Research Article

Amar Shukla, Ankit Verma, Hussain Falih Mahdi*, Tanupriya Choudhury*, and Thipendra Pal Singh

Path reader and intelligent lane navigator by autonomous vehicle

https://doi.org/10.1515/pjbr-2022-0117
received November 25, 2022; accepted March 30, 2023

Abstract: Internet of Things (IoT) is a physical network of physical devices, such as widgets, structures, and other objects, which can store program, sensors, actuators, and screen configurations to allow the objects to assemble, control, display, and exchange data. The aim of this research was to develop an autonomous system with automated navigation. Using this approach, we are able to make use of deep neural networks for automatic navigation as well as the identification of pot holes and road conditions. Additionally, it displays potholes in traffic and the correct lane on the screen. The system stresses how important it is to select the path from one node to the next.

Keywords: convolution neural network, pavement condition, congestion condition, pothole condition, traffic light condition

*** Corresponding author: Hussain Falih Mahdi,** Computer and Software, College of Engineering, University of Diyala, Baqubah, Iraq, e-mail: hussain.mahdi@ieee.org

*** Corresponding author: Tanupriya Choudhury,** CSE Department, Symbiosis Institute of Technology, Symbiosis International University, Pune, Maharashtra, 412115, India; School of Computer Science, University of Petroleum and Energy Studies (UPES), Dehradun, 248007, Uttarakhand, India; CSE Department, Daffodil International University, Daffodil Smart City, Birulia 1216, Bangladesh; CSE Department, Graphic Era Hill University, Dehradun, 248002, Uttarakhand, India, e-mail: tanupriya1986@gmail.com, tanupriya@ddn.upes.ac.in, tanupriya.choudhury@sitpune.edu.in

Amar Shukla: School of Computer Science, University of Petroleum and Energy Studies (UPES), Dehradun, 248007, Uttarakhand, India, e-mail: ashukla@ddn.upes.ac.in

Ankit Verma: Wartin Labs Technologies LLP, H-187 WorkWings, Noida, Uttar Pradesh 201301, India, e-mail: avankitverma.2011@gmail.com

Thipendra Pal Singh: School of Computer Science Engineering and Technology, Bennett University, Greater Noida, Uttar Pradesh, 201310, India; School of Computer Science, University of Petroleum and Energy Studies (UPES), Dehradun, 248007, Uttarakhand, India, e-mail: thipendra@gmail.com, tpsingh@ddn.upes.ac.in

1 Introduction

Globally, nearly 3,287 people die daily due to car accidents. One of the major reasons behind this is the driver sleeping in the car or trying to stop the car when it is at a very high speed. Just as the industrial revolution freed humanity from physical drudgery, artificial intelligence (AI) has the potential to free humans from mental drudgery.

To reduce the number of accidents that occur on a daily basis, it is critical to reduce the amount of human error; it will be extremely fascinating if all we have to do is fit our destination into our schedule and keep working until we reach our goal without making any mental or physical mistakes.

The use of a self-driving car can not only prevent accidents, but also provide self-relief for minor daily activities.

The Internet of Things (IoT) is a network of standard items such as motorized vehicles, the Internet, televisions, and other contraptions that are specifically linked together, enabling new types of correspondence between things and people as well as between things themselves. Building the IoT has advanced in recent years, adding another estimation to the universe of data and correspondence movement estimations.

Home automation or Smart Homes can be portrayed as presentation of improvement inside the home condition to give settlement, solace, security, and essential capacity to its inhabitants. In addition, figuring out how to improve home condition can give broadened singular satisfaction. With the presentation of the IoT, the examination and use of home mechanization are getting progressively standard.

Self-driving cars can help persons who are unable to drive on their own due to infirmities such as blindness. According to studies, a driver's mistake is cited as a reason in 94% of crashes, and self-driving vehicles can help eliminate driver error. Because it does not require rest like people and can operate constantly for hours, it can increase traffic congestion, save fuel, and reduce greenhouse gas emissions.

Reduced travel time: Travel by a car should be safe whether the car is going slowly or rapidly. Higher speeds

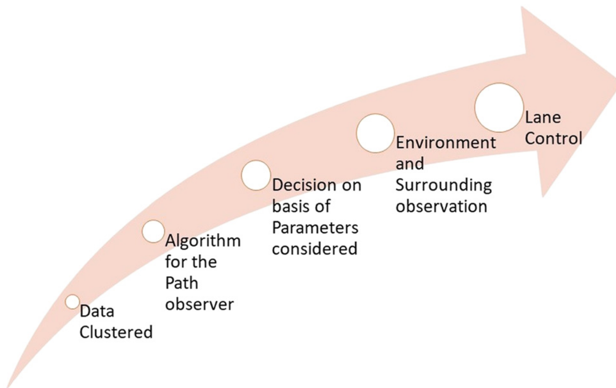

Figure 1: Flow diagram of the proposed process.

are likely to be possible, because computers will eliminate human error as a cause of accidents. Less expensive insurance: If car insurance companies join the car movement, your rates could decrease significantly. Risk allocation depends more on the vehicle than the driver, so you can expect insurance premiums to go down. Redirecting our emergency services' efforts and resources will allow us to redirect our emergency services to where they are needed.

In order to further approach with the process, the structural diagram will show the entire basic procedure of the progress of the study.

The Figure 1 contains the basic flow information where the data are gathered from the environment and then these data are further processed and resulted in the different maps of path observer, then the suitable path is chosen, which taking the consideration of the lane tracking and then the control is there while moving in the lane through reading of the environment.

2 Literature review

We can better examine and grasp new research projects if we have a broader view of the algorithms that can be deployed alongside existing ones.

