International Journal of Computational and Electronic Aspects in Engineering
Volume 6, Issue 3, July 2025, pp 121 - 133
https://doi.org/10.26706/ijceae.6.3.20250602

Development of A Concept for A Driverless Vehicle Using an Artificial Neural Network

Sadeq Thamer Hlama [1,*], Zaid Hamid Alkhairullah[2], Abeer Naser Faisal [3]

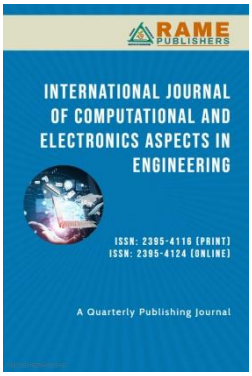
[1] Department of Computer Science, College of Computer Science and Information Technology, University of Sumer, Dhi-Qar, Iraq

[2] Ministry of Education, General Directorate of Vocational Education, Department of Vocational Education in Thi Qar, Iraq

[3] Department of Computer Information Systems, College of Computer Science and Information Technology, University of Sumer, Dhi Qar, Iraq

sadeqthamer1976@gmail.com[1], zaid.sw82@gmail.com[2], abeernaser13@gmail.com [3]

[*]Correspondence: sadeqthamer1976@gmail.com



Article – Peer Reviewed
Received: 30 April 2025
Accepted: 15 June 2025
Published: 2 July 2025

Cite this article: Sadeq Thamer Hlama, Zaid Hamid Alkhairullah, Abeer Naser Faisal, "Development of a concept for a driverless vehicle using an artificial neural network", *International Journal of Computational and Electronic Aspects in Engineering*, RAME Publishers, volume 6, issue 3, pp. 121-133, 2025.
https://doi.org/10.26706/ijceae.6.3.20250602

Abstract: Recent developments in self-driving cars and smart control would make the idea of an autonomous vehicle a viable one. Nonetheless, using of such cars in traffic and some other hostile surroundings successful requires effective solution of a package of navigation, vision, and control problems. In all this, our ultimate goal is to come up with cost effective methods that are operationally sound which in turn can help the larger research community to take self-driving automobiles offering business potential seriously. However, what we are in need of, is a ploy that could transform the traditional motorists into discouraging using self-driving cars and at the same time works out the sorting of the present passenger cars into the future research work. In this vein, the current discussion develops a modular mechanical structure that could be manufactured in especially fast fashion and adapted on a significant portion of modern automobiles. The design brief is an intermediate stage of development on the road to the production of wholly autonomous cars. With the use of commercially available actuators, we prove that it is possible to transform homegrown automobiles of the sort found in abundance in the parking lot of whatever rest stop or convenience store you happen to drop by, quite reasonably to include most of the privately owned vehicles, into autonomous models. In the context of the motor vehicle's automation, motors are often used as actuators Considering the fact that the electro-mechanical motors did not work, pneumatic system was introduced to make one predetermined step automated. Mechanical setup of autonomous cars is the most important aspect and should provide a possibility to make regular updates as the car is designed to work under extremely changing conditions. In order to measure the feasibility of such strategy, two further convolutional neural networks were re-implemented, allowing to conduct an objective comparison of the performance, technical severity and architecture of the proposed system with regards to the reference networks. The architecture is proposed to 300 AlexNet units and three PilotNet units (Increased feature extraction capacity, slightly reduced control model replication) and operate at a bigger scale than the prevalent models of neural networks that are available today. On the one hand, it has fewer architectural complexities compared to its peers and hence experiences longer latencies, and slower inference time; on the other hand, the system still ranks equally well on the autonomous-driving metrics reported by two similar benchmarks. By meeting this performance objective and at the cost of no longer requesting high-speed computing hardware, the model promotes not only cost-efficiency, scale, and total affordability..

Keywords: Deep neural network, end-to-end learning, embedded systems, machine learning, camera, autonomous driving, and convolutional neural network.

1. Introduction

There are four possible cube-shaped components of the autonomous driving system (detectors, understanding subsystem, intending subsystem, and control of automobile, figure (1) Currently, the vehicle is employing its array of detectors to conduct a worldwide survey. The understanding block's output is sent into the preparation subsystem, where it's employed for behavior preparation and long-term route planning. The controller module monitors your vehicle's progress along the predetermined path and issues directives to the relevant subsystems to keep things moving smoothly. Research in machine learning, and deep learning in particular, has led to many technological applications and discoveries across many fields. System learning has a considerable effect on the car industry and the progress toward fully autonomous vehicles. Many automotive technologies will reach maturity at the same time as cars gain more autonomy in the workplace. Delivery trucks, as well as other types of robot cars and robots, may park in warehouses as they wait to be sent. This project's primary goal was to provide a solution for autonomous driving on a lightweight automobile stage using just basic hardware components, processing power, and storage. We expect to find a light deep neural network (DNN), an end-to-end neural system that can soon carry out the job of autonomous driving to the representative track, despite our knowledge of hardware constraints. A version of the network used for inference may be deployed on a stage of hardware with limited processing power [1].

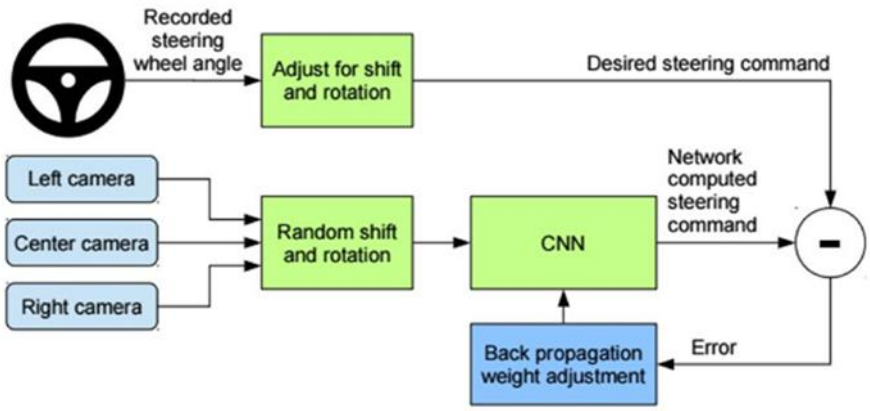


Figure 1: Block diagram of an end-to-end autonomous driving system[1].

