Image Processing & M.L Based Driverless Car with Image Detection System

[1] **Ketan Bodhe,** [2] **Himanshu Taiwade,** [3] **Janhvi Ingle,** [4] **Juhi Patre,** [5] **Kajal Singh,** [6] **Sakshi Polkamwar**, [7] **Shreeya Bajirao**

[1,2] *Professor, Department of Computer Science and Engineering, Priyadarshini Institute of Engineering and Technology, Nagpur, Maharashtra, India*
[3,4,5,6,7] *Student, Department of Computer Science and Engineering, Priyadarshini Institute of Engineering and Technology, Nagpur, Maharashtra, India*

ABSTRACT

Every year, 1.25 million people are projected to die in traffic accidents around the world. Humans' failure to pay attention to road signs and follow the rules is a major cause of accidents. A signboard detection system has been installed to avoid this problem. This technology could be beneficial in recognizing certain domains like classrooms, traffic signals, colleges, hospitals, offices, and so on, as well as potentially saving many lives. This study describes the building of a low-cost prototype of a small self-driving automobile model utilizing simple and easily available technologies. In this prototype, the Raspberry Pi controller and H-bridge drive two DC motors to enable vehicle automation. Sonar sensors, image processing, computer vision, and machine learning have all been used in intelligent systems. We propose using a pattern matching methodology to create a self-driving vehicle to overcome the challenge.

Keyword: - *Raspberry Pi, Lane detection, Traffic light detection.*

1. INTRODUCTION

According to a survey conducted by the International Road Federation (IRF), India has the highest number of road fatalities, accounting for 10% of all road accidents worldwide. Road accidents are caused by human mistake in 78 percent of cases. New designs for self-driving vehicles are being developed as a result of rapid technical advancements in order to ensure accident-free transportation. Google, Uber, and Tesla are leading the global effort to design and manufacture self-driving vehicles. A self-driving car (also known as a driverless automobile) does not require human assistance and is capable of sensing its environment on its own. To detect the road, barriers, pedestrians, and other things in the environment, a multitude of sensors are combined and utilised.

This machine was made with a Raspberry Pi. The method of recording and identifying signboards mainly relies on digital image processing. Raspberry Pi designed the framework to make learning coding and computer programming easier for young people. The device provides real-time information from road signs that indicate the most difficult and vital jobs. Drivers would benefit greatly from our system's capacity to recognize, interpret, and infer road traffic signs. Because it is difficult to forecast what kind of conditions the vehicle will encounter, the algorithm must be able to achieve the intended output even when the conditions are unfavorable to obtain a high level of precision All road signs are placed in certain regions to protect the safety of all drivers.

The goal of road signs is to keep everyone safe on the road. Due to changes in weather conditions or viewing angles, traffic signals can be difficult to observe until it's too late. The purpose of an automated road sign identification system is to recognize and identify one or more road signs from a camera's live color images.

2. PROPOSED SYSTEM

According to the World Health Organization, approximately 1.25 million people die in traffic accidents each year, or 3287 people every day, and 20-50 million people are maimed and incapacitated for the rest of their lives (WHO). It is difficult to introduce contemporary systems in our country because they are manufactured in other countries. In some foreign countries, driving takes place on the other lane, whereas we drive on the left. India's traffic flow is far more congested than in other affluent countries, hence current systems struggle to work on these routes. The new technology uses a pattern matching process in which cameras detect a unique pattern that will be printed on the roads. This pattern will be recorded by the camera and processed by a Raspberry Pi before ordering the automobile to drive in the desired direction. The camera can also take ambient photos to identify various adjacent obstacles; if the obstacles become too close to the car or are going to collide with it, the car will come to a stop before the obstacle in front of it moves. Special patterns will be deployed alongside the route to assess what type of road is present ahead.