How is technology helping transform the world? There have been great changes in the technology of self-driving and automatic cars since the 1920s when the first ever radio-controlled vehicle/car was introduced. In the years following to this, many automatic electric cars were seen on the roads that were powered by embedded circuits, and by 1960 automatic cars which had same electronic guiding system came in the picture. In the 1980s, vision-guided autonomous vehicles (AVs), which were a great achievement of the technology at that time, were introduced. The similar or slightly modified forms of this technology are still being used today [1].

We can increase people's trust in driverless cars. Although public trust is crucial to widespread adoption, this is the main obstacle. The goal of this study is to determine which variables are most important in increasing the use of driverless cars. The study found that a vehicle's ability to meet performance expectations and its reliability were important adoption determinants, according to quantitative research. The major concerns that were raised had to do with privacy, such as location, security, and the like [2].

Robotic platforms allow for incremental development of manual processes, and path planning has become a critical area even if the environment inside and outside the building is unknown. Our challenge is to invent ways to make the algorithms as intelligent or pre-established as possible, and to arrive at our destination in the most efficient manner. One of the most significant issues in this field is finding a path that is free of static and dynamic obstacles. In this article, a methodology is proposed to cover the critical points and reach the initial key point in a dynamic environment with the implementation details of the robotic platform. The primary computation is taking place inside the Raspberry Pi B + module, and other modules include compass, wheel encoders, and ultrasonic sensors [3].

Convolution neural networks (CNNs) are used to create a self-driving automobile based on monocular vision. The authors of this research sought to develop a method for modeling raw input photos to a specific steering angle predicted by the CNN. CNN was trained using data acquired from the vehicular road/platform Raspberry Pi 3 and a front connected camera, as well as photos of the road and time-synchronized steering wheel angle gained through manual driving. Whether road markers were present or not, the speed reached was 5–8 km/h [4].

CNN maps raw images that have been taken from a front attached camera on the vehicle directly to the steering wheel. This system works very well on traffic-filled roads with or without markings on the road. The only training data provided were human steering angles which were then used to predict the particular angle at which the car should be steered. This system has smaller networks and better performance as minimal number of processing steps are required and this performs better as components can self-optimize [5].

Efficient and extremely compact CNNs were generated in their study, which makes use of a novel sparse connection topology. Because of the sparseness of inter layer filter dependencies, this results in a significant reduction in processing power consumption as well as a reduction in the number of parameters required, without sacrificing CNN accuracy. The article's findings indicated that the system's accuracy was greater than that of CNN's cutting-edge architecture. When compared to previous models, the model required 40% less parameters and was 31% faster on the CPU, while preserving greater or similar efficiency [6].

By comparing different models of CNN while implementing them on a self-driving car, they test which model is the best and proves to be the most efficient in a simulated environment. The CNN has been trained by the manually obtained data by driving a car and using previously obtained data from end-to-end deep learning techniques. When training is done, the CNN is tested in the driving simulator by checking its ability to reduce the distance traveled by the car to go to the center, heading error, and root mean square error. The conclusion drawn was that adding long-short term memory layers in CNN produced better steering of the car which took into account the previously predicted value by CNN and not just the new predicted value or a single instance [7].

Level 2 automatic cars are implemented by the authors by taking the inputs from the front-facing camera on the vehicle and feeding them as steering inputs. The network requires minimal human intervention as maximum variable features are learnt from the camera inputs themselves. The data set used is from NVidia and Udacity, and when the CNN is given real inputs it can adapt to real environment driving given a controlled environment. The setup consists of an ultrasonic sensor that will detect obstacles and an red green blue depth camera working at 10 HZ which outputs a steering angle [8].

OShea and Nash [9] have described the various Artificial Neural Networks (ANNs) and their types, most significantly CNN. CNNs are mostly used to solve difficult image-driven tasks that require pattern recognition. These have precise and simple architecture and are easy to implement; this study gave great insight into ANN and especially CNN.

Unlike typical cars [10], self-driving cars can park anywhere. Instead, they can drive, fly, or cruise (circle around). Vehicles are enticed to work together to clog roads. According to San Francisco's downtown data, self-driving cars might roughly treble the number of vehicles entering, leaving, and inside cities. Planned travels extend due to parking and cruising. Parking subsidies may have the unintended consequence of worsening congestion. According to the study's conclusions, the introduction of congestion pricing in cities in the near future will be heavily reliant on AVs. Congestion pricing should incorporate a time-based penalty as well as a distance- or energy-based fee to internalize various externalities associated with driving.

Vehicle speed, eye-gazing, and hand gestures [11] all reveal a driver's purpose and attentiveness. The appearance and behavior of a car indicate to passengers whether the driver is likely to pay attention to the road. This research aims to enable passengers to comprehend and express their autonomous car awareness and intent to pedestrians, which might be difficult if explicit interfaces are avoided. The idea of an AV's mission and awareness to pedestrians was conducted. Four user interface prototypes were designed and tested on Segway and cars. It is possible to taste, touch, smell, and hear things out in the environment and combine the senses to do so.

Deep learning-based vehicle [12] control systems are becoming more common. Before building a vehicle controller, engineers must rigorously test it under various driving conditions. Recent improvements in deep learning algorithms promise to solve challenging non-linear control problems and transfer knowledge from earlier events to new situations. These significant advances have gotten little attention. This study uncovers current and valuable information on intelligent transportation systems, which is vital for the field's future. Control and perception are interwoven in this research.

Modern autonomous [13] driving systems rely on historical mapping. Although prevalent in cities, precise maps are difficult to develop, preserve, and transmit. Rural areas have high turnover, making exact mapping challenging. A self-driving automobile was tested in the countryside to ensure its functionality. The car uses its local sensing system to detect its road conditions. This system calculates a car's distance and speeds through recursive residual filtering and odometer, allowing it to navigate complex road networks easily.