The desire to build autonomous cars only increases every day, and the expanded interest in this direction is associated with high levels of protection, functionality, durability, and pleasure. Driverless cars are especially fit in busy areas and highways whereby they are able to enhance the traffic. The accidents caused by other road users or pedestrians might also be reduced because of such vehicles. Various labs around the world are dedicated to developing smart systems to be used in driverless car applications such as the Center of Automotive Research in Stanford University (CARS).TORCS is characterized as a vibrant academic commons, uniting a large number of program developers into a community along with the users. Also, it provides the structural background whereby the yearly competitions are regularly held, an inherent component in many global summits. The application runs on a shared environment that has races, which are made of multiple cars on different monitors. The person in charge of an agent representing a certain vehicle can release any given client module. At the 2013 Genetic and Evolutionary Computation Seminar the agent called Gazelle was entered in the TORCS contest. In this ecosystem, every car is controlled by a process that constantly analyzes the current state of the car; based on the information about the environment, i.e., the track, car state and competitor state.

The upper screen which categorizes the client automobile and provides relevant statistics including its spatial layout, time passed since the race has been started, the optimal time needed to finish a certain lap, and a set of auxiliary indicators. The lower display also changes the outlook, allowing you to observe the other cars that are potentially rivals and that you will have a chance of seeing here and there. Each screen will be accompanied by localized statistics: in this screen on the top we will see the levels of gears and speed of laps, whereas in this screen at the bottom we will find among other information the angle of rotations and acceleration as regards further frames. The analysis framework in the background behind this interface takes off the EPIC control developed by Guse and Vrajitoru, which in turn was submitted in the GECCO 2009 competition. Shortly all in all EPIC settles two underlying duties: to decide some good rotational angle in each frame and to pick up optimal acceleration in the next frame.

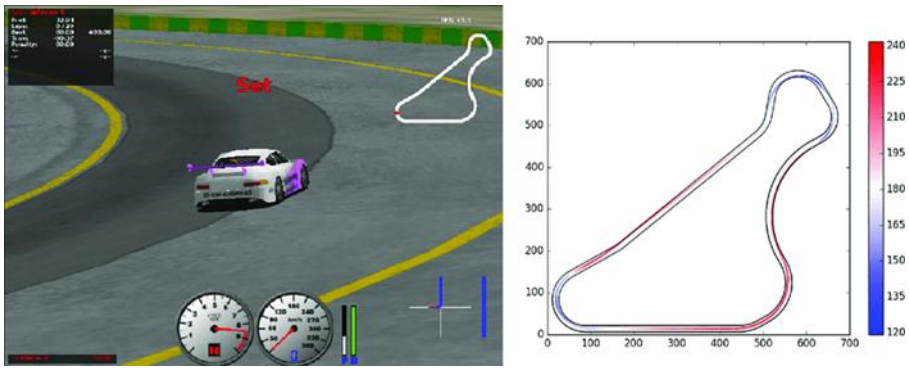


Figure 2: TORCS image during the race

The control unit, in the proposed case, calculates the target angle according to the available space, which is free in all respects, before it. There is also a sharp-twist detecting mechanism where the goal rate is recalibrated in order to ease an imminent sharp twist so that the vehicle sits in the desired lane. It uses hill-climbing method to optimize the rate parameters to make adaptations to new tracks. Even more important, the application of system-learning methods to the embedded hardware supports two independent trends: both the introduction of new hardware systems with the capacity to support light machine-learning architectures, as well as the versions, that can be applied to low-performance hardware, and the processing of data needed to run machine-learning inference. It should be mentioned that most of the current prominent learning-based systems on autonomous driving were prototyped on actual cars. In most of them, the inference engine of the system-learning is situated in the back of the car. Other approaches put more emphasis on very deep neural networks, something that with existing computational resources can be computationally prohibitive.

The goal of the project was to create a more extensive and, in a sense, lighter deep neural system that, on the one hand, could provide similar autopilot results in terms of quality and capabilities to famous modern solutions and, on the other hand, it would be permitted to be deployed in an embedded system. This change, or cheap neural architecture, will be specially appreciated by car robots and other robots used to deliver goods, equipment, or anything to perform industrial activities, and so on, so computers do not always have unlimited computing capabilities.

J-Net architecture comparison shows that although the J-Net architecture is small and less computational demanding when compared to other architecture, it gives comparable results in autonomous-driving tasks. Deployment of this domain indicates that lightweight versions of convolutional neural networks are needed since the onboard processors of the embedded automotive fields do not have the potential hardware resources that will facilitate the implementation of a large-scale deep learning. Such features are also supported by the Epic library created by the Gazelle motorist whose modular structure offers using a wide choice of opportunities to create track predictions with the specific aim of optimizing working behavior. One of the notable techniques utilized in the context is path segmentation whereby the path has been broken into segments that are disjoint polygons which belong to a set of directive and geometric classes that are predefined. The overall movement is then computed by the controller using these simple polygons.

This control is simple and contains modules that restrict gear altering, steer moves, pedal rankings, and other functions. Your opponent modifier is the most important component of this job. This modifies the driving behavior of opponents who are nearby, correcting both the steering and braking controls instantly. AUTOPIA, a more recent control, is used for the racing car contest. It provides the entire driving structure, including six main tasks: steering controller, pedal controller and steering controller, stuck place manager, target rate conclusion and opponent modifier, as well as a learning module. It provides a strong and simple structure, especially for the opponent modifier that uses heuristic rules. Many learning methods are used to find the best path for your car in order to reduce the time required to complete the race. This article will present an empirical learning method. This evolutionary strategy is self-adaptive and can be used to determine the parameters that will affect the mark rate. It's easy to apply and generalize. The driver does not have an opponent management system. A second control that works with an evolutionary learning process has been presented. This knob uses a technique based on evolutionary learning to plot out the most direct path to your car. In order to better arrange the path ahead of your car, a new learning approach has lately adopted the usage of hyper heuristics. The technique takes a real-valued optimization approach to the TORCS-based automotive system and analyses how various methods perform. One may use a hyper-heuristic framework to govern the combination of two heuristics. Hyper-

heuristics have been successful in the TORCS setting. There will also be the incorporation of artificial neural networks, which are widely recognized as a reliable learning platform amongst software engineers. The control generates an iteration using the NNs to establish the path and mark rate of the vehicle. The NNs were trained using data collected from a player. The work is rewarding since it requires deciding on alternative courses of action, but the potential rate is smaller than that of those travelling along the exact same routes. In this study, we present a neural system with the depth of a book that can operate a vehicle without human intervention. Additionally, embedded automotive systems may be developed using J-Net. We also provide details on the unbiased research of J-Net and the outcomes of reimplementation of PilotNet. To begin, a deep neural network structure was published by J-Net. Production of J-Net has begun. Once again, the version has been trained and is ready for autonomous driving. The goal is to provide a neutral assessment of the network architecture. The models are trained with all the information we have gathered so far. In a simulated setting, the trained models may be employed for autonomous driving. There are three different models of complex brain networks shown, all of which may be used to demonstrate autonomous driving. Videos of three heavy-duty neural system models driving autonomously over a typical path in a driving simulator serve as the source material for this article. Parameters of autonomous driving are evaluated quantitatively and qualitatively throughout this inference, and the results are presented.

