Automotive Science and Engineering, Vol. 16, No. 2, (2026), 5094-5106

Iran University of Science and Technology

Automotive Science and Engineering

Journal Homepage: ase.iust.ac.ir

ISSN: 2008-9899

Design and Implementation of an Intelligent Path-Finding Robot with Line-Following and Obstacle-Avoidance Capabilities

Rahmatulah Karimipour[1], Mansour Hakimelahi[2*], Masoud Masih-Tehrani[3]

[1]M.Sc. Student, Department of Mechatronics Engineering, Ahlul Bayt (AS) International University, Tehran, Iran

[2]Faculty Member, Department of Mechatronics Engineering, Ahlul Bayt (AS) International University, Tehran, Iran

[3]Faculty Member, School of Automotive Engineering, Iran University of Science and Technology, Tehran, Iran

ARTICLE INFO

Article history:

Received : 4 Aug 2025

Accepted: 3 May 2026

Published: 28 Jun 2026

Line-Following Robot; Obstacle Avoidance; PID Control; Arduino Uno; Infrared and Ultrasonic Sensors

ABSTRACT

In this study, an intelligent path-finding robot that simultaneously follows a predefined path and avoids obstacles is designed, modelled, and implemented on a low-cost embedded platform. The primary objective is to improve the navigation performance of small-scale mobile robots by fusing infrared (IR) and ultrasonic measurements and by tightly coordinating the control algorithm with the hardware architecture. An Arduino Uno R3 serves as the central processing unit; an L298N driver regulates the speed and direction of two geared DC motors; and a servo-actuated steering mechanism, a three-module IR array, and two HC-SR04 ultrasonic sensors complete the perception and actuation chain. A PID path-following controller is formulated on the differential-drive kinematic model, discretised at the firmware rate of 50 Hz, and verified in the Proteus environment prior to prototype construction. In simulation, the controller recovers a 60 mm initial offset on the straight path within 1.24 s, tracks an S-shaped curve with a root-mean-square error of 4.8 mm, and executes the complete obstacle bypass and path-recovery manoeuvre. Experimental evaluation on straight, curved, and obstacle-laden tracks, with 20 trials per scenario, produced success rates of 100%, 95%, and 90%, respectively, while the deviation between simulated and measured behaviour remained below 5%. The proposed path-recovery strategy re-identified and rejoined the main line in every successful obstacle trial without operator intervention. The results confirm that tightly integrated, low-cost sensing and control can deliver reliable autonomous navigation for educational, internal-transport, and small-scale automation applications.

[DOI: 10.22068/ase.2026.727]

1. Introduction

In recent decades, the rapid progress of electronics, control engineering, and artificial intelligence has opened new horizons in the science of robotics [1]. Among the branches of this discipline, mobile robots have secured a prominent position in manufacturing, transportation, medical services, and education owing to their capacity for autonomous operation and productivity enhancement [2]. Within intelligent vehicle research, accurate path tracking and robust motion control remain active topics, as evidenced by recent coordinated control strategies for autonomous vehicle path tracking [3] and

*Corresponding Author
Email Address: hakim@abu.ac.ir
https://doi.org/10.22068/ase.2026.727

optimal control allocation for the lateral motion control of in-wheel motor vehicles [4], while learning-based robust control frameworks continue to improve vehicle stability under uncertain conditions [5].

One of the most practical classes of mobile robots is the line-following robot, which tracks a predefined path and reacts to the obstacles encountered along its route [6]. Integrating line tracking with obstacle avoidance on a single platform markedly improves navigation reliability in shared and semi-structured environments [7]. Among the available sensing options, the combination of IR arrays for path detection and ultrasonic transducers for obstacle ranging is regarded as one of the most effective low-cost solutions [8]; in parallel, machine-learning-based perception continues to expand the capabilities of automotive and robotic systems [9], [10].

The primary objective of this research is to design and implement an intelligent robot capable of autonomously following a designated path while simultaneously avoiding obstacles. To this end, an Arduino Uno R3 microcontroller is employed as the central processing unit, an L298N motor driver controls the speed and direction of the geared motors, and a combination of IR and ultrasonic sensors captures environmental data. The system is analysed in the Proteus simulation environment, implemented in the Arduino IDE, and validated on a physical prototype.

The significance of this study stems from the observation that most existing low-cost robots perform only one of the two functions, either path following or obstacle avoidance [11]. Integrating both capabilities on a single platform improves precision, response time, and operational stability [12], [13], and is directly relevant to warehouse automation, internal transport, and flexible production lines [14], where energy-aware and degradation-conscious operation is also attracting growing attention [15]. This research further demonstrates that a logically designed control algorithm, executed on inexpensive components, can deliver stable and reliable navigation suitable for real-world applications [16].

The main contributions of this work are as follows. First, a complete low-cost mechatronic design, covering the sensing chain, actuation, power system, and control software, is documented with full technical specifications to support reproducibility. Second, a PID path-following controller is formulated on the kinematic model of the robot, and its closed-loop performance is quantified through Proteus-based and model-based simulation before implementation. Third, a dedicated path-recovery strategy is developed to re-identify and rejoin the main line after every obstacle-avoidance manoeuvre, a capability that is usually described only briefly in comparable studies. Finally, the integrated system is evaluated experimentally on straight, curved, and obstacle-laden tracks, and its performance is compared with similar platforms reported in the literature. The remainder of the paper is organised as follows: Section 2 presents the research method and the system design; Section 3 reports and analyses the experimental results; Section 4 discusses the findings and the limitations of the study; and Section 5 concludes the paper.

2. Research Method and System Design

2.1. Design Methodology and System Architecture

This research is of an applied developmental nature, with the primary objective of designing and implementing a practical prototype of an intelligent line-following robot equipped with obstacle-avoidance capabilities. The design process consists of three main stages: simulation, prototype construction, and performance evaluation under laboratory conditions. In the first stage, the conceptual design of the system was carried out in the Proteus 8 environment to verify the integrity of communication among the hardware components and to evaluate the logic of the control algorithm. The control algorithm was then implemented in the Arduino IDE environment and uploaded to the Arduino Uno R3 board. In the final stage, a physical prototype of the robot was constructed and tested on straight, curved, and obstacle-laden paths.

