Journal of Engineering and Computer Sciences
Qassim University, Vol. 15, No. 1, pp. 71-94, (January 2022)

Development of Autonomous Cars using Raspberry Pi and Google Accelerator for Machine Learning Based Lane Detection

Rayan Hamza Alsisi[1], Mohamed Uzair[1], Nader Oudah[1], Youssef Mabrouk[1], Omar Alnawajha[1], Arshad Karimbu Vallappil[1], Marwan Hadri Azmi[2], and Hafiz Abdul Wajid[1]
[1]Faculty of Engineering, Department of Electrical Engineering, Islamic University of Madinah, Madinah 41411, Saudi Arabia.
[2]Faculty of Engineering, Department of Electrical Engineering, Universiti Teknologi Malaysia, Johor Bahru 81310, Malaysia

(Received 4/1/2023, accepted for publication 1/9/2023)

Abstract. Autonomous Vehicles (AV) are the smart cars of the future anticipated to be driverless, efficient, and crash-avoiding ideal urban cars. Software complexity, real-time data analytics, verification, and testing are among the more significant challenges in autonomous driving technology. This article presents practical experience and valuable insight into the above-mentioned challenges by developing a lab-scale autonomous car prototype using Rasberry Pi and Google accelerator. The full description of the car, including its technical specifications, the hardware and software design procedures, and the lab-scale circuit for testing, are discussed in detail. The developed prototype is equipped with the machine learning-based lane detection algorithm. The performance of the installed lane detection algorithm is verified by testing the car prototype using the lab scale circuit.

***Keywords*:** Autonomous Vehicle, Artificial Intelligent, Machine Learning, Computer Vision

1. **Introduction**

With around 2 billion vehicles predicted to roam the road by 2030, road safety is one of the essential agendas of governments around the world. Currently, measures to enhance road safety have been carried seriously, with the help of technologies like Closed-Circuit Television (CCTV) cameras, road sensors, and more [1]. However, despite these efforts, based on data from the World Health Organization (WHO) [2], road accidents have caused around 1.25 million deaths yearly. WHO has also projected that these numbers will hit 1.8 million by 2030 [3]. Furthermore, it has been reported in [4] that the number of road accident fatalities in the United States alone was 32,000 in 2014 and increased to more than 35,000 in 2015, which demonstrates that human errors can still occur even with the assistance of the currently deployed technologies. Innovative technologies such as connected and autonomous cars are actively studied to help reduce these human errors and subsequently improve life-threatening situations on the road.

An autonomous car is a computer-controlled car that can mainly guide itself on the road, make its own decision, and react to the changes in its surroundings. All these functions are being operated by the car without human interaction. The autonomous car should be supported by features such as navigation and path planning, maneuver control, and neighborhood awareness, which includes object detection, lane detection, self-positioning, and lane spotting [5]. Lane detection plays an integral part in the operation and safety of autonomous cars via lane keeping and lane departure control systems, ensuring the car is safely located in its lane and subsequently minimizing the collision on the road. Machine learning and deep learning mechanisms have been actively identified to potentially support lane detection features in an autonomous car [6][7]. Within this domain of research, several approaches have been proposed [8]. Among the work is the work of [9], which proposes a multi-sensor data system utilizing a deep neural network to detect the lane in a 3D space. The Convolutional Neural Network (CNN) and its waveform are analyzed in detail for lane detection [10]. In another set of works [11], stereo vision and CNN are developed in the lane detection and classification strategy for the autonomous car. Finally, in [12], a recurrent neural network is proposed to perform the autonomous car's features of lane detection and collision avoidance.

The idea of the autonomous car, despite its challenging features, opens up new innovative applications and presents consumers with safety, ease of use, comfort, and value-added services. To study the impact of these technologies, small prototypes of the autonomous car are built to analyze the performance of the car in the control laboratory environment. A prototype vehicle equipped with drive-shaft and steering encoders, a scanning laser range finder and a passive RFID reader is used as a mapping tool, applicable to underground mining, which combines odometry, laser range scanners, and RFID beacons [13],[14]. A lab-scale dump truck equipped with a Light Detection and Ranging (LIDAR) sensor is used to perform the navigation triangulation, in combination with reflective beacons placed on apriori known positions to demonstrate the feasibility of autonomous underground navigation [15]. The work of [16] builds the prototype shuttle car and the lab-scale and mock mine environment and provides preliminary results on autonomous navigation in the mining industry [16]. The autonomous car prototype in [16] is equipped with a series of cameras for the surrounding views, and LIDAR tracking is used for collision avoidance and emergency brakes.

To the best of our knowledge, there are still no autonomous car prototypes in academia that study the practical aspect of the lane detection features of an autonomous car. To add to this line of study, our work develops the autonomous car prototype using Rasberry Pi with the Google accelerator, implementing an AI-based technique for lane detection problems. This paper conveys practical experience developing and testing the lab-scale autonomous car prototype. It includes the full description of a laboratory-scale autonomous car with a detailed machine learning-based lane detection algorithm, its design procedure, performance results on the lab-scale circuit, and a summary and conclusion

2. Impacts of Autonomous Vehicles on Intelligent Transport

Autonomous vehicles will greatly impact the societal, health, environmental, and safety domains of the intelligent transport industry. AVs are generally designed to ensure improved performance in different situations. However, eliminating various risks, such as accidents, is still not feasible. Therefore, let us briefly discuss the promising positive impacts of AVs and their associated risks.

