The Indonesian Journal of Computer Science
www.ijcs.net
Volume 14, Issue 2, April 2025
https://doi.org/10.33022/ijcs.v14i2.4849

Vision-based Obstacle Detection and Motor Speed Control for Autonomous Driving Systems

Aye Nilar Win[1], Zin Mar Lwin[2], Tin Tin Hla[3]
ayenilar.ec2010@gmail.com[1], dr.zinmar80@gmail.com[2], tintinhla99@gmail.com[3]
[1,3]Department of Electronic Engineering, Mandalay Technological University, Myanmar
[2]Department of Remote Sensing, Mandalay Technological University, Myanmar

Article Information

Received : 21 Apr 2025
Revised : 28 Apr 2025
Accepted : 30 Apr 2025

Keywords

vehicle detection, motor speed control, MobileNet SSD

Abstract

Autonomous driving systems rely on robust perception and control mechanisms to navigate safely in dynamic environments. This study presents a vision-based approach for obstacle detection and motor speed control using MobileNet SSD object detection model. The system utilizes a camera module to capture real-time video frames, which are processed to detect and classify obstacles. Based on the detected objects' position and distance, an adaptive motor speed control algorithm adjusts the vehicle's velocity to ensure collision avoidance and smooth navigation. The implementation is tested on a Raspberry Pi-based platform with an integrated motor control system, utilizing PWM signals for speed regulation. MobileNet SSD offers a lightweight, faster alternative for real time inference. Experimental results demonstrate the system's effectiveness in detecting obstacles and dynamically adjusting speed in response to environmental conditions. This approaches enhance autonomous vehicle safety and efficiency, making it suitable for real-world applications in self-driving technologies.

A. Introduction

Autonomous driving systems rely on advanced perception and control mechanisms to navigate safely in dynamic environments. One of the key challenges in autonomous navigation is the ability to detect obstacles accurately and adjust motor speed accordingly to prevent collisions. Traditional sensor-based approaches, such as LiDAR and ultrasonic sensors, offer obstacle detection capabilities but often come with limitations such as high cost, complexity, and reduced performance in certain environmental conditions.

This research focuses on developing a vision-based obstacle detection and motor speed control system using MobileNet SSD model. The system employs a camera module to capture real-time video data, which is processed to detect obstacles. Based on the detected obstacles' position and distance, a motor speed control mechanism dynamically adjusts the vehicle's speed using PWM signals, ensuring smooth navigation and collision avoidance. The overall system flow is shown in Figure 1.

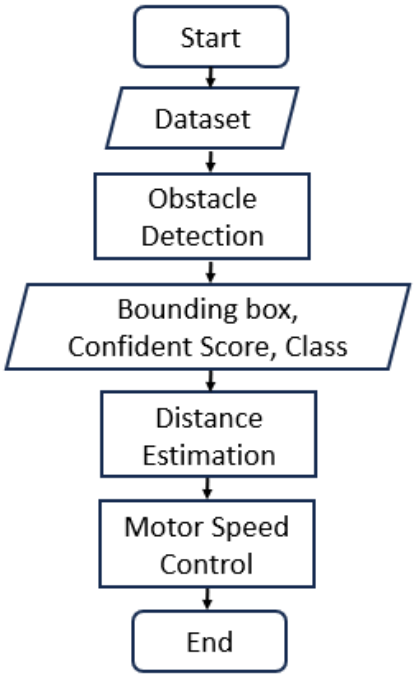

Figure 1. Overall system flow of the proposed System

The proposed approach is implemented on a Raspberry Pi-based platform, integrating computer vision with motor control for an efficient and cost-effective autonomous driving system. The study aims to enhance the safety and reliability of autonomous vehicles by leveraging deep learning-based vision techniques. The following sections will discuss the system architecture, implementation details, and performance evaluation of the proposed method.

The organization of the study includes five sections: the first section is introduction of the obstacle detection and motor speed control system. The next section follows related works of the system, section C discusses the methodology of the proposed system, results and discussion follow the next section D, and the final one conclude the research.

B. Related Works

Obstacle detection is a crucial task in autonomous navigation systems, robotics, and intelligent transportation systems. Over the years, various techniques have been proposed to improve the accuracy, efficiency, and reliability of obstacle detection. Traditional methods relied on handcrafted features and geometric models using image processing techniques, while recent advancements leverage deep learning-based approaches to automatically extract meaningful features from sensor data. The integration of sensors such as cameras, LiDAR, and radar has significantly enhanced obstacle detection capabilities. This section reviews the existing studies on obstacle detection, highlighting the evolution of techniques, the role of sensor fusion, and the impact of deep learning models in improving detection performance.

Obstacle detection in rail transit system is criticle to ensure oprational safety especially in challenging environmental conditions such a dense fog. Traditional detection methods often struggled with missed and false detection due to poor image quality in low-visibility scenarios. To overcome these limitations, J. Chen et. al., [1] proposed a Multi-Scale Adaptive YOLO (MSA-YOLO) algorithm that enhances obstacle detection performance under foggy conditions. The method combines image enhancement techniques with deep learning-based object detection to improve accuracy in adverse weather environments. The MSA-YOLO algorithm integrates six image enhancement filters such as defog, white balance, gamma correction, contrast adjustment, tone mapping, and sharpening to improve image quality before detection. However, determining optimal hyperparameters for these filters is challenging due to varying environmental conditions. To address this issue, the authors introduced a multi-scale adaptive module that dynamically adjusts filter hyper parameter, improving the overall image enhancement process. The enhanced images are then processed by the YOLO object detection model to accurately identify obstacles on rail transit tracks. This approach outperform traditional rail transit detection methods in foggy scenarios reducing both missed and false detection.

Automated vehicles in industrial environments require reliable obstacle detection systems to ensure operational safety and efficiency. Traditional obstacle detection methods often rely on distance-measuring laser scanners, which are highly accurate but not commonly integrated into commercial vehicles. To address this limitation, M. Wenning et. al., [2] proposed an anomaly-based obstacle detection system using camera-based computer vision algorithms, leveraging existing monocameras typically installed in modern vehicles. This approach eliminates the need for additional hardware while providing a robust solution for obstacle detection in industrial settings. The proposed method utilizes an anomaly detection algorithm to identify obstacles by learning the normal appearance of the vehicle's path. The model consists of a pre-trained feature extractor and a shallow classifier, which estimates the probability of occurrence for each image region. The algorithm is trained on images captured during a single run through the industrial environment, without requiring pixel-level annotations or large datasets. This makes the approach highly efficient compared to traditional semantic segmentation methods, which require extensive labeled data for training. This

method represents a cost-effective and salable solution for enhancing safety in automated guided vehicles and industrial autonomous systems.

The development of self-driving car prototypes has gained significant attention in autonomous vehicle research, aiming to enhance road safety and driving efficiency. The study in [3] designed and visualized a prototype self-driving car system using Raspberry Pi. The project showed the process of stop sign and traffic light detection system with Lane detection system. The system is designed to enable autonomous navigation and decision-making under various environmental conditions. The proposed system employs image processing-based lane detection algorithms to accurately identify lane markings, ensuring safe navigation along the road. Although the lane detection method demonstrates high effectiveness in clear weather conditions, its performance degrades in adverse weather scenarios such as rain and fog**.** The proposed self-driving car prototype represents a significant step toward developing intelligent transportation systems capable of making real-time decisions for safe and efficient autonomous driving.