The designed system consists of three cooperating parts: the hardware, the software, and the control algorithm. The hardware comprises the sensor set, the motor driver, and the Arduino Uno R3 microcontroller; the software processes the data received from the sensors and generates the control signals for the motors and the steering servo; and the control unit performs real-time decision making based on the fused sensor data. Figure 1 shows the overall view of the assembled platform, and Figure 2 presents the corresponding control block diagram. As illustrated in Figure 2, the closed loop is formed by the IR array and the two ultrasonic sensors on the feedback side, by the PID-based decision logic executed on the Arduino Uno R3, and by the L298N motor driver and the steering servo on the actuation side, while

[DOI: 10.22068/ase.2026.727]

the battery feeds the power and logic rails through dedicated regulators.

Figure 1. Overall view of the developed robotic platform.

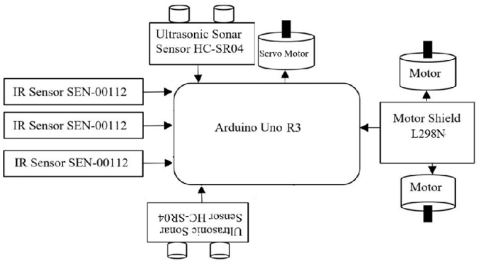

Figure 2. Control block diagram of the robot: the IR array and the two ultrasonic sensors provide the feedback, the Arduino Uno R3 executes the decision logic, and the L298N driver and the steering servo realise the corrective actions.

For the design and implementation of the line-following robot, the following components were utilised: an Arduino Uno R3 board, an L298N motor driver, geared DC motors, a servo motor, IR sensor modules, ultrasonic sensors, and a rechargeable lithium-ion battery pack. The main technical specifications of the resulting platform, including the dimensions, mass, power consumption, and the operating ranges of all sensors, are summarised in Table 1.

2.2. Hardware Design

In the hardware section, the Arduino Uno R3 board is used as the central processing unit. This microcontroller receives data from the IR and ultrasonic sensors and transmits PWM (pulse-width modulation) signals to the motors. To control the speed and rotational direction of the motors, an L298N motor driver is employed, which supports high-current loads and provides bidirectional control for two independent motors. The wiring and physical layout of the electronic components are shown in Figure 3.

2.2.1. Arduino Uno R3 Board

In the design of this robot, the ATmega328P microcontroller, embedded in the Arduino Uno R3 board, is used as the primary programming and processing unit. This microcontroller executes the robot's assigned tasks according to the uploaded algorithm. The ATmega328P is based on the Harvard architecture, which separates program memory from data memory and thereby optimises the overall performance; instructions stored in program memory are executed sequentially [17]. The board operates at 16 MHz and provides 32 KB of flash memory, 2 KB of SRAM, 1 KB of EEPROM, 14 digital input/output pins, six of which support PWM, and six analogue inputs [18]. Programming the ATmega328P normally requires an external circuit, such as an optional oscillator, together with a development environment running on a computer; the Arduino Uno R3 board already integrates all the necessary circuitry, so the user only needs to know the Arduino pinout configuration to upload the program directly onto the board [18].

2.2.2. L298N Motor Driver

The L298N motor driver consists of two transistor H-bridges, allowing bidirectional control of two DC motors with a continuous current of up to 1 A per channel and a peak current of up to 2 A [8]. The module operates within a supply range of 5 to 35 V, integrates a 5 V regulator that biases its logic side, and regulates the motor speed through the PWM signals generated by the Arduino, while its direction pins select the sense of rotation of each motor. In this project, the two geared DC motors of the drive axle are connected to the two output channels of the L298N module.

2.2.3. DC Geared Motor

Geared DC motors are among the most widely used actuators in line-following robots, since the combination of a DC motor with a compact gearbox increases torque and reduces output speed. This configuration yields smoother, controllable, and stable motion, particularly when the robot negotiates curves or narrow paths. In line-following robots, geared DC motors not only provide the necessary driving power but also enhance the precision of speed and directional control. Owing to their low current consumption, compact size, and cost-effectiveness, they are an ideal choice for both educational applications and small-scale industrial projects. Each motor of the present platform is rated for 6 V operation, provides a gear ratio of 1:48, and delivers a no-load output speed of approximately 200 rpm, a

[DOI: 10.22068/ase.2026.727]

combination that balances tractive effort and velocity for the 1.2 kg robot.

2.2.4. Servo Motor

Servo motors are electromechanical actuators that provide precise rotational motion for controlling the angular position of machine components, and they are widely used for accurate control of speed, acceleration, and angular position in robotic steering systems [19]. In this project, a standard SG90 micro servo, operating at 4.8-6 V with a torque of approximately 1.8 kg·cm and a transit speed of about 0.1 s per 60°, controls the front wheels of the robot. The servo receives its angular commands from the Arduino, which maps the PID correction to steering angles between 60° and 120°, thereby ensuring accurate steering and path correction.

2.2.5. Infrared Sensor

In this project, three TCRT5000-based reflective IR modules are used for line detection. Each module operates at 5 V, draws approximately 20 mA, and offers an adjustable detection range of 1-25 mm; it outputs both an analogue reflectance level and a digital threshold signal [20]. The sensor operates by emitting infrared light from its own LED and measuring the intensity of the reflected radiation, which allows bright and dark regions, including the guideline itself, to be identified. The black line is detected by the array of three modules mounted beneath the chassis between the wheels, with a lateral spacing of 25 mm between adjacent modules, which represents the minimum configuration required for effective line following. The working principle relies on the fact that white surfaces reflect most of the incident infrared radiation whereas black surfaces absorb it; the receivers convert the reflected intensity into a position estimate that is transmitted to the Arduino board for processing [8].

2.2.6. Ultrasonic Sensor

Obstacle ranging is performed by two HC-SR04 ultrasonic transducers. Each sensor houses a transmitter and a receiver within the same unit, operates at 5 V with a nominal current of 15 mA, emits bursts at a frequency of 40 kHz, and measures distances between 2 cm and 400 cm with a resolution of approximately 3 mm and an effective beam angle of about 15°. The transmitter emits a series of ultrasonic pulses that, upon striking an object, are reflected back toward the source; by measuring the time interval between transmission and reception, the distance between the sensor and the object is obtained from the time-of-flight relation [8]. Two sensors are installed at the front of the chassis, one on the left and one on the right, so that the robot can determine on which side an obstacle lies; this side discrimination is essential for selecting the direction of the bypass manoeuvre.