We start with the clear and significant benefits of AVs, which is enhanced traffic safety compared to vehicles with drivers. This feature will result in many subsequent positive impacts, such as a considerable drop in accident ratios globally and enhanced safety for pedestrians, side walkers, and cyclists. AVs can also be considered green technology with massive potential in reducing noise and air pollution, as soon most AVs will be electric. The direct impact of AVs is even more significant to the driver, such as less stress and tired from long hour driving or when stuck in traffic jams. When it is applied to public transport, it improves the utilization of commute times and increases social interactions since AVs are viable to all irrespective of age, can or cannot drive, easy mobility to venues and events, Etc. Already mentioned benefits will also result in a stress-free live style, subsequently improving overall health.

While the positive impacts of AVs are huge, there are also risks associated with the large-scale deployment of AVs in the future. One direct impact might drain away certain types of jobs, especially those who drive for a living. Based on the statistical study by Goldman Sachs, it is estimated that there could be up to 25,000 jobs lost a month in the US. With truck driving as one of the most common occupations in the US, that figure could turn into over 300,000 job losses per year. On the other hand, an estimated 1.7 million truckers could also be replaced by self-driving counterparts in the UK alone.

Besides drivers, there is also a reduction of jobs in law enforcement, as the automation of cars enhances the safety domains by reducing and eliminating traffic violations. Table I lists the advantages and disadvantages the future of AVs will offer to the environment, society, human health, and safety [17-21].

Table (1). Advantages and disadvantages of AVs

Domains	**Positive Impact**	**Reason**	**Negative Impact**	**Reason**
Societal	Ease of access	1. Door-to-door access for people of all ages (kids, adults, or elderly) 2. Enhanced access to venues for intellectual, pleasure, or social exchange	Job cuts on a mega-scale specifically in the transportation sectors	Public transport vehicle drivers are no longer needed.
	Wider mobility	For people who cannot drive because of disabilities and serious illness		
	Less number of cars per family	Single AV can perform multiple rounds for all your tasks daily reducing the number of cars on roads		
	Commuting time utilization	Riders can utilize commute time by working or relaxing		

	Enhanced affordability	Reduced ride costs enabling middle- or lower-class community		
Health	Improved health	1. Less stress - Long hours of driving directly affect human health as driving is a stressful job. Therefore, overall health is improved consisting of physical, mental, etc. 2. Fewer noise - Less cars on roads will reduce noise excessively, which as per research related to hypertension, cardiovascular health, etc.		
Environment	Less pollution	1. Gas consumption is significantly reduced with AVs directly ensuring less pollution I. Less number of AVs will be on the road II. AVs will utilize optimized routes from destination A to B III. Optimized performance of acceleration and braking systems 2. Moreover, future AVs will be electric vehicles reshaping		

		the transportation industry completely		
Safety	Significant reduction in traffic accidents because of driver-related crashes	Many violations made by the drivers are completely avoided such as 1. Over speeding 2. Impaired driving 3. Traffic violations 4. Distractions (use of cell phones etc.)	Job cuts related to the law enforcement unit.	Better transportation infrastructure reducing fines
	Increased pedestrian safety	Traffic violations such as not following traffic signals will drop to zero with driverless cars which is not possible in case of a driver on steering control		

3. Concept Design: Camera with Google Accelerator Unit (Gau) Connected to Rasberry Pi Processor

3.1 The concept design of the autonomous car prototype:

Fig. 1 presents the main hardware and software components to design and build the autonomous car prototype. First, the proposed design consists of a camera installed at the front end of the car prototype with the computer vision function capturing the video of the front sight. The camera and its function act as the vision for the designed prototype, and the captured video after being processed via the artificial intelligent method is used to steer and maneuver the forward movement direction. The steering and forward movement manoeuvring is performed by the DC motor. To perform the AI processing on the contents of the captured video frame, Python programming with an OpenCV package installed and running on the Raspberry Pi and Google accelerator is used.

The Raspberry Pi, refer to Fig. 4, is a single printed circuit board with four cores achieving a speed of 1.5GHz. Moreover, it is equipped with either 2 or 4 GB of RAM capable of processing and analyzing photos and instructing proper command to the DC Motors attached to the wheels of the car. However, the Raspberry Pi unit can analyze one frame per second, making it unvital to control the motors. This limitation is partially improved by connecting the Google Accelerator unit to the Raspberry Pi unit via a USB port which can process 12 frames per second. The Coral USB Accelerator adds an Edge TPU coprocessor to the system, which includes a USB-C socket to connect with a host computer for accelerated Machine Learning (ML) inferencing. The onboard Edge TPU is a small Application Specific Integrated Circuit (ASIC) designed by Google that accelerates TensorFlow Lite models in a power-efficient manner. This can perform 4 trillion operations per second using only 2 watts of power, as explained in the block diagram.
Moreover, software of Raspbian, Python, OpenCV, TensorFlow, and Keras is installed and used for the execution of the codes with details as follows:

- To program the operating system of the Rasberry PI and its graphical user interface.
- Python is used to control the manoeuvring function for the DC motors of the built prototype
- OpenCV package is utilized to process the image captured by the camera.
- TensorFlow is an end-to-end open-source machine learning platform compatible with Google Accelerator.

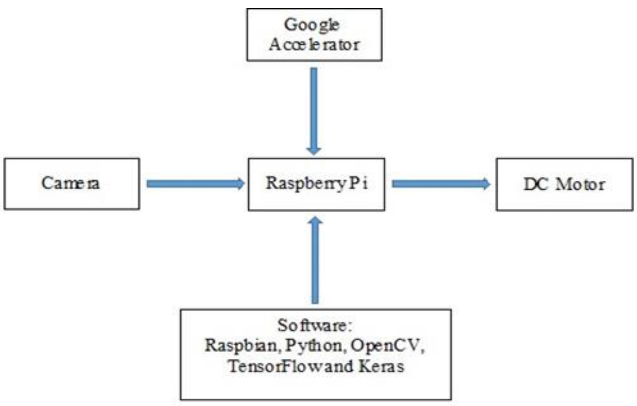


Fig. (1). Main components of the proposed autonomous car prototype.

