Alkadhum Journal of Science, Vol. 1, No. 2 (2023)

Open Access ISSN: 3005-2386 (Online), ISSN: 3005-2378 (Print)

IRAQI Academic Scientific Journals

Alkadhum Journal of Science (AKJS)

Journal Homepage: https://alkadhum-col.edu.iq/JKCEAS

Alkadhum Journal of Science

Artificial Neural Network-Powered, Driverless Vehicle Concept Development

[1]Sahar R. Abdul kadeem, [2]Ali H. Naser*, [1]Ahmed R. Hassan, [3]Ghufran Abbas Betti

[1]National University of Science and Technology, Nasiriyah - Iraq

[2]Computer Department, Computer Sciences and Mathematics College, University of Thi-Qar - Iraq

[3]Electrical and Computer Engineering Department, University of Tabriz - Iran

Article information

Article history:
Received: November, 1, 2023
Accepted: December, 1, 2023
Available online: December, 14, 2023

Keywords:
convolutional neural network,
embedded systems,
deep neural network

Corresponding Author:
Ali H. Naser
prog.ali95@gmail.com

DOI:
https://doi.org/10.61710/akjs.v1i2.63

Abstract

Autonomous cars are now possible due to significant advances in robotics and intelligent control systems. Before these vehicles can safely operate in traffic and other hostile environments, there are many navigation, vision, and control issues. We want techniques that are both cost-effective and efficient, so that the field of research and academia may fully embrace self-driving cars. Within this scenario, we need something that can convert people to autonomous automobiles and include existing vehicles so that academics and explorers can access them. This study proposes a flexible mechanical layout that can be assembled in a short time and installed in most modern automobiles; it can also be used as a stepping stone in the development of autonomous vehicles. Using various actuators, conventional automobiles can be converted into autonomous vehicles. In the context of motor vehicle automation, motors are often used as actuators. In addition to motors, a pneumatic system was developed to automate the predetermined steps. An autonomous vehicle's mechanical arrangement is crucial, and it must be regularly updated and built to be robust in the face of dynamic conditions. We re-implemented two additional convolutional neural networks in an effort to conduct an objective test of their proposed network and compare our system's structure, technical complexity, and performance test during autonomous driving with theirs. This predicted network is around 250 times larger than the Alex Net network and four times larger than Pilot Net after training. Although the complexity and measurement of the publication's system are lower than other models that contribute lower latency and greater speed throughout inference, the operation was claimed by our system, which achieved autonomous driving with an equivalent efficacy as that achieved with two other models. The projected deep neural system reduces the need to infer ultra-fast computational hardware. This is important for cost efficiency, scale, and cost.

1. Introduction

There are four possible cube-shaped components of the autonomous driving system (detectors, comprehension subsystem, aiming subsystem, and vehicle control, Figure (1). Currently, the vehicle is conducting a global survey using its array of detectors. The understanding block output is sent into the preparation subsystem, where it is employed for behaviour preparation and long-term route planning [1]. 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 towards fully autonomous vehicles [2]. Many automotive technologies will reach maturity at the same time as cars gain more autonomy in the workplace. Delivery trucks and other types of robot cars and robots may park in warehouses as they wait to be sent. The fundamental objective of this research was to create a solution for lightweight automotive stage autonomous driving with minimal hardware components, processing power, and storage [3]. In spite of our familiarity with hardware limitations, we anticipate discovering a lightweight deep neural network (DNN), an end-to-end neural system capable of soon performing autonomous driving to the representative course. A version of the network used for inference may be deployed on a stage of hardware with limited processing power [4].