2.2.7. Power Supply

The rechargeable battery used in this project belongs to the class of lithium-ion batteries, which are widely employed in portable devices such as electric vehicles, computers, electronic tools, and mobile phones, and whose electrical behaviour under load and vibration remains an active research topic [21]. In this system, an 11.1 V 3S lithium-ion pack composed of three 18650 cells with a nominal capacity of 2200 mAh provides the required power for the robot's operation. The battery feeds the L298N motor input directly, while integrated regulators supply 5 V to the Arduino, the sensors, and the servo. Under nominal operating conditions the platform draws approximately 15 W, which corresponds to an autonomy of roughly 1.5 h of continuous operation; these figures, together with the dimensions and mass of the robot, are critical for assessing the suitability of the platform for real-world deployment [15]. Table 1 summarises the main specifications of the developed robot.

Table 1. Main specifications of the developed robot.

Item	Specification
Overall dimensions	25 cm × 15 cm × 10 cm (L × W × H)
Mass (including battery)	1.2 kg
Wheelbase / track width	18 cm / 14 cm
Microcontroller	Arduino Uno R3 (ATmega328P, 16 MHz, 32 KB flash)
Motor driver	L298N dual H-bridge, 5-35 V, 1 A cont. / 2 A peak per channel
Drive motors	2 × 6 V geared DC motors, 1:48 gear ratio, ~200 rpm no-load
Steering actuator	SG90 micro servo, 4.8-6 V, ~1.8 kg·cm, 0.1 s per 60°
Line sensors	3 × TCRT5000 IR modules, 5 V, 1-25 mm range, 25 mm spacing
Ranging sensors	2 × HC-SR04 ultrasonic, 5 V, 40 kHz, 2-400 cm, ~15° beam
Battery	11.1 V 3S Li-ion, 3 × 18650 cells, 2200 mAh
Power consumption	~15 W (nominal)

[DOI: 10.22068/ase.2026.727]

Item	Specification
Nominal forward speed	0.25 m/s
Control loop rate	50 Hz

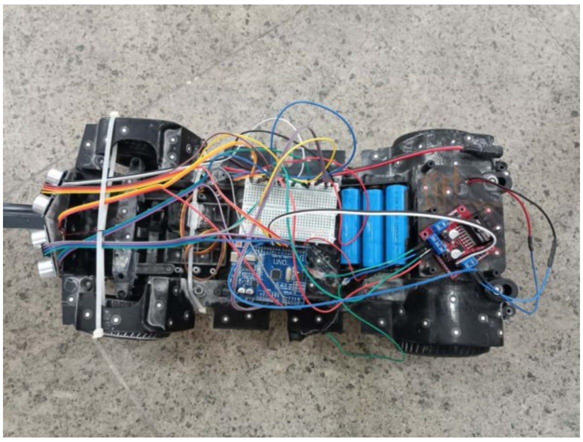

Figure 3. Connection of the hardware components, including the IR array, the ultrasonic sensors, and the Arduino board.

2.3. Software Design and Control Algorithm

The robot was programmed in the Arduino IDE environment. The overall software logic is structured around three main operational modes. In the line-following mode, the robot analyses the outputs of the three IR modules and navigates according to the detected line position. In the obstacle-avoidance mode, the robot continuously monitors the distance data of the two ultrasonic sensors and, when an obstacle is detected within the predefined threshold, stops and changes direction to avoid a collision. In the path-return mode, the robot re-identifies the main path after bypassing the obstacle and resumes its forward motion.

2.3.1. Kinematic Model and PID Control Law

The path-following controller is designed on the kinematic model of the robot. With the servo steering the front axle and the two geared motors driving the rear axle, the platform is accurately described by the single-track (bicycle) model:

$$\dot{x} = v\cos\theta,\ \ \dot{y} = v\sin\theta \tag{1}$$

$$\dot{\theta} = \frac{v}{L}\tan\delta \tag{2}$$

where (x, y) is the position of the robot in the path frame, θ is its heading angle, v is the forward speed, L is the wheelbase (0.18 m), and δ is the steering angle of the front axle. The state vector of the path-following problem is therefore $[x, y, \theta]^T$, and the control objective is expressed through the lateral error:

$$e(t) = y_{ref}(t) - y(t) \tag{3}$$

where y_{ref} denotes the lateral coordinate of the path centreline; the reference value of the error is zero, which corresponds to perfect centring of the sensor array over the line. PID control remains the most common approach for low-cost line-following platforms owing to its simplicity and low computational footprint [22]. In this algorithm, the outputs of the sensors are received as feedback inputs and compared with the reference value, and the resulting error signal is processed by the PID controller, which generates the steering correction:

$$u(t) = K_p e(t) + K_i \int_0^t e(\tau)d\tau + K_d \dot{e}(t) \tag{4}$$

where u is the steering correction and K_p, K_i, and K_d are the proportional, integral, and derivative gains. The command is saturated at the servo limits of ±30° around the central position of 90°, and a differential wheel-speed correction Δ proportional to u is applied to the two driven wheels to support the steering action:

$$v_R(t) = v_0 - \Delta(t),\ \ v_L(t) = v_0 + \Delta(t) \tag{5}$$

where v_0 is the nominal common-mode speed of 0.25 m/s. The gains were tuned on the prototype through a manual procedure: starting from a proportional-only controller, the gain was increased until a sustained oscillation was observed (an ultimate gain of approximately 3°/mm); the gain was then halved, as suggested by the Ziegler-Nichols guidance, the derivative action was added to damp the response, and a small integral action was introduced to remove the residual steady-state offset on long straights. The final gains were selected as K_p = 1.5°/mm, K_i = 0.1°/(mm·s), and K_d = 0.8°·s/mm. To reject the sensor noise in the derivative term, the error signal is filtered by a first-order exponential moving average, and an anti-windup mechanism freezes the integrator whenever the steering command saturates. Fuzzy and sliding-mode extensions of classical PID schemes have been shown to further improve robustness in vehicle applications [23], [24]; the present platform retains a linear PID structure to preserve the simplicity and the low computational cost required by the 8-bit microcontroller.