One of the main functions of the software is to pre-process the captured image from the camera. Moreover, for essential feature extraction such as map values, kernel convolution techniques are used, given by the following equation 1

$$G[m,n] = (f * h)[m,n] \\ = \sum_j \sum_k h[j,k]\, f\,[m-j, n-k] \tag{1}$$

where the input image is denoted by f and kernel by h, with rows and columns of the resulting matrix marked with indices m and n, respectively. The convolutional process starts with setting a filter, which will be applied to the image, one pixel at a time.
Then, the following filtering Equation 2 is used to minimize the size of the considered image for efficient processing by converting the image into several smaller layers

$$[n_I n_I n_C] * [f_I f_I f_C] \\ = [n_I n_I n_C] * [f_I f_I f_C] \\ = \left[\left[\frac{n+2p-f}{s} + 1\right], \left[\frac{n+2p-f}{s} + 1\right], n_f \right] \tag{2}$$

with image size (n), filter size (f), number of channels in the image (nc), padding (p), stride (s), and number of filters [nf]. In order to process more images in less time, each colored image is divided into three layers, and each layer of the colored image consists of three primary colors. As a result, utilizing a higher number of filter increases is directionally proportional to layers and inversely proportional to the dimensions of these layers. Fig. 2 provides an example of how the filter is applied to reduce the dimensions of the RYB color image.

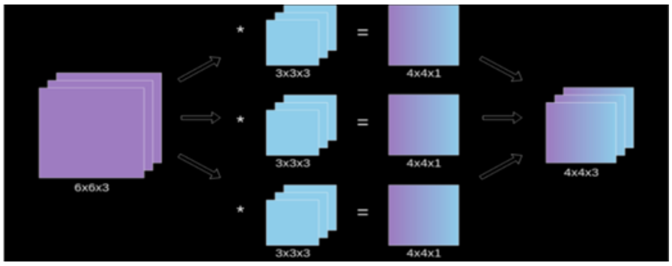


Fig. (2). Filtering to reduce the dimensions of a color image [22]

Simulation techniques are an essential part of electronic circuit design, providing insight into a designed circuit's operation prior to its being built. It also provides invaluable input to design optimization and verification and can highlight problems that result from component and interconnect placement. In this work, the software Proteus is used to design and simulate the autonomous car prototype's electronic circuit. Fig. 3 presents the electronic circuit design and simulation of the servo motor driver that controls the motor of the right and left wheels of the prototype.

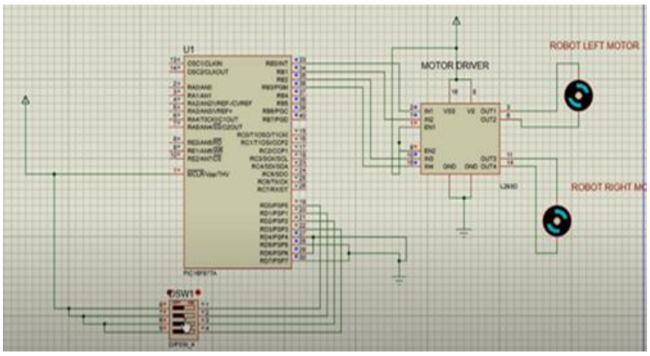


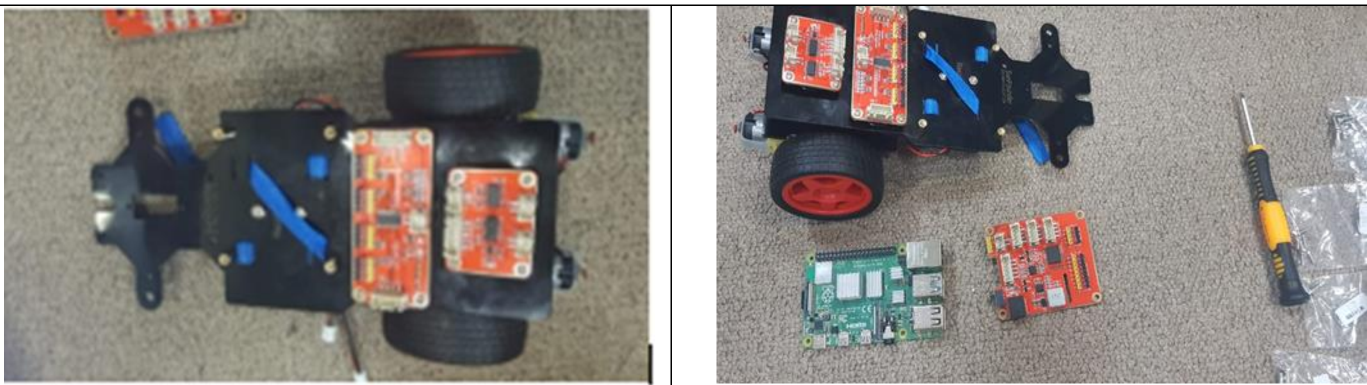
Fig. (3). Electronic circuit design for the driver of the back wheel motors

3.2 Assembly of the prototype:

Now we are ready to assemble the AV prototype. There are two main pieces of hardware that need to be integrated to build the prototype. First is the Raspberry Pi, which acts as the brain of the car, with features such as USB inputs, GPIO pins, and its connectivity to wireless networks through Bluetooth and WIFI, shown in Fig. 4.

