR motors, an L298 motor driver, and a buck converter and three 3.7V battery**.**

Raspberry Pi 4 Model B

The Raspberry Pi 4 Model B is a low-cost, powerful single-board computer offering significant improvements over its predecessor, the Pi 3 Model B+, with features like a faster processor, more RAM options, and enhanced connectivity, including dual-band Wi-Fi, Bluetooth 5.0, and Gigabit Ethernet.

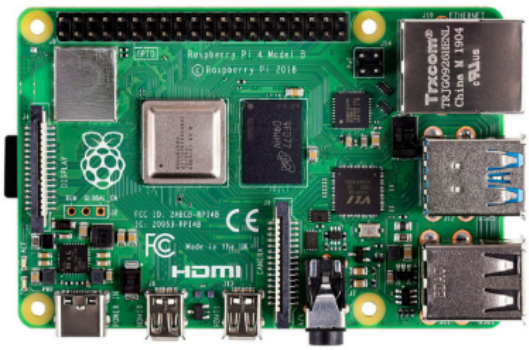

Figure 5. Raspberry Pi 4 Model B

L298N Motor Driver

L298N Motor Driver Module is a high power motor driver module for driving DC and Stepper Motors. This module consists of an L298 motor driver IC and a 78M05 5V regulator. L298N Module can control up to 4 DC motors, or 2 DC motors with directional and speed control.

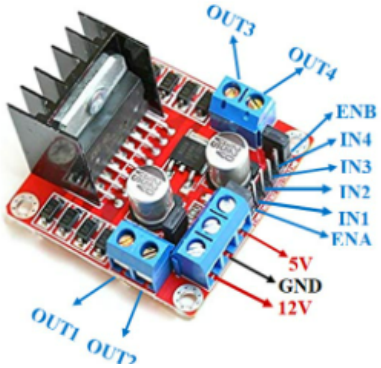

Figure 6. L298N Motor Driver

GM25-13 CPR Motor

The GM25-13cpr motor, often referred to as the JGA25-370B, is a mini DC geared motor with an encoder, designed for applications requiring force and speed control, typically operating at 12V DC and 130RPM.