R. Fang and C. Cai [4] proposed a computer vision-based scheme that integrates deep learning techniques to achieve real-time obstacle detection and target tracking. The system combined ResNet-18 model for obstacle detection in the vehicle environment and YOLOV3 for real time target tracking, offering a robust solution for autonomous navigation. The method utilized to extract high-level features, enabling accurate obstacle detection in complex scenes. The YOLOv3 model is employed for real-time target tracking, which allows the vehicle to continuously follow a tracked object. The entire system is deployed on an NVIDIA Jetson Nano motherboard, a low-power embedded platform suitable for real-time autonomous applications. To ensure stable and precise navigation, the vehicle's steering and movement are controlled using a Proportional-Integral-Derivative (PID) algorithm. This algorithm adjusts the vehicle's motion based on detected obstacles and tracked targets, enhancing the system's overall stability and accuracy.

A monocular vision-based algorithm is proposed in [5] to detects obstacles and generates obstacle-aware regions to facilitate safe navigation in unknown environments. This approach leverages deep learning techniques to estimate depth information from a single camera view, making it a more cost-effective alternative to binocular systems. The algorithm consists of three main stages. First, a deep encoder-decoder network is employed to predict a disparity image from a monocular input image, estimating depth information. The predicted disparity image is then categorized into three classes—obstacle, road, or obstacle-free—by combining V-disparity analysis with a fuzzy inference system. This classification enables the identification of obstacle-aware regions within the visual perception field. Based on the detected obstacle regions, the system generates intermediate waypoint to create a flyable path that ensures the multicopter's safety by maintaining appropriate margins around detected obstacles. The study highlights the feasibility of using monocular vision for autonomous navigation, especially in scenarios where weight, power consumption, and cost are critical factors.

In the application of self-driving cars, convolutional neural networks (CNNs) are widely used for autonomous navigation through real-time perception. proposed an autonomous car model that utilizes Convolutional Neural Networks

(CNNs) for predicting vehicle steering directions based on camera images. The system employs Raspberry Pi 4 Model B as both the control and processing unit, offering a compact and cost-effective solution for autonomous navigation. The model captures real-time images from the Raspberry Pi camera, which are processed by the trained CNN to predict the appropriate steering direction. The Raspberry Pi then sends control signals to the L298n motor driver to adjust the car's movement [6].

The detection of atypical aviation obstacles is crucial for ensuring air traffic safety om rapidly developing urban areas near airports. An automated approach for identifying and classifying atypical aviation obstacles is proposed using unmanned aerial vehicles (UAV) and YOLO algorithm. The UAV equipped with high-resolution cameras provide up-to-date spatial data and the YOLO algorithm enables real-time obstacle detection [7]. In paddy field, an obstacle detection and tracking system [8] is established using a machine vision system mounted on a transplanter to detect moving obstacles such as humans and water buffalo. The system combines an improved YOLOv3 model with deep SORT for accurate obstacle detection and real-time tracking.

The study in [9] provides a comprehensive review of robust control techniques for Permanent Magnet Synchronous Motors (PMSMs). PMSMs exhibited nonlinear characteristics with time-varying parameters, making precise speed regulation challenging. The study focuses on robust speed controllers designed to ensure accurate speed tracking, minimal overshoot, and strong disturbance rejection. Through a comparative analysis, the study outlines the evolution and trends in PMSM speed regulation methods, offering valuable insights for optimizing motor control performance in industrial applications. In the research [10] , the authors proposed an enhanced integral sliding model control (ISMC) system to regulate the mechanical speed of an induction motor under model uncertainties and load torque variations. The ISMC provides a faster speed convergence rate and reduces the chattering effect through a switching sigmoid function.

C. Research Method

The proposed system integrates MobileNet SSD model for real-time obstacle detection with a motor speed control mechanism to ensure safe navigation. The methodology consists of three main stages data acquisition, obstacle detection, and motor speed control.

Dataset Preparation

The Common Object in Context (COCO) dataset [11] is a large-scale, widely used benchmark dataset for object detection, segmentation, and image captioning. It was developed by the Microsoft COCO project and is extensively used for training and evaluating deep learning models. It consists of 80 object categories, including people, vehicles, animals, and household items. The dataset provides bounding box, instance segmentation masks, keypoints, and image captions. The dataset contains 118000 images of training, 5000 images of validation, and 40000 images of testing. For real-world testing, a monocular RGB camera mounted on the autonomous vehicle captures real-time video frames for obstacle detection. The

collected image dataset includes various road condition, lighting scenarios, and obstacle types to enhance model robustness. Data preprocessing techniques such as image resizing, normalization are applied to improve detection performance.

Object Detection using MobileNet SSD

The MobileNet SSD network [12] is a lightweight and efficient object detection framework designed for resource-constrained devices. It combines the MobileNet architecture, which uses depthwise separable convolutions to reduce computational complexity, with the SSD framework [13], which enables detection at multiple scales. MobileNet SSD is known for its fast inference speed and moderate accuracy, making it suitable for real-time applications like self driving cars.

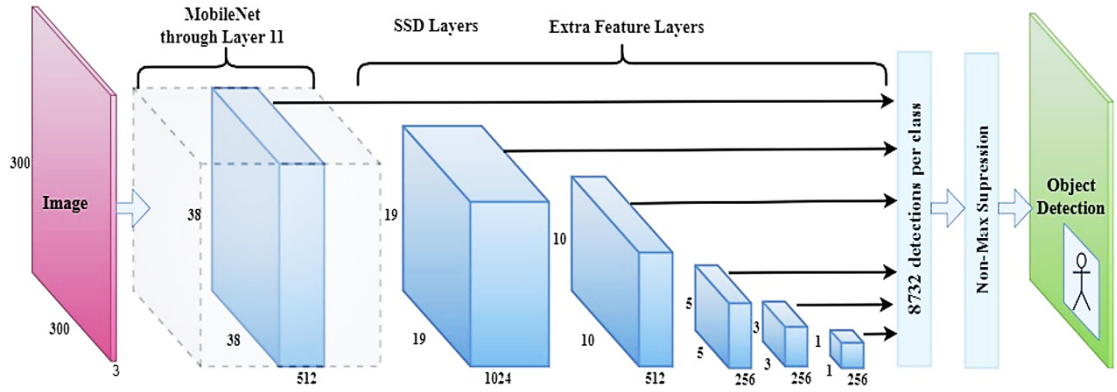

Figure 2. The architecture of MobileNet SSD

In the vision-based obstacle detection system, MobileNet SSD combines the efficiency of MobileNet with the SSD framework for real-time object detection, making it suitable for edge devices with limited computational resources as shown in Figure 2. The architecture of the model consists of backbone network (MobileNet) and detection head (Single Shot detector). This model reduces computational complexity by splitting standard convolutions into depthwise convolution by applying a single filter per input channel and point wise convolutions by combining outputs using 1×1 convolutions.