In this paper, we will discuss previous work, the structure of the proposed system, the design of the system, a comparison between previous systems and the system used, and the results and future work.

2. Related work

Deep learning is not merely a subcategory of machine-learning, it is part of an extended family of machine-learning procedures that depend upon the profiling of data. One layer within a deep neural system defines its representations relative to the faster ones created by its preceding layers. Convolutional networks therefore become a typical spot of gradients among the neural architectures. In the contexts of deep learning, CNNs are often described as a rigorous combination of three core concepts: 1) Local representative regions, 2) Shared weights, 3) plasma or time sub-sampling. The combined principles impart invariance of scale, shift and distortion to the network. CNNs have become popular because of processing the data that come in several arrays colour picture, talk, sound spectrogram, and lately, video. All these modalities are characterized by localized spatial or temporal associations besides carrying a common computing cost within time or space units. Their performance is also enhanced by pooling mechanisms, and strategic up-scaling of multiple layered architectures. Although, in practice, CNNs have become one of the minimal reactory sites of visual scene content assessment, and their use becomes as universal in sociology, as in industry and business. These include automotive, surveillance, security, smart home software, and retail automation. One of the first deep models to work well is convolutional neural networks. It also helped to create many systems that could be used in critical industrial applications. Convolutional neural networks were later used to create optical character recognition and handwriting recognition tools. The latest uses of CNNs are endless.

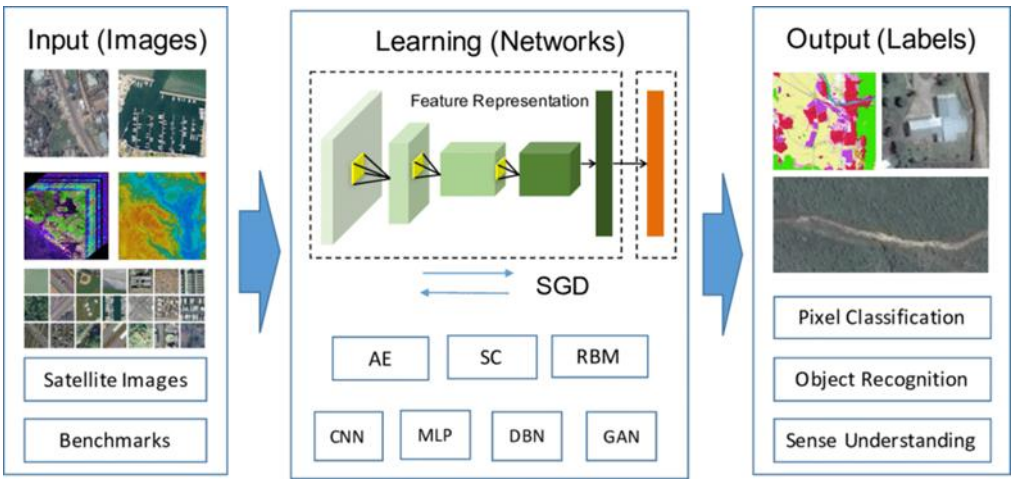


Figure 3: Flow of deep learning development

Convolution neural networks have advanced greatly thanks to the efforts of the Image Net Big Scale Visual Recognition Challenge. The winning designs in this competition are cutting edge in the field of deep learning and neural networks. They serve as a springboard for creative thinking and a basis for rethinking existing problems. Automated machine learning, video-to-video synthesis, deep learning using artificial information, and go playing are all examples of seminal works in the field of profound learning. The first serious attempts to create autonomous vehicles were launched in the 1950s. The first completely autonomous automobiles appeared on the road in 1984. The war for autonomous cars waged by the Defense Advanced Research Projects Agency, the Grand Challenge competitions held in 2004 and 2005, and the Urban Challenge held in 2007 were all pivotal in establishing that robots are capable of performing the difficult, individual job of driving. While it is true that autonomous vehicle prototypes may be tested on public roads, numerous issues still need to be resolved before the technology can be widely used. Several obstacles stand in the way of fully autonomous vehicles at the moment. Detector combination, advanced preparation choices, full autonomous drive learning, full autonomous drive reinforcement learning, and human-machine interaction are all on the list. In this study, we'll compare and contrast the various deep learning architectures utilized in autonomous cars in a systematic way. Sensor and detector combinations used in autonomous cars will also be detailed.

3. System for Fully-Autonomous Driving

The development of a self-driving strategy involved end-to-end learning framework in our investigation. The method is also defined to be rudimentary in that it plays with the raw pixel inputs and the resultant output in terms of steering. Managed by the physical attributes of the vehicle, mainly, the steering wheel, the system is conditioned to work only on camera data which is in real-time.

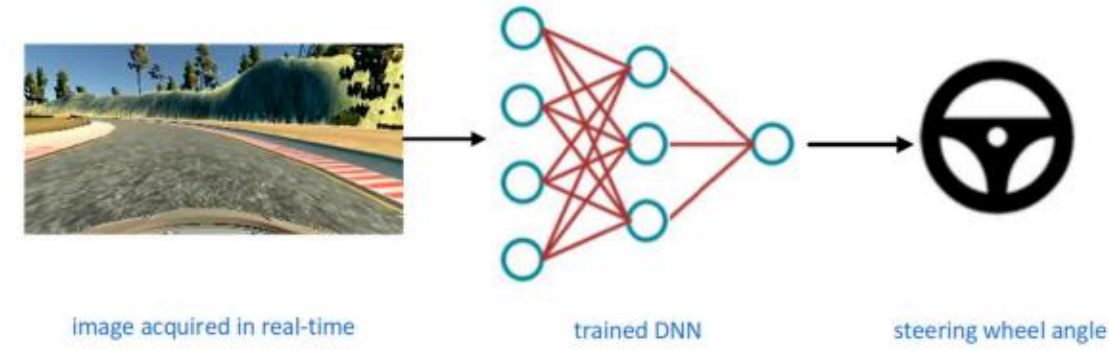


Figure 4: Autonomous driving in real time to do this, the trained deep neural network model is given data from the main camera.