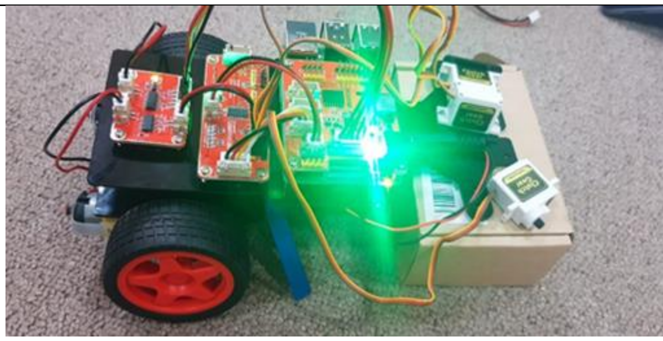
The second hardware is the robotic car components from the body chassis, wheels, gear motor, servo motor, battery chase, camera, and PCB board, i.e. Robot Hat, motor drive module, and PCA9865. Fig. 5 presents all of the steps to assemble the prototype.

Fig. (4). Layout of Raspberry pi

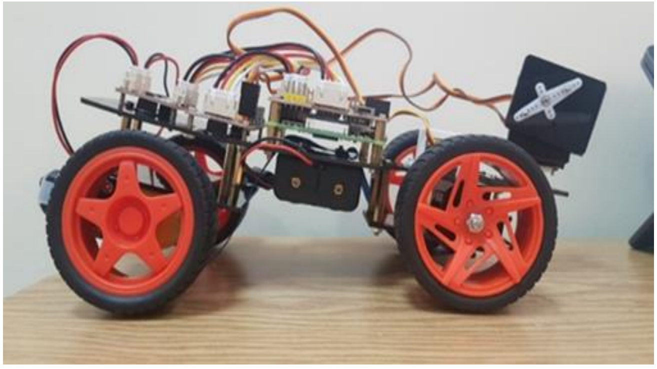


(a) (b)

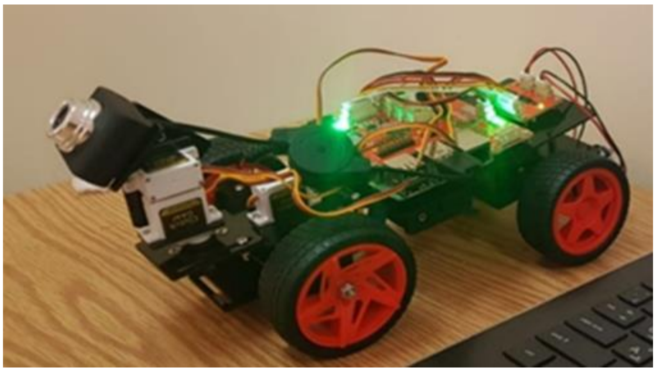
(c)



(d)



(e)



(f)

Fig. (4). Assembly steps for the prototype. (a) Assembling the main chassis and rear motors. (b Preparing Raspberry and Hat. (c) Installing Raspberry Pi's Hat. (d) Preparing the front wheels. (e) Installing the front wheels. (f) Adding the camera.

3.3 Route Optimization and Control:

This subsection present and explain the components of the robotic car that are used for route control and optimization.

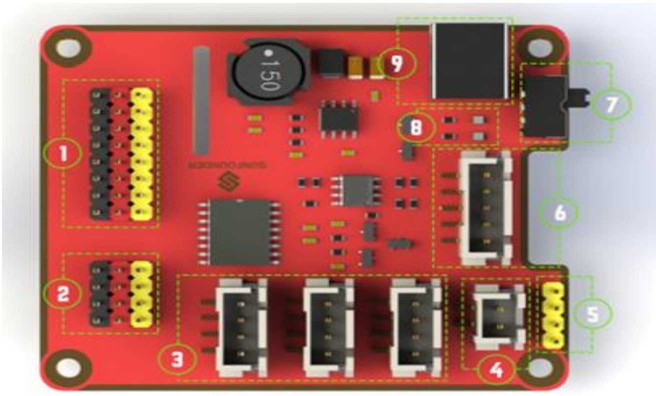


Fig. (6). Robot Hat

As shown in Fig. 6, a Robot Hat is used to design the 40-pin GPIO Raspberry PI, which can power Raspberry PI using a lithium 18650 battery. Moreover, Robot Hat can manage power even if the Raspberry got its power from an external source using a Type-c cable. Furthermore, it contains an integrated circuit PCF8591 which is I2C communication, a protocol used to send signals from Raspberry PI to motors. Moreover, the PCA9865 16-channel 12-bit I2C bus shown in Fig. 7 is the Pulse Width Modulation (PCM) driver used to control servo motors attached to the front wheels of the car.

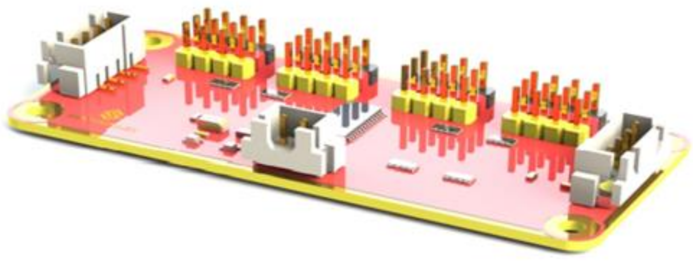


Fig. (7). PCA9865

Also, a motor driver module with the following specifications shown in Fig. 8 is used:

i. Power motor control port: include pins for supplying the chip and the motors

ii. PWM input for motors: PWM signal input for adjusting the speed of the two motors

iii. Motor Output Port: output port for two motors.