2.3.2. Decision-Making Logic

The overall decision-making process of the robot is summarised in the flowchart of Figure 4. The main loop, executed at 50 Hz, first reads the

[DOI: 10.22068/ase.2026.727]

two ultrasonic channels: if the measured distance on either side falls below the 30 cm threshold, the robot enters obstacle-avoidance mode, determines the free side from the pair of readings, and executes the bypass manoeuvre. Otherwise, the IR array is read, the lateral error is computed, and the PID law of (4) produces the steering and differential-speed commands: if the line is centred, the robot moves straight ahead; if the line is to the left, it turns right; and if the line is to the right, it turns left. The loop repeats until the end point of the path is reached.

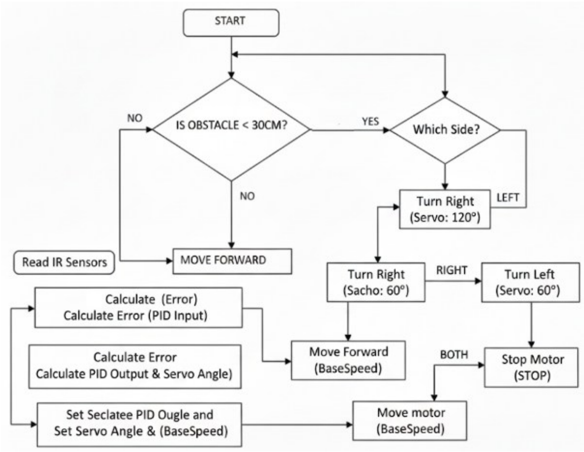

Figure 4. Flowchart of the robot operational logic, including obstacle detection, line-following, and path recovery.

To make the implemented behaviour reproducible, Algorithm 1 presents the pseudocode of the complete control logic.

Algorithm 1. Line following with obstacle avoidance and path recovery.

```
initialise Kp, Ki, Kd, v0, threshold dth = 30 cm
loop at 50 Hz until the end point is reached
   dL ← distance(ultrasonic, left);  dR ← distance(ultrasonic, right)
   if dL < dth or dR < dth then          // obstacle detected
      stop;  side ← free side (larger reading)
      steer to side;  advance until both readings are clear
      execute PATH RECOVERY (Section 2.3.3)
   else
      read IR array;  e ← lateral error estimate [mm]
      u ← Kp·e + Ki·∫e + Kd·ė                  // PID law, (4)
      δ ← clamp(90° + u, 60°, 120°)            // servo command
      vR ← v0 − kΔ·u;  vL ← v0 + kΔ·u          // differential speeds, (5)
      command servo δ;  command motors (vR, vL)
   end if
end loop
```

2.3.3. Path-Recovery Strategy

The automatic return to the main path after an avoidance manoeuvre is one of the key innovations of the developed system and is described in detail in this section. When the bypass phase ends, that is, when both ultrasonic channels report clear distances for a predefined number of consecutive samples, the robot does not steer back blindly. Instead, it decelerates to 40% of the nominal speed and initiates a controlled re-identification procedure: the steering servo sweeps from 60° to 120° in discrete steps of 10° while the controller monitors the IR array. As soon as an outer module of the array detects the line, the sweep stops, the side of the detection is recorded, and the robot steers toward the detected side along a bounded-curvature arc.

The line-tracking loop is re-activated at the moment of the first detection, so the PID controller smooths the rejoining trajectory and suppresses the overshoot; this behaviour is visible in the simulated trajectory of Figure 6(c) and was replicated in the experiments. If no module detects the line during a complete sweep, the robot uses the sign of the last known error, stored when the line was last seen, to continue a wider search arc, limited to five attempts, before halting and signalling a path-loss fault. This memory-based fallback distinguishes the present implementation from the simple stop-and-restart approaches reported for comparable low-cost platforms [6], and it ensured that, in all successful obstacle trials, the robot rejoined the line without operator intervention.

2.3.4. Simulation in the Proteus Environment

Before constructing the physical prototype, the complete circuit was simulated in the Proteus 8 environment. Figure 5 shows the simulated schematic, in which the Arduino Uno R3, the L298N driver, the three IR modules, the two HC-SR04 sensors, the steering servo, and the drive motor are interconnected exactly as in the realised hardware. During this simulation, the communication between the sensors, the motor driver, and the Arduino was examined, the PWM outputs were verified for every error condition, the ultrasonic timing logic was validated, and the correct functionality of the implemented control algorithm was confirmed.

[DOI: 10.22068/ase.2026.727]

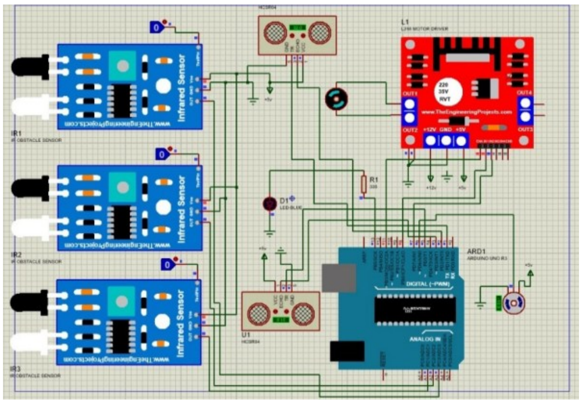

Figure 5. Circuit performance simulation in the Proteus environment.

The closed-loop performance of the path-following controller was additionally evaluated through a model-based simulation of (1)-(5), executed at the discretisation rate of the real firmware (50 Hz) and including the measurement noise of the IR array and the rate limit of the servo. Figure 6 summarises the results for the three reference scenarios. On the straight path (Figure 6(a)), the controller recovers from an initial lateral offset of 60 mm within a settling time of 1.24 s, with a residual error below 2 mm; on the S-shaped curved path (Figure 6(b)), the root-mean-square tracking error is 4.8 mm; and in the obstacle scenario (Figure 6(c)), the robot completes the full detect-bypass-recover sequence and returns to the line with a final error below 2 mm. The corresponding error-convergence histories are shown in Figure 6(d). These simulation results provided the quantitative baseline against which the experimental behaviour of the prototype was assessed in Section 3.

