Grenze International Journal of Engineering and Technology, June Issue

Integrating Autonomous Line Following Model using PID Controller with Enhancing Location Intelligence

M. Niharika[1] and Prof. Charanraj B R[2]
[1-2]Computer Science, PES University, Bangalore, India
Email: niharikamukesh1223@gmail.com, brcharanraj@gmail.com

Abstract—**It presents Autonomous Line Following Model, leveraging the transformative capabilities of Internet of Things (IoT) technology. The integration of real-time object detection ensures dynamic tracking of surrounding objects, while precision lane following guarantees adherence to a predefined path through a Proportional-Integral-Derivative (PID) algorithm. Live monitoring is achieved through Webcam by image processing, Convolution Neural Networks (CNNs), enabling remote access to the vehicle's camera feed. This multifaceted approach not only amplifies location intelligence but also provides high-quality live-streaming capabilities, significantly enhancing the model's real-world applicability. Project provides accident prevention using different sensor simulation whereas the vehicle is stopped slowly and enabling the HRMS watch module enables driver status detecting the heart monitoring function of the driver The synergy of these components underscores the immense potential of IoT in autonomous systems, exemplifying efficient and intelligent navigation.**

***Index Terms*— PID, Autonomous car, CNN, HMS, image processing.**

I. INTRODUCTION

In the rapidly evolving landscape of autonomous systems, our 2024 paper introduces an Autonomous Line Following Model, capitalizing on the transformative capabilities of Internet of Things (IoT) technology. This innovative model goes beyond conventional autonomous navigation by seamlessly integrating cutting-edge features such as real-time object detection, precision lane following monitoring through a Proportional-Integral-Derivative (PID) algorithm, live monitoring through Webcam by image processing, Convolution Neural Networks (CNNs), enabling remote access to the vehicle's camera feed. This multifaceted approach not only amplifies location intelligence but also provides high- quality live-streaming capabilities, significantly enhancing the model's real-world applicability with high-quality live- streaming capabilities. The system architecture incorporates Arduino Uno microcontroller that is essential for precise lane following and obstacle recognition, which enables the car to follow a predefined route by using PID Algorithm majorly controlling the function of hardware components. Raspberry Pi microcontroller is essential for live streaming with image processing, which is directly connected to a webcam which control manual and automatic mode of driving that provides accident prevention using different sensor simulation and execute precise movements.

II. LITERATURE REVIEW

In this sectionof ou r paper, we have discussed various ways to design an IoT-based AI surveillance car. The

Grenze ID: 01.GIJET.10.2.999_1

research has highlighted that the model can be built in several ways using different microcontrollers having its own strengths and weaknesses that need to be taken into account when designing the system. Moreover, we have pointed out that the choice of microcontroller also affects the programming language used to develop the prototype. For example, the programming language used by Arduino is C/C++, the scripting language used by NodeMCU is Lua, and the programming languages used by Raspberry Pi include Python, C, and Java [1]. There are many ways to design this prototype, some are listed below.