This AI product features [14] should assist in minimizing traffic congestion, road accidents, and social exclusion. Future human transportation will have AI-powered drivers. Despite its apparent benefits, people are still wary about driverless cars. People's trust in machines may help build autonomous systems. This study assesses the acceptance of autonomous technologies. That is, future studies should examine user trust and approval. Changes to the roadway and subsurface infrastructure impacted traffic, community attitudes and concerns, potential transferable behaviors and requests, other business models, and strategy. Malaysian law enforcement agencies must identify critical elements to investigate AV manipulators' conspiracy claims appropriately.

A family of nonlinear [15] under-actuated systems was found to be soluble. The vehicle's lateral dynamic control system incorporates the usage of forwarding and backward controls. Even if the findings of theoretical studies

Table 1: Detailed study of the current models with various parameters

Study	Model	Approach	Issues and challenges	Advantages
[17]	AVE (LIDAR)	Object classification and fusion approach proceeded	Object classification and fusion approach proceeded	Better fusion approach for identification of object
[18]	Multi-Level Cloud System (AV)	Vanet-based approach, where tactile internet over the radio active network	Validity testing in the real-time environment	Model induces the new mode of cloud era to establish the new idea
[19]	Taxonomy (AV)	Tested the relation of driver and AV, by using society of automotive engineers levels	Testing environment should have multiple modes of testing environment for validation	Excellent mode of problem identification and executed well
[20]	Optimal Scheduling (AV)	Used mixed integer linear programming for optimized scheduling of vehicles	Multifunctionality with more environment can be taken for consideration	Generalized social problem is well addressed with modern technology
[21]	AV Impact (AV)	Chosen the calibrated models and evaluated the effective vehicle percentage	Effectiveness can be improved and analysis parameter could be increased more	This model provided the optimal solution in all the scenario of impact
[22]	Autono Vi-Sim	Contains rapid development and testing of vehicle configurations	Marginal tendency of testing could be updated	Good model for approach to test and develop
[23]	Collision Avoidance (AV)	Novel emergency steering-control strategy approached using decision making and motion control layer	Validation and testing environment in the multiple parameters can be adopted	Emergency steering control could avoid lot of accident as we face in this era
[24]	AV Path tracking	It proceeded with predictive model control, to control the vehicle deviation and balancing	Testing and accuracy of validation can be improved	Model satisfies the objective controlling deviation and balancing of the vehicle
[25]	Self-Diagnosis System	They approached the IoT-based learning model diagnosis of data controlling	Effectiveness of the model can be improved	Model contains advanced mode of AI and IoT-based features for the solution of the problem

on AV lateral control can be applied to multiple circumstances, the results can still be used in other applications.

In the study, the performance of the closed-loop system was compared to that of a typical human driver.

AVs will completely [16] revolutionize ground transportation. In the future, new cars that can judge and drive themselves are expected to replace traditional cars. Sensors help self-driving cars sense and comprehend their surroundings, whereas 5G allows them to sense and comprehend distant environments. Local perception, like human perception, can be helpful for short-range vehicle control. Despite the fact that people's perspectives have broadened, they can still prepare for the future and drive with greater caution while adhering to a set of norms (safety, energy management, traffic optimization, comfort). Faults can emerge as a result of background noise, ambient circumstances, or manufacturing problems, regardless of how well an electronic sensor has previously worked. The most practical solution to the shortcomings of individual sensors is for them to be integrated. The goal of this research is to talk about performance optimization for local automated driving systems in automobiles.

Table 1 contains the multifunctionality of the AV and the strategic issues and challenges which can really be taken as the premium objective to work in this article are as follows:

- An AV with a specific algorithm should contain more amount of parameters for validation and testing.
- The involvement of the Deep Learning and AI model can be enriched more to make the system more agile and updated.
- It should contain more fusion-based approaches for innovative vehicle systems.
- Effectiveness and accuracy in the fusion approaches can be increased.

3 Problem formulation

As mentioned in the literature and through the different challenges we received from the survey, to do the automatic navigation, we need to have specific road measures. These measures are kept in consideration; one by one we discussed the standards and their factor which could affect the data processing. The following observation should be taken as reference.

- We need to verify all the road conditions and the feasibility of the data receiving and processing through the system.
- Then, by considering all the parameters, we need to design the mode of solution.

Table 2: Pavement condition

Pavement condition	Characteristics	Roughness IRI (mm/m)
PV1	Good driving limit exceeded	≤1.39
PV2	Smooth surface	1.4–2.69
PV3	Uneven surface condition	2.7–4.19
PV4	Border of road uneven	4.2–5.59
PV5	Irregular road, undrivable conditions	>5.6

PV1: very good pavement, PV2: good pavement, PV3: fair pavement, PV4: poor pavement, PV5: very poor pavement.

Table 3: Congestion condition

Congestion condition	Characteristics	Congestion factor
CC1	No traffic measurability	≤1
CC2	Slight rush	≤2
CC3	Specific crowd in area	≤3
CC4	Surface of the road is not prevalent	≤4
CC5	Surface is so much even and undrivable	≤5

CC1: very good congestion condition, CC2: good congestion condition, CC3: fair congestion condition, CC4: poor congestion condition, CC5: very poor congestion condition.

Some factors that we will consider in optimization of fuel-efficient routes are as follows:

– International Roughness Index (IRI) is pavement roughness. The roughness parameter in Table 2 is calculated by the vertical oscillations of the vehicle chassis per road section (generally 100 m). Its unit is mm/m. This table describes the roughness factor for the different surface parameter where it ranges up to 5.6 where it contains the different characteristics and satisfies all the conditions.

Taking the shortest route and avoiding (Table 3) congested overcrowded paths should be followed since they might result in increased fuel consumption. Congestion is graded on a scale of 1 to 5. In this table, there is a congestion factor for several conditions of measurable factor that contain distinct features; if the ranges vary within the range, the choice factor for choosing the road can be deviated or chosen.