The detection head SSD detects objects with multi-scale feature maps at multiple resolutions using feature maps from different layers. An anchor boxes are used to predefined aspect ratios to localize objects of varying shapes. The output is class probabilities and bounding box offsets for each anchor box. This model provides a computationally efficient solution for real-time obstacle detection in autonomous driving systems, particularly suited for resource-constrained edge devices.

Distance Estimation

The system estimates the distance to the obstacle using the detected bounding box size and a precalibrated depth model. A calibration process is conducted by capturing images of objects with known dimensions at various distances. A relationship between bounding box size and distance is established.

Monocular depth approximation is achieved using bounding box scaling. The distance *D* to an obstacle is inversely proportional to its pixel height *h* in the image illustrated in Equation (1):

$$D = \frac{W \times f}{w} \tag{1}$$

where *D* is the distance to the obstacle, *W* is actual width of the object, and *f* is the focal length of the camera obtained through calibration. The parameter *w* is the width of the detected object bounding box.

Motor Speed Control

Motor speed control is the process of regulating the speed of an electric motor to achieve desired performance. In the context of vision-based obstacle detection system, motor speed control involves dynamically adjusting the speed of the DC motor based on the proximity of detected obstacles. The Raspberry Pi processes input from the camera and ultrasonic sensor, then sends signals to the motor driver to either increase, decrease, overtaking, or stop the motor speed. It ensures smooth navigation, prevents collisions, and enhances the system's overall responsiveness and safety. The circuit diagram of the system is shown in Figure 3.

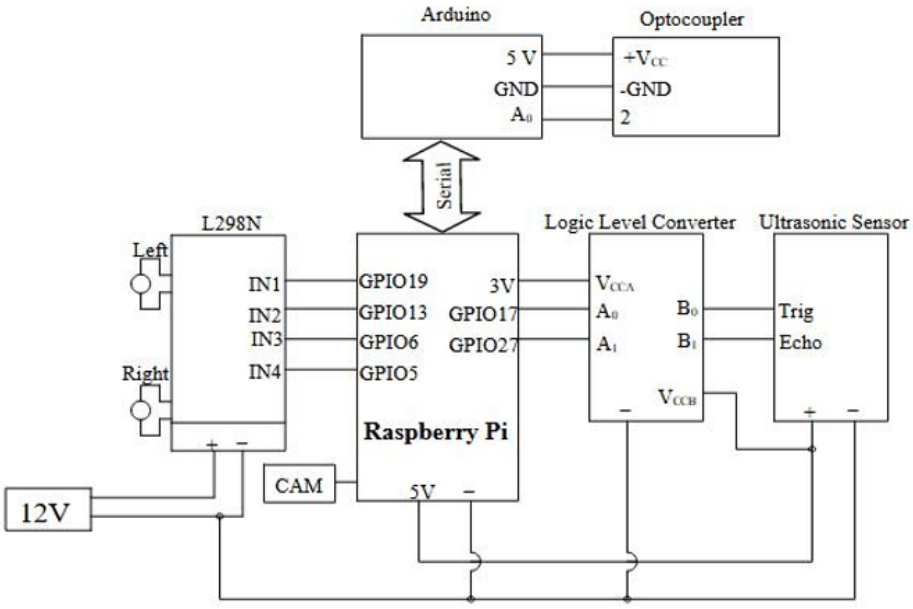

Figure 3. Overall circuit diagram for the proposed system

The proposed system is designed by the controller Raspberry Pi 4B which serves as the central processing unit, handling obstacle detection and controlling motor speed based on detected obstacles. The system is powered by a 12V power source, which supplies power to the L298N motor driver for driving the DC motors. Multiple GPIO pins on the Raspberry Pi are used to interface with different components: GPIO19, GPIO13, GPIO6, GPIO5. These are connected to the IN1, IN2,

IN3, and IN4 pins of the L298N motor driver, these control the direction and speed of the left and right motors. GPIO17, and GPIP27 are connected to the logic converter to handle communication between the Raspberry Pi and the ultrasonic sensor.

The camera module is connected to the Raspberry Pi to capture real-time video frames for obstacle detection using MobileNet SSD. The L298N motor driver controls two DC motors such as left and right. It receives control signals from the Raspberry Pi GPIO pins to manage motor direction and speed. The ultrasonic sensor helps measure the distance to obstacles. It uses a "Trig" pin to send out ultrasonic pulses and an "Echo" pin to receive the reflected signals.

The logic level converter bridges the voltage difference between the Raspberry Pi 3.3V logic level and the ultrasonic sensor 5V logic level. It ensures proper signal transmission between the two components. The optocoupler isolates different parts of the circuit, protecting sensitive components like the Raspberry Pi from high-voltage spikes. It ensures electrical isolation between the input and output signals. The arduino is connected to the Raspberry Pi via a serial connection. It handles additional sensor data or motor control signals, enhancing the system modularity.

The Raspberry Pi processes video frames for obstacle detection and uses distance data from the ultrasonic sensor to adjust motor speed. The L298N driver receives commands from the Raspberry Pi to control motor movements, ensuring smooth navigation in response to detected obstacles.

The optocoupler works as an electrical isolator between the motor control circuit and the Arduino. The input side of optocoupler has an LED that lights up when voltage is applied across +Vcc and -GND. The output side has a phototransistor that turns ON/OFF based on LED light that creats a signal for the Arduino. It connects +Vcc to the Arduino 5V pin, -GND to the Arduino GND pin, and place a current-limiting resistor in series Vcc to protect the optocoupler internal LED. In the phototransistor side, it add a pull-up resistor between A0 (analog pin) and the Arduino 5V to ensure a stable signal. When the motor control circuit sends a signal to the optocoupler, the internal LED lights up. This light activates the phototransistor that allows current to flow on the output side, sending a signal to the Arduino A0 pin. The Arduino reads this signal and adjusts the motor speed accordingly using PWM. The optocoupler transmits PWM signal from the microcontroller to a motor driver for controlling the motor speed.

To ensure safe navigation, the system integrates an adaptive speed control mechanism using a proportional-integral-derivatice (PID) controller. The bounding box coordinates from MobileNet SSD are converted into distance estimates using a calibrated depth approximation. Based on obstacle distance and classification, the system dynamically adjusts the motor speed. If no obstacle detected, the system maintain normal speed. If the system detects obstacle at moderate distance, it takes gradual speed reduction. If the detected obstacle is in close proximity, the system immediate braking or full stop. The PID controller fine-tunes motor speed to ensure smooth transitions and prevent abrupt stops, improving driving stability.

The PID controller adjusts the motor PWM in equation (2) duty cycle based on the error (e(t)) between the target RPM from obstacle distance and actual RPM

from encoder feedback. The method of the motor control system to ensure adaptive speed regulation based on real-time obstacle detection. The PID controller gains are tuned empirically or via optimaization algorithms to balance responsiveness and stability.

$$u(t) = K_p \cdot e(t) + K_i \cdot \int_0^t e(\tau)\, d\tau + K_d \cdot \frac{de(t)}{dt} \tag{2}$$

Where,

u(t): output PWM signal (0-100% duty cycle)
e(t) = Target RPM – Actual RPM: instantaneous error
Kp, *Ki*, *Kd*: Tuned gains

D. System Implementation and Testing

The obstacle detection and speed control system is deployed on an embedded hardware platform Raspberry Pi with real-time processing capabilities. The study employs on python programming language with tensorflow library and open CV library. The framework of the real-time obstacle detection and motor speed control is illustrated in Figure 4.

