Line Follower Mobile Robot for Surveillance Camera Monitoring System

Wira Hidayat bin Mohd Saad[1], Saiful Anuar bin Abd. Karim[2], Norliana Binti Azhar[1], Zahariah Manap[3], Yew Yuan Soon[1], Masrullizam Mat Ibrahim[1]

[1]Faculty of Electronic and Computer Engineering, Universiti Teknikal Malaysia Melaka, Durian Tunggal, Melaka, Malaysia

[2]Department of Electrical Engineering, Politeknik Ibrahim Sultan, 81700 Pasir Gudang. Johor, Malaysia

[3]Faculty of Engineering Technology, Universiti Teknikal Malaysia Melaka, Durian Tunggal, Melaka, Malaysia.

wira_yugi@utem.edu.my

***Abstract*— The purpose of this study is to demonstrate the development of a mobile surveillance camera monitoring system. This robotic system is introduced to assist in solving the limited coverage problem facing by the conventional surveillance camera which is usually installed at a fixed position. Line follower is chosen to provide a mobile movement of the surveillance monitoring system. Other than surveillance purpose, the camera also functions to detect any kind of obstacle on the route. A specific route and task are designed for the robotic system to test its functionality. It shows that the robotic system manages to complete the designed track and manage to detect the obstacle**

***Index Terms*— Arduino; Labview; Line Follower Robot; Monitoring System; Webcam.**

I. Introduction

Nowadays, the surveillance camera is being installed everywhere as they are planted in the streets, shops, restaurants, hotels, and sometimes even in the refrigerator. It indicates how important it is for our society to keep us feel safe. Different people tend to have different ways of thinking on the importance of surveillance camera. Statistically speaking, it proves that surveillance systems can control a criminal rate [1]-[3].

Existing surveillance systems are said to have several drawbacks, such as high expenses to replace old cameras to higher resolution cameras, large storage for saving data files, computation power in processing data, and so on [4]. Since the camera only covers its specific point of view, the coverage of the conventional surveillance system is quite limited. Thus, in this project a mobile surveillance camera on the line follower robot is introduced to have increased region of coverage than a single fixed position surveillance camera as line follower features introduced specific route mobility to the surveillance camera monitoring system. On top of that, the function of the camera installed not only can be used for surveillance purposes but also can be used to monitor the route for the obstacle.

As time passed, more version of line-follower robot has been proposed. Most of them introduce changes to the conventional line-follower robot, which enable it to function better with a certain type of improvements such as path planning [6][7][8][9], smoother movement [10], and faster signal processing speed [11]. In this study, the focussed is more on the structure development and basic functionality of each of the components of the system. The conventional line follower technique that is utilizing the light sets of infrared transceivers are used to verify the position of the line and to decide the turning of the wheels using On-Off control system.

. This paper is divided into five sections including this section. The second section describes the development of the entire system for surveillance and line follower. Section three focussed on the demonstration and testing of the system functionality and last section consist of the conclusion and future work.

II. Methodology

The methodology of the project is divided into three stages of work progress. The first stage is the development of a line following robot using Arduino. The second stage is the development of monitoring system using LabVIEW, and lastly combine both the Arduino and LabVIEW into a line following monitoring system. The project flowcharts are shown in Figure 1 and Figure 2 for the first stage and second stage respectively. The combination of both systems is shown in Figure 5.

A. Line follower system

In this stage, the line follower system for the robot is developed. This includes the basic structure and working principle of a line-follower by using Arduino platform as the main controller.

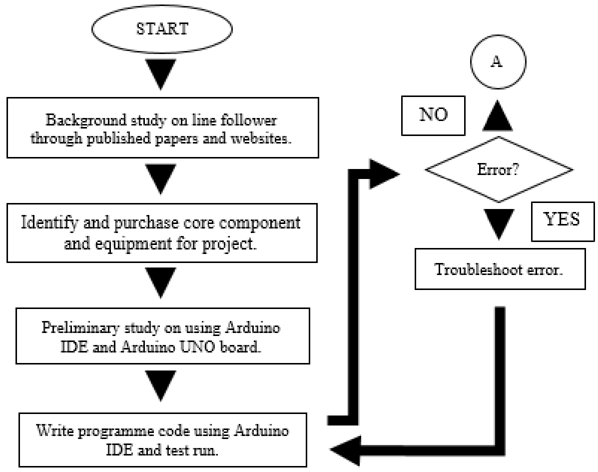


Figure 1: Flowchart of the line following robot development process using Arduino Uno controller

The line follower system in this mobile system is guided by using four pairs of an infrared transceiver that projected to the ground in order to follow the track line. For now it is moved based on the on-off system where the decision of the wheel rotation is solely on the detection of the line by four pairs of infrared sensors based on Table 1.