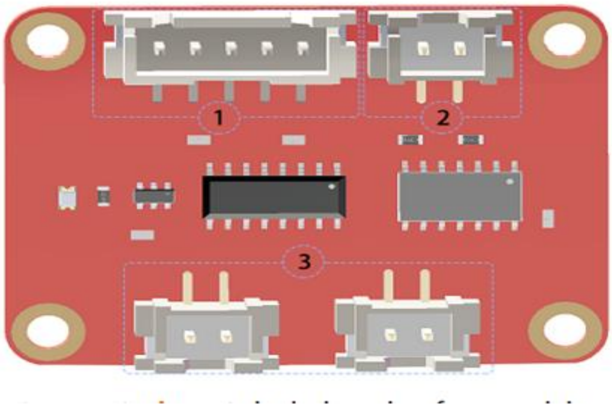


Fig. (8). Motor Driver Module

Fig. (9). Clutch Gear SF006C for Servo Motor

The clutch gear shown in Fig. 9 is a digital servo with a DC motor inside the core with specifications given in Table II. The steering gear reducer will automatically clutch and protect the product from damage and normal load with applied load. Fig. 10 on the other hand depicts the DC gear motor used to control the speed of the clutch gear.

Finally, a wide-angle USB camera that acts as the input component for the robotic car with the function to capture the video image of the lane is presented in Fig. 11.

Table (2). Specification of motor

Item	V=4.8V	V=6.0V
Consumption (No Load)	$\leq 50mA$	$\leq 60mA$
Stall Current	$\leq 550mA$	$\leq 650mA$
Rated Torque	$\geq 0.6kgf.cm$	$\geq 0.7kgf.cm$
Max. Torque	$\geq 14kgf.cm$	$\geq 1.6kgf.cm$
No Load Speed	$\leq 0.14sec/60^{\circ}$	$\leq 0.12sec/60^{\circ}$

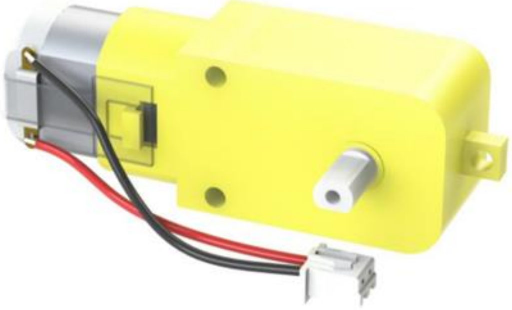


Fig. (10). Dc Gear Motor

Fig. (11). USB Camera

4. Development of Lane Detection Algorithm

This section explains the development of the lane detection algorithm programmed in our built autonomous car prototype. Lane detection is an integral part of an advanced driving assistant system. The accurate identification of lanes is the foundation of driving assistant systems like lane departure warning systems and lane change maneuver estimation.

The lane detection algorithm uses the snapshot image from the wide-angle camera. Based on the output of the lane detection algorithm, the corresponding angle of the steering wheel is computed and sent to the front server motor to ensure the prototype autonomously maneuvers the built prototype car within the lab-scale track.

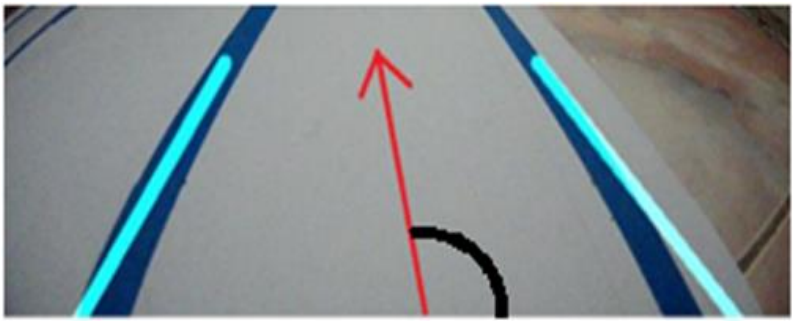


Fig. (12). Two lines generated as an output from the lane detection algorithm

Fig.12 illustrates the two lines generated from the executed lane detection algorithm. Each line can be represented using the (X, Y) coordinates of the pixel at both ends of the line as in Eq. 3 and 4

$$X_k = [x_{1,k} \quad x_{2,k}] \tag{3}$$

$$Y_k = [y_{1,k} \quad y_{2,k}] \tag{4}$$

where subscripts 1 and 2 represent the left and right lines, respectively. As a result, the objective of the lane detection algorithm is to obtain these four coordinates, two for each line. To produce the two lines, image processing techniques are utilized to process the snapshot image to reduce computational complexity. Firstly, the RGB snapshot image is converted to one layer grayscale image to reduce the size of the image and reduce the complexity of the algorithm. Second, the noise is removed from the grayscale image by using Gaussian blurring technique [23]. This technique enhances the difference between pixel, which improve the detection of the lane. Third, canny edge detection [24] is used to extract useful structural information, i.e., the line structure of the lane, which also significantly decreases the quantity of data to be processed. Finally, the lane structure and the four coordinates of the left and right lines in the image are identified by using the Hough transform method [25-27].