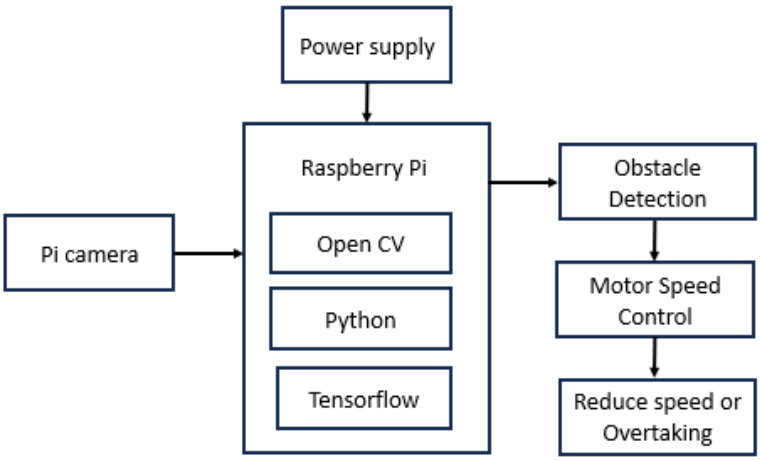

Figure 4. The real-time obstacle detection and motor speed control framework

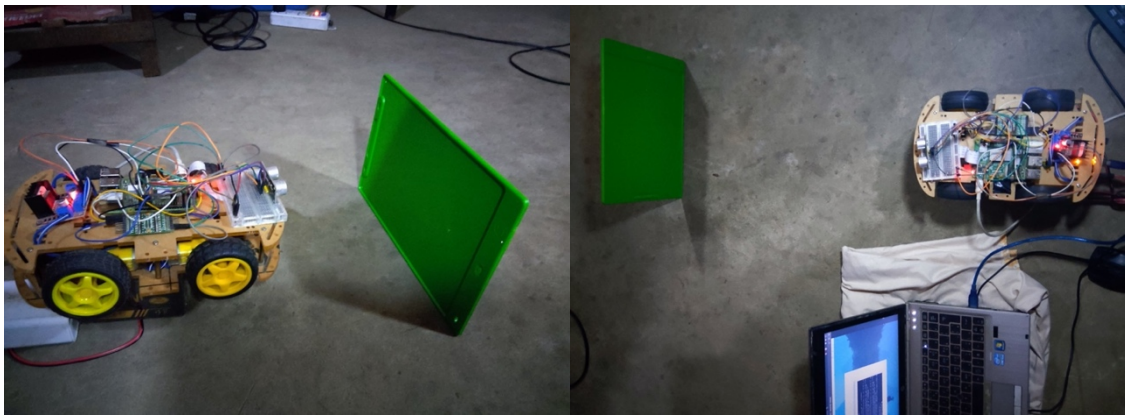

Figure 5. Testing for detecting obstacle

The paper designs real-world testing on an autonomous vehicle prototype as shown in Figure 5 under different driving conditions near or moderate distance. The deep learning model, MobileNet SSD is employed to detect obstacles in real-time. The detection pipeline follows these steps:

Model Training: A labeled dataset is used to train MobileNet SSD, ensuring it can identify obstacles such as pedestrains, vehicles, and roadblocks. The model was trained for 30 epochs, batch size 32, learning rate 0.001 using Adam optimizer on Laptop with 64 bits Window 11 operating system, 11th Gen Intel(R) Core(TM) i7-1165G7@2.80GHz, and RAM 8 GB.

Inference Process: The trained model processes incoming frames, outputting bounding boxes, confidence scores, and class labels for detected objects. A Raspberry Pi v2 camera captures video at 30 FPS. Frames are resized to 300×300 pixels for MobileNet SSD model. Non-maximum suppression (NMS; IoU threshold: 0.45) filters redundant detections.

Obstacle Classification: Based on the detected obstacle type and its position relative to the vehicle, decisions are made for speed adjustments.

Evaluation Method for Motor Speed Control

In the context of the motor speed control system, revolution per minute (RPM) is a critical metric for real-time motor speed adjustment that ensure smooth acceleration or deceleration based on obstacle proximity. Optical or magnetic encoders attached to the motor shafts generate pulse proportional to rotational speed. The onboard microcontroller Raspberry Pi counts encoder pulses per unit time to calculate RPM as in Equation (3).

$$\text{RPM} = \frac{\text{Number of pulses per second} \times 60}{\text{Number of pulses per revolution (PPR)}} \tag{3}$$

The target RPM is inversely proportional to obstacle proximity, the closer obstacles, the lower RPM. By tightly coupling RPM control with vision-based obstacle detection, the system achieves adaptive, safe, and efficient autonomous navigation.

E. Result and Discussion

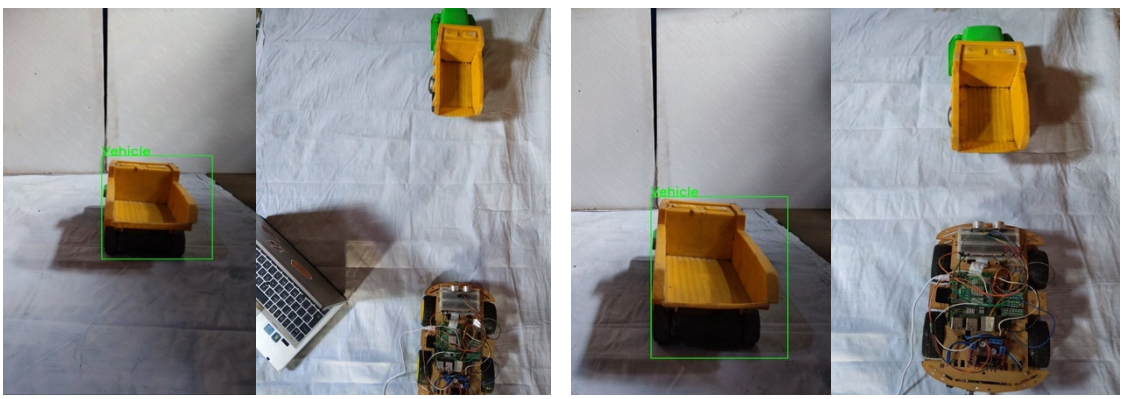

Figure 6. Real-time obstacle detection

The MobileNet SSD model is trained on COCO dataset and test the real-time obstacle detection work. In the experimentation, the system uses the parameter value Kp = 1.0, Ki = 0.1, and Kd = 0.05. The distance threshold value set to 20 cm to calculate whether the obstacle is near or not. In Figure 6, the first one is normal driving condition and the second is near obstacle and closely overtaking condition.