Table 1
Condition of the sensor detection on the robot direction.

Sensor → Direction	IR1	IR2	IR3	IR4
Left	1	0	0	0
Curve left	1	1	0	0
Turn left	1	1	1	0
Right	0	1	1	1
Curve left	0	0	1	1
Turn left	0	1	1	1
Straight	Others condition			

B. Camera monitoring system using

The construction of a monitoring system is done wirelessly connecting the NI MyRIO device to the PC using Wifi and monitored by using LabView. A webcam is connected to the MyRIO device USB port as a monitoring aid.

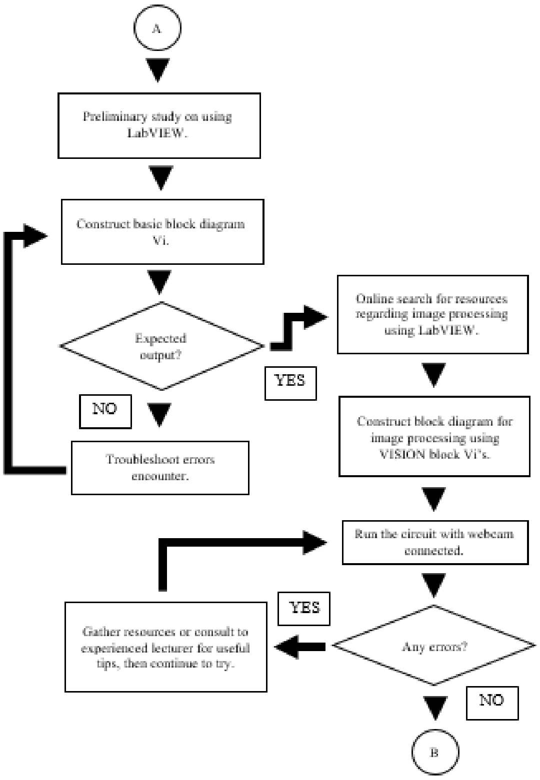


Figure 2: Flowchart in constructing the monitoring system using LabVIEW

In the construction of this monitoring system, some block Vis are used, such as the 'IMAQdx Open Camera VI', 'IMAQdx Configure Grab VI', 'IMAQdx Grab2 VI', and 'IMAQdx Close Camera VI'. Figure 3 shows the block diagram to display an image captured from webcam and viewed on display.

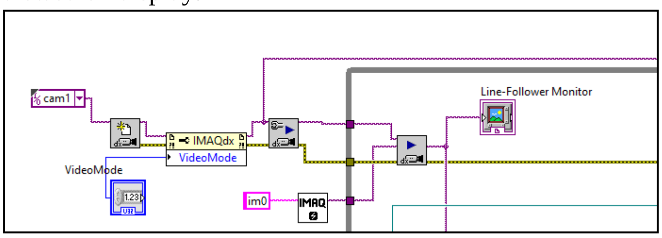
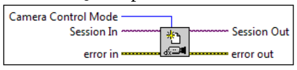
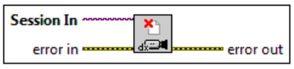
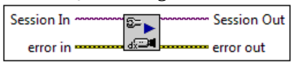
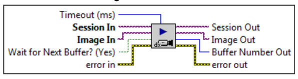


Figure 3: Block diagram to capture and display images from the webcam.

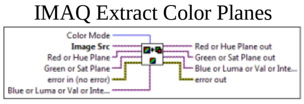
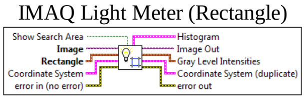
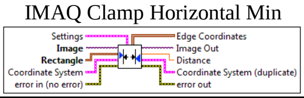
The brief information on the Vision and Motions Block VIs from LabVIEW is shown in Table 2. Every each of the VIs plays an important role to in the monitoring system.

Table 2
The function of basic Vision and Motion Block VIs.

IMAQdx Block VIs	Function
IMAQdx Open Camera VI 	Create a camera as an input and loads configuration file of the camera.
IMAQdx Close Camera VI 	Close the created camera input.
IMAQdx Configure Grab VI 	Configure and starts to obtain images from the input camera.
IMAQdx Grab 2 VI 	Acquires most current frame to image out. Used to display the images from the input camera at the image out.