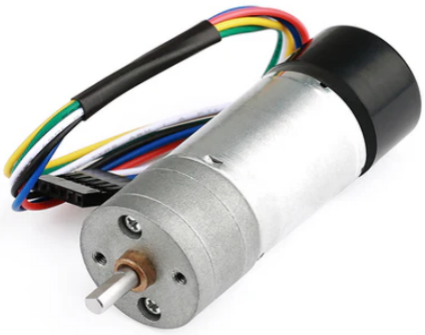

Figure 7. GM25-13cpr Motor

D. Circuit Configuration

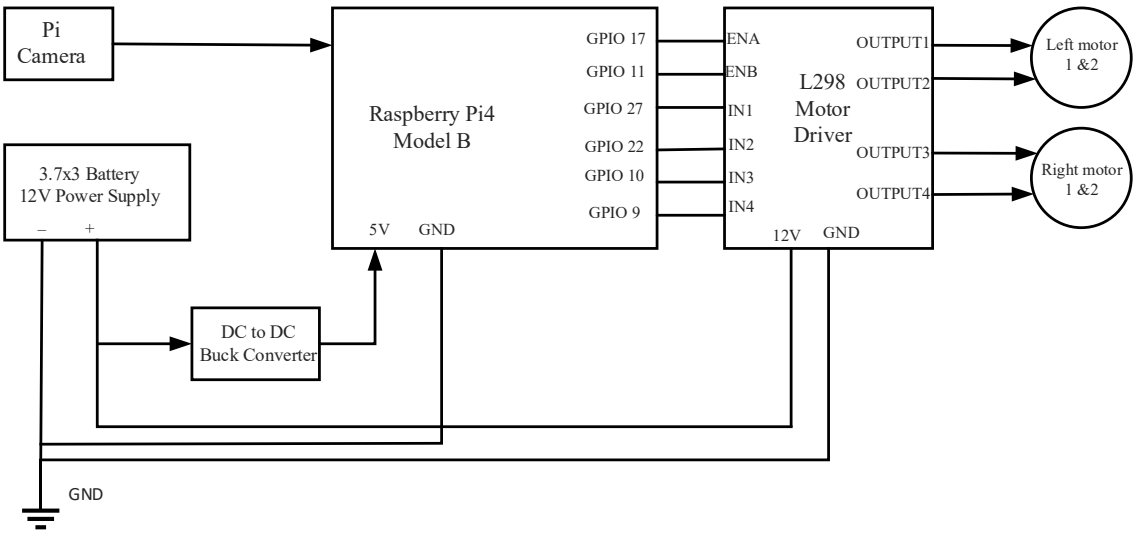

Figure 8. Circuit Connection Diagram of the Human Detection Robot

The circuit connection diagram of the human detection robot is shown in Figure 8. The system is powered by a 12V battery (3.7V x 3 cells = 12V). A DC-DC buck converter steps down the 12V to 5V to power the Raspberry Pi 4 Model B. The L298 motor driver receives 12V directly from the battery to power the motors.The Raspberry Pi serves as the controller for the system. It receives input from the Pi Camera for object/human detection. The motor movement is controlled via GPIO pins, which send signals to the L298 motor driver. The L298 motor driver is responsible for controlling the left and right motors.The ENA (Enable A) pin is used to control motor speed using PWM (Pulse Width Modulation).The IN1, IN2, IN3, and IN4 pins determine the direction of the motors. IN1 and IN2 control the left motors.IN3 and IN4 control the right motors. The motor driver's OUTPUT1 and OUTPUT2 are connected to the left motor. OUTPUT3 and OUTPUT4 are connected to the right motor. The Pi Camera is connected to the Raspberry Pi. It captures images or video for human detection or object tracking. The Raspberry Pi processes the image data and controls motor movement accordingly.

E. Evaluation Metric for Human Detection

The confusion matrix evaluates the performance of a machine learning model by comparing predicted and actual classifications. It is a matrix that categorizes predictions into four groups based on the relationship between actual and predicted values: True Positives (TP), True Negatives (TN), False Positives (FP), and False Negatives (FN).

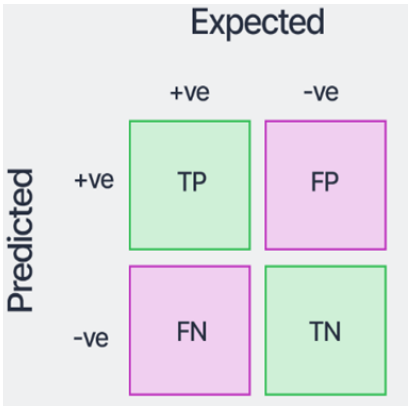

Figure 9. Confusion Matrix

The definitions of the confusion matrix is shown in Figure 9. True Positive (TP) refers to the number of positive samples that are correctly classified, while True Negative (TN) represents the number of negative samples correctly identified. False Positive (FP) indicates the number of negative samples mistakenly classified as positive, and False Negative (FN) denotes the positive samples that are incorrectly classified as negative. If accuracy is defined as the ratio of correctly predicted objects to the total predictions, the formula can be expressed as:

$$\text{Accuracy} = (TP+TN)/(TP+TN+FP+FN) \quad (6)$$

F. Result and Discussion

An experimental setup of a mobile robot is shown in Figure 10 that appears to be designed for human detection or autonomous navigation. The Pi Camera takes real-time video input. Raspberry Pi runs an object detection algorithm to identify objects or humans. Motor speed and direction are adjusted using the motor driver. The motor encoders are used in the system for precise speed control.

The PWM (Pulse Width Modulation) duty cycle vs. motor speed characteristics in a DC motor control system is shown in Table 1. It highlights how the motor responds to different PWM duty cycles and provides insight into the accuracy of speed control. This table presents experimental data showing how varying the PWM duty cycle affects a DC motor's voltage, desired speed, actual speed, and speed error. PWM Duty Cycle (%) indicates the percentage of time the PWM signal is high (ON) during each cycle. Higher duty cycles provide more power to the motor.