We can save money by taking a route with less traffic lights (Table 4), as opposed to spending a few minutes standing in line to change signals and wasting fuel. This can be classified on a scale of 1–5. These are the most important factors to consider while determining the best driving parameters for safety. It also defines the conditions that are beneficial for moving from one node to another by optimizing and selecting the best conditions for the journey.

The path has fewer potholes. Potholes on the road can be categorized on a scale of 1 to 5 (Table 5). Potholes are

Table 4: Traffic light availability

Light condition	Characteristics	Traffic factor	Light
TL1	Light visibility is good; all the objects in the navigation are clearly visible	≤1	
TL2	All the objects in the navigation are visible	≤2	
TL3	Driving visibility is there	≤3	
TL4	Uneven driving visibility	≤4	
TL5	No light visibility, uneven traffic conditions	≤5	

TL1: very good traffic light condition, TL2: good traffic light condition, TL3: fair traffic light condition, TL4: poor traffic light condition, TL5: very poor traffic light condition.

Table 5: Pothole condition

Pothole condition	Characteristics	Road factor
PC1	Smooth driving condition with no potholes	≤1
PC2	Rare pothole in long distance travelled	≤2
PC3	Pothole is minor, driving can be possible	≤3
PC4	Lot of potholes, slow and steady driving can be done	≤4
PC5	Lot of potholes, uneven phase for driving	≤5

also a key element in determining whether or not to travel on a road, and these conditions may be taken to avoid catastrophic accidents and to offer a smooth driving experience for visitors who travel on that route. Potholes have distinct circumstances in the road where it disrupts the smooth driving element in the road and these are also different factors which are required for determining the road conditions and evaluating the distance.

3.1 Designing and development

First, calculate the optimized distance by consulting the effecting pavment condition (EPV) from equation 1. These factors helps to analyze the proper navigation. We will convert the required map into a graph where each place on potholes, the map will be depicted as the node on the graph. This helps to detect the potholes in accurate manner.

A car has to enter the area or the place where it wants to start the journey and set the destination area. If the car has to go from one place to another, then all the routes possible according to the map are depicted in the form of Figure 2 as shown. What is being added with each distance here is the factor effecting pavment condition (FEPV) value that will help us find the best fuel-efficient path in the final route.

$$\begin{aligned} &\mathrm{Fepv} = D_\mathrm{p} + (\mathrm{Pr} + \mathrm{Cf} + \mathrm{TL} + \mathrm{Rf})/4D_\mathrm{p} = \mathrm{Distance},\\ &P_\mathrm{r} = \text{Pavement roughness},\\ &C_\mathrm{f} = \text{Congestion factor},\\ &T_\mathrm{l} = \text{Traffice light factor},\\ &R_\mathrm{f} = \text{Road factor}. \end{aligned} \tag{1}$$

After this we have to design an algorithm for the aforementioned problem, and this project also aims to choose the best optimized algorithm which will be used to find the shortest path between two points entered by the user (Table 6).

Table 6: Finding out the navigation path through the proposed approach

Algorithm	Finding out the navigation path through the proposed approach

Input: *Enter the Distance Nodes values*
Output: *Path Navigator and observed value*

```
  Begin
      If nij = 0 if i = j Dij = 0 length (ni,nj) Cij = 0 otherwise NULL
          for K = 0 to A-1
          for J = 0 to A-1
          nij(K + 1) = min(nij(k)),Epv(nij(k) + nij(k) + dij(k)
      End
      for
  End
  for
End
```

4 Methodology

First, we need to do the setup of the car as shown in Figure 2 with the required hardware requirement. Take all the four motors and connect jumping wires with them. Since our H bridge can only handle only two motors at a time, connect two motors with each other at one time. Assemble all the motors in the plastic plates. Remember to cross-couple to let the motors move in the same direction. open source computer vision library stores the image in the form of the BGR color format; however, we need to change it to RGB color format which is important to adjust the settings of the view we have. We will use a setup function for our camera to stabilize, then we will take region of interest around these four corners.

We will take a sample, region of interest as shown in Figure 3. For this we will define the region we want identify for lane to get focused by camera to move car in forward direction. In the implementation process, we will

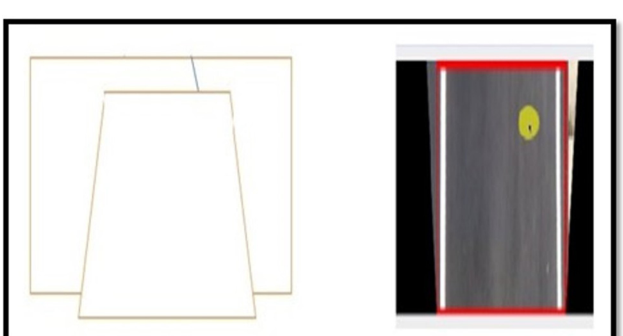

Figure 2: These two figures contain the frame for the region of interest and actual region of interest calculated.

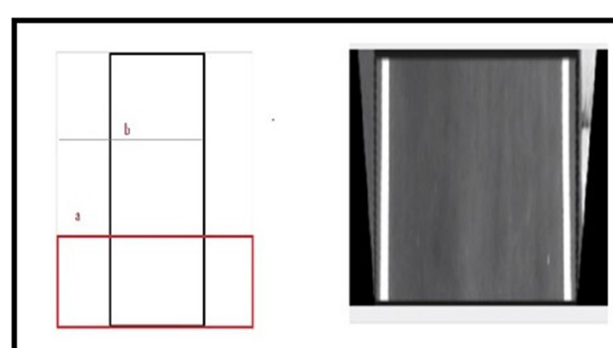

Figure 3: Calculation of the right and left position of the lane and the gray scale image.

convert our RGB image into gray scale to get the clear vision through camera.