Some of the image filtering VIs are added into the system circuit to improve the captured image quality. In this project, the additional filter VI of 'IMAQ Extract Color Planes', 'IMAQ Light Meter (Rectangle)', and 'IMAQ Clamp Horizontal Min' are being implemented into the video monitoring block circuit. The function of these filter blocks is stated in Table 3.

Table 3
The function of image filtering block Vis

IMAQ Block VIs	Function
IMAQ Extract Color Planes 	Extract selected planes (RGB, HSL, HSV, or HSI) from an image.
IMAQ Light Meter (Rectangle) 	Measure the pixel intensities within a selected rectangular region.
IMAQ Clamp Horizontal Min 	Detects and measure the horizontal distance of vertical sides.

The connection of the IMAQ Light Meter (Rectangle) VI to the output image of IMAQ Extract Color Planes is as shown in Figure 4. A drawn rectangular frame is required for this light meter VI to work, therefore, a constant is first created at the 'Image out' port of IMAQ Light Meter VI, then the property node of ROI block is enabled to visible the frame drawing on the display monitor at the front panel. By using either global rectangular or contour shapes is allowed in order to draw preference region on display in the front panel. 'Unbundle' function block is used to split the output port of 'ROI' property node into multiple ports. 'Index Array' function block is used to specify x- and y-coordinates for a rectangle shape and bundle up those indexed coordinates to connect to the 'IMAQ Light Meter (Rectangle) VI.

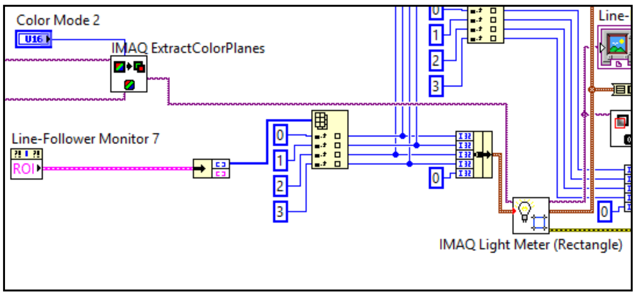


Figure 4: Connection to IMAQ Light Meter (Rectangle).

C. Arduino Plus LabVIEW

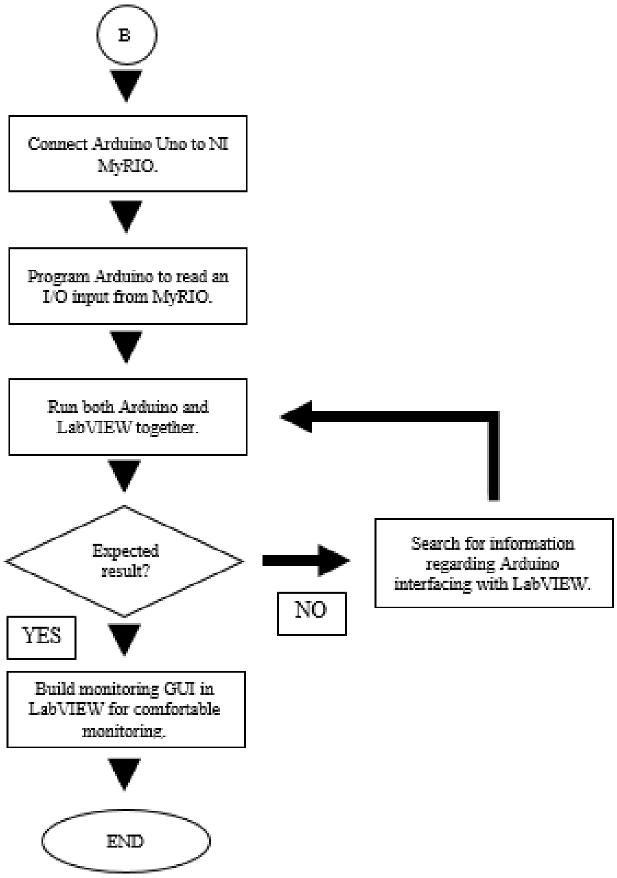


Figure 5: Flowchart of combining both systems which are line follower using Arduino Uno and monitoring system using MyRIO .