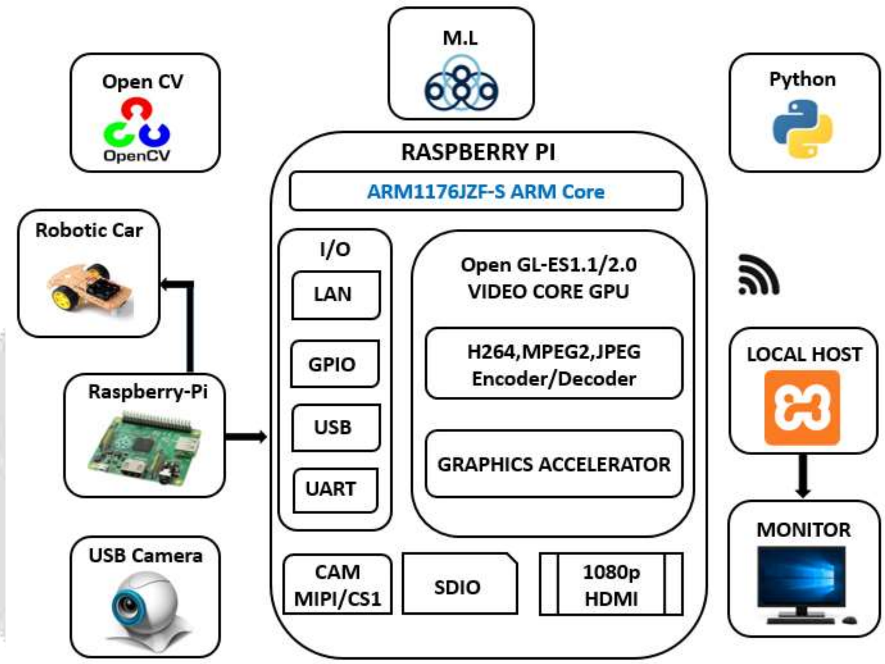

Figure-1: Block Diagram

3. METHODOLOGY

A succession of steps has been taken to construct a real-time signboard detection and recognition system. To construct an effective signboard detection and identification system, we employ the following methodologies:

The Raspberry Pi 3 controller's General-Purpose Input Output (GPIO) pins are used to connect the H-bridge motor driver and a sensor, such as an ultrasonic sensor. The web camera is mounted via the USB port on the Raspberry Pi board. Using the IC MCP3008 Analog to Digital Converter, the accelerometer sensor's analog output is converted to digital and sent to the Raspberry Pi 3's GPIOs.

The H-Bridge driver circuit regulates the movement of these motors in either a clockwise or anticlockwise direction after receiving control signals from the Raspberry Pi controller. Obstacles in the region are detected and measured by ultrasonic sensors in the front and infrared sensors on the left and right sides of the automobile.

An IR sensor module at the front controls lane following. The x-axis value of the accelerometer sensor located on top of the car is set to detect the driverless car's tilt, and it is connected to the controller through the IC MCP3008 ADC. A USB webcam is used in conjunction with a Python framework to track real-time artefacts and respect traffic laws.

The Raspberry Pi 3 is bundled with all of the programmes for implementing image processing methods, which are written in Python.

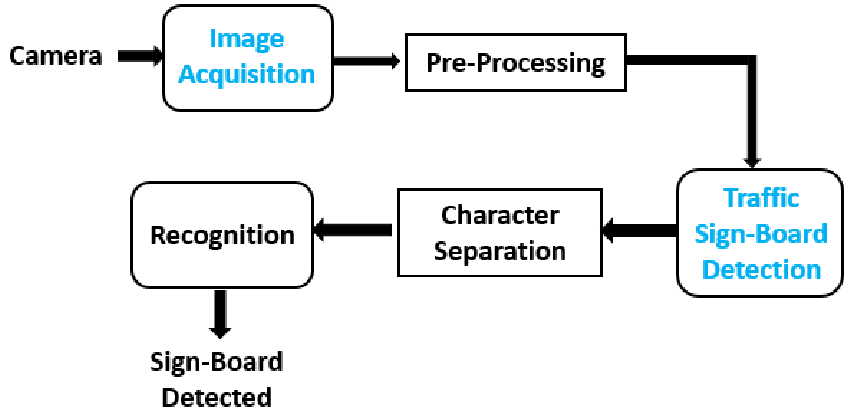

Figure- 2: Image Processing & Sign Recognition

The controller board is connected to an Android phone through Wi-Fi, allowing the phone to be utilized as an input device for commands and program execution. Raspberry Pi transmits a message as well as a location connection to the phone number and email address supplied in the file using Twilio. Using a mobile hotspot, the user can view the Raspberry Pi's desktop and monitor the car from anywhere, and the programming is done in Python using the Integrated Development and Learning Environment (IDLE). All self-driving car functionalities, such as Lane Detection and Following System (LDFS), Traffic Light Detection System (TLDS), Real-Time Object Detection System (RTODS), and Accident Warning System, can be selected by entering a code on the user's mobile phone.