III. Components

1. Arduino-UNO: The Arduino Uno is a widely-used microcontroller board in various electronic projects, including remote-controlled vehicles. Its compact design and compatibility with various electronic components make it a popular choice. Bluetooth technology enables wireless communication between devices, and it is commonly used in remote-controlled devices such as remote-controlled vehicles. Compared to wired connections, Bluetooth offers more control and efficiency Arduino board. The Bluetooth module then translates these signals into electrical signals that control the movements of the remote-controlled car.
2. Webcam: The transmitting unit captures continuous video footage using a camera installed on the robotic vehicle and wirelesslytransmits it to the receiving unit. It consists of an Arduino Uno, an L293D motor driver and TTGher motor have been employed, and an IR sensor has been added to detect obstacles in its path. A buzzer has also been includedto provide audio feedback to the user, and a Li-ion battery is used as a power source to make the car portable. and a power supply battery.
3. L293D motor driver: The L298N motor driver is a popular integrated circuit that can be used to drive DC motors in both directions, meanwhile, the receiving unit is responsible for receivingthe video and control signals from the transmitting unit and controlling the movement of the robotic vehicle. It includes an HC-12 receiver, a motor driver IC, two DC The HC- 12 module is utilized for wireless communication with a range of up to 1.8 km. The proposed system enables Six different types of movements, namely stop, forward, backward, left, right and backward.
4. Raspberry Pi: The Raspberry Pi is an affordable, compact computer that can connect to a monitor or TV, and uses a standard keyboard and mouse. A prototype remote-controlled (RC) vehicle has been developed for surveillance purposes that can transmit live video and be controlled over the internet. The Raspberry Pi used to control the RC car is equipped with Wi-Fi connectivity to connect to a web server.
5. Webcam: A USB webcam is connected to the Raspberry Pi to enable live video broadcasting, while a motor controller connected to the Pi controls the car's movements. The user can access a web interface that provides both live video streaming and an RC controller panel. This interface enables the user to remotely control the vehicle and view live video feeds while exploring the area.
6. Battery and Jumpers:: Three AA batteries power the RC car, and a 37-watt power bank supplies 5 volts and 2.1 amps to the Pi. It permits the transfer of electrical signals between elements.
7. Buzzer: The buzzer is a device that is capable of generating sound. by using an IR sensor that can detect the object's distance, and we have connected it to the microcontroller board. When the distance reading is less than 10 cm, the microcontroller sends a signal to the buzzer to generate a sound Buzzer
8. TT-Geared motor: The TT-Geared motor provides the necessary driving force to move the car. With its high torque output and compact design, it is an excellent choice for applications that require precise control and manoeuvrability.
9. Heartbeat Sensor: consists of a light-emitting diode and a detector like a light detecting resistor or a photodiode. The heartbeat pulses cause a variation in the flow of blood to different regions of the body. When tissue is illuminated with the light source
10. Irsensor: They are often utilize infrared light-emitting diodes (IR LEDs) to emit infrared radiation and infrared detectors to receive and analyze the reflected or emitted radiation.

IV. Methodology and Results

The objective of this researchpaper is to design and implement an Internet of Things (IoT)-based artificial intelligence (AI) surveillance car with obstacle detection, lane following, and live streaming capabilities the system employs an ultrasonic sensor for obstacle detection, IR sensors Series for line following with pid algorithm, and Webcam for live streaming The performance of the system is evaluated using various scenarios to ensure its functionality, reliability, and accuracy.

The IR sensors series are placed underneath the car are angled toward centreline, left and right sides, respectively that detect the black tape and send signals to the Arduino UNO, which in turn sends a signal to the L293D motor driver with car wheel which is connectedto the car's DC motors controllingthe speed and direction which is programmed by PID Algorithm. Enabling Web CAM connected to the access point IP address of the cam which will open a web page with a live video stream from the camera.

- Lane Following Accuracy: The accuracy of a lane follower with 180 RPMof greatest reactiontimes were 0.84 and
- 1.63 will vary depending on a number of variables, environmental conditions can have an impact on response time.
- Obstacle Detection Accuracy: The accuracy of obstacle detectionusingan ultrasonic sensor depends on the prototype is designed to detect obstacles within a range of 10 cm.
- Live Streaming: The delay rate for live streamingwith an QHM495LM webcam with Image resolution interpolated to 25 mega pixels with 6 light sensor, 16 MP Image Resolution, USB Interface and Night Vision Focus Range: 4 cm to infinity is impacted by a number of factors, including the quality of the internet connection, the stream resolution, the microcontroller's processing capability, and the system's programming. Transmit video at up to 1920X1080 FHD resolution and 30 FPS, the greatest latency for live streaming was measured to be 15 seconds.
- HMS System: emergency/push button continuously measures the BPM through a pulse sensor, and if the reading is ever below 45 BPM (Beats per Minute), through a Bluetooth signal transmission, the car shifts from manual to self-driving (automatic) mode. If at any point of time if this button is pressed, the car transitions from manual to automatic mode. Autonomous car system: The car at any point of time, is in one of the two modes: Manual or Autonomous In manual mode For lane detection, Initially the pi camera takes images of the line and using image processing the techniques the binary image is used to find the Regionof Interest. Canny edge detectionis used to extract the relevant information so as to reduce the amount of data being processed. Using Hough line transform, the lines are detected and the midpoint is calculated. Then Based on the collected data, the car is parked accordingly.