The coding of the functional line following robot is configured in order to establish a digital i/o pin communication in between NI MyRIO device and Arduino Uno controller. Using LabVIEW, the same block diagram created for webcam monitoring, a digital i/o pin of the MyRIO device is selected to act as a control switch to either stop or resume the movement of the line following robot.

One of the digital i/o pins of Arduino Uno is set as an input pin and connected to one of the MyRIO output pins to read the digital instruction from the MyRIO device whether to continue moving or not due to the obstacle. The Arduino Uno is set to run the line-follower when the detected digital signal from MyRIO is HIGH, otherwise the robot stops moving and stays still.

Figure 6 shows the software development and hardware connection for the communication between the two different controllers. The IR line sensor, servo motors, servo motor driver, and Arduino Uno are connected in serial. The webcam is connected to MyRIO, while one of the digital input/output pins from both Arduino and MyRIO is used in the alternate control option in order to allow MyRIO to take manually RUN or STOP the line follower robot.

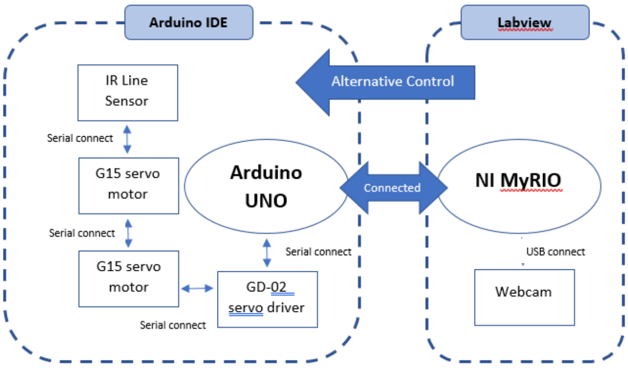


Figure 6: Graphical illustration on the hardware and software structure of project prototype.

III. RESULTS AND ANALYSIS

A. Line Following Performance

The overall testing result of the prototype robot was satisfied especially when it is being tested on the straight line. In the Arduino coding, the wheel mode of the servo motor is set to the maximum speed, which is 100 revolutions per minute. When the prototype robot is sensing the corner path, the robot is programmed to have one wheel slowed down to half of the original speed to ensure smooth turning performed by the robot.

However, during the preparation of the line path one corner of the loop path is purposely designed to have a sharper corner to observe the reaction of the robot. As a result, the line sensor missed sensing the line at the designed sharp corner. This is due to the robot moving speed could not match to the supposed turning speed, plus the fast detection of the Arduino processor, before the robot was able to finish a turn, the robot is already off-tracked and lead to zero output from infrared sensor as it detects no line.


Figure 7: Prototype robot missed the track at the corner but back to the line after that.

One way to solve this problem is that instead of making the line-follower stop when the infrared sensors detect no line, the coding of the prototype is modified to turn to a specific direction when especially when facing the scenario where either all the infrared sensors sensing the line or all of them sense no line. Figure 7 shows the successful back-on-track after it missed the line at the sharp corner.

C. LabVIEW Monitoring Front Panel

After finish constructed the block diagram for the monitoring system, the front panel of this LabVIEW project is rearranged and well decorated to give a comfortable monitoring experience as shown in Figure 8.

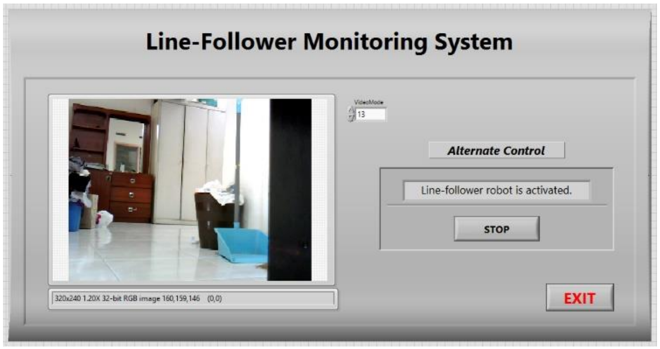


Figure 8: Monitoring Front Panel on LabVIEW.

Overall, this front panel buffers the real-time images through the webcam connected to the NI MyRIO which is wirelessly connected to LabVIEW. The alternate control gives the user the highest authority to hold the line-followers movement, or to resume its movement. The video mode control above the alternate control area determines the resolution as well as the frame per second image processing speed of the webcam.