We define the threshold manually, initially setting a specific value and creating a histogram for all values above this threshold. These are converted into white pixels, while all remaining values become black pixels. The next step, involves identifying all edges and corners within the lane using the Canny edge detection technique. This process facilitates easier object identification for our autonomous car.

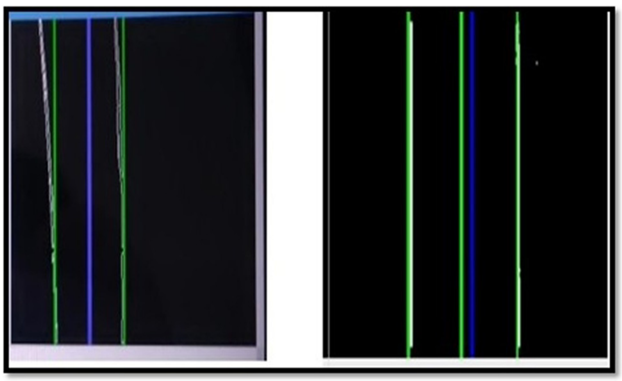

Figure 4: Calculation of the center of the lane and calibrating lane center with frame center.

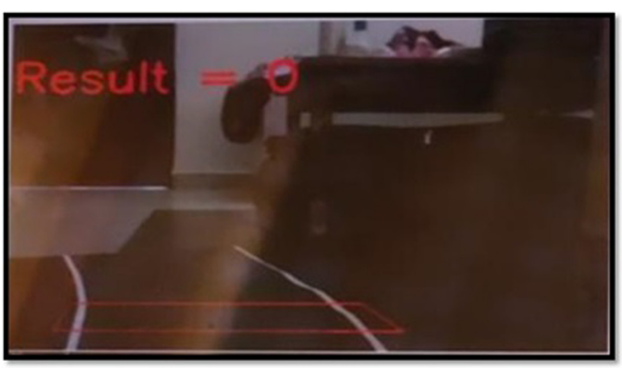

Figure 5: Difference between lane center and frame center.

Prior to image processing, we convert our RGB image in Figure 4 into gray-scale for easier manipulation. We define the threshold manually by setting a specific value. Those greater than the threshold are turned into white pixels, while all remaining values are turned into black pixels.

The next step is finding all the edges and corners coming in the lane so that it can help our car identify objects easily, with the help of canny edge detection. Canny edge detection basically detects the sudden change in the image gradient. For getting canny edges, we will apply sobel operator on our threshold image.

In sobel operator, suppose Gx is an image pixel where each pixel contains the horizontal derivative and Gy is an image pixel where each pixel contains the vertical derivative, then $G = \mathrm{sqrt}(Gx^2 + Gy^2)$ where G represents the image gradient. Then we will find the exact position of the lane, *i.e.*, right position and the left position. The next step is finding the left position and the right position of our lane where our autonomous car will move in during its journey.

Green lines depict the lane finder. In the next step, we will find the lane center using the left lane position and the right lane position. The blue color in Figure 5 shows the lane center.

In the next step, we will calibrate our lane center with frame center (Figure 5). The green line depicts the lane center and blue line depicts the frame center; we will shift frame center towards the left so that it can calibrate with lane center. In the next step, we will move our autonomous car in different directions and check the difference between lane center and frame center as a result.

In the following stage, we will use CNNs (Figure 6). CNNs are a type of neural network that have proven to be particularly effective in picture recognition and categorization.

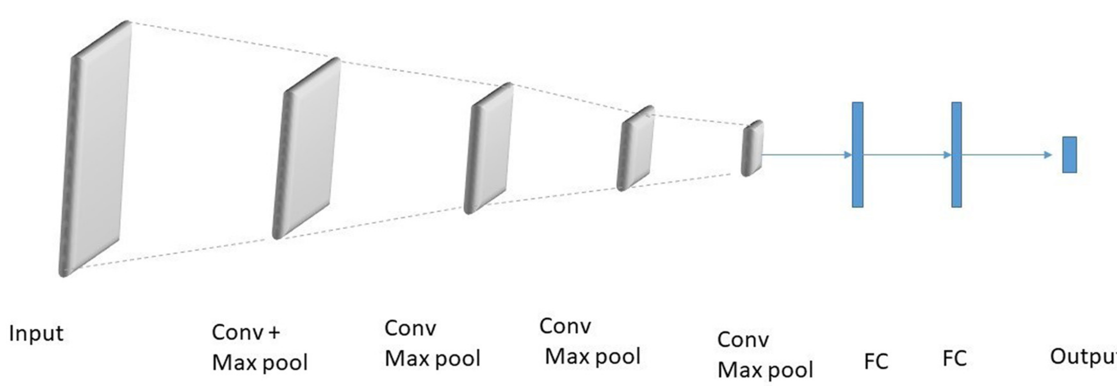

Figure 6: CNN model.

The CNN design categorizes and is primarily utilized for character recognition jobs, such as Classification, Convolution, Filters, Non-Linearity (ReLU) Activation Function, Pooling, or Sub Sampling (Fully Connected Layer).

Since we are dealing with CNN, we will deal here with conv2D, MaxPool2D layer which is present in Keras, and after we have imported the sequential model we will first add some convolutions layer. We will first add convolution layer with 32 filters for the first two convolution layers having 5x5 kernel matrix filter, which can be involved on original image for extracting the important feature from image.

The kernel matrix is applied on complete image matrix. We have now incorporated a down-sampling filter, specifically Max2D, which reduces the image's dimensions. This process effectively shrinks the size of the image, simplifying further manipulation and analysis.

Next, we must decide upon the pooling size. It is critical to select the pooling dimension as well. Also, we are using convolution and pooling in this layer to allow our model to learn more information.

Next, we will add two more convolution layers, with 64 filters and down sampling, at the conclusion. To get things started, after we have finished pooling, we will go on to the dropout layer, which is a regularization approach that randomly sets the weights of a section of the nodes in the layer to zero. The last step involves dealing with the feature that causes certain nodes to randomly disconnect from the network. This necessitates the remaining network to reach a distributed solution.