A. Working Flow

PID Algorithm for Lane Following

- Proportional (P): Explain the proportional component's role in responding to the current error. The proportional term contributes to the correction proportionally to the error magnitude.
- Integral (I): Discuss the integral component's role in handling accumulated past errors over time. It eliminate any steady-state error that may persist.
- Derivative (D): Explain how the derivative component anticipates future errors by considering the rate of change of the error. It helps dampen oscillations and improve system stability.

$$\text{Output} = Kp * \text{error} + Ki * \int \text{error}\, dt + Kd * d(\text{error})/dt$$

RC Car control in Manual Driving Mode

In Manual Driving Mode it involves communication between Car and Manual Controller. The Manual Controller is fitted with RF Transmitter and the car is fitted with RF receiver. Based on signal transmitted by the manual controller from RF Transmitter, the car receives the signal and performs actions accordingly.

RC Car control in Automatic Mode

In Self Driving mode there is serial communication between python in raspberry pi and Arduino UNO. Using image processingalgorithms, the lanes is recognized and python sends signals to Arduino UNO based on image processing algorithms and the Arduino responds accordingly.

Image Processing for the live Feed from Webcam

For detection of Lane lines, we use Open CV and by capturing the road through Pi Camera, certainimage processing technique is applied to the video.

Gray Scaling:

Method convert continuous-tone image to an made up of Pixels, 3 channel color image has Red, Greenand Blue channels and in this each Pixel is a Combination of three intensity levels, whereas gray scale image has only 1 channel and Pixel has only one intensity value ranging from 0 → 255. So, by using a gray Scale image processinga single channel is faster than 3 channel and less computational intensify [4].

gray = cv2.cvtcolor(lane_image, cv2.color_RGB2GRAY) (1)