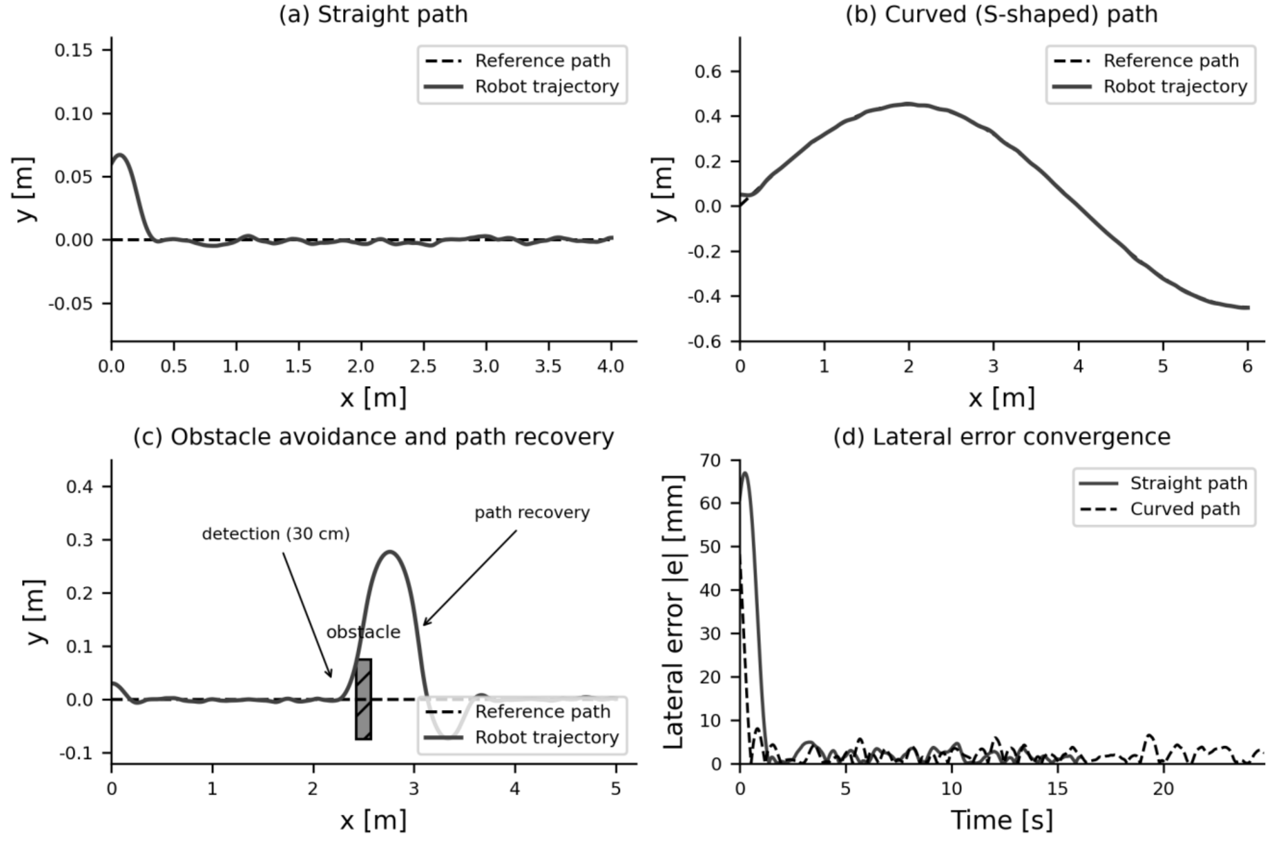

Figure 6. Simulation results of the PID path-following controller: reference and simulated trajectories on (a) the straight path, (b) the curved path, and (c) the obstacle scenario, and (d) the corresponding lateral-error convergence.

2.3.5. Construction of the Physical Prototype

After the functionality of the system was verified in the simulation environment, the physical prototype of the robot was constructed, as shown in Figure 7. The robot's body was made of lightweight yet durable plastic to reduce the overall weight while maintaining movement stability. The motors, sensors, and control circuits were mounted on the chassis, and the test track was prepared as a 30 mm-wide black line on a white surface for path detection.

[DOI: 10.22068/ase.2026.727]

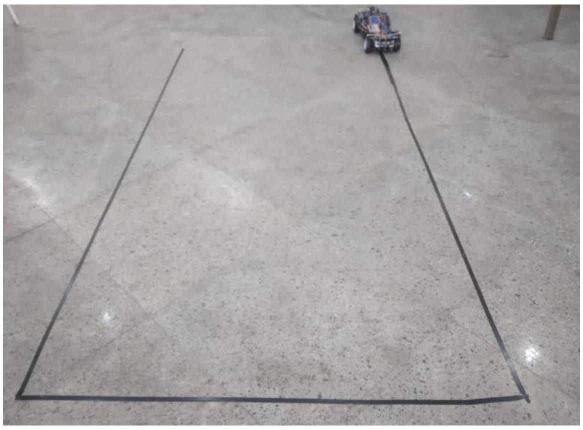

Figure 7. Physical prototype of the robot developed for experimental testing.

3. Results and Analysis

3.1. Experimental Setup and Conditions

After the design and implementation of the hardware and software components, the intelligent line-following robot was tested in a real-world environment to evaluate its performance under various conditions. The experiments were conducted on three types of track: a straight path, a curved path, and a path containing obstacles. The main objective of these tests was to assess the accuracy of the robot in line tracking and its response speed in obstacle detection. Each scenario was repeated 20 times, and every trial was performed under identical, documented conditions: an indoor laboratory illuminated by fluorescent lamps, providing an ambient illumination of 400-600 lux at the track surface, a matte white PVC running surface, a constant ambient temperature of approximately 24 °C, and the nominal forward speed of 0.25 m/s. A trial was counted as successful when the robot completed the track without operator intervention, without leaving the line for more than 2 s, and without colliding with any obstacle. The success rate, the standard deviation computed over four blocks of five consecutive trials, and the test conditions are reported in Table 2.

Table 2. Experimental results of the robot under the documented laboratory conditions (20 trials per scenario; four blocks of five consecutive trials).

Test No.	Path type	Trials	Successful runs	Success rate (%)	SD (%)
1	Straight path	20	20	100.0	0.0
2	Obstacle path	20	19	95.0	5.8
3	Curved path	20	18	90.0	11.5

3.2. Experimental Results on the Straight Path

In the first experiment, a straight track with a black line on a white background was designed. The robot successfully followed the path in all 20 trials without any deviation, and no errors were observed in line detection. The success rate on the straight path was therefore 100%, fully consistent with the algorithm's design and programming. This result demonstrates the complete coordination between the infrared sensors and the control system and confirms the reliability and precision of the implemented hardware-software integration. The measured behaviour also matched the simulated response of Figure 6(a), with a deviation of only 2.1% between the simulated and the measured lateral-error profiles.