A self-driving automobile must be able to recognize, monitor, and discriminate diverse routes in order to achieve the core goal of lane recognition and tracking. The LDFS uses three infrared sensors (IR1, IR2, and IR3) mounted on the self-driving car and connected to the Raspberry Pi controller to determine where the car is in relation to the yellow line in the middle of the lane.

A lane on the road has been planned with a yellow line drawn in the center for the projected self-driving car to follow. When IR1 and IR3 are low, a black lane is identified, and when IR2 is high, a yellow line is identified. The user's cell phone serves as a control mechanism for starting and directing the vehicle's motion. All of the automobile's sensors are triggered, and the motor starts, when the user enters the car and keys in 1 on his phone. The car starts moving ahead as soon as IR2 detects a yellow colour line. Both IR1 and IR2 detect a yellow line when the car is prone to shifting to the right, and the automobile shifts to the left before only IR2 sees yellow. IR2 and IR3 detect a yellow line when the car begins to turn left, then the vehicle shifts to the right until only IR2 detects yellow. As a result, the car self-corrects to the centre of the lane and goes forward as long as IR2 detects yellow alone. When the car starts to turn left, IR2 and IR3 detect a yellow line and shift the vehicle to the right until only the driver is left. Yellow is detected by IR2. As a result, the car self-corrects to the centre of the lane and goes forward as long as IR2 detects yellow alone.

The red, yellow, and green colors of the traffic light system are detected and monitored using computer vision principles. Object Detection System in Real-Time (RTODS) Object detection is used in a variety of applications, including autonomous vehicles, defense, surveillance, and industrial applications. Because it can run on video and produces a satisfactory result, the Single Shot Detector (SSD) is an excellent alternative.

4. WORKING & ARCHITECTURE

The process flow of the real-time object detection system is as follows: if an object is detected by the IR sensor on the left, the car should move right, and vice versa. When dynamically calculating its distance from the object, the front-mounted ultrasonic sensor identifies and overcomes small obstacles.

On the Raspberry Pi, Real-Time Object Detection is implemented using the OpenCV Deep Neural Network (DNN) and Mobile Net-SSD algorithms. In Python, the technique is implemented using the OpenCV library. Training and test datasets with ground truth bounding boxes, such as Caffe-datasets and associated class labels, are required for the Mobile Net SSD methodology.

The camera's input frames are in BGR format, which is transformed to equivalent Hue Saturation Values in BGR colour space (HSV). The range of values for Hue (representing colour) in OpenCV is 0 – 179, for Saturation (representing colour intensity/purity) it is 0 – 255, and for Value (representing colour brightness) it is 0 – 255 , the value range (which represents colour brightness) is 0 to 255. The colour range to be monitored is defined according to the requirements, and then morphological changes on the colour are applied to reduce noise from the binary image. The colours are then contoured using a rectangular delimited line called contour to distinguish each colour.

It outputs a grayscale image with each pixel indicating how closely its surroundings resemble the template. Contours are defined as a line that links all spots with the same intensity along the image's boundaries. Shape analysis, calculating the size of an object of interest, and detecting objects are all possible with contours. As a result, detecting an object with many feature maps increases the size of the item to be detected. To guarantee that the network learns what constitutes an inaccurate detection, a 3:1 ratio of negative to positive predictions is employed during preparation instead of all negative predictions.

This module's accuracy is determined by the thresholding values altering. It shows how closely a thresholding value resembles a recognized or common value. To determine if a value is correct, compare it to the agreed-upon value. And it's possible that these values were produced. For signboard identification and recognition, it offers an accurate result for each image acquired by the camera.

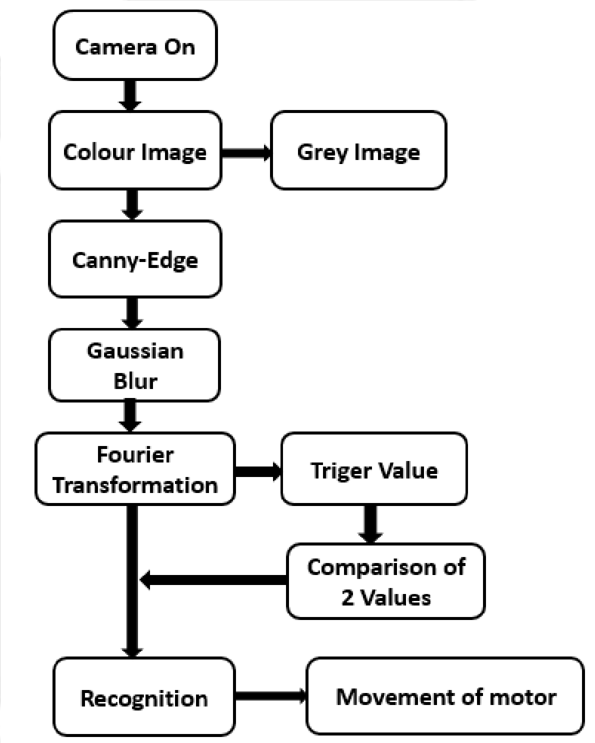

Figure-3: Flow Chart

We're working on a prototype that's simple to understand and implement. The following are the various components of the architecture and their functions:

1. **Contour-** Contour is a term for the continuous joining points that form a similar strength or color curve. It can be used to identify and recognize objects as well as analyze their shapes. In OpenCV, contour recognition is comparable to recognizing a white object against a dark background.
2. **Canny Edge Detection-** In image processing, edge detection is a technique for establishing the borders of artifacts inside images. It works by sensing brightness discontinuities. It is mostly used for picture segmentation and data extraction in domains such as computer vision, machine vision, and image processing.
3. **Gaussian Blur-** This is the result of using a Gaussian function to blur an image. It's a common visual software effect that's used to reduce visual noise and detail.
4. **Fourier Transform-** The Fourier Transform is a useful image processing tool for breaking down images into sine and cosine components. The Fourier Transform's uses include image processing, image filtering, picture restoration, and image pressure, to name a few.
5. **Raspberry Pi-** The Raspberry Pi is a low-cost computer that uses a regular keyboard and mouse and connects to a display or television. It's a powerful small device that, in the end, allows individuals to learn how to program in Scratch and Python.

6. **Ultrasonic Sensors-** An ultrasonic sensor is an electronic device that measures the distance between two objects using ultrasonic sound waves and converts the reflected sound into an electrical signal. Ultrasonic waves travel more quickly than audible sound waves (for example the sound that people can hear).
7. **Relay Motors-** Relays are electromechanical or electronic switches that open and close circuits. It opens and closes connections in another circuit to regulate one electrical circuit. When a hand-off contact is consistently open (NO), there is an open contact when the hand-off isn't energized, as transfer graphs indicate.

5. RESULT & DISCUSSION

1. Image Pre-processing Performance

To minimize storage space and reduce computing complexity, the original images are scaled down into pixels. In the suggested methodology, the RGB segmentation methodology is used to do picture pre-processing following the picture acquisition phase. In the suggested method, a filter is applied to each channel threshold area to pick only those portions of the image where pixel values are within the range of the target item. To smooth down the image and fill up the smaller areas, the median filter is utilized.

2. Traffic Sign Detection Performance

Using extracted data (center and area) from each of the final selected choices, the image is used to design their range of pixel values, region, and form. Only red-colored traffic indicators should be included in the proposed procedure.

Several difficulties were discovered during the studies that were affecting the detection efficiency. The sun segments and illuminates the red color of the traffic sign in a consistent manner, according to the proposed detecting method. The RGB model's color segmentation attribute, which is used to compare RGB values, causes this.

3. Recognition Performance

After color segmentation and form matching, the proposed method clusters the entire image using semi-supervised SVM. Intensity correction and histogram equalization are applied to conventional traffic sign photos to lessen the influence of changeable illumination. The processing time for the recognition system is 0.18 seconds. The system takes 0.43 seconds to run on average.

4. The System's Outcomes

The implemented technique is very accurate but sluggish on a 700 MHz Broadcom processor; the overall average time for both detection and recognition is 1.5 frames per second (fps). The detection step improves the number of photos that can be analyzed because it is based on form and runs faster than recognition. To avoid errors caused by prior images, the processor should be paused for a short time. The biggest issue with shape-based detection is the existence of obstructions around speed signs. Python-OpenCV is a wrapper over the original C/C++ code that makes it simple to use. However, there are native Python script written functions that are not available in OpenCV, which significantly slows down performance.