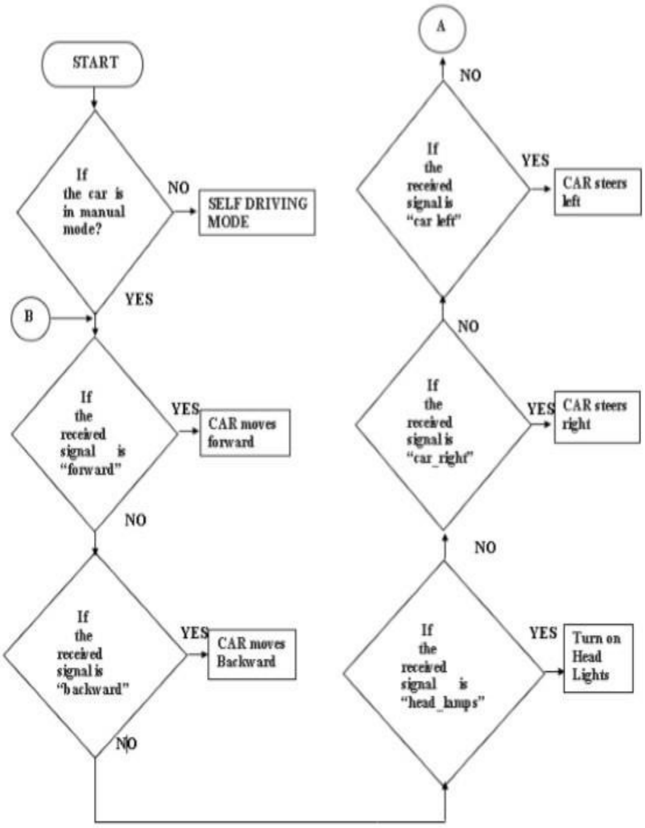

Fig 1. Manual Mode Flowchart

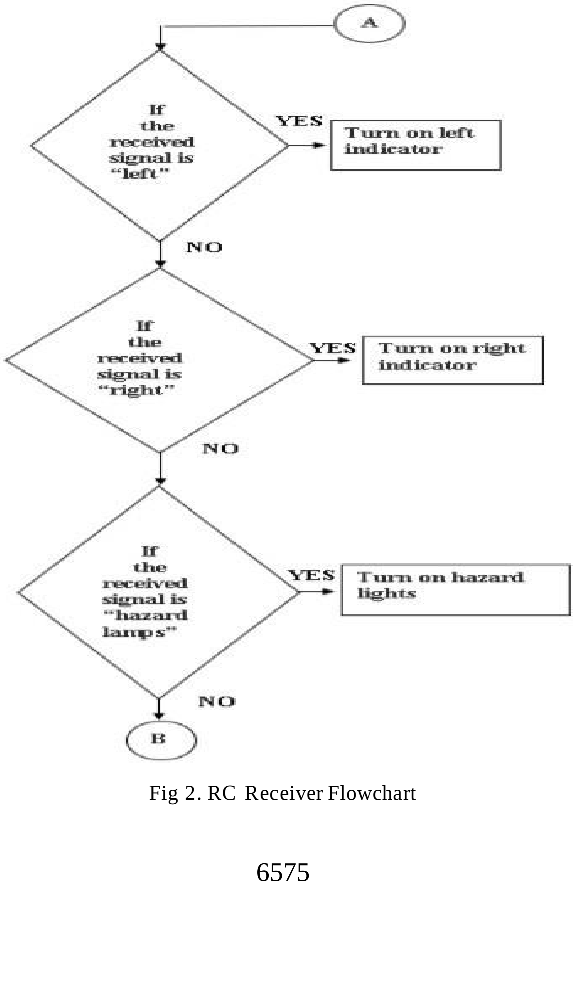

Fig 2. RC Receiver Flowchart

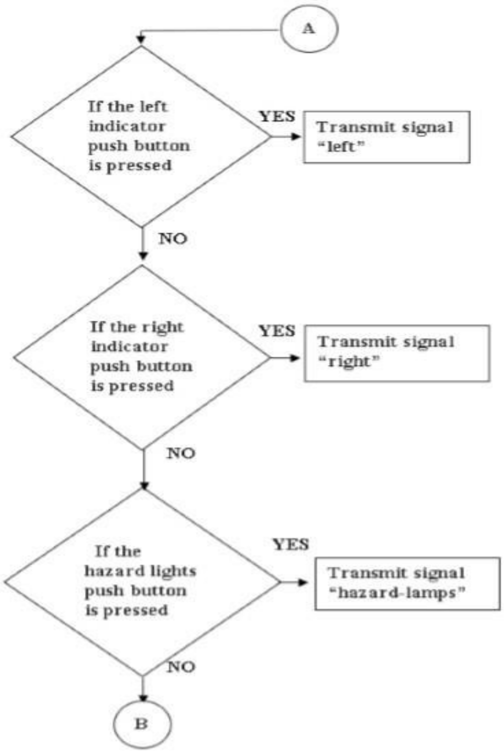

Fig 3. RC Transmitter Flowchart

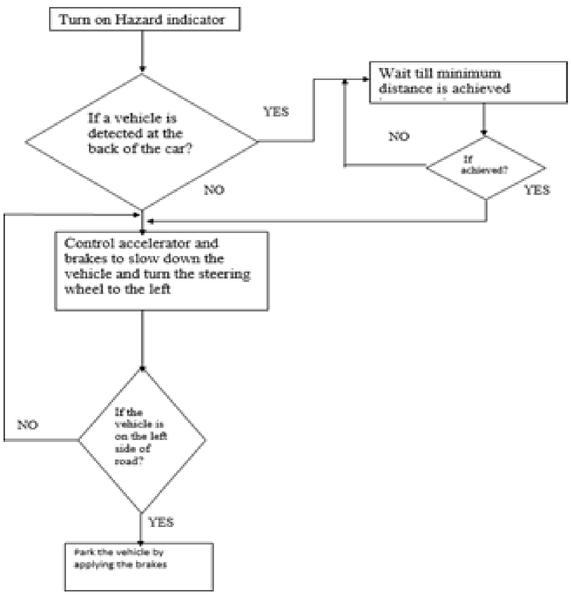

Fig 4. Autonomous mode working flow

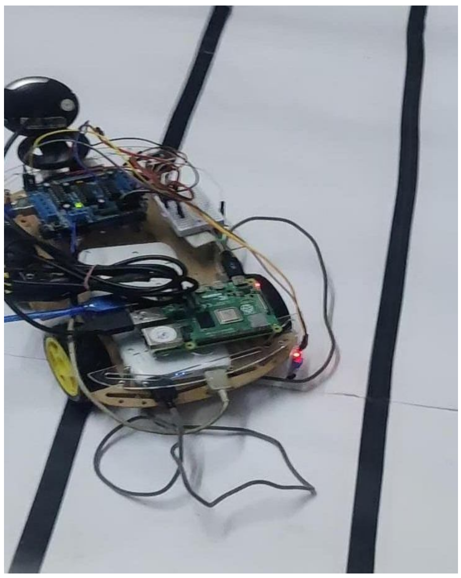

Fig 5: Left turn and Park

Gaussian Blur

A Gaussian smoothing is the end result of blurring an image by means of Gaussian function. It is extensively used impact in graphics software, commonly to lessen image noise and decrease detail. If noise is present it can detect false edges and effects edge detection. Image is a collection of Pixels, where each of the pixel in the gray scale image is represented by a single number that describes the brightness of that particular pixel.