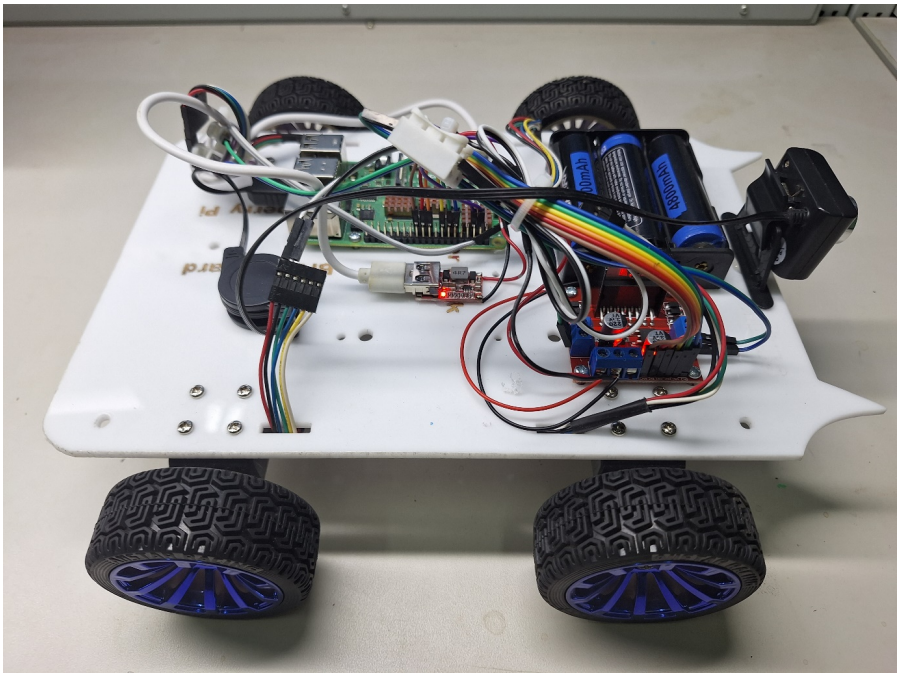

Figure 10. Experimental Setup of Human Detection Robot

Table 1. Duty Cycle vs. Motor Speed Characteristics

PWM Duty Cycle(%)	Voltage(v)	Desired Motor Speed (RPM)	Actual Motor Speed (RPM)	Error (%)
20%	2.4	26	24	7.6%
40%	4.8	52	49	5.8%
60%	7.2	78	74	5.1%
80%	9.6	104	99	4.8%
100%	12	130	125	3.8%