Experiment Without PID Controller

Figure 7 represents the motor speed control over time for 15 minutes. It shows RPM values as the system detects obstacles at a threshold distance of 20 cm, with the actual driving distance being 9 cm. As the system detects obstacles closer to the threshold value, the motor speed is reduced to maintain safety. The periodic dips in the RPM show the system slowing down in response to obstacles.

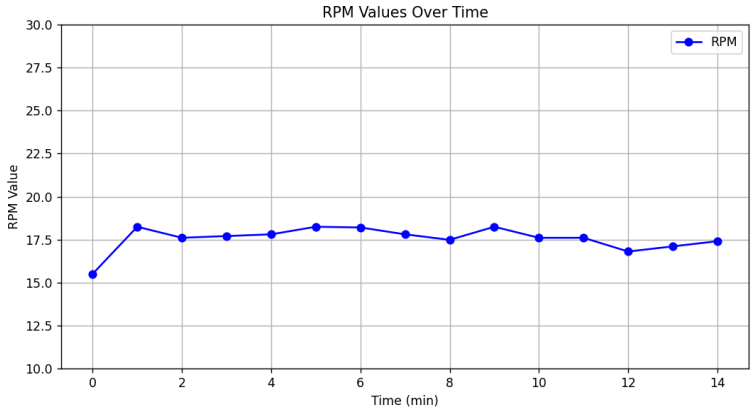

Figure 7. Motor speed control over time without PID (obstacle distance 9 cm)

The RPM values in Figure 8 are relatively higher and more stable compared to driving at 15 cm. Since the distance is greater, the system doesn't need to slow down as frequently, resulting in fewer fluctuations and more consistent motor speed. The threshold value of 20 cm means that the motor only reduces speed when the obstacle is within that range. Since the driving distance is 15 cm, the motor operates near the threshold most of the time, allowing it to maintain higher RPM without frequent breaking. The system demonstrates a balance between responsiveness and maintaining steady speed.

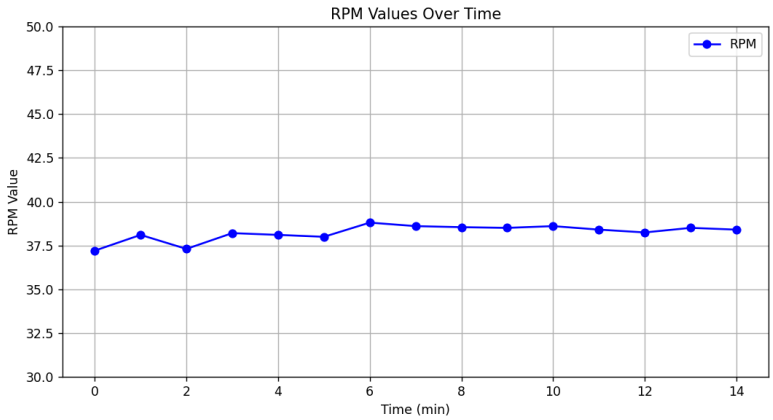

Figure 8. Motor speed control over time without PID (obstacle distance 15 cm)

To slow down a motor-driven system when detecting a closely approaching obstacle, a simple feedback mechanism is implemented without PID controller. Using ultrasonic sensor, the system can trigger a predefined speed reduction when the obstacle enters a critical threshold distance (20 cm). Figure 9 illustrates the speed reduction when detecting the obstacle in distance of 9 cm.

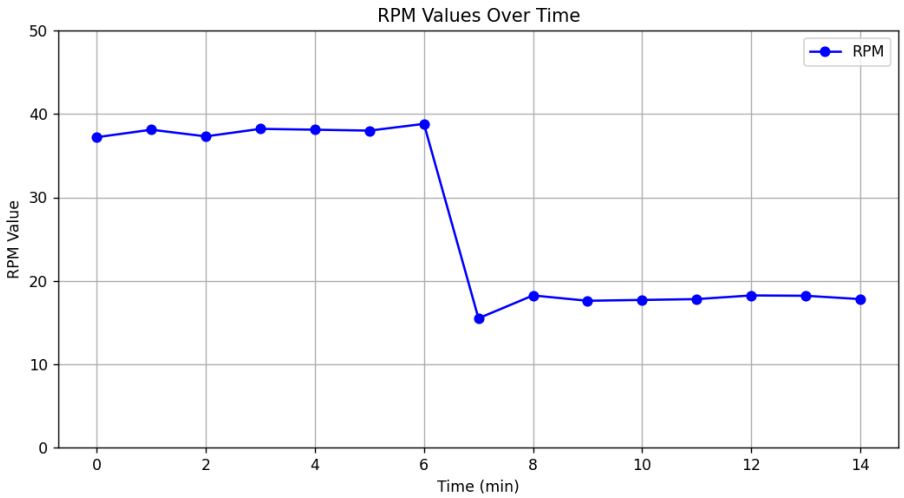

Figure 9. Reduce speed in closely obstacle without PID

Experiment With PID Controller

Figure 10 and 11 show the RPM values over time using PID controller, and the data looks quite stable with minor fluctuations. Proportional control adjusts the motor speed in proportion to the error between the desired and actual RPM. Integral control addresses accumulated past errors by summing them over time. This helps eliminate steady-state error, ensuring the RPM reaches and stays at the desired value. Derivative control predicts future errors by calculating the rate of change of the error. It helps dampen the system, preventing overshoot and improving stability. As shown in figures, the stability of RPM suggests that the PID controllers is tuned well, prevention oscillations and overshoots. The minor variations could be due to external disturbances, sensor noise, or slight changes in load, which the PID controller manages effectively.

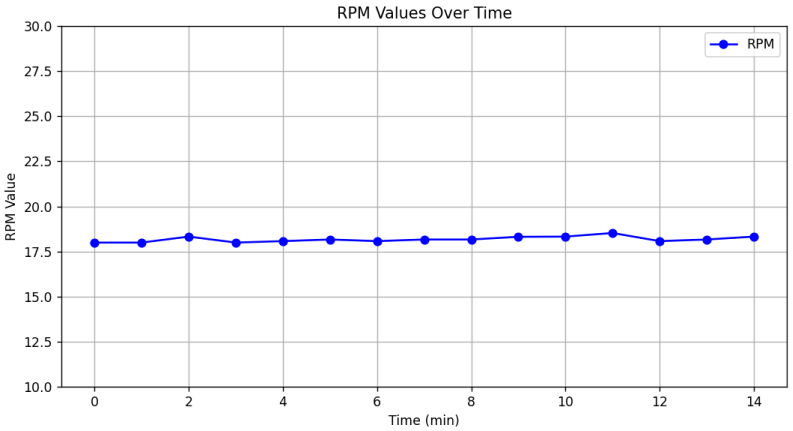

Figure 10. Motor speed control over time with PID (obstacle distance 9 cm)

In Figure 11, the obstacle distance is 15 cm, which is near the threshold value. Since the distance is closed to the threshold, the system does not need to apply emergency braking or rapid deceleration. The motor continues running at a relatively steady speed. Because the obstacle is far enough, the controller does not intervene aggressively. The RPM stays close to the setpoint, ensuring smooth motor operation.