Gray = cv2.cvtcolor(lane_image, cv2.color_RGB2GRAY) (2)

Blur = cv2.GaussianBlur(gray, (5,5),0) (3)

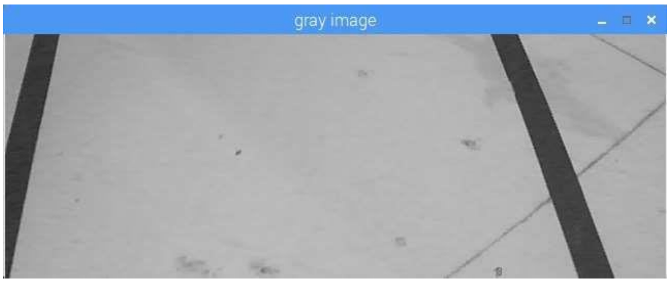

Fig 6: Grey Scaling

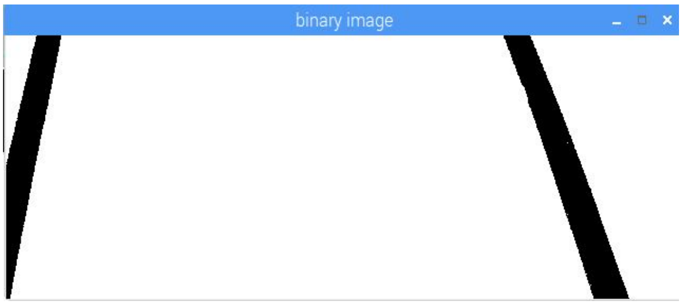

Fig 7: Binary image

Canny Edge Detection

Edge detection corresponds to the region in an image where there is a sharp change in intensity or sharp change in color between adjacent pixels in an image. To compute the gradience in an image one must recognize to represent an image in 2-D coordinate space X and Y. X – width of the image. Y – width of the image. Product of X and Y is the total number of Pixels in an image to Perform a derivative f(x, y) on our function X and Y directions Therefore, by measuring the change in intensity with respect to adjacent pixels.