Someone behind the wheel drove the vehicle and measured the resulting steering angles. The primary goal here is to amass data for training the DNN variant. We utilized the autonomous driving simulator. Manual (training) driving included a single driver controlling the vehicle using a pointing device such as a mouse, joystick, keyboard, or keyboard. The data collection was also gathered mechanically. While using the manual driving approach, we recorded camera images and steering values for each chassis. The visuals served as the feature set and the guiding parameters for the tag gathering process. The vehicle's speed increased as a result of this simplification. This training procedure yielded valuable data for the neural system, which eventually became capable of driving a car with no further assistance from a human. Cloning behavior is the term for this phenomenon. Second, we used this information to train the autonomous driving system's deep neural network to make future steering wheel predictions. Real-time autonomous driving was performed in the same simulated environment as the one used during training. During sovereign driving, success was determined by how much of the time the vehicle spent on the representative path. Our autonomous driving framework is organized like this.

4. Design

Pneumatic Circuit: As it can be seen in the pneumatic circuit diagram in Figure 1, it contains three cylinders, such as brakes, accelerator, and clutch, and five/three management control valve that controls all of these elements. The valves can be controlled with the help of a microcontroller. The reservoirs are actuated by simply turning the knobs and they can

operate independently or together depending on the needs of the truck or car. Signals received are transmitted to the microcontroller.

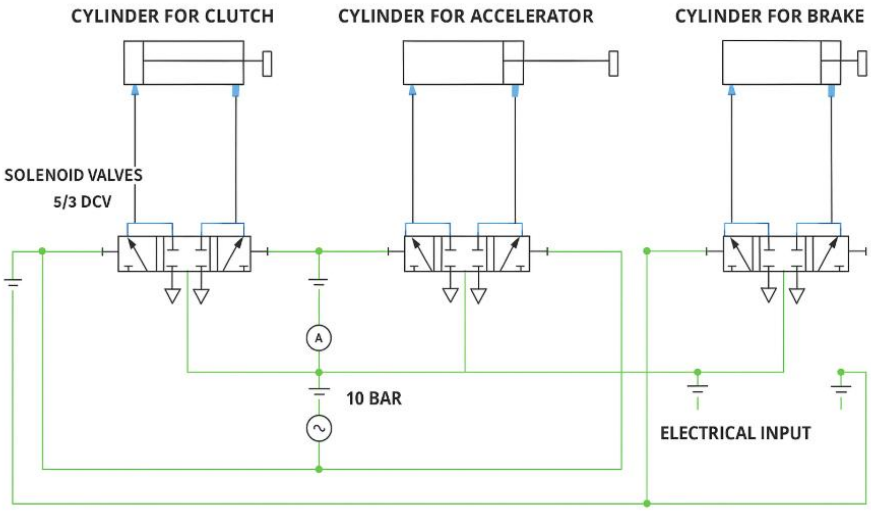


Figure 5: Pneumatic Circuit [26]

The accompanying cad drawing provides the outline of the framework, mounting, and overall arrangement of a three-tank system. Every iteration will have three tanks; the frame is then mounted to the car chassis, and the electronics added as shown in Fig. (6).

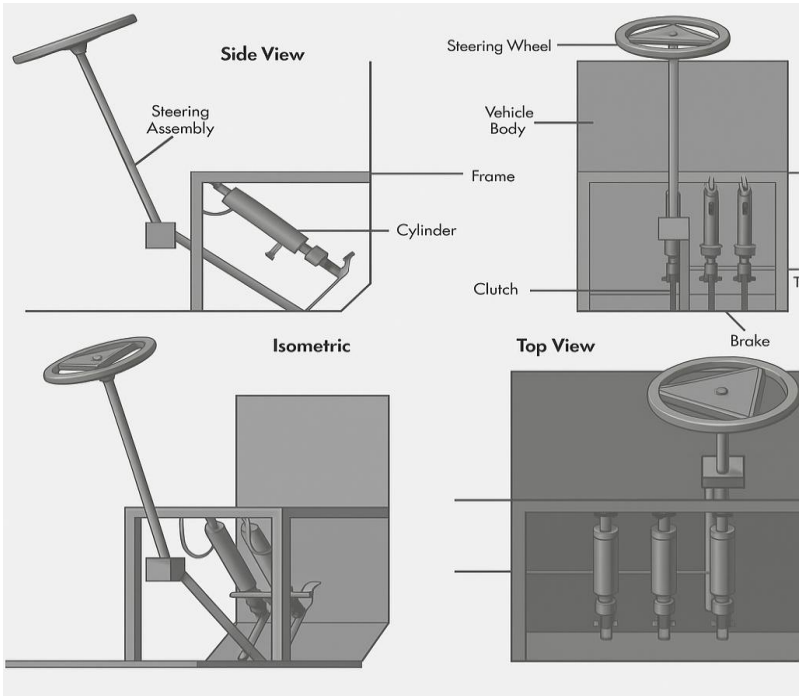


Figure 6: Cad drawing accelerator, brake and clutch respectively [26].

The pneumatic cylinders are connected into this the suggested cad version is fabricated in the Car and can be analyzed with Pneumatic inputs in Fig. (7) Management control valves and are subsequently joined to the air reservoir.

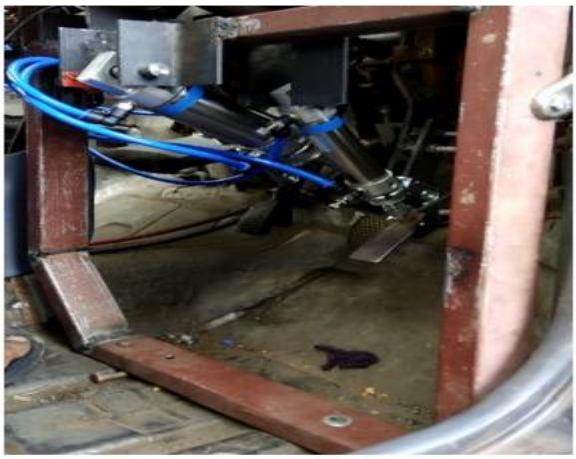


Figure 7: Fabricated Parts [26].

5. Steering System

To create the stepper motor and all of the equipment pushing the steering group, the cad version was designed. The engine shaft mounts the driving gear and the powered gear. A framework is used to attach the engine to the car's body. The framework will not allow the gears to detach and can be used to keep them in place. An encoder wheel can be used to locate the feedback from this spinning wheel. This is used to determine the feedback of the wheel and convert it into closed-loop feedback. As shown in Fig. 9, the cad model clearly defines the location of the meeting, and drive, as well as the framework where the engine is mounted. (9) The engineered frame can be fabricated and placed in the framework, as shown in Figure. (10). The engine is run at different speeds to analyze the frame. The engine could be contained within the framework at high speeds and the gears may have been engaged throughout. Steering is an essential part of the automobile's dynamics. It controls the navigation and direction of the vehicle. Most cars use hydraulic steering, leaving behind the old mechanical system. A few cars also now have electric power steering systems. Electric power steering offers a variety of benefits, including less maintenance, flexibility, speed, energy conservation, and a favorable environment. Electric steering is used in new technologies like automated parking and driving. A driverless vehicle could have the ability to automate this steering wheel.