The maximum resolution for the webcam used in this project supports up to 1600 x 1200 pixels but has a low frame per second of only 5 fps. High video resolution provides a sharper image, however due to its high resolution, the processor needs longer time to process them thus resulting in delay during monitoring. Apart of the video resolution, image processing speed of a webcam also tends to affect the overall performance. High frame per second give smooth transition of the image thus resulting in smoother graphics especially when the webcam capturing moving objects, or the webcam itself is moving. Low frame per second can cause blurred images easily, unless everything stays still or move in extremely low speed, otherwise the image captured would be a waste of memory.

C. LabVIEW Front Panel for Image Filter Processing

Apart from the original monitoring, there is also the front panel for further process on the images (Figure 8). The original images captured by the webcam undergoes further image filtering process to enable the horizontal distance detection (especially to detect the width of the line path ahead), and light intensity data collection.

Figure 9: Monitoring front panel for horizontal line width detection (left) and light intensity measurement (right).

The user can draw a random size of rectangular on display to filter only the portion of the area bounded by the drawn rectangle. The parallel red lines in Figure 10 is the vertical line detector, where it works by detecting potential vertical line path within the drawn rectangle. The clear-to-see green rectangle in Figure 11 is the selected region where the total light intensity within the region is calculated (in pixel). The 4-row table beside it gives the reading of mean intensity, standard deviation, minimum intensity, and maximum intensity of that selected region.

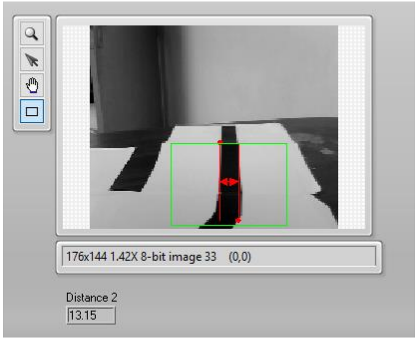


Figure 10: Parallel red lines detect the dark color path.

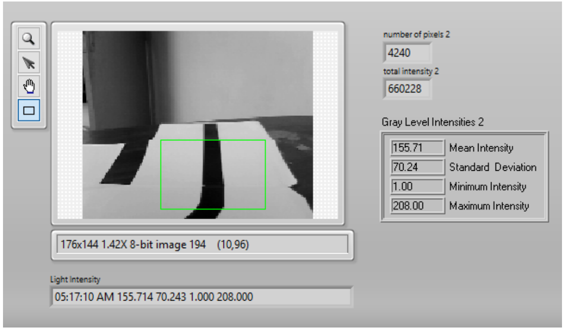


Figure 11: Light intensity data recorded within the drawn green rectangle.

This filtering process is tested under certain situations. The following figures show the output results obtained and displayed on the front viewing panel.

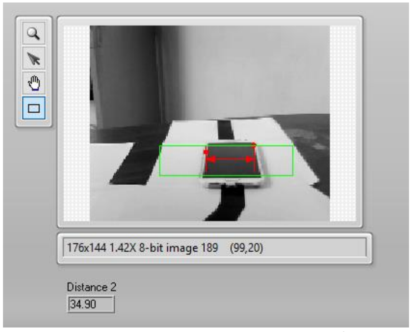


Figure 12: A mobile phone is placed on top of the line path.

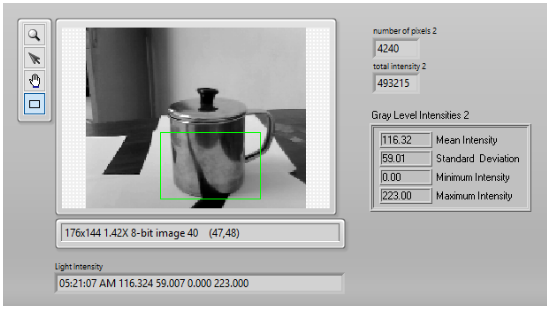


Figure 13: Reflective metal cup is placed within the green rectangular frame.

By referring to Figure 12, the width of the screen is determined to be 34.90 pixels out of the resolution width of

176 pixels, which is about 20% of the total horizontal pixels of the image. The actual width of the mobile screen is 7 cm, meaning that the actual surrounding width captured by the webcam is about 35 cm wide.

Referring to Figure 13, Even though a reflective medium is placed in front of the webcam but it does not seem to have greater light intensity reading, mostly due to the low light condition of the surrounding. The light intensity reading will be higher if there is a whiter region within the green rectangular frame.