3.3. Experimental Results on the Curved Path

In the second experiment, a curved path with successive bends was designed to evaluate the robot's ability to navigate turns and maintain its trajectory. The robot demonstrated satisfactory performance under these conditions, maintaining accurate line tracking throughout most of the course; however, a slight reduction in speed and control authority was observed while negotiating the sharpest turns, and the path detection accuracy in this test was approximately 90% (18 successful runs out of 20, with a standard deviation of 11.5%), indicating stable and reliable behaviour even under dynamic path conditions. The analysis of the two failed trials attributed the deviations to three coexisting sources. At the sensor level, the discrete 25 mm spacing of the IR modules quantises the measurable error, so on sharp curves the line can temporarily leave the array footprint before a correction is issued. At the mechanical level, the finite transit time of the steering servo (about 0.1 s per 60°) introduces a phase lag that becomes significant at high curvature, and the resulting overshoot occasionally caused the robot to cross the line. At the environmental level, minor reflections from the locally glossy areas of the surface reduced the contrast margin of the outer modules. These observations are consistent with the largest simulation-experiment deviation, measured at 4.6% in this scenario (Figure 8(b)).

3.4. Experimental Results on the Obstacle Path

In the third experiment, obstacles were placed along the path to evaluate the robot's ability to detect and avoid objects using the ultrasonic sensors. The robot successfully detected obstacles located within the 30 cm detection range in all trials, selected the free side correctly, and adjusted its trajectory to prevent collisions. The path-recovery strategy of Section 2.3.3 was executed in every successful trial, and the robot rejoined the main line without operator intervention. In a few specific cases, a slight delay in the robot's response was observed, which was attributed to the processing time of the ultrasonic sensor data and to the echo-cycle scheduling of the two ranging channels. Overall, the operational accuracy in this scenario was 95% (19 successful runs out of 20, with a standard deviation of 5.8%), indicating a highly reliable obstacle-detection and avoidance capability.

3.5. Overall Performance and Comparison with Similar Studies

The analysis of the results demonstrates that the integration of the infrared array and the two ultrasonic sensors significantly enhances the accuracy, stability, and reliability of the system under various operating conditions. Compared with conventional line-following robots that rely solely on IR sensors, the proposed design effectively minimised the errors caused by light reflection and by the blind zone in front of the robot. Figure 8(a) summarises the success rates of the three scenarios together with their standard deviations, and Figure 8(b) compares the simulated and the experimental outcomes: the deviation remained below 5% in all scenarios (2.1%, 3.4%, and 4.6% on the straight, obstacle, and curved paths, respectively), confirming the predictive value of the simulation model.

Moreover, the simulation results obtained in the Proteus environment showed a strong correlation with those of the real-world experiments. The deviation between the simulated and the experimental data was less than 5%, indicating the high precision and validity of the developed model. In terms of energy efficiency, the use of the L298N motor driver contributed to improved power management and to the prevention of motor overheating during the repeated trials, and the measured autonomy of approximately 1.5 h on a single charge was sufficient for a complete experimental session. Overall, the robot's performance across all test conditions confirmed the robustness of the system design and the accuracy of the implemented control algorithm.

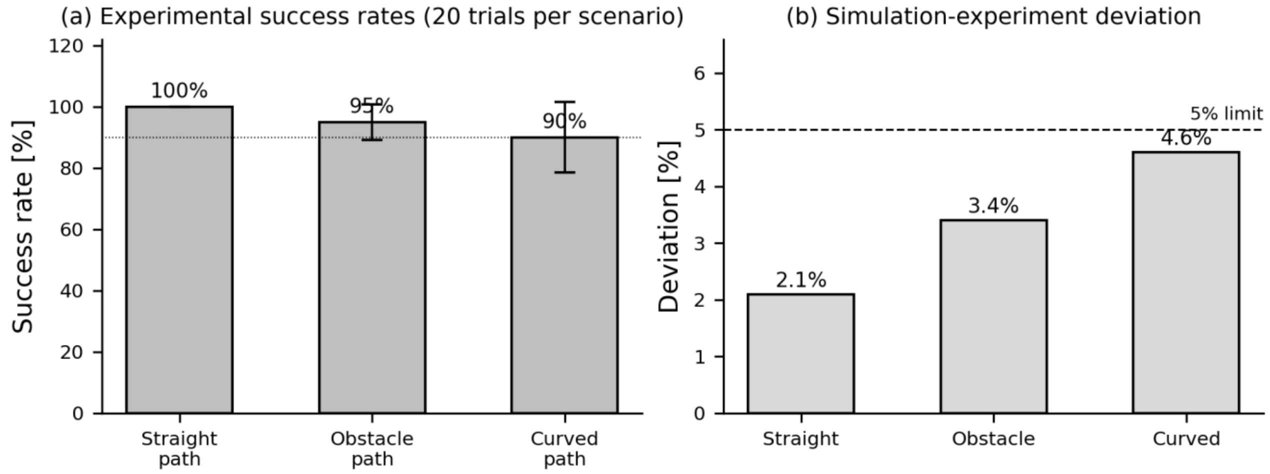

Figure 8. Experimental performance summary: (a) success rates of the three test scenarios (error bars denote one standard deviation over four blocks of five trials) and (b) the deviation between the simulated and the experimental outcomes.

Table 3 compares the proposed platform with similar line-following robots reported in the recent literature. The comparison shows that the proposed robot combines three features that are rarely available together in this cost class: side-discriminating dual ultrasonic ranging, a PID controller formulated on an explicit kinematic model with fully reported gains, and a documented path-recovery strategy, while achieving success rates that are competitive with the cited platforms.

Table 3. Comparison of the proposed robot with similar platforms reported in the literature.