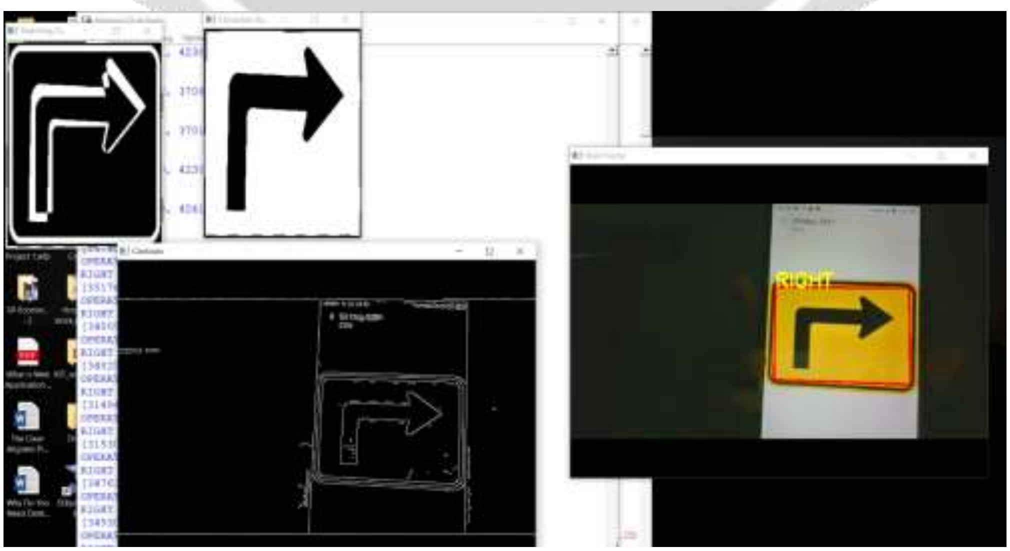

Figure-4: Sample output

6. CONCLUSION

The work in this study is divided into two components, labeled "detection" and "recognition," which are similar to other applications in the field. Shape-based algorithms were utilized for detection since color-based segmentation is substantially less reliable than shape-based segmentation. To recognize and identify these potentially harmful road signs, a database of road sign patterns and restrictions is used. Complex techniques (such as Eigenfaces, SURF-SIFT template matching, and Fuzzy matching) were not chosen because of the Raspberry Pi's limited computing capability, despite their availability in the OpenCV library. In order to keep Raspberry Pi functioning smoothly, other classifiers such as k closest neighbor and Euclidean distance were chosen for this project.

Detection, monitoring, and recognition are all linked; recognition removes false positives, while detection narrows the options and enhances the accuracy of the system. Another important part is maintaining the results; as a result, when additional databases are produced throughout time, it must be improved.

The main idea is to use an input photo as a starting point to recognize and classify traffic signals. Our technology is totally automated, removing the need for manual labor. Human error is reduced, and precision, processing speed, and reliability are enhanced as a result of the automation process. This paper outlines a way for developing a self-responding system.

7. REFERENCES

[1] Chandan G, Ayush Jain, Harsh Jain and Mohana, Real-time object detection and tracking using deep learning and OpenCV, III International Conference on Inventive Research in Computing Applications (ICIRCA 2018).
[2] Karti Balasubramanian and Arunkumar, Object recognition and obstacle avoidance robot, Chinese Control Design & Conference (CCDC 2009).
[3] Renjith.R B.Tech Student (Dept. of Electronics and Communication) Sree Narayana Gurukulam College Ernakulam, India ren1234ith@gmail.com. IEEE (2017)
[4] Fahim Bin Basheer, Jinu J Alias and Mohammad Favas C, Design of accident detection and alert system for motor cycles, International Conference on Automobile Engineering (IEEE 2013).
[5] Tiple Anjali Hemant, Tiple Hemanth P and Gauravsagar Hanumanth, Prototype of autonomous car, 2018 International Journal of Engineering Research in Electronics and Communication (IJERECE 2018).
[6] Tiago Moura, Antonio Valente and Vitorphilipe, The traffic sign recognition for autonomous robot, 2014 International Conference on Autonomous Robot System and Competitions (ICARSC 2014).
[7] R. M. Swarna Priya (B)·C. Gunavathi (B)·S. L. Aarthy School of Information Technology and Engineering, Vellore Institute of Technology, Vellore, Tamil Nadu, Indi. SPRINGER (2019)
[8] Daniel L. Rosenband. ―Inside Waymo's Self-Driving Car: My Favorite Transistors‖, Symposium on VLSI Circuits Digest of Technical Paper 2017.
[9] R. Tiwari (&) D.K. Singh Lovely Professional University, Phagwara, Punjab, India. SPRINGER (2018)
[10] Mochamad Vicky Ghani Aziz Control and Computer System Laboratory School of Electrical Engineering and Informatics, ITB Bandung, Indonesia.IEEE(2017).