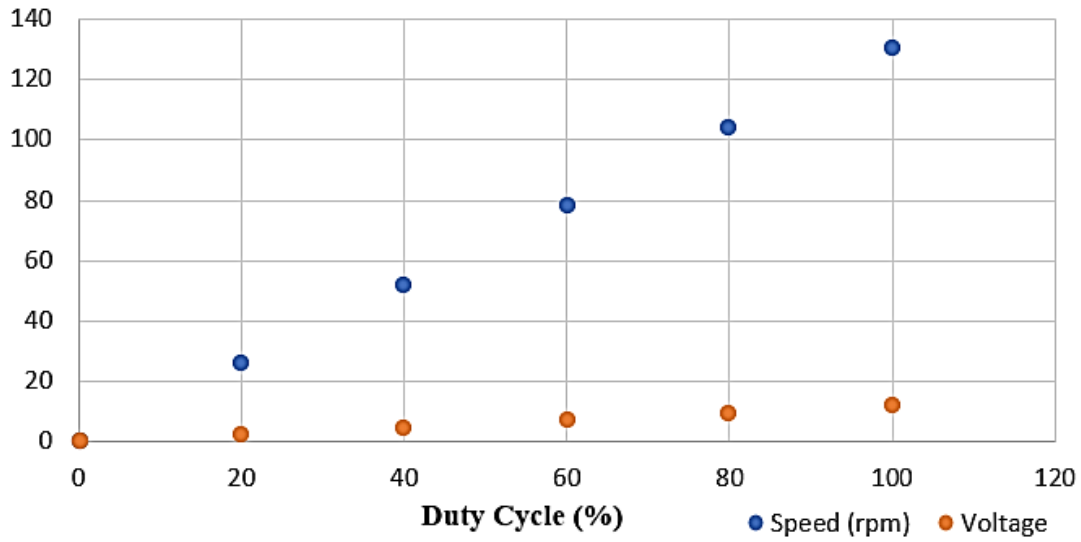

Figure 11. Motor Speed and Voltage at Different Duty Cycle

The relationship between PWM Duty Cycle (%) and the corresponding values of Motor Speed (RPM) and Voltage (V) is shown in Figure 11. X-axis (Duty Cycle %) represents the PWM (Pulse Width Modulation) duty cycle values ranging from 0% to 100%. PWM controls how much voltage is delivered to the motor, which in turn affects its speed. Y-axis represents two different metrics: Blue dots is Motor speed in RPM. Orange dots is Voltage applied to the motor. The motor speed increases almost linearly with the increase in duty cycle. At 20% duty cycle, speed is around 26 RPM. At 100% duty cycle, speed reaches 130 RPM. This indicates that the PWM duty cycle effectively controls the motor speed. Voltage also increases proportionally with duty cycle. At 20% duty cycle, voltage is 2.4V, increasing to 12V at 100% duty cycle. This shows that the effective voltage across the motor increases as the PWM duty cycle increases. Voltage (V) is the effective voltage delivered to the motor at each duty cycle. It increases proportionally with the duty cycle. Desired Motor Speed (RPM) is the target or reference speed expected at each corresponding duty cycle. Actual Motor Speed (RPM) is the real motor speed measured during operation. Error (%) is the percentage difference between desired and actual speed. Voltage and speed increase with duty cycle.At 20% duty cycle(2.4V), desired speed is 26 RPM and actual speed is 24 RPM. At 100% duty cycle (12V), desired speed is 130 RPM and actual speed is 125 RPM. At low duty cycles (20%), the error is relatively higher (7.6%).At high duty cycles (100%), the error reduces significantly (3.8%).

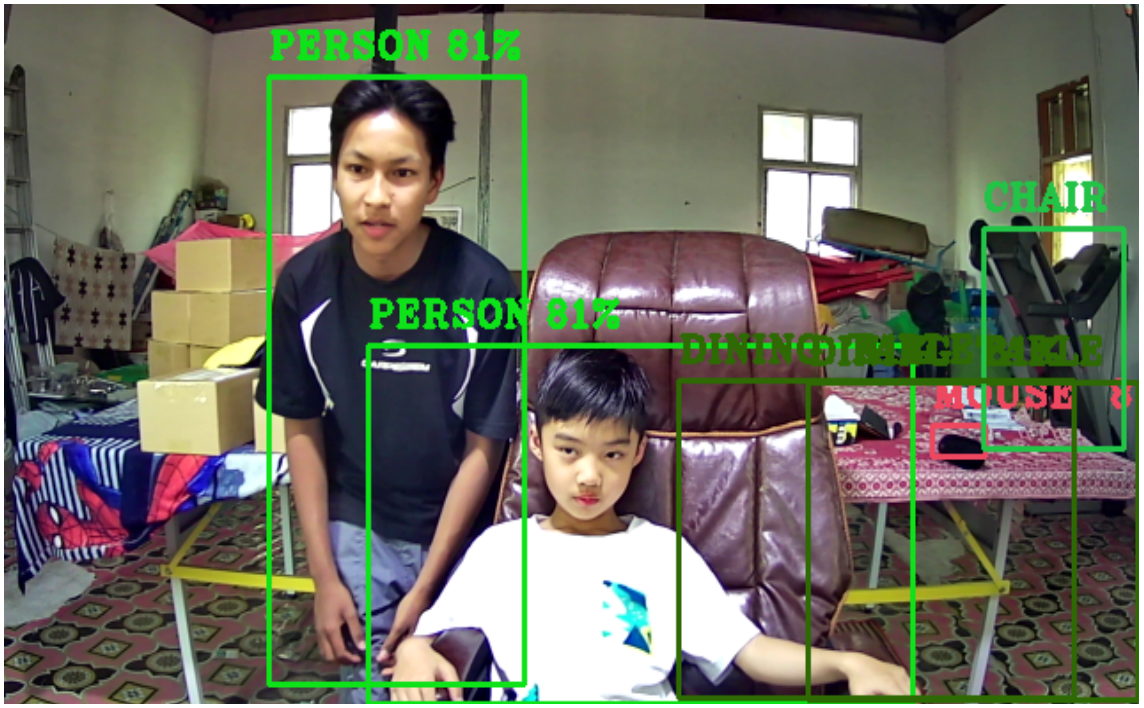

Figure 12. YOLOv8 Object Detection Output(at 2 feet distance)

The output of the YOLOv8 (You Only Look Once, version 8) object detection model applied to an indoor environment using a Pi camera is shown in Figure 12. The system is part of a human detection robot setup and is used to detect multiple objects in real time. Two persons is detected, each labeled with 81% confidence. Each object is surrounded by a bounding box and labeled with the class name and confidence score. The camera was placed at a distance of 2 feet from the persons. Under this setup, the YOLOv8 model achieved 99% detection accuracy.The accuracy was evaluated based on 100 total testing samples.This means 99 images were correctly detected, and only 1 image was incorrectly classified or missed. The high accuracy at close range (2 feet) indicates strong performance of YOLOv8 for short-range human detection tasks.