Study	Platform	Line sensing	Ranging	Control	Obstacle avoidance	Path recovery

[DOI: 10.22068/ase.2026.727]

Study	Platform	Line sensing	Ranging	Control	Obstacle avoidance	Path recovery
[8] (2020)	Line-follower robot	IR reflectance array	1 ultrasonic	Curvature-based	Yes	Not reported
[22] (2021)	Differential-drive robot	Reflectance array	None	Modified PID	No	No
[12] (2023)	Nursing-assistant robot	IR line sensing	Ultrasonic	Rule-based	Yes	Not reported
[20] (2024)	Path-following robot	IR modules	None	Threshold logic	No	No
[7] (2025)	AGV for stock distribution	Colour detection	None	Embedded logic	No	Not applicable
This work	Differential drive + servo steering	3 IR modules (25 mm array)	2 ultrasonic (side-discriminating)	PID on kinematic model	Yes	Yes (documented)

Video recordings of the straight-path, curved-path, and obstacle-avoidance experiments are available from the corresponding author upon reasonable request.

4. Discussion

4.1. Sources of Tracking Error

The experimental campaign identified three categories of error sources, whose relative weight explains the difference between the 100% success of the straight path and the 90% success of the curved path. The first category is sensor-level noise and quantisation: the IR array resolves the lateral position with an accuracy of approximately ±12.5 mm at the array level, and the analogue readings carry a noise component of about 2 mm, which the derivative term of the PID law amplifies unless the exponential filter described in Section 2.3.1 is applied. The second category is mechanical: the transit time of the steering servo introduces a delay of about 0.1 s per 60° of steering, which at the nominal speed of 0.25 m/s corresponds to a travelled distance of roughly 25 mm before the commanded correction is fully realised; on sharp curves this lag produces overshoot, and the gearbox backlash adds a small dead band around the centred position. The third category is algorithmic: the 50 Hz loop imposes a 20 ms sampling latency, and the saturated steering command limits the achievable corrective curvature; both effects grow in severity with the local path curvature.

These three mechanisms act jointly: the failed curved-path trials exhibited a repeated pattern in which the line left the sensor footprint during a sharp bend (sensor limitation), the subsequent correction arrived with a measurable delay (mechanical limitation), and the saturated steering command produced an overshoot across the line (algorithmic limitation). Addressing the joint effect therefore requires a combined solution, such as curvature-adaptive speed reduction or a look-ahead error formulation, which is left for future work.

4.2. Agreement between the Simulation and the Experiment

The deviation between the simulated and the experimental outcomes remained below 5% in all scenarios, which validates the model-based design flow adopted in this research: the kinematic model of (1) and (2), combined with the discretised PID law and the measured noise and rate-limit figures, predicted the experimental behaviour of the platform with engineering accuracy. The residual deviation is attributed to the effects that the simulation deliberately neglects, namely the rolling friction and the wheel slip on the PVC surface, the small dead band of the servo, the nonlinearity of the IR reflectance transition near the black-white boundary, and the timing jitter of the ultrasonic echo scheduling. The close agreement between the Proteus verification and the measured behaviour indicates that the simulation stage can be used with confidence as a design-verification tool for future iterations of the platform, reducing development time and hardware iterations.

4.3. Limitations of the Study

Although the results demonstrate satisfactory and stable system performance, several limitations were encountered during the design, construction, and testing phases. From a physical standpoint, the platform measures 25 cm × 15 cm × 10 cm and weighs 1.2 kg including the battery, which is well suited to laboratory and desktop tracks but restricts operation in very tight layouts, where the minimum achievable turning radius becomes the binding constraint. From an energetic standpoint, the nominal power consumption of approximately 15 W and the resulting autonomy of 1.5 h limit long-duration missions; larger battery packs

[DOI: 10.22068/ase.2026.727]

would increase the mass and reduce the payload margin, so a careful trade-off is required for deployment scenarios that demand extended operation.

From a sensing standpoint, the front servo motor exhibited a slight response delay during directional changes, which momentarily slowed the robot's motion during turns and produced minor deviations between the actual and the desired trajectories. The ultrasonic sensors showed reduced measurement precision when detecting obstacles with angled or non-perpendicular surfaces, because the irregular reflection and scattering of the ultrasonic waves weakened the returned echo; soft or sound-absorbing materials and very low obstacles are likewise not reliably detected. Variations in the ambient light intensity and in the ground surface finish also had noticeable effects on the accuracy of the IR array, occasionally causing minor deviations from the intended path. Finally, the control layer operates at a fixed nominal speed and follows a single guideline without junction logic, and the platform has not yet been evaluated outdoors or under dynamic obstacles. Within these boundaries, the system remains well matched to its target applications in education, internal transport, and small-scale automation.

5. Conclusions

In this research, an intelligent line-following robot was designed, simulated, and implemented with the capability of simultaneously tracking a predefined path and avoiding obstacles. The system design employed a three-module IR array for path detection and two HC-SR04 ultrasonic sensors for obstacle recognition, with the Arduino Uno R3 as the central processing unit and the L298N motor driver, together with a servo-based steering mechanism, providing precise control over the speed and direction of the geared DC motors. The PID path-following controller was formulated on the kinematic model of the platform, tuned on the prototype, and validated first in the Proteus environment and then experimentally.

The experimental campaign, comprising 20 trials per scenario under documented laboratory conditions, produced success rates of 100% on the straight path, 95% on the obstacle-laden path, and 90% on the curved path, with standard deviations of 0.0%, 5.8%, and 11.5%, respectively. The deviation between the simulated and the experimental data remained below 5% in all scenarios, confirming the validity and the engineering accuracy of the proposed design flow. The path-recovery strategy completed the detect-bypass-rejoin sequence in every successful obstacle trial without operator intervention, and the analysis of the failed trials attributed the residual errors to identifiable sensor, mechanical, and algorithmic sources, providing a clear direction for further improvement.

These findings confirm that the proposed system possesses high stability and performance, and that the integration of multiple sensors effectively reduces motion and decision-making errors. The documented specifications, the reported control gains, and the pseudocode make the platform a reproducible, low-cost reference for robotics education, internal transportation, and small-scale automation.

Future work will focus on three directions. First, more advanced decision-making algorithms, including machine-vision-based perception, which has proven effective for object detection and scene understanding in automotive applications [10], [25], and learning-based path planning [1], [16], will be investigated to allow operation in dynamic and unstructured environments. Second, sensors with higher accuracy and better environmental adaptability will be incorporated to improve detection reliability and control stability. Third, a camera-based machine-vision module will be evaluated for recognising colours, patterns, and shapes along the path, thereby enhancing the autonomy and the versatility of the platform; a curvature-adaptive speed profile constitutes a further promising refinement, since the error analysis of Section 4.1 indicates that reducing the speed on sharp bends would directly mitigate the dominant failure mechanism.