Once these coordinates are obtained, the next task is to determine the direction for the autonomous car prototype. In Fig.12, the direction for the autonomous car can be determined by generating the third lines represented using the (X, Y) coordinates as in Eq. 5, 6, 7, and 8

$$\mathrm{X_{avg}} = [\mathrm{x_{1,avg}} \quad \mathrm{x_{2,avg}}] \tag{5}$$

$$\mathrm{Y_{avg}} = [\mathrm{y_{1,avg}} \quad \mathrm{y_{2,avg}}] \tag{6}$$

where for $i \in \{1,2\}$

$$\mathrm{x_{i,avg}} = \frac{\mathrm{x_{i,l}} + \mathrm{x_{i,r}}}{2} \tag{7}$$

$$\mathrm{y_{i,avg}} = \frac{\mathrm{y_{i,l}} + \mathrm{y_{i,r}}}{2} \tag{8}$$

By using the computed (X_{avg}, Y_{avg}), the direction can be represented using the steering angle θ, which can be computed using Eq. 9

$$\theta = \tanh^{-1}\left(\frac{\mathrm{y_{1,avg}} - \mathrm{y_{2,avg}}}{\mathrm{x_{1,avg}} - \mathrm{x_{2,avg}}}\right) \tag{9}$$

Algorithm 1: Lane detection and steering angle algorithm

Input: Image S, m x n pixels

$$\mathrm{S} -> \begin{bmatrix} s_{1,1} & \cdots & s_{1,n} \\ \vdots & \ddots & \vdots \\ s_{m,1} & \cdots & s_{m,n} \end{bmatrix}$$

Output : (1) The coordinate (X_k, Y_k) of the lane in the image S, where $k \in \{l, r\}$ represents the left and right lines,

(2) The steering angle θ

Procedure:

If (S ->empty),

return (X_k, Y_k) empty;

else

P = grayscale (S); change the image to one layer of grayscale image

P = GaussianBlur(P); remove image noise

P = Canny(P); apply Canny Edge Filter

(X_k, Y_k) = Hough Line (P); apply Hough Transform

for $i \in \{1,2\}$

$$x_{i,avg} = \frac{x_{i,l+}x_{i,r}}{2}$$

$$y_{i,avg} = \frac{y_{i,l+}y_{i,r}}{2}$$

$$\theta = \tanh^{-1}\left(\frac{y_{1,avg} - y_{2,avg}}{x_{1,avg} - x_{2,avg}}\right)$$

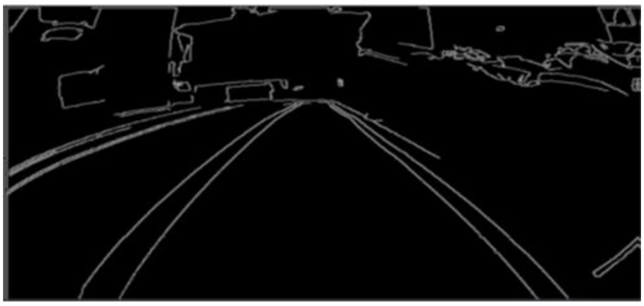
end

return (X_k,Y_k) and θ

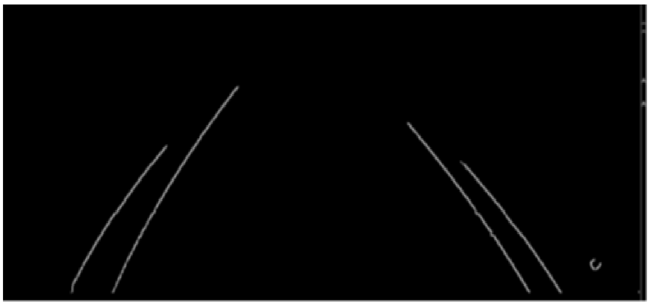
end

(a)

(b)

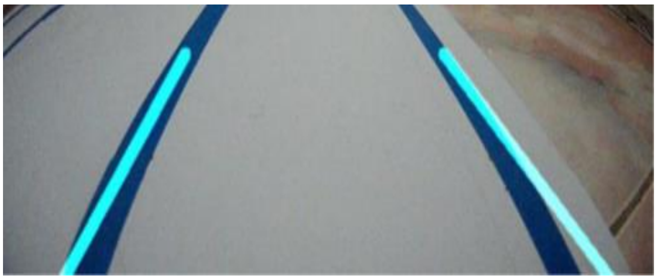


(c)



(d)

(e)



(f)

Fig.(13): Output of the processed image to detect the lane using Algorithm 1. (a) The original image from the wide-angle camera. (b) The blurred and grayscale image. (c) The canny image of the grayscale image with a clear line structure. (d) The cropped canny image of the lanes.

(e)The two left, and right lines from Hough transform. (f) The estimated left and right lines identify the lane in the original image.

5. Results and Analysis

This section shows the result of the processed images from Algorithm 1 given in Fig.13. Algorithm 1 is implemented using Python programming with the OpenCV library. The three fundamental techniques of the Gaussian blurring, canny edge detection, and Hough transform are executed using the build functions in the OpenCV library optimized for computer vision applications. The three layers RGB original image in Fig. 13(a) are firstly converted to the first layer of the Gaussian blurred grayscale image shown in Fig.13(b). When applying the cv.canny() function, the image after executing the canny edge detection is given in Fig.13(c). Based on the canny image, we can see that the lane is situated at the bottom of the image, which can then be cropped, as shown in Fig.13(d). This observation is very important as it illustrates the importance of the wide-camera tilting in the prototype of the autonomous car. The camera should be tilting downward towards the road more, such that fewer edges are generated at the top part of the canny images, which existed because of the background of the original image. One practical solution to reduce the complexity of Algorithm 1 is to begin cropping the bottom part of the three layers RGB image first before converting it to a grayscale image [28]. As a result, the Gaussian blurring and canny edge detection are only implemented to the smaller size of the grayscale image. Next, applying the Hough transform function, HoughLinesP(), to the canny image of the lane, Algorithm 1 produces the estimated left and right lines depicted in Fig.13(e). Furthermore, using these two estimated lines, the steering wheel angle is computed, and then feedback to the front wheel motor for front maneuver.