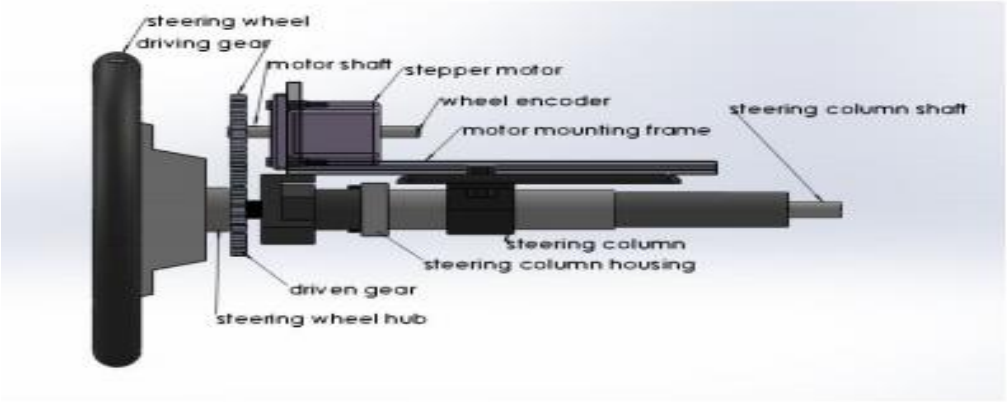


Figure 8: cad model [26].

6. The J-Net Architecture

The dimensions of the very first convolutional input were 320×64×3 (width × height × channels). We then applied three normalization operations along with scattering transformations for better generalization. A kernel of size 5×5 was employed using 32 feature maps in the initial convolutional layer. For richer feature extraction, three convolutional layers

were chosen in total. The initial convolutional layer was configured with 32 filters, followed by 64 filters in the second and 128 filters in the third layer.

Parameter efficiency was enhanced by deepening the network architecture. Increasing the depth of a neural network generally improves performance due to its ability to learn more abstract feature representations. In computer vision, deeper networks have demonstrated better efficiency because they can model hierarchical structures more effectively. For example, lower layers of a deep neural network detect simple features like edges or textures, while deeper layers capture more complex structures such as curves or object parts.

In this context, the layers were also designed to extract relevant environmental features required for autonomous navigation on a representative road trail. Since the task demanded more abstract and non-trivial features—beyond basic geometric primitives—we used a structure that includes three convolutional layers, followed by one flattening layer, and two fully connected layers. This configuration ensures robust feature capture and decision-making capacity while maintaining computational efficiency.

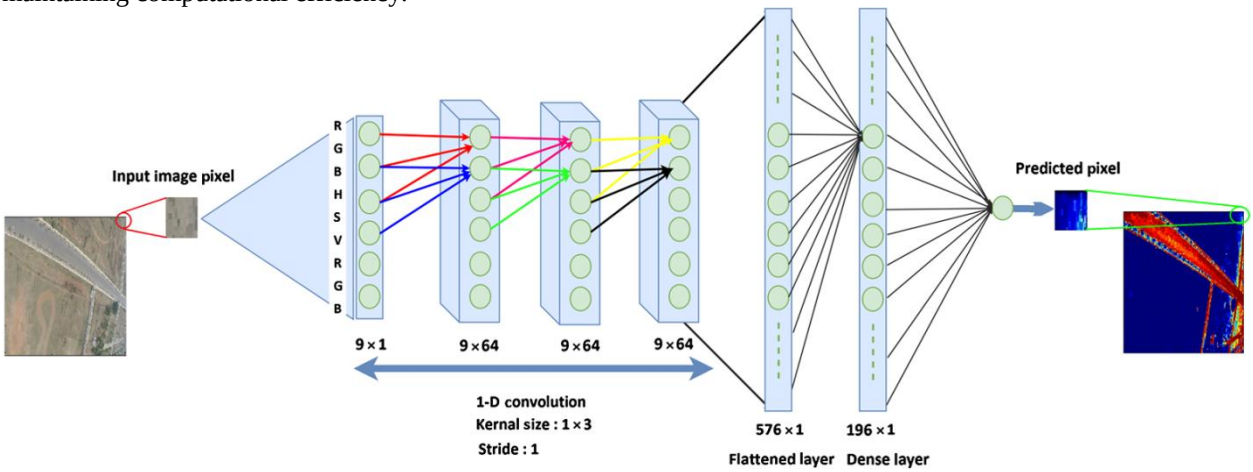


Figure 9: Architecture of a proposed deep neural network J-Net that can be used to drive autonomously.

Upper pooling has the first benefit: it does not rely on trainable parameters. This reduces the risk of overfitting, enhancing generalization. MaxPooling often yields a more robust feature representation. The downside is that it becomes computationally more challenging when convolutions operate at lower strides. Pooling layers also introduce new hyperparameters such as the pooling region size and stride length. The pooling layer operates independently on each depth slice of the input and spatially reduces its dimensions.

In this implementation, MaxPooling with a 2×2 kernel was selected. This downsamples each depth slice by a factor of 2, effectively reducing 75% of the activations while preserving the depth dimension. This technique helped reduce the number of trainable parameters, even though the core spatial structure of the feature map remained largely unaffected.

The same MaxPooling configuration was applied after the first and second convolutional layers. Following the second convolution, a third convolution layer with a kernel size of **5×5** was introduced. This setup resulted in approximately 17,920 trainable parameters at that stage. Each convolutional layer was paired with the ReLU activation function, as mentioned previously.

The final structure of J-Net comprised three convolutional layers, each followed by MaxPooling of size 2×2. After the last convolution, additional MaxPooling operations brought the total number of trainable parameters to approximately 31,280. These nodes were then connected to the final network layers, as we were constructing an agent-learning system.

The flattening layer that followed contained no trainable parameters; it only reshaped the multi-dimensional feature maps into a 1D array for the fully connected layers. The final DNN architecture ended with two fully connected layers:

- The first fully connected layer had 10 output nodes,
- The second and final layer had 1 output node used for regression in steering angle prediction.

To evaluate model performance, we re-implemented three benchmark networks: LeNet-5**,** AlexNet**,** and PilotNet, each trained on an identical dataset. These models served as baseline comparators to J-Net for the purpose of autonomous driving simulation.