When it comes to increasing generalization and controlling over fitting, this strategy works well. (ReLu) is an abbreviation for maximal activation function (0,*x*). The rectifier activation function is used to introduce non-linearity into the system.

The flatten layer is used to convert the final feature mappings into a single 1D vector representation. It will be necessary to flatten the layers once they have been convolution and max pooled in order to use completely connected layers. Essentially, it combines into the convolution layer all of the previously trained local properties.

As an alternative to digging deeper, we built an ANN classifier based on the properties of the previous layer. The final layer produces a distribution of the likelihood of each class, which is displayed on the screen.

Table 7: Analysis of pavement by considering the EPV factor

Pavement level	Level	Noise	SS	RG	LC	EPV	Optimal path
PV1	0	0	0	0	0	NULL	0
PV2	0.25	1	1	0	0	Marginal	0.0055
PV3	0.5	1	1	0	0	Good	0.0066
PV4	0.75	1	1	1	0	Pleasant	0.0074
PV5	1	1	1	1	1	Best	0.0084

PV: pavement level, SS: smoothness, RG: roughness, LC: localization.

5 Result and analysis

The process has been thoroughly evaluated the crucial aspects of the path by testing various parameters. While updating the results, these parameters were analyzed in relation to the condition of the pavement.

Table 7 contains the optimal path efficiency considering the EPV factor in the various dimension of pavement level, and this table contains the level which defines the five constraints from 0 to 1. This table defines the optimized path by

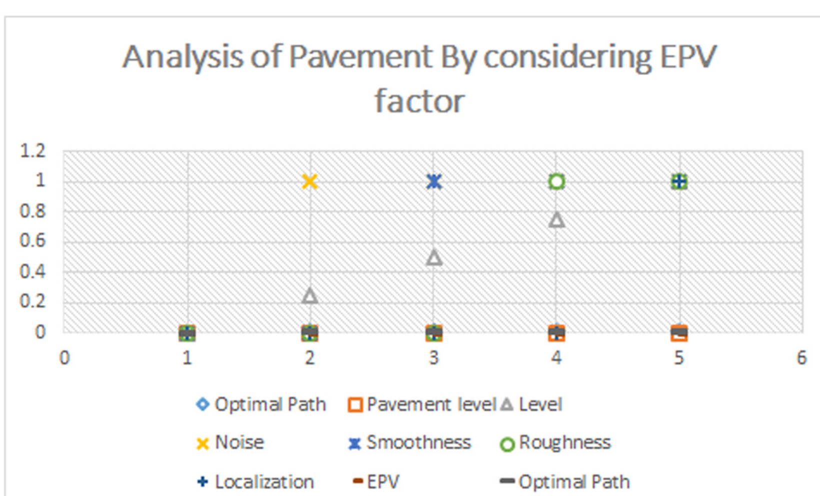

Figure 7: Optimal path by consideration of EPV factor and pavement level with different parameters.

Table 8: Analysis of congestion condition by considering the EPV factor

Congestion condition	Traffic system	Object identification	Noise	EPV calculation	Optimal path
CC1	0	0	0	0	0
CC2	23	22	33.4	26.1	0.66
CC3	22	28	38	29.3	0.79
CC4	25	29	39	31	0.81
CC5	46	45	50	47	0.88

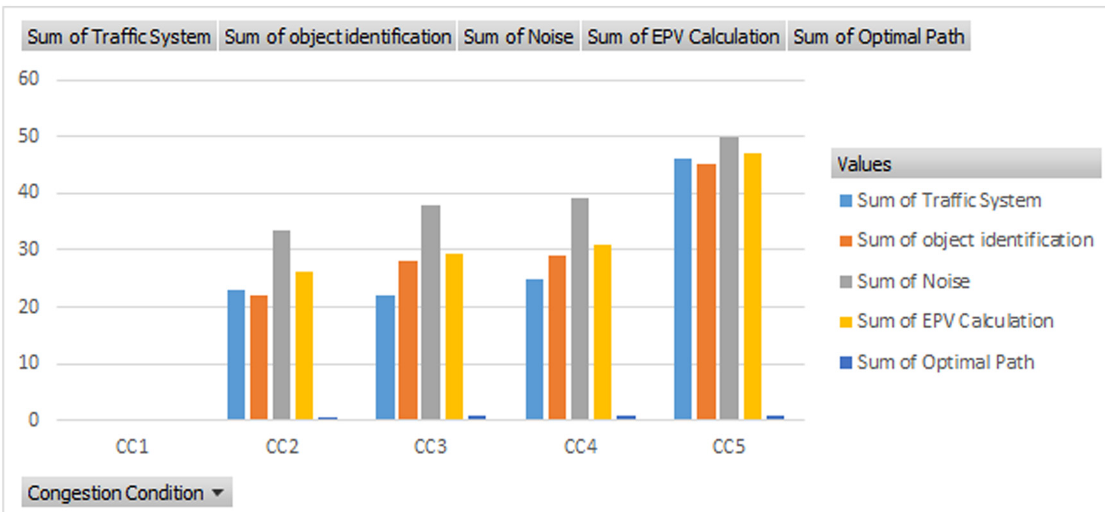

Figure 8: Analysis of the congestion condition.

checking the notation of the noise value of the road, smoothness, roughness, localization, and finally, the EPV factor.

Figure 7 contains the detail analysis for the optimal path relation with the EPV and payment condition and describes the feasible nature for the optimal decision-making by the system.

Table 8 contains the analysis of the congestion condition considering the EPV factor, and there was feasible observation in the levels 5 and 4 where the driving condition is best by considering the multifactor analysis (Figure 8).