$$\textit{gray} = \textit{cv2.cvtcolor(lane image, cv2.COLOR\_RGB2GRAY)} \quad (4)$$

$$\textit{blur} = \textit{cv2.GaussianBlur(gray, (5,5),0)} \quad (5)$$

$$\textit{canny} = \textit{cv2.canny(blur,50,150)} \quad (6)$$

Fig 8: Eroded image

Region of Interest and Hough Transform

Region of interest helps us to process only the particular region required and rest all will be ignored used for detection of regular curves like lines, circles, ellipses etc.

$$\textit{Height} = \textit{image.Shape [0]} \quad (7)$$

$$\textit{Polygons} = \textit{np.array([[(200,height),(1100,height),(550,250)]])} \quad (8)$$

Detection of Mid-point

Using Hough transform, we will be able to detect the left and right lines of the road, then we take one coordinate point (x1,y1) on the left line and one coordinate point (x2,y2) on right line which has equal y coordinate value i.e. y1=y2.Then we calculate the distance between the two points using the distance formula.

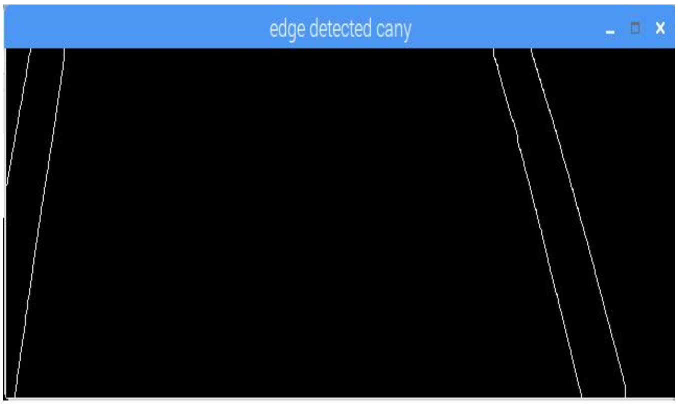

Fig 9: Edge detected canny image

$$distance = \sqrt{(x2 - x1)2 + (y2 - y1)2} \qquad (9)$$

After calculating the distance, we find the midpoint using the formula

$$mid = (distance)/2 \qquad (10)$$

We add the midpoint value i.e. mid to x1 value to get midx1

$$midx1 = mid + x1 \qquad (11)$$

mid x1 value contains the x coordinate value of midpoint of the lane. After calculating the midx1 value we take a reference point x ref which is the center value of width of screen. Original. The following pseudo code is used for knowing the position of the car in the lane.

Pseudo code

If xref < midx1 then

write data "right" into serial communication

if xref > midx1 then

write data "left" into serial communication

if xref == midx1 then

write data "straight" into serial communication

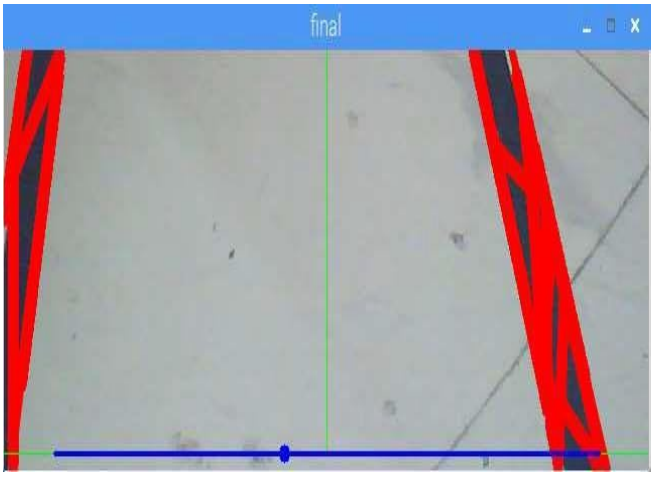

Fig 10: Midpoint between Lanes image

V. Future Scope

The future scope of this project includes the development of a surveillance car using Raspberry Pi instead of Arduino UNO. Raspberry Pi is a more powerful and versatile microcontroller that can do more complex tasks and add new functions to your system. Additional features we plan to include:

- Object recognition: Systems can be enhanced to recognize specific objects in the environment and respond accordingly. For example, it can be trained to detect specific vehicles or people and send alerts when detected.
- GPS Tracking: Equipping your system with a GPS module allows you to track its location and send this data to a remote server. This feature is particularly useful for surveillance and monitoring applications where real-time tracking is essential.
- Voice Control: Add a microphone and speaker to enable voice control of your car. It allows users to operate the car using voice

VI. CONCLUSION

In conclusion, this study successfully demonstrates the design and implementation of an IoT-based AI surveillance vehicle equipped with obstacle detection, tracking, and live streaming capabilities. Using an ultrasonic sound sensor for fault detection, an IR sensor with a PID algorithm for the following bands, and a live web camera, the system efficiently integrates hardware and software components to achieve reliable and accurate performance. The accuracy of the system evaluated under different conditions shows promising results with minimal distortion despite environmental impacts. Obstacle detection accuracy of 10 cm ensures timely response to avoid collision. Although the live streaming capability is affected by the quality of the internet connection and other factors, it achieves satisfactory performance with a maximum latency of 15 seconds. The integration of the HMS system improves the car's safety features by enabling automatic mode transitions based on the driver's health. The automatic system provides accurate navigation and parking by using advanced image processing techniques for lane detection. Overall, this project not only demonstrates the potential of IoT and AI to improve the performance of autonomous vehicles, but also lays the foundation for future developments aimed at more complex and multifaceted autonomous systems.

REFERENCES

[1] Chaudhari, Jagruti, Asmita Desai, and S. Gavarskar. "Line following robot using arduino hospital 2019" 2nd International Conference on Intelligent Communication and Computational Techniques (ICCT). IEEE, 2019

[2] P. Murmu, H. Paul, J. J. Roopa, and A. J. Timothy, "A Novel modernistic techniques in women security system using ESP32 and Arduino Uno," in 2019 2nd International Conference on Signal Processing and Communication (ICSPC), 2019, pp. 330- 33.

[3] Sonal, G., Raninga, P., and Patel, H. (2018), "Design and implementation of RGB color line following robot," in Proceedings of the International Conference on Computing Methodologies and Communication, ICCMC 2017, 2018, vol. 2018-January, pp. 442–446.

[4] The work "Neural controller of autonomous driving mobile robot by an embedded camera" was presented at the 4th International Conference on Advanced Technologies for Signal and Image Processing (ATSIP 2018), no. Figure I, pages 1–5, by H. Omrane, M. S. Masmoudi, and M. Mesmoudi.

[5] Miguel Tupac-Yupanqui, et al. "Exploiting Arduino features to develop programming competencies." 20602-20615, IEEE Access 10 (2022).

[6] Nelson Rai et al. "Automotive with Bluetooth remote control and Arduino." IEEE Trends in Engineering 33 (2016): 381-384. [3]

[7] Michael Yosua, Joni Welman, and Simatupang. "A remote controlled car using wireless technology." 1.2 (2016) of the Journal of Electrical and Electronics Engineering

[8] "Railroad Accidents: Common Causes, Statistics and Prevention". [Online]. Available: https://www.sidgilreath.com/learn/railroad - accidents-causes.html. [Accessed: 14-Mar.-2020].

[9] Nelson Rai et al. "Automotive with Bluetooth remote control and Arduino." IEEE Trends in Engineering 33 (2016): 381-384. [3]

[10] Michael Yosua, Joni Welman, and Simatupang. "A remote controlled car using wireless technology." 1.2 (2016) of the Journal of Electrical and Electronics Engineering.