A lab-scale track is used to verify and test the performance of the assembled autonomous car prototype. Fig.14 shows the circuit containing several types of line segments with different angles and bends to imitate actual lanes in the real application. Using this track, the testing of our autonomous car prototype is conducted at the speed of 30 meters per second. At this speed, we found that a 13-frame-per-second video is sufficient for our AI lane detection algorithm to work and subsequently safely maneuver the car. Fig.15 captured some of the lane detection output of the test at several locations in the tracks.

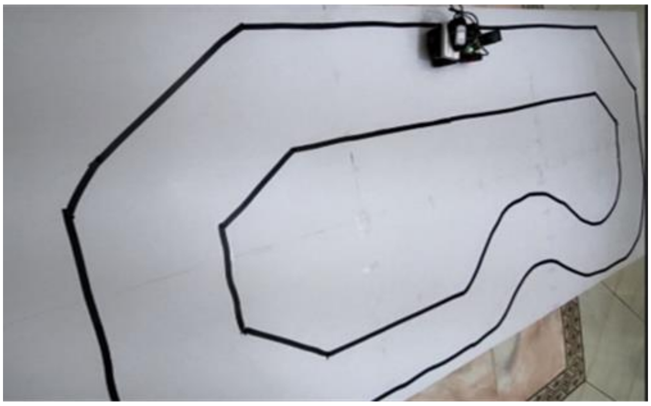


Fig. (14): Lab-scale track for testing the built autonomous car prototype.

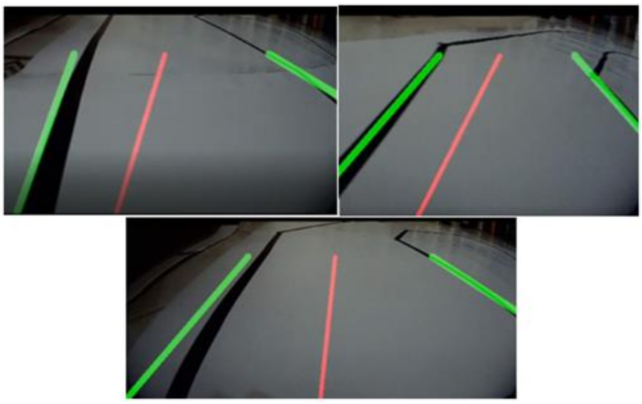


Fig.15: Output of the generated lane from Algorithm 1 for several locations during the testing.

6. Conclusions and Future Works

In this article, we have conducted detailed and comprehensive real-time verification and testing on autonomous driving in the laboratory using an assembled autonomous car prototype. The complete hardware description and the procedure for designing and developing the autonomous car prototype are presented. Utilizing the computing and processing capability of the Rasberry Pi and Google Accelerator, the designed prototype of the autonomous car is equipped with the machine learning-based lane detection algorithm. Utilizing a 13 frame per seconds image from the integrated camera, the assembled prototype can detect the line segments of the lab scale circuit and act accordingly by calculating the steering angle for autonomous driving maneuvers. Extending this testing and verification of the developed autonomous car prototype at different speeds will be the subject of interest for future work. To conclude, this work provided valuable insights and practical experience in developing autonomous driving technologies and dealing with the complexity of integrating software and hardware components.

7. References

[1] J. A. Guerrero-Ibanez, S. Zeadally, and J. Contreras-Castillo, “Integration challenges of intelligent transportation systems with connected vehicle, cloud computing, and Internet of Things technologies,” IEEE Wireless Commun., vol. 22, no. 6, pp. 122–128, Dec. 2015.

[2] WHO. (2016). The Road Traffic Death Rate by Who Region and Income Level. [Online]. Available:
http://www.who.int/gho/road_safety/en/

[3] M. Gross, “A planet with two billion cars,” Current Biol., vol. 26, no. 8, pp. R307–R310, 2016.

[4] K. Henry, Traffic Fatalities Up Sharply in 2015, NHTSA, Washington, DC, USA, 2016.

[5] R. Hussain and S. Zeadally, "Autonomous Cars: Research Results, Issues, and Future Challenges," in IEEE Communications Surveys & Tutorials, vol. 21, no. 2, pp. 1275-1313, Second quarter 2019.

[6] Y. Zhang, Z. Lu, X. Zhang, J. H. Xue and Q. Liao, "Deep Learning in Lane Marking Detection: A Survey," in IEEE Transactions on Intelligent Transportation Systems, vol. 23, no. 7, pp. 5976-5992, July 2022

[7] M. Jefri Muril, N. H. Abdul Aziz, H. A. Ghani, and N. A. Ab Aziz, "A Review on Deep Learning and Nondeep Learning Approach for Lane Detection System," 2020 IEEE 8th Conference on Systems, Process and Control (ICSPC), 2020, pp. 162-166, doi: 10.1109/ICSPC50992.2020.9305788.

[8] K. Muhammad, A. Ullah, J. Lloret, J. D. Ser, and V. H. C. de Albuquerque, "Deep Learning for Safe Autonomous Driving: Current Challenges and Future Directions," in IEEE Transactions on Intelligent Transportation Systems, vol. 22, no. 7, pp. 4316-4336, July 2021, doi: 10.1109/TITS.2020.3032227.

[9] M. Bai, G. Mattyus, N. Homayounfar, S. Wang, S. K. Lakshmikanth, and R. Urtasun, “Deep multi-sensor lane detection,” in Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst. (IROS), Oct. 2018, pp. 3102–3109.