The LeNet-5 model, unfortunately, showed weak performance—it failed to follow the driving path and was excluded from further evaluation. Both AlexNet and PilotNet exhibited robust navigation capabilities across the simulated track. PilotNet, in particular, demonstrated consistently high inference stability.

The system design comparison between J-Net**,** AlexNet, and PilotNet is illustrated below. Convolution operations were found to be computationally expensive, especially when aiming for a lightweight, deployable solution. Convolution adds a significant number of nodes and associated weights to the system. One mitigation strategy was to apply strided convolutions, allowing the filters to move by 2 pixels, thus reducing feature map size and computational cost.

However, reducing sampling frequency risks information loss. An alternative is pooling. Instead of skipping convolutions altogether, we used local pooling operations to combine neighborhood values. To reduce dense layer dimensions, we adopted MaxPooling after each convolutional stage, comparing local regions in each feature map and retaining the maximum activation.

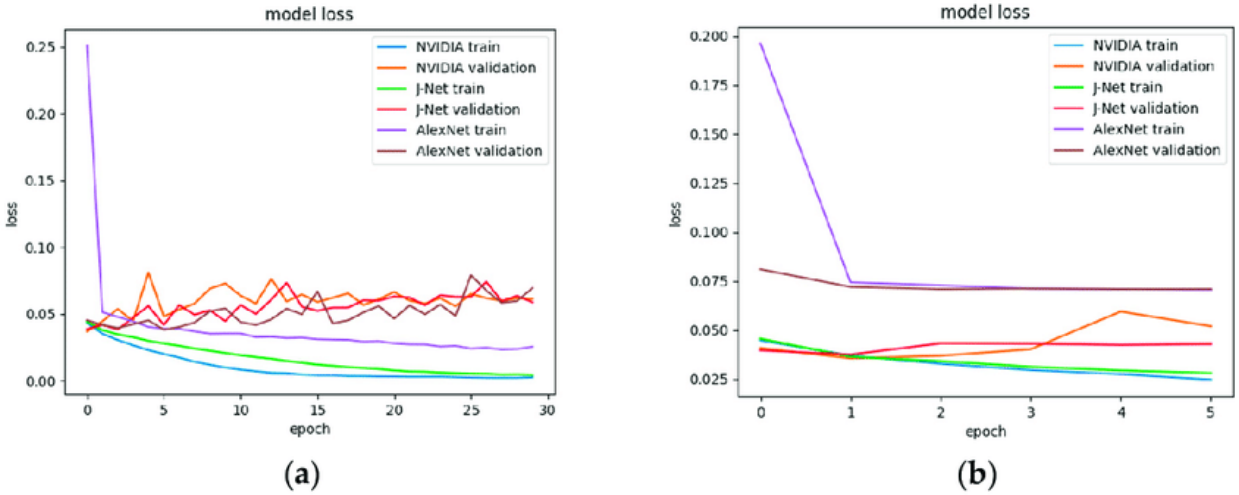


Figure 10: Here we compare the various deep neural network architectures we deployed and utilized for full-stack autonomous driving: (**a**) AlexNet; (**b**) PilotNet; (**c**) J-Net.

Details about AlexNet and PilotNet Reimplementation's: In our job, we implemented the AlexNet structure with the intent of learning autonomous driving. There are only two parallel computing pipelines in the original AlexNet architecture. This is because the original AlexNet had only two Graphics Processing Units (GPUs) were used to manage the heavy computational load, and we divided the convolution operations into two sections. We had adequate hardware capabilities to train a modified AlexNet model using a single modern GPU. As such, we adopted the structure of AlexNet but employed only a single stream of convolutions, maintaining the use of filter configurations compatible with the original implementation.

One major distinction between the original and our version was the input resolution. AlexNet originally used images sized 224×224 pixels, whereas in our case, the resolution was 320×160 pixels, and after preprocessing and border cropping, the effective input size became 320×64 pixels. This required us to make architectural adjustments to ensure compatibility.

When applying AlexNet to our data, we encountered layer overlap issues due to the reduced spatial size. The original model's early three pooling layers significantly decreased spatial resolution, but since our inputs were already reduced in size, further pooling risked eliminating crucial features. To resolve this, we removed the first two pooling layers from the original design and retained only a single MaxPooling layer after the first convolutional layer, along with 384 feature maps at that stage.

We tested our AlexNet variant using a dataset captured under the same environmental conditions as the original, ensuring a consistent benchmark. One key difference was the parameter count: while original AlexNet has over 60 million parameters, our streamlined variant used fewer than 3.2 million parameters. Furthermore, the original AlexNet concludes with 1,000 output neurons (for ImageNet classification), whereas our model ends with a single output node—linked directly to a steering control module.

NVIDIA PilotNet (sometimes referred to as NVIDIA CNN) is a vertically optimized convolutional neural network created by NVIDIA Corporation for autonomous vehicle applications, developed under the DARPA DAVE platform. Its architecture includes a normalization layer, followed by five convolutional layers and four fully connected layers.

The core structure remained unchanged in our reimplementation, but notable differences existed in layer width and parameter volume. While the sequence of operations—convolution, flattening, and dense layers—was preserved, the input dimension was reduced to enhance speed and compatibility with embedded systems.

Our variant of PilotNet was subjected to 30 full optimization cycles, alongside a lightweight version that ran for 6 optimization epochs. Interestingly, a sharp spike in validation loss was observed around the fifth epoch, indicating a clear inflection point in the learning process. Based on this behavior, training was terminated early and typically converged within 4 to 5 epochs for the lightweight variant.

For J-Net, the decision was made to extend training to 6 epochs based on empirical performance gains. This moderate-length training cycle allowed for sufficient convergence without overfitting and helped in streamlining the parameter pathways during training. Even though the required number of epochs for similar CNN models ranged between 4 and 10, the six-epoch configuration proved most effective for our simulation setup.

To support optimization and efficiency, we also applied hyperparameter pruning techniques, which simplified network training and improved stability. This design choice aligned with expectations from prior evaluations, which indicated that J-Net, with approximately 1.4 million trainable parameters, consistently emerged as the most balanced and efficient model for our autonomous driving task.

7. Results and Discussions

To examine how it stacks up against other deep neural networks like AlexNet and PilotNet, we ran a comparison on J-Net. This was done to ensure the expected results of the publishing layout. Each of the three network architecture models was developed and trained using the identical initial set of data. For the purpose of autonomous driving, these models were utilised to draw conclusions from the simulation. The trained version's size and quantity of trainable parameters were much larger than the untrained version's, and the system's performance was significantly different, making the operator follow suit.