Table 9 analyzes the traffic condition and pothole condition and contains the major feature balance for the optimal detection of the path to identify the feasibility for the driving. The table describes the conditions and reading obtained during the testing phase in the road.

Table 9: Analysis of pothole condition and traffic light condition by considering the EPV factor

Pothole	TL	TS	OI	noise	EPV	Optimal path
PP1	TL1	8	12	8	9	0.3
PP2	TL2	28	27	46.9	34	0.68
PP3	TL3	29	54	56	46	0.74
PP4	TL4	29	59	39	42	0.87
PP5	TL5	46	45	50	47	0.96

TL: traffic light, TS: traffic system: OI: object identification, OP: optimal path.

Figure 9 describes the functional analysis for the optimal travel path for deciding the final route on the basis of pothole condition and the traffic light condition by using the EPV factor.

The figure discusses the setup model, and this model is tested in the different road structure and domain. And it shows the promised observation in the strategic road condition.

Table 10 contains the comparison of the existing system in terms of the following parameters: sensor fusion, perception, localization, mapping, and efficiency. The proposed system shows better competency in deciding the path with the existing system.

5.1 Cost of hardware components

The whole component which is required for the creation of these projects is very optimal as compared to the other components which were used. Table 11 contains the cost of the hardware components in USD and INR.

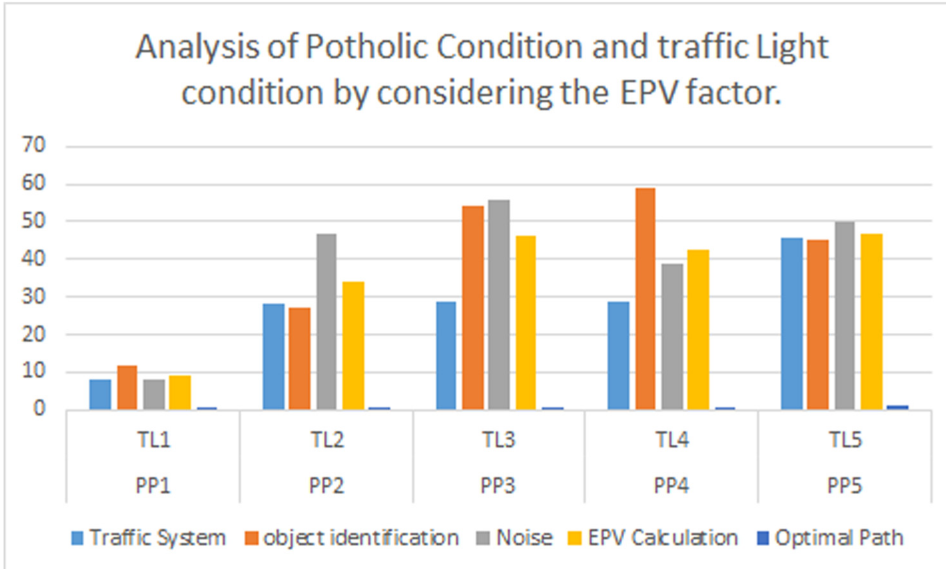

Figure 9: Analysis of pothole condition and traffic light condition.

Table 10: Existing system comparison

Model	SS	PC	LC	MP
AV linear quadratic Gaussian (LQG) control [26]	✓	✗	✓	✓
Hybrid cost and time path AV [27]	✓	✗	✓	✓
Physics-based path AV [28]	✓	✓	✗	✓
Predictive maneuver AV [29]	✓	✗	✓	✓
Proposed system	✓	✓	✓	✓

SS: sensors, PC: perception, LC: localization, MP: mapping.

Table 11: Cost analysis

S.No	Hardware name	INR	USD
1	Arduino UNO R3	700	9.43
2	Robot wheel	20	0.27
3	Wooden board	50	0.67
4	Rasbiri pipe	2,348	31.64
5	Connecting pipes	10	0.13
6	Artificial lanes	200	2.69
	Total	3,328	44.83

6 Conclusion

Based on the findings from the system analysis, our team attempted to compare the results based on sensors, perception, and localization. We suggested that the system have a multifunction choice for choosing the pathway, taking the EPV component into account slightly to ensure it includes the most prevalent path. With this approach, we were able to identify the most efficient approach to take while discussing in a hostile environment. This was an effective way to determine the most effective course of action as it takes into account multiple factors such as sensors, perception, and localization. Furthermore, it further strengthens our team's commitment to providing maximum efficiency with minimum effort.

Funding information: The authors state no funding involved.

Author contributions: All authors have accepted responsibility for the entire content of this manuscript and approved its submission.

Conflict of interest: The authors state no conflict of interest.

Informed consent: Informed consent was obtained from all individuals included in this study.

Ethical approval: The conducted research is not related to either human or animals use.

Data availability statement: Data sharing is not applicable to this article as no datasets were generated or analysed during the current study.

References

[1] J. Borenstein and Y. Koren, "Obstacle avoidance with ultrasonic sensors," *IEEE J. Robot. Autom.*, vol. 4, no. 2, pp. 213–218, 1988.

[2] Y. Wang, E. K. Teoh, and D. Shen, “Lane detection and tracking using B-Snake,” *Image Vis. Comput.*, vol. 22, no. 4, pp. 269–280, 2004.

[3] N. P. Pawar and M. M. Patil, “Driver assistance system based on Raspberry Pi,” *Int. J. Comput. Appl.*, vol. 95, no. 16, p. 16, 2014.

[4] T. D. Do, M. T. Duong, Q. V. Dang, and M. H. Le, “Real-time self-driving car navigation using deep neural network,” In *2018 4th International Conference on Green Technology and Sustainable Development (GTSD)*, 2018, pp. 7–12.

[5] M. Bojarski, D. Del Testa, D. Dworakowski, B. Firner, B. Flepp, P. Goyal, et al., “End to end learning for self-driving cars,” *arXiv preprint arXiv:1604.07316*, 2016.