Figure 13. YOLOv8 Object Detection Output(at 15 feet distance)

Figure 14. YOLOv8 Object Detection Output(at 25 feet distance)

The real-time detection of a person in an indoor environment using an object detection system is shown in Figure 13. The bounding box labeled "PERSON 62%" shows the system has successfully detected a person with 62% confidence. The person is located 15 feet away from the camera. Out of 100 total test samples, the detection model correctly identified persons in 98 of them, yielding a 98% accuracy for this distance. The detection model performs very well even at a

distance of 15 feet, demonstrating robust accuracy (98%) in real-world, cluttered environments. Figure 14 showcases the object detection model identifying multiple objects in an indoor environment, with a specific focus on person detection at a long distance. The model detected a person at the center of the image, highlighted with a bounding box labeled "PERSON 61%." he detection confidence is 61%, indicating the model is less certain than at shorter distances. The person is 25 feet away from the camera.This is a long-range detection, challenging due to reduced resolution and possible occlusion. From 100 total testing samples taken at this 25-foot range, the model correctly detected the person in 96 cases, resulting in 96% detection accuracy. Despite the low confidence score (61%) for this specific frame, the overall accuracy at 25 feet remains high (96%), demonstrating the robustness of the detection model even at longer ranges.

G. Conclusion

The human detection robot, utilizing the YOLOv8 model and COCO dataset, demonstrates high accuracy in detecting humans at varying distances.The accuracy remains exceptionally high, with 99% at 2 feet, 98% at 15 feet, and 96% at 25 feet.This indicates the reliability of the system in different scenarios, making it suitable for applications requiring precise human detection, such as recuse operation, security, surveillance, and automation.

H. Acknowledgment

The author is deeply gratitude to Dr. San Yu Khaing, (Acting) Rector of Mandalay Technology University for the continuous support of the research. The author wishes to extend grateful thanks to Dr. Tin Tin Hla, Professor and Head, Electronic Engineering Department, Mandalay Technology University for her kind help and invaluable suggestions. The author would like to express her sincere deepest gratitude to Dr. Maung Aye, supervisor for giving his suggestions, advice, supervision and support throughout the paper writing period. The author specially thanks all her teachers from the Department of Electronic Engineering, Mandalay Technological University for the development of this paper. Especially, the author would like to thank her parents for their help and encouragement and also thanks all her friends.

I. References

[1] M. Vajgl, P. Hurtik, and T. Nejezchleba, ‘‘Dist-YOLO: Fast object detec tion with distance estimation,’’ Appl. Sci., vol. 12, no. 3, p. 1354, Jan. 2022, doi: 10.3390/app12031354.

[2] J. Yu and H. Choi, “YOLO MDE: Object detection with monocular depth estimation”, Electronics, 2022, vol. 11, no. 1, p. 76.

[3] B. Strbac, M. Gostovic, Z. Lukac, and D. Samardzija, ‘‘YOLO multi-camera object detection and distance estimation,’’ in Proc. Zooming Innov. Consum. Technol. Conf. (ZINC), May 2020, pp. 26–30, doi: 10.1109/ZINC50678.2020.9161805

[4] Z. J. Khow et al.,“Improved YOLOv8 Model for a Comprehensive Approach to Object Detection and Distance Estimation”, IEEE Access, 2024, Vol. 12, pp. 63574-63767.

[5] Vandit Gajjar, Ayesha Gurnani,Yash Khandhediya,., “Human Detection and Tracking for Video Surveillance: A Cognitive Science Approach,” IEEE International Conference on Computer Vision Workshops,2017

[6] Hashim Maso Kahily and A.P Sudheer “Real Time Human Detetion and tracking from a Mobile Armed Robot using RGB-D Sensor” World Conference on Futuristic Trends in Research and Innovation for Social Welfare, 2016 IEEE , ISSN 978-1-4673-9214-3L

[7] H. Xu, X. Lv, X. Wang, Z. Ren and R. Chellappa, "Deep Regionlets for Object Detection," arXiv:1712.02408v1, December 2017..

[8] H. Chen, Y. Wang, G. Wang and Y. Qiao, "LSTD: A Low-Shot Transfer Detector for Object Detection," arXiv:1803.01529v1, 2018.

[9] J. Huang, A. Fathi, V. Rathod, I. Fischer, C. Sun, Z. Wojna, M. Zhu, Y. Song, A. Korattikara, S. Guadarrama and K. Murphy, "Speed/accuracy trade-offs for modern convolutional object detectors," arXiv:1611.10012v3, pp. 1-21, April 2017.

[10] Prajakta A Patil, Prachi A Deshpande, “Moving Object Extraction Based on Background Reconstruction”, International Journal of Innovative Research in Computer and Communication Engineering, Vol. 3, Issue 4, April 2015.