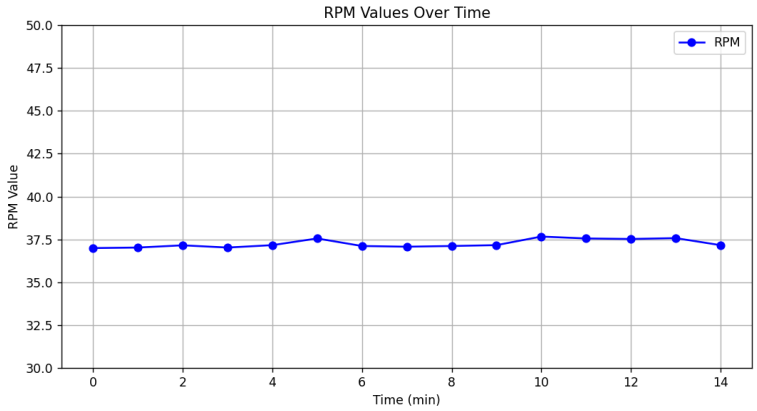

Figure 11. Motor speed control over time with PID (obstacle distance 15 cm)

In Figure 12, the system reaches to an obstacle detected within the threshold, and the PID controller is reducing the motor speed. At the start, the RPM is stable around 38 to 40 RPM. This indicates that the motor is running at its normal speed, with no immediate obstacles detected within the threshold. When the obstacle enters the threshold distance, the PID controller reduces the motor speed rapidly to avoid a collision. The RPM drops sharply from about 38 RPM to around 18 to 20 RPM.

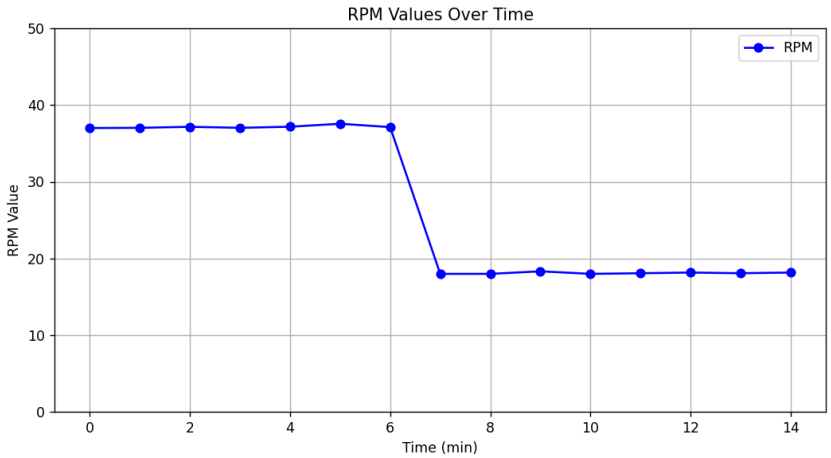

Figure 12. Reduce speed in closely obstacle with PID

PID Controller Response Results

Figure 13 shows the PID controller response to changes in the setpoint over time, with moderate Ki and Kd, it balances responsiveness and stability. Overshoot and oscillation are controlled but not completely eliminated. The controller adapts well to step changes in the setpoint.

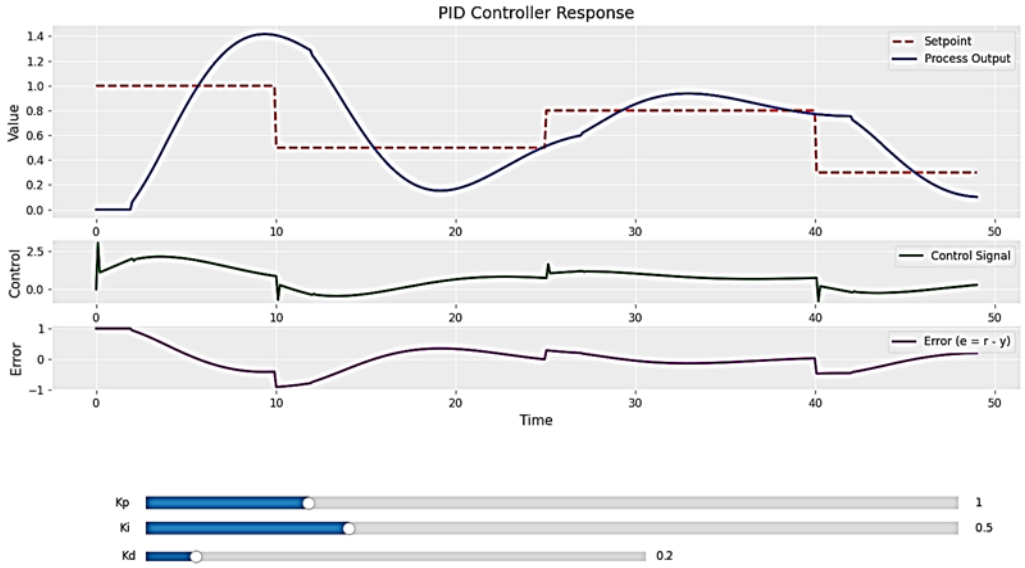

Figure 13. PID response with moderate integral and derivative gains

Figure 14 illustrates the PID controller response when only proportional control (P-control) is used, Ki = 0, Kd = 0, and Kp = 1. When it is compared to Figure 13 (with moderate Ki and Kd), this response is more stable but less accurate. It lacks the correction strength provided by the integral component and the damping of the derivative component.

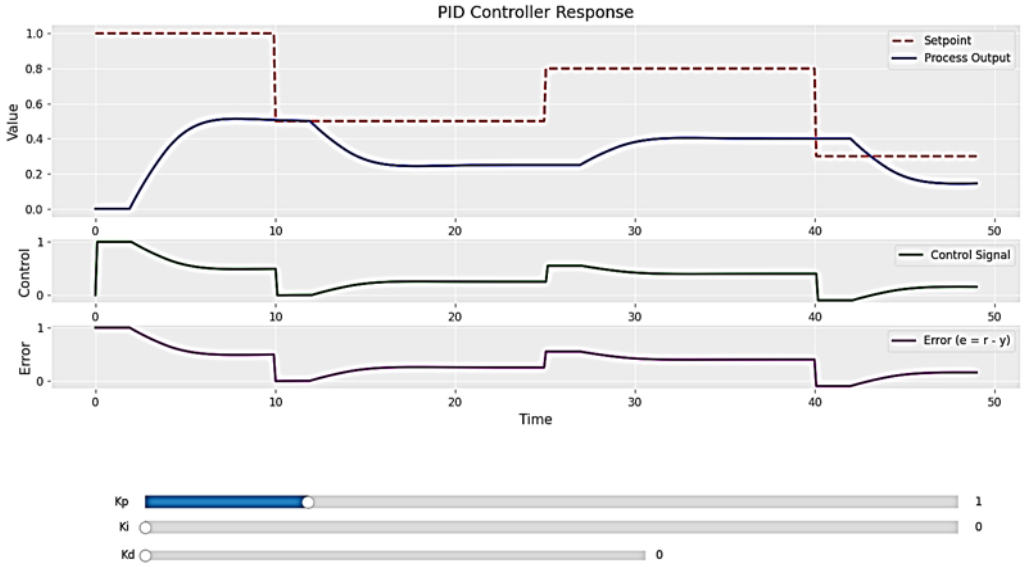

Figure 14. PID response with only proportional control

Figure 15 demonstrates the PID controller response with balanced gains, Kp = 1, Ki = 0.1, and Kd = 0.05. These values indicate a typical PID configuration with all three components active and moderately tuned. This system eliminates the steady-state error. It is smoother and more stable than the moderate gain PID (from Figure 13). It shows the benefit of tuning PID gains appropriately for stable and responsive control.