Two static constants, or weights or model parameters, form the backbone of the implementation of these neural system units. The system's computing capacity is instantaneously defined by the structure and interactions between nodes. In conventional neural networks, the connections between neurons are incomplete. Among the key distinctions between natural and synthetic neural networks is this. The complexity of the system depends on the structure and filtering capabilities of the network's hidden layers. This is true of an untrained design model.

These neural networks are shown together with their layer structure and the total number of tunable parameters. Compared to AlexNet's 58,940,000 and PilotNet's 410,000 trainable parameters, and J-Net’s 205,000 trainable parameters, the differences are noteworthy. The system described in this study, called J-Net, contained around half as many trainable parameters as PilotNet and roughly 290 times fewer than AlexNet.

Throughout the process, we also performed several floating-point calculations. The following numerical criteria led to this conclusion:

- 58.94 million multiplications in AlexNet,
- 312.45 million multiplication operations in PilotNet, and
- 147.10 million multiplication operations in J-Net.

These models were also compared in terms of their scale:

- The memory requirements for PilotNet and J-Net variants were approximately 5.1 MB and 2.3 MB respectively,
- while AlexNet required 475.8 MB.

Each model was trained with the same optimizer, loss function, and data set. Variations were observed in terms of overfitting behavior, which is reflected in the training loss and identification ratios. Each version also required a different number of epochs to converge. The size of the trained version impacts the inference performance, especially in systems constrained by memory-limited embedded hardware. These factors are directly influenced by the system structure, including differences in layer dimensions, arrangement, and inter-layer connectivity.

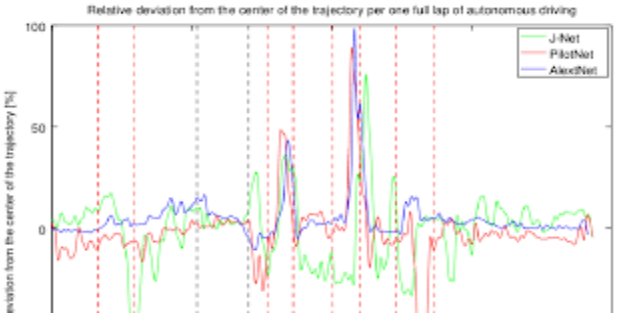


Figure 11: Percentage of a lap driven autonomously that deviates from the centered trajectory.

Histograms may be used for a statistical analysis of driverless car data. In the context of long-term assessments, this is crucial information. A histogram of all J-Net Driving data is now available. The biggest fluctuations were associated with the J-Net market, indicating that this network had the lowest degree of curve similarity. When compared to other networks, including those used for autonomous driving, the Internet's oscillations were far milder. Nonetheless, there were noticeable curve oscillations in this version. Despite the fact that both instructional unusual occurrences were about a hundred percent off course, this was not the case. Histogram showing relative departure from this trend. According to these results, the best consistent driving conditions were given by AlexNet for sovereign driving. It made extensive use of the subtle fluctuations in the centre as well. Curve management, on the other hand, reveals that trajectories sometimes deviate considerably from the mean. Although this detour was legal, it caused the car to take a different path from the one we had planned for self-driving vehicles.

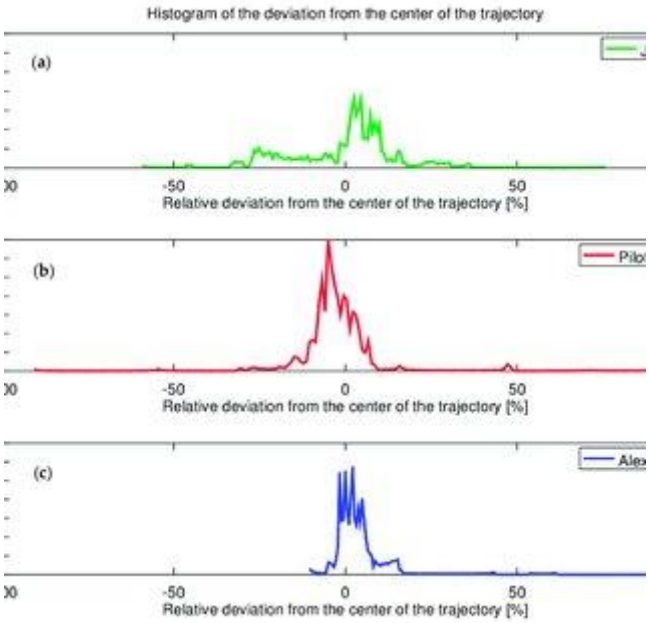


Figure 12: Histograms showing the amount of time an autonomous vehicle strays from the center of its trajectory over the course of a single lap. (a) J-Net; (b) Pilot Net; (c) AlexNet.

8. Conclusion

Tablets that can infer from and train system learning units have allowed for the creation of innovative solutions to previously intractable challenges. Machine learning systems that may be readily deployed on more compact and

inexpensive embedded platforms are in high demand for a wide range of industrial applications. Using a cheap version in terms of computing power and memory tools, machine learning units may be deployed on low-performing hardware stages. To do this, the neural network model structure should be meticulously planned. A growing trend is the creation of lighting networks that are compliant with demanding hardware specifications. As a result, new and improved hardware has been developed, and specialised chip units for system learning and deep learning have been developed. A neural system that can be trained for autonomous driving is described in this article. The goal of this research was to develop a compact, deep neural system that could be easily inferred and implanted in the automobile's later phases of production. It'll pave the way for highly autonomous vehicles. To accomplish autonomous driving on a typical route with far less computing resources than other well-known methods, we built and implemented J-Net. The answer presented in your paper is the most significant result of your study. In spite of its simple construction, it performs well in computing benchmarks. The algorithm becomes more difficult to understand as more and more actions are carried out throughout each iteration. In contrast to the other neural networks tested in this research, ours is more complicated, suggesting that same outcomes may be accomplished with fewer operations. J-drawbacks Unfortunately, there is one area where the Net falls short: it cannot adapt to ever-changing, complicated environments. We use raw camera pictures and steering data to train our model, but we ignore the vehicle's speed during training to save training time. This directly impacts the maximum speed at which an autonomous vehicle may go. It might be possible to train the J-Net to foresee future vehicle speeds. When applied to both speed and steering predictions, the same strategy may lead to improved precision in both. Input images in real time determine everything. A unique framework must be built and refined to safely anchor the car's mechanical parts, such as the accelerator, clutch, steering wheel, and brakes. The inputs necessary for the system to operate in the vehicle can come from the vision module. The technologies in question are completely controllable and need no downtime or human error. There might be a lag in the steering wheel's rotation due to the engine driveway. In the future, the vehicle's equipment system might be automated to enable faster speeds. The use of stepper motors and pneumatic cylinders would allow this to be accomplished.