[6] Y. Ioannou, D. Robertson, R. Cipolla, and A. Criminisi. “Deep roots: Improving CNN efficiency with hierarchical filter groups,” *CoRR, abs/1605.06489*, 2016.

[7] J. del Egio, L. M. Bergasa, E. Romera, C. Gómez Huélamo, J. Araluce, and R. Barea, “Self-driving a car in simulation through a CNN,” In *Workshop of physical agents*, 2018, pp. 31–43.

[8] A. Agnihotri, P. Saraf, and K. R. Bapnad, “A convolutional neural network approach towards self-driving cars,” In *2019 IEEE 16th India Council International Conference (INDICON)*, 2019, pp. 1–4.

[9] K. O'Shea and R. Nash, “An introduction to convolutional neural networks,” *arXiv preprint arXiv:1511.08458*, 2015.

[10] A. Millard-Ball, “The autonomous vehicle parking problem,” *Transp. Policy*, vol. 75, pp. 99–108, 2019.

[11] K. Mahadevan, S. Somanath, and E. Sharlin, *Communicating awareness and intent in autonomous vehicle-pedestrian interaction*, New York, NY, USA, Association for Computing Machinery, 2018.

[12] S. Kuutti, R. Bowden, Y. Jin, P. Barber, and S. Fallah, “A survey of deep learning applications to autonomous vehicle control,” *IEEE Trans. Intell. TransportatiSyst.*, vol. 22, no. 2, pp. 712–733, 2021.

[13] T. Ort, L. Paull, and D. Rus, “Autonomous vehicle navigation in rural environments without detailed prior maps,” In *2018 IEEE International Conference on Robotics and Automation (ICRA)*, 2018, pp. 2040–2047.

[14] N. Adnan, S. M. Nordin, M. A. bin Bahruddin, and M. Ali, “How trust can drive forward the user acceptance to the technology? In-vehicle technology for autonomous vehicle,” *Transportation Res. Part. A: Policy Pract.*, vol. 118, pp. 819–836, 2018.

[15] J. Jiang and A. Astolfi, “Lateral control of an autonomous vehicle,” *IEEE Trans. Intell. Veh.*, vol. 3, no. 2, pp. 228–237, 2018.

[16] J. Fayyad, M. A. Jaradat, D. Gruyer, and H. Najjaran, “Deep learning sensor fusion for autonomous vehicle perception and localization: A review,” *Sensors*, vol. 20, no. 15, pp. 4220–4220, 2020.

[17] H. Gao, B. Cheng, J. Wang, K. Li, J. Zhao, and D. Li, “Object classification using CNN-based fusion of vision and LIDAR in autonomous vehicle environment,” *IEEE Trans. Ind. Inform.*, vol. 14, no. 9, pp. 4224–4231, 2018.

[18] M. Khayyat, A. Alshahrani, S. Alharbi, I. Elgendy, A. Paramonov, and A. Koucheryavy, “Multilevel service-provisioning-based autonomous vehicle applications,” *Sustainability*, vol. 12, no. 6. p. 2497, 2020.

[19] R. McCall, F. McGee, A. Mirnig, A. Meschtscherjakov, N. Louveton, T. Engel, et al., “ A taxonomy of autonomous vehicle handover situations,” *Transportation Res. Part. A: Policy Pract.*, vol. 124, pp. 507–522, 2019.

[20] S. A. Fayazi and A. Vahidi, “Mixed-integer linear programming for optimal scheduling of autonomous vehicle intersection crossing,” *IEEE Trans. Intell. Veh.*, vol. 3, no. 3, pp. 287–299, 2018.

[21] D. Stanek, R. T. Milam, E. Huang, and Y. A. Wang, Measuring autonomous vehicle impacts on congested networks using simulation. *Transportation Research Board 97th Annual Meeting*; 2018.

[22] A. Best, S. Narang, L. Pasqualin, D. Barber, and D. Manocha, “Autonovi-sim: Autonomous vehicle simulation platform with weather, sensing, and traffic control,” In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops*, 2018, pp. 1048–1056.

[23] X. He, Y. Liu, C. Lv, X. Ji, and Y. Liu, “Emergency steering control of autonomous vehicle for collision avoidance and stabilisation,” *Veh. Syst. Dyn.*, vol. 57, no. 8, pp. 1163–1187, 2019.

[24] C. Sun, X. Zhang, Q. Zhou, and Y. Tian, “A model predictive controller with switched tracking error for autonomous vehicle path tracking,” *IEEE Access*, vol. 7, pp. 53103–53114, 2019.

[25] Y. Jeong, S. Son, E. Jeong, and B. Lee, “An integrated selfdiagnosis system for an autonomous vehicle based on an IoT gateway and deep learning,” *Appl. Sci.*, vol. 8, no. 7, p. 1164, 2018.

[26] K. Lee, S. Jeon, H. Kim, and D. Kum, “Optimal path tracking control of autonomous vehicle: Adaptive full-state linear quadratic Gaussian (LQG) control,” *IEEE Access*, vol. 7, pp. 109120–109133, 2019.

[27] H. Fazlollahtabar and S. Hassanli, “Hybrid cost and time path planning for multiple autonomous guided vehicles,” *Appl. Intell.*, vol. 48, no. 2, pp. 482–498, 2018.

[28] B. Sebastian and P. Ben-Tzvi, “Physics based path planning for autonomous tracked vehicle in challenging terrain,” *J. Intell. Robotic Syst.*, vol. 95, no. 2, pp. 511–526, 2019.

[29] Q. Wang, B. Ayalew, and T. Weiskircher. “Predictive maneuver planning for an autonomous vehicle in public highway traffic,” *IEEE Trans. Intell. Transportation Syst.*, vol. 20, no. 4, pp. 1303–1315, 2019.