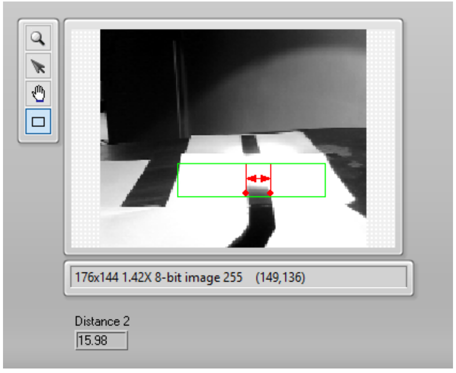


Figure 14: Bright light shines on the dark line path (test path detection).

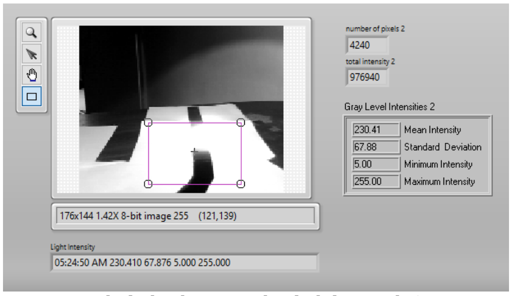


Figure 15: Bright light shines on the dark line path (test intensity).

A torchlight was used to shine on the dark surface of the line path as shown in Figure 14, yet the filter block was still able to detect the line width. Light intensity reading in Figure 15 showed great increased when the white color pixel in the frame increased.

IV. Conclusion

At the end of this project, a fully functional prototype of a line-follower surveillance camera monitoring system is successfully designed and constructed. This robot is capable of following the line path prepared with a high degree of efficiency and has very low possibility of getting off-track subjected to have a clear path to the destination.

In the future, this project concept can be further integrated into intelligence patrol robot that is suitable to use in any residential area to provide large area high-efficiency monitoring which fit the limitation of surveillance cameras, or present monitoring systems. Implementing this type of monitoring robot could reduce the requirement of installing high-cost surveillance camera around the houses.

Acknowledgment

Authors would like to thank Machine Learning & Signal Processing (MLSP) research group under Center for Telecommunication Research and Innovation (CeTRI) of Universiti Teknikal Malaysia Melaka (UTeM) and Faculty of Electronics and Computer Engineering (FKEKK), UTeM for the use of existing facilities to complete this project and funding provided under 'UTeM Zamalah Scheme'. Authors would also like to thank Ministry of Higher Education (MOHE), Malaysia for sponsoring this work under project RAGS/1/2015/TK0/UTEM/02/2.

References

[1] N. G. La Vigne, S. S. Lowry, J. A. Markman, and A. M. Dwyer, "Control and Prevention."
[2] K. Kobayashi, K. Iwamura, K. Kaneda, and I. Echizen, "Surveillance Camera System to Achieve Privacy Protection and Crime Prevention," pp. 6–9, 2014.
[3] A. Isnard, "Australian Institute Of Criminology Can Surveillance Cameras Be Successful In Anti-Social Behaviours ?," no. August, 2001.
[4] L. Chaisorn and Y. Wong, "Video Analytics for Surveillance Camera Networks," vol. 2, 2013.
[5] J. Thirumurugan, "Line Following Robot for Library Inventory Management System," pp. 1–3, 2010.
[6] T. Fujiwara and Y. Iwatani, "Interactions with a Line-Follower : an Interactive Tabletop System with a Markerless Gesture Interface for Robot Control," pp. 2037–2042, 2011.
[7] K. M. Hasan, C. E. Discipline, C. Engineering, and P. Putra, "Sensor Based Autonomous Color Line Follower Robot with Obstacle Avoidance," pp. 598–603, 2013.
[8] P. Goel, G. Arora, and V. K. Panchal, "Incorporating Perception Radius to the Line Follower Robot," 2014.
[9] I. N. Rodrigues, C. Luiz, S. De Melo, and V. Mello, "FPGA hardware architecture with parallel data processing to detect moving objects using the background image subtraction technique," pp. 841–846.
[10] M. Engin, "Path planning of line follower robot," pp. 1–5, 2012.
[11] F. Ortega-zamorano, J. M. Jerez, D. U. Muñoz, R. M. Luque-baena, L. Franco, and S. Member, "Efficient Implementation of the Backpropagation Algorithm in FPGAs and Microcontrollers," vol. 27, no. 9, pp. 1840–1850, 2016.