[10] Y. Y. Ye, X. L. Hao, and H. J. Chen, “Lane detection method based on lane structural analysis and CNNs,” IET Intell. Transp. Syst., vol. 12, no. 6, pp. 513–520, Aug. 2018.
[11] W. Song, Y. Yang, M. Fu, Y. Li, and M. Wang, “Lane detection and classification for forward collision warning system based on stereo vision,” IEEE Sensors J., vol. 18, no. 12, pp. 5151–5163, Jun. 2018.
[12] J. Li, X. Mei, D. Prokhorov, and D. Tao, “Deep neural network for structural prediction and lane detection in traffic scene,” IEEE Trans. Neural Netw. Learn. Syst., vol. 28, no. 3, pp. 690–703, Mar. 2017.
[13] N. J. Lavigne and J. A. Marshall, “A landmark-bounded method for large-scale underground mine mapping,” J. Field Robot., vol. 29, no. 6, pp. 861–879, 2012.
[14] N. J. Lavigne, J. A. Marshall, and U. Artan, “Towards underground mine drift mapping with RFID,” presented at the 23rdCan. Conf. Elect. Comput. Eng., 2010.
[15] M. Azizi and E. Tarshizi, "Autonomous control and navigation of a lab-scale underground mining haul truck using LiDAR sensor and triangulation - feasibility study," 2016 IEEE Industry Applications Society Annual Meeting, 2016, pp. 1-6.
[16] V. Androulakis, J. Sottile, S. Schafrik and Z. Agioutantis, "Concepts for Development of Autonomous Coal Mine Shuttle Cars," in IEEE Transactions on Industry Applications, vol. 56, no. 3, pp. 3272-3280, May-June 2020, doi: 10.1109/TIA.2020.2972786.
[17] Fahez, Kenway; Economic Impact of self-driving cars July 25, 2019 [Online]. Available: https://www.bosch.com/stories/economic-impact-of-self-driving-cars/
[18] D. Edward, F. Morgan; Safety Analysis of Free driving cars, Oct, 2017
[19] A. Devis, Sayan Dey; Safety Analysis of Autonomous Vehicles, Sep. 2018
[20] S.Harv, Social impacts of autonomous cars, [Online]. Available: https://www.semanticscholar.org/paper/A-Brave-New-World/
[21] By H. Marsotn; Levels of Autonomous cars; [Online]. Available:https://www.govtech.com/fs/automation/autonomous-vehicles-to-have-huge-impact-on-economy-tech-sector.html
[22] Piotr Skalski; Gentle Dive into Math Behind Convolutional Neural Networks; [Online]. Available:
https://towardsdatascience.com/gentle-dive-into-math-behind-convolutional-neural-networks-79a07dd44cf9
[23] J. Immerkaer, "Use of blur-space for deblurring and edge-preserving noise smoothing," in IEEE Transactions on Image Processing, vol. 10, no. 6, pp. 837-840, June 2001, doi: 10.1109/83.923280.

[24] P. Bao, Lei Zhang, and Xiaolin Wu, "Canny edge detection enhancement by scale multiplication," in IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 27, no. 9, pp. 1485-1490, Sept. 2005, doi: 10.1109/TPAMI.2005.173.
[25] Mark Haidekker, "The Hough Transform," in Advanced Biomedical Image Analysis , IEEE, 2011, pp.211-235, doi: 10.1002/9780470872093.ch7.
[26] Ian Bogost; Will Robocars Kick Humans Off City Streets?; [Online]. : https://www.theatlantic.com/technology/archive/2016/06/robocars-only/488129/
[27] Mathhew Joseph: How Driverless Cars Will Change the Feel of Cities. [online]: https://www.theatlantic.com/technology/archive/2017/11/life-in-a-driverless-city/545822
[28] Lei Zhang: Self Driving Car-A new Legal Frontier? [online]:https://www.whitecase.com/publications/article/self-driving-car new-legal-frontier

تطوير سيارات ذاتية القيادة باستخدم Google Accelerator و Raspberry Pi لاكتشاف المسار القائم علي التعليم الآلي

ريان حمزة السيسي، محمد عزير، نادر عودة، يوسف مبروك، عمر النواجحة، أرشد كريمبو فالابيل، حافظ عبد الواجد
كلية الهندسة، قسم الهندسة الكهربائية، الجامعة الإسلامية بالمدينة المنورة، المملكة العربية السعودية
مروان هادري عزمي
كلية الهندسة، قسم الهندسة الكهربائية، الجامعة التكنولوجية الماليزية، ماليزيا

(قدم للنشر في 2023/1/4 م ، وقبل للنشر في 2023/9/1 م)

ملخص البحث. المركبات ذاتية القيادة هي السيارات الذكية للمستقبل، والتي من المتوقع أن تكون بدون سائق، و فعالة، وتتجنب الإصطدامات كسيارات مثالية. يعد تعقيد البرامج وتحليلات البيانات في الوقت الفعلي والتحقق والإختبار من بين التحديات الأكثر أهمية في تكنولوجيا القيادة الذاتية. تقدم هذه المقالة خبرة عملية ورؤية قيمة للتحديات المذكورة أعلاه من خلال تطوير نموذج أولي لسيارة ذاتية القيادة على نطاق المختبر. تمت بالتفصيل مناقشة الوصف الكامل للسيارة، بما في ذلك مواصفاتها الفنية، وإجراءات تصميم الأجهزة والبرامج، ودائرة الإختبار على نطاق المختبر. تم تجهيز النموذج الأولي المطور بخوارزميات اكتشاف المسار القائمة على التعلم الآلي. يتم التحقق من آداء خوارزميات اكتشاف المسار المثبتة عن طريق اختبار النموذج الأولي للسيارة باستخدام دائرة المقياس المعملي.