Declaration of Conflicting Interests

The author(s) declared no potential conflicts of interest with respect to the research, authorship, and/or publication of this article.

AI-Assist Statement

During the preparation of this work, the authors used a large language model (GLM, developed by Z.ai) in order to improve the language and the readability of the manuscript and to assist in cross-checking the bibliographic details. After using this tool, the authors reviewed and edited the content as needed and take full responsibility for the content of the published article.

[DOI: 10.22068/ase.2026.727]

Acknowledgements

The authors received no specific grant for this research from any funding agency in the public, commercial, or not-for-profit sectors. The authors would like to thank Ahlul Bayt (AS) International University and the Iran University of Science and Technology for providing the laboratory facilities used in this study.

References

[1] V.D.C. Le Duc Hanh, Path following and avoiding obstacle for mobile robot under dynamic environments using reinforcement learning, Journal of Robotics and Control (JRC), Vol.4, No.2, (2023).

[2] M.S.N. Manjunath, Path planning and obstacle overtaking robot using controller, International Journal of Engineering Research & Technology (IJERT), Vol.9, No.8, (2020).

[3] M. Dehghan Manshadi, B. Mashadi, Coordinated control of torque vectoring and active rear camber for autonomous vehicle path-tracking under limit conditions, Automotive Science and Engineering, Vol.16, No.2, (2026), pp.5010-5023.

[4] M. Sarani, B. Mashadi, M. Majidi, Lateral motion control of in-wheel motor EVs via optimal energy-efficient control allocation, Automotive Science and Engineering, Vol.16, No.1, (2026), pp.4937-4952.

[5] A. Khosravian, M. Masih-Tehrani, A. Amirkhani, S. Ebrahimi-Nejad, Robust autonomous vehicle control by leveraging multi-stage MPC and quantized CNN in HIL framework, Applied Soft Computing, Vol.162, (2024), 111802.

[6] G. Chandrappa, Design and implementation of a digital twin for a line-follower robot, Master's Thesis, Riga Technical University, Riga, (2025).

[7] L.H. Huynh, D.T. Duong, L. Vincent, Design and implementation of a line-tracking robot for autonomous navigation, EAI Endorsed Transactions on Sustainable Manufacturing and Renewable Energy, Vol.2, No.2, (2025).

[8] A. Bendimrad, B. El Amrani, Design and implementation of line follower and obstacle detection robot, International Journal of Power Electronics and Drive Systems (IJPEDS), Vol.11, No.1, (2020), pp.160-168.

[9] M. Baghaeian, E. Abbasi, Improving defect detection in GDXray castings via inverse problem-based deep learning, Automotive Science and Engineering, Vol.16, No.1, (2026), pp.4922-4936.

[10] A. Khosravian, A. Amirkhani, H. Kashiani, M. Masih-Tehrani, Generalizing state-of-the-art object detectors for autonomous vehicles in unseen environments, Expert Systems with Applications, Vol.183, (2021), 115417.

[11] Z. Li, X. Yin, X. Li, S. Tang, Research into autonomous vehicles following and obstacle avoidance based on deep reinforcement learning method under map constraints, Sensors, Vol.23, No.2, (2023), 844.

[12] M.T.A. Rahim, M. Ashraf, N. Kumar, U. Kumar, A.K. Singh, Line following with obstacle avoidance autonomous nursing assistant robot, International Research Journal of Modernization in Engineering Technology and Science, Vol.5, (2023).

[13] B. Kishore, K. Pavan Kumar, G. Tharun Kumar, K. Mounish, M. Durga Rao, Design and implementation of an embedded model with line following feature and live streaming, Journal of Engineering Sciences, Vol.15, (2024).

[14] T.C. Do, D.T. Dinh, Development and implementation of autonomous mobile robots for warehouse applications, International Journal of Mechanical Engineering and Robotics Research, Vol.14, No.3, (2025), pp.360-366.

[15] M.A. Rahbari, M. Salehpour, M. Bahramkhoo, S. Gohari Rad, A. Alijani, Integrated optimization of ride comfort and battery degradation in pure electric vehicles with active suspension, Automotive Science and Engineering, Vol.16, No.2, (2026), pp.5024-5041.

[16] H. Yang, X. Teng, Mobile robot path planning based on enhanced dynamic window approach and improved A* algorithm, Journal of Robotics, Vol.2022, (2022), 2183229.

[17] B. Hartono, E.A. Putri, Line follower based container robot as a solution to improve service mobility in Universitas Tanjungpura, Journal of Electrical Engineering, Energy, and Information Technology (J3EIT), Vol.11, No.3, (2023), pp.102-113.

[18] D.H.S. Razi, AVR Microcontrollers (in Persian), Kian Academic Publications, Tehran, Iran, (1401 AP).

[19] B. Bae, D.-H. Lee, Design of a four-wheel steering mobile robot platform and adaptive steering control for manual operation, Electronics, Vol.12, No.16, (2023), 3511.

[DOI: 10.22068/ase.2026.727]

[20] O.R. Solavat, Path following robot using Arduino and IR sensors, International Research Journal of Engineering and Technology (IRJET), Vol.11, No.5, (2024).

[21] S. Ghulami, M. Hakimollahi, Experimental investigation of mechanical vibration effects on lithium-ion battery state-of-charge estimation using ensemble machine learning models, Automotive Science and Engineering, Vol.16, No.2, (2026), pp.5000-5009.

[22] S. Oguten, B. Kabas, PID controller optimization for low-cost line follower robots, arXiv preprint, arXiv:2111.04149, (2021).

[23] H. Nazemian, M. Masih-Tehrani, Hybrid fuzzy-PID control development for a truck air suspension system, SAE International Journal of Commercial Vehicles, Vol.13, No.1, (2020), pp.55-69.

[24] A. Emami, M. Masih-Tehrani, A. Amirkhani, A. Najafi, Robust fuzzy sliding mode controller for the full vehicle semi-active suspension system considering actuator dynamics, IET Intelligent Transport Systems, Vol.19, No.1, (2025).

[25] A. Amirkhani, A. Khosravian, M. Masih-Tehrani, H. Kashiani, Robust semantic segmentation with multi-teacher knowledge distillation, IEEE Access, Vol.9, (2021), pp.119049-119066.

[DOI: 10.22068/ase.2026.727]