9. Future Enhancement

Embedding this network in an electrical stage with constrained hardware resources and minimal chip power is a future goal. All robot-cars for warehouses and delivery trucks have been offered as prospective ultimate use cases for its proposed end to end learning system. In this study, we provide a lightweight DNN alternative that paves the way for deployment on embedded mobile platforms with hardware that meets the stringent requirements of low power consumption, cheap cost, and small footprint that are all essential in industrial settings.

References

[1] LeCun Y., Bottou L., Bengio Y., Haffner P. “Gradient-Based Learning Applied to Document Recognition”:2020.

[2] Simard D., Steinkraus P.Y., Platt J.C. “Best practices for convolutional neural networks applied to visual document analysis”:2021.

[3] Shin H., Roth H., Gao M., Lu L., Xu Z., Nogues I., Yao J., Mollura D., Summers R. “Deep Convolutional Neural Networks for Computer-Aided Detection”: CNN Architectures, Dataset Characteristics and Transfer Learning:2020.

[4] Pathak D., Krähenbühl P., Donahue J., Darrell T., Efros A.A. Context Encoders: “Feature Learning by Inpainting”:2021.

[5] Karpathy A., Toderici G., Shetty S., Leung T., Sukthankar R., Li F.-F. “Large-Scale Video Classification with Convolutional Neural Networks”:2000.

[6] Chi J., Kim H.-C, “Prediction of Arctic Sea Ice Concentration Using a Fully Data Driven Deep Neural Network”:2019.

[7] Russakovsky O., Deng J., Su H., Krause J., Satheesh S., Ma S., Berg A.C. “Imagenet large scale visual recognition challenge”:2000.

[8] Simonyan K., Zisserman A. “Very deep convolutional networks for large-scale image recognition”:2018.

[9] Szegedy C., Liu W., Jia Y., Sermanet P., Reed S., Anguelov D., Erhan D., Vanhoucke V., Rabinovich A. May.23.2022.

[10] He K., Zhang X., Ren S., Sun J. “Deep residual learning for image recognition”.2019.

[11] Visin F., Kastner K., Cho K., Matteucci M., Courville A., Bengio Y. Renet: “A recurrent neural network-based alternative to convolutional networks”. 2000.

[12] Zoph B., Vasudevan V., Shlens J., Le Q.V. “Learning Transferable Architectures for Scalable Image Recognition”. 2021.

[13] Acuna D., Ling H., Kar A., Fidler S. “Efficient Interactive Annotation of Segmentation Datasets with Polygon-RNN++”.Mar.16.2022.

[14] Wang T.C., Liu M.Y., Zhu J.Y., Liu G., Tao A., Kautz J., Catanzaro B. "Video-to-video synthesis".Nov.5.2023.
[15] Bojarski M., Del Testa D., Dworakowski D., Firner B., Flepp B., Goyal P., Jackel L., Monfort M., Muller U., Zhang J., et al. "End to end learning for self-driving cars". 2021
[16] Bojarski M., Yeres P., Choromanska A., Choromanski K., Firner B., Jackel L., Muller U. "Explaining how a deep neural network trained with end-to-end learning steers a car".2000.
[17] Mehta A., Adithya S., Anbumani S. "Learning end-to-end autonomous driving using guided auxiliary supervision". 2019.
[18] Chen Y., Wang J., Li J., Lu C., Luo Z., Xue H., Wang C. "LiDAR-Video Driving Dataset: Learning Driving Policies Effectively".Jan.8.2023.
[19] Ramezani Dooraki A., Lee D.-J. "An End-to-End Deep Reinforcement Learning-Based Intelligent Agent Capable of Autonomous Exploration in Unknown Environments".Jun.3.2022.
[20] Krizhevsky A., Sutskever I., Hinton G.E. "Imagenet classification with deep convolutional neural networks". 2023.
[21] Udacity, Inc. "Self-Driving Car Simulator". [(accessed on 5 November 2018)].
[22] Goodfellow I., Bengio Y., Courville A. "Deep Learning". The MIT Press; Cambridge, MA, USA: 2020.
[23] Aggarwal C.C. "Neural Networks and Deep Learning. Springer International Publishing; Cham, Switzerland".2021.
[24] Chollet F. "Deep Learning with Python. Manning Publications"; Shelter Island, NY, USA: 2021.
[25] Sutton R.S., Barto A.G. "Reinforcement Learning", The MIT Press; Cambridge, MA, USA: 2020.
[26] LeCun Y., Boser B., Denker J.S., Henderson D., Howard R.E., Hubbard W., Jackel L.D. "Back propagation applied to handwritten zip code recognition". May.23.2023.
[27] P. Goodman, "Advantages and disadvantages of driverless cars," Nov. 22 2022.
[28] A. A. Jose, C. A. S. Pillai et al., "A novel approach for scheduling and routing of the self-guided vehicles in mesh topology using velocity control and alternate path techniques," 2021.
[29] J. M. Anderson, "Self-driving vehicles offer potential benefits, policy challenges for lawmakers," Jan. 6 2019.
[30] Geem, M.H.,On strongly continuous ρh-semigroup,Journal of Physics: Conference Series, 2019, 1234(1), https://doi:10.1088/1742-6596/1234/1/012109 .
[31] Geem, M.H.,Hassan, A.R.,Neamah, H.I., 0-Semigroup of g-transformation Journal of Interdisciplinary Mathematics , 2025, 28(1), pp. 311–316.
[32] Alsaeedi, A.H., Al-Mahmood, H.H.R., Alnaseri, Z.F. et al. Fractal feature selection model for enhancing high-dimensional biological problems. *BMC Bioinformatics* 25, 12 (2024). https://doi.org/10.1186/s12859-023-05619-z .
[33] Shaimaa H.Mohammad, Israa Z. Chyad Alrikabi, Hayder Rahm Dakheel al- fayyadh, "Number Plate Recognition System Based on an Improved Segmentation Method", International Journal of Computational and Electronic Aspects in Engineering, RAME Publishers, vol. 6, issue 1, pp. 42-50, 2025. https://doi.org/10.26706/ijceae.6.1.20250207 .
[34] Serri Ismael Hamad, "Utilizing Convolutional Neural Networks for the Identification of Lung Cancer", International Journal of Computational and Electronic Aspects in Engineering, vol. 6, issue 1, pp. 35-41,2025. https://doi.org/10.26706/ijceae.6.1.20250206.
[35] Hiyam Hatem, "improved deep learning models for plants diseases detection for smart farming", international journal of computational and electronic aspects in engineering, vol. 6, issue 1,pp. 10-21, 2025. https://doi.org/10.26706/ijceae.6.1.20250204 .