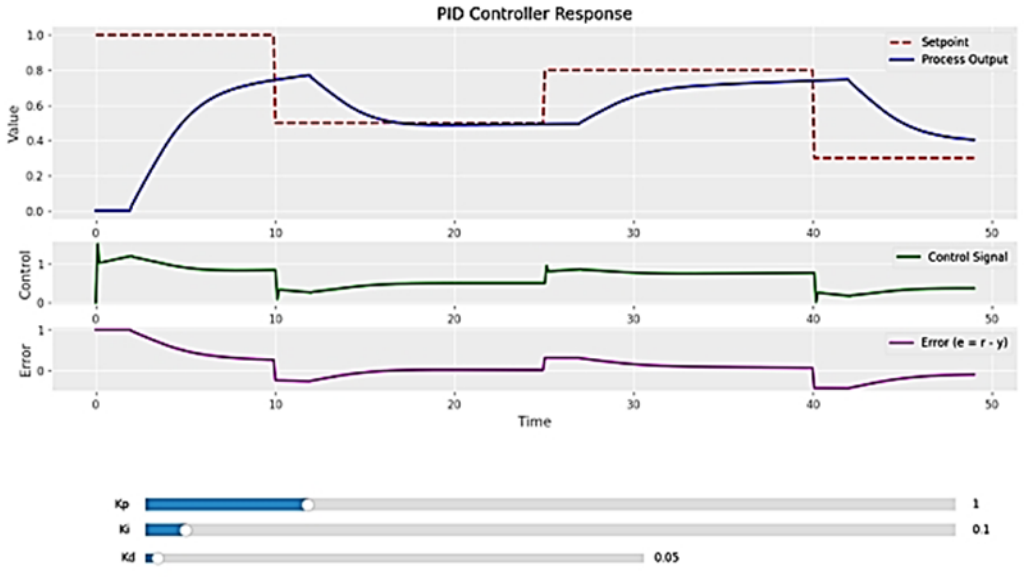

Figure 15. PID response with balanced gains

Figure 16 demonstrates the behavior of a system controlled by a PID controller with increased gain values. The system is kept stable and the setpoint is tracked fairly well, indicating that the controller has been well-tuned to be responsive without being overly aggressive.

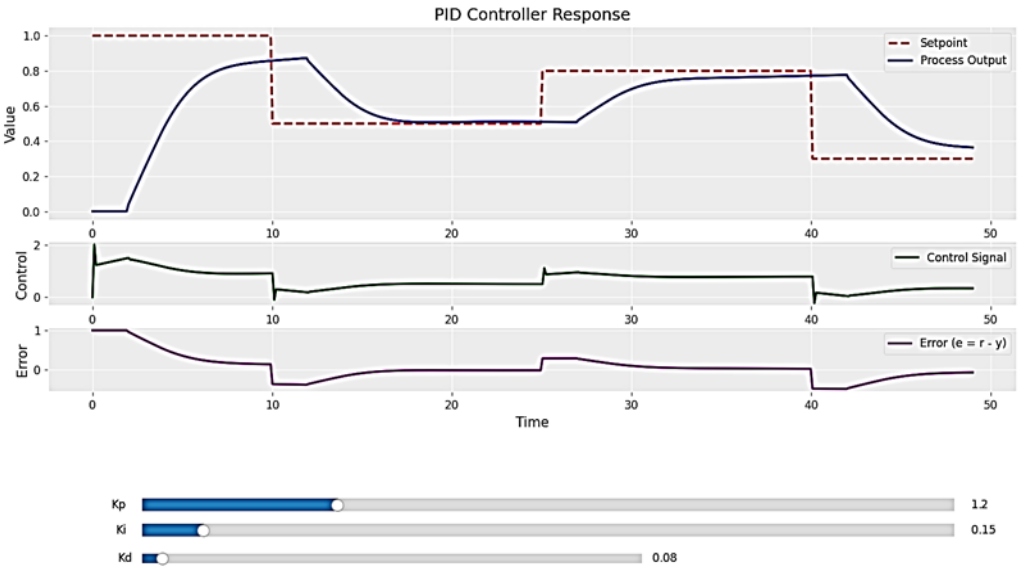

Figure 16. PID response with slightly higher gains

In Figure 17, it illustrates how a PID controller behaves when the tuning parameters are set to relatively high values, i.e., the controller is “aggressively” tuned. This kind of tuning is useful when fast tracking and minimal lag are desired, but it comes with the risk of instability if not carefully managed. If a PID controller is being tuned for a self-driving car’s speed or position, this figure suggests that aggressive tuning can be helpful in fast response scenarios, such as quick overtaking, but it must be used cautiously to prevent oscillation or system strain.

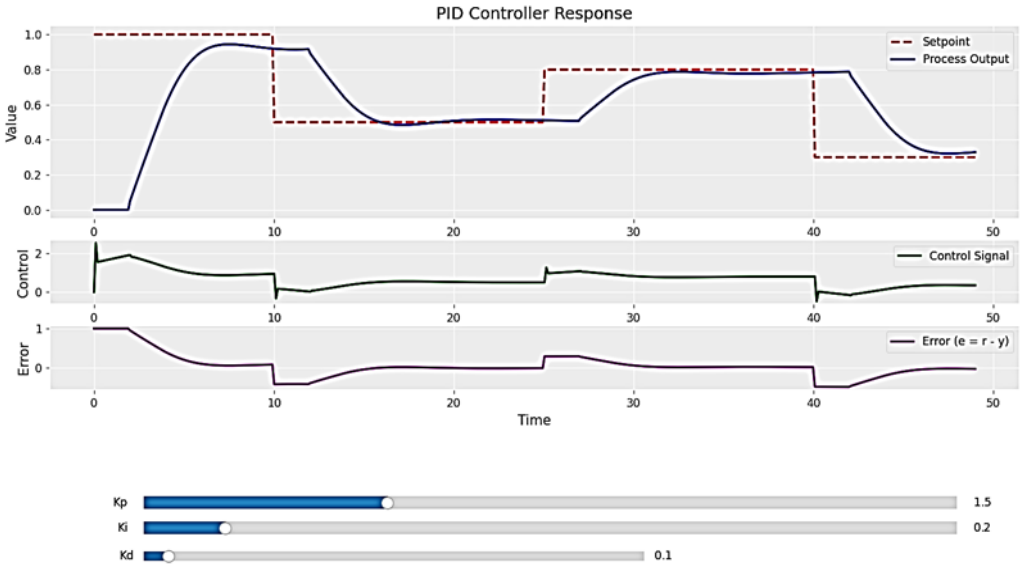

Figure 17. PID response with aggressive gains

Experiment in Overtaking Condition

When the system detects an obstacle within the 20 cm threshold, the PID controller immediately reduces the motor speed to avoid collision. The RPM drops from high RPM to low RPM. This reduction allows the system to carefully approach to obstacle while assessing the environment for a safe overtaking maneuver. While reducing speed, the system evaluates conditions for overtaking using input sensors. The system adjusts the motor speed and wheel direction to smoothly execute the overtaking maneuver. The PID controller increases RPM slightly to gain enough speed to overtake. Figure 18 shows the overtaking condition when the system detected obstacle.

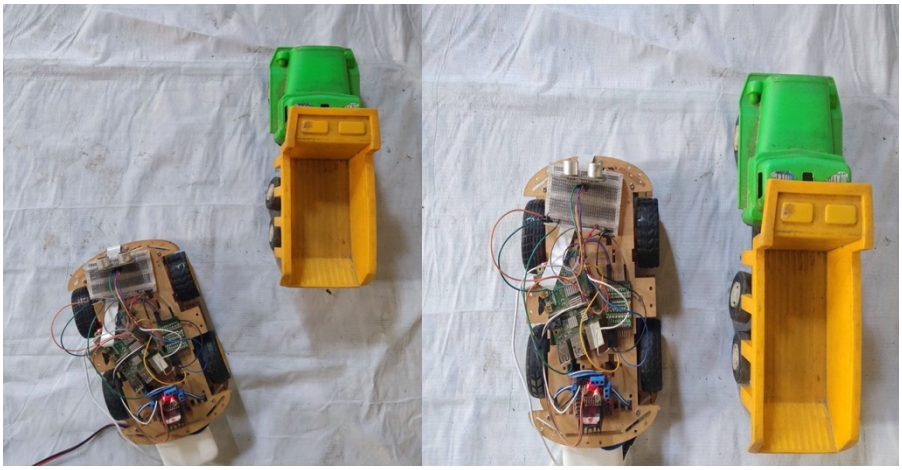

Figure 18. Reducing speed and overtaking condition

The Raspberry Pi continuously monitors the surroundings to detect any potential obstacles. If an obstacle is detected within the safety threshold (20 cm), the system gradually reduces the speed to prevent collision. The PID controller adjusts the speed smoothly rather than applying abrupt braking, that ensuring a

controlled deceleration. If the obstacle remains stationary or too close, the system overtakes the previous vehicle or it may completely stop the motor until it is safe to proceed. The Raspberry Pi sends Pulse Width Modulation (PWM) signals to motor driver, which regulates the DC motor speed accordingly.

F. Conclusion

This study focuses on a vision-based system for obstacle detection and motor speed control in autonomous driving system. The MobileNet SSD provides a lightweight, faster alternative for detecting smaller or less complex objects, allowing efficient processing on the Raspberry Pi. The detected objects are classified based on their distance and size, influencing the vehicle navigation and speed control decisions. The motor speed control mechanism ensures smooth operation by dynamically responding to obstacles in the environment. The system is tested in real-world autonomous navigation scenarios that demonstrates real-time obstacle detection, accurate speed control, and stable performance. For future improvements, integrating additional sensors such as LiDAR or ultrasonic sensors enhances detection accuracy, especially in challenging conditions like low light or occlusions. Further optimization of computational performance allows better real-time processing for embedded applications. This research contributes to the ongoing development of intelligent and adaptive autonomous driving systems.

G. Acknowledgment

The author is deeply gratitude to Dr. San Yu Khaing, Rector of Mandalay Technological University for her help, encouragement, support and guidance. The author wishes to extend grateful thanks to Dr. Tin Tin Hla, Professor and Head, Electronic Engineering Department, Mandalay Technological University for her kind help and invaluable suggestions. The author would like to express thanks to her supervisor Dr. Zin Mar Lwin, Professor and Head, Remote Sensing Department, Mandalay Technological University for her very detailed checks, grateful encouragement, continuous patience and true-line guidance. The author specially thanks to all her teachers from Department of Electronic Engineering, Mandalay Technological University, for the development of this paper. Especially, the author would like to thanks her family for their help and encouragement and also thanks all her friends.

H. References

[1] J. Chen, D. Li, W. Qu, and Z. Wang, “A MSA-YOLO Obstacle Detection Algorithm for Rail Transit in Foggy Weather,” Appl. Sci., vol. 14, no. 16, p. 7322, Aug. 2024, doi: 10.3390/app14167322.

[2] M. Wenning, T. Adlon, and P. Burggräf, “Anomaly detection as vision-based obstacle detection for vehicle automation in industrial environment,” Front. Manuf. Technol., vol. 2, p. 918343, Aug. 2022, doi: 10.3389/fmtec.2022.918343.

[3] S. Agrawal, V. Kumar, and A. K. Dey, “SELF DRIVING CAR USING RASPBERRY PI,” vol. 12, no. 4, 2024, International Journal of Creative Research Thoughts.

[4] R. Fang and C. Cai, "Computer vision based obstacle detection and target tracking for autonomous vehicles," MATEC Web Conf., vol. 336, p. 07004, 2021, doi: 10.1051/matecconf/202133607004.
[5] H.-C. Chen, "Monocular Vision-Based Obstacle Detection and Avoidance for a Multicopter," IEEE Access, vol. 7, pp. 167869–167883, 2019, doi: 10.1109/ACCESS.2019.2953954.
[6] Y. Sarada Devi, S. Sathvik, P. Ananya, P. Tharuni, and N. Naga Krishna Vamsi, "Vision-Based Obstacle Detection and Collision Prevention in Self-Driving Cars," J. Phys. Conf. Ser., vol. 2335, no. 1, p. 012019, Sep. 2022, doi: 10.1088/1742-6596/2335/1/012019.
[7] M. Lalak and D. Wierzbicki, "Automated Detection of Atypical Aviation Obstacles from UAV Images Using a YOLO Algorithm," Sensors, vol. 22, no. 17, p. 6611, Sep. 2022, doi: 10.3390/s22176611.
[8] Z. Qiu et al., "Vision-Based Moving Obstacle Detection and Tracking in Paddy Field Using Improved Yolov3 and Deep SORT," Sensors, vol. 20, no. 15, p. 4082, Jul. 2020, doi: 10.3390/s20154082.
[9] K. Ullah, J. Guzinski, and A. F. Mirza, "Critical Review on Robust Speed Control Techniques for Permanent Magnet Synchronous Motor (PMSM) Speed Regulation," Energies, vol. 15, no. 3, p. 1235, Feb. 2022, doi: 10.3390/en15031235.
[10] F. Shiravani, P. Alkorta, J. A. Cortajarena, and O. Barambones, "An Enhanced Sliding Mode Speed Control for Induction Motor Drives," Actuators, vol. 11, no. 1, p. 18, Jan. 2022, doi: 10.3390/act11010018.
[11] https://cocodataset.org/
[12] https://danghoangnhan.github.io/MobileNetArchitecture/.
[13] https://developers.arcgis.com/python/guide/how-ssd-works/.