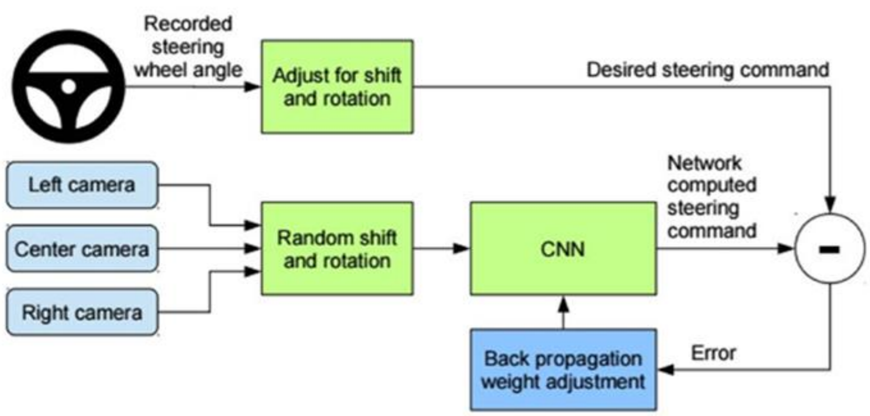

Figure 1: End- to-end autonomous transportation system block diagram

The challenge of creating safe, useful, long-lasting and enjoyable autonomous vehicles has boosted interest in the field more and more every day. Highways and other heavily travelled areas are perfect for the deployment of driverless vehicles [5]. The risk of accidents caused by other vehicles or pedestrians may also be reduced by autonomous vehicles. The Centre for Automotive Research at Stanford University (CARS) is one of numerous research facilities worldwide devoted to the creation of intelligent systems for autonomous vehicles. TORCS attracts a large user base of programmers and is used to host exciting competitions at annual conventions across the world [6]. The software provides a host that runs races with several cars on multiple screens simultaneously. The provider of a particular vehicle's activity may choose to publish a customer module. Gazelle, our driver, is taking part in the Genetic and Evolutionary Computation Seminar's 2013 TORCS competition. The cars are controlled by algorithms that evaluate information about the track, their opponents' vehicles, and their own performance to decide where they stand in the competition [7], Figure (2).

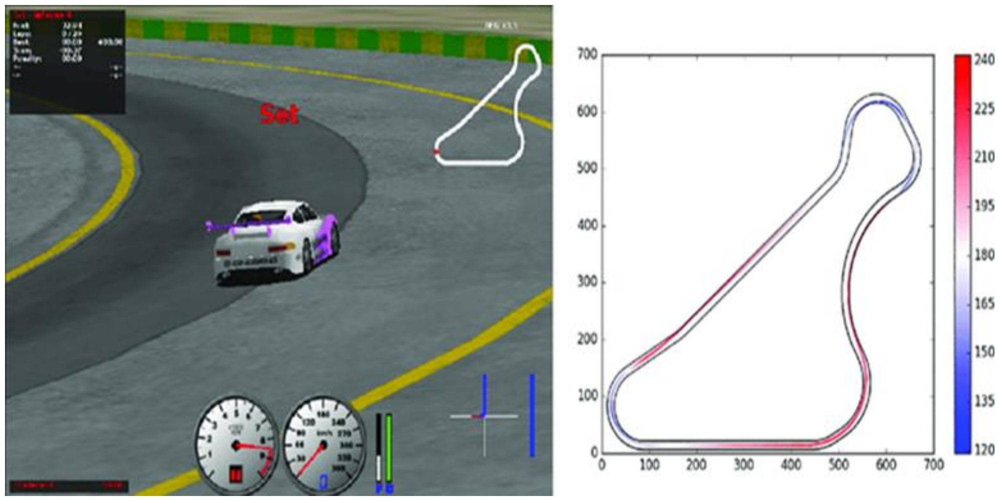

Figure 2: TORCS image during the race

Your customer's car and its specifics, such as its current location, how much time has elapsed since the race began, and the best time to complete a lapse, are displayed at the top of the screen. The low screen displays a different perspective on the race, which may make it easier to spot rival vehicles. We might even be able to locate some data on the car, such as its top speed and gear ratios [8]. EPIC control, as demonstrated by Guse and Vrajitoru, is central to the task at hand in this paper. The submission of Epic to the GECCO 2009 competition. The Epic driver relies on two parts: calculating the mark rate from the next frame and calculating the mark angle for rotation in each frame. The commanded angle is determined by the amount of clear air ahead, which is taken into account by the controller [9]. It also has a technology that detects fast turns and adjusts the target speed accordingly so that the vehicle does not veer off the road. To fit the speed parameters into new music tracks, it employs hill climbing strategies. Two new developments stem from incorporating system learning solutions into embedded hardware systems: the development of cutting-edge hardware systems able to design lightweight machine-learning architectures and variants that are compatible with low-performance hardware and the processing of data required for machine-learning inference [10]. Most well-known learning algorithms for autonomous driving systems have been developed for actual automobiles. The inferential system learning version is often installed in the trunk. Other methods rely on extremely deep neural networks, which are resource-intensive to compute. To achieve comparable performance for autonomous driving, we set out to develop a larger alternative, even a lighter profound neural system. However, this will make the embedded platform configuration possible. The embedded automotive systems and robot automobiles designed to move products will adopt this lightweight approach [11].

The findings demonstrate that J-Net is just as effective in autonomous driving as competing models, despite its lesser size and complexity. A computationally lightweight solution to the DNN used for autonomous driving is crucial due to the ongoing deployment of the embedded automotive stage [12]. This is because embedded technologies may be constrained by on-board computers' inability to run cutting-edge deep learning models. The Epic code used by the Gazelle driver improves both these areas. The goal of any of the numerous approaches to acquiring track projections is to optimise the operation. The path segmentation method is a good illustration of this. Specifically, this means segmenting the route into known polygons. The track is then reconstructed by your control system from these polygons [13]. This control is straightforward and incorporates modules that limit things like gear shifting, steering movements, and pedal orders. The most crucial aspect of this role is the opponent modifier. Nearby opponents' driving behaviour is altered as a result, with instantaneous adjustments made to their steering and braking [14]. The most up- to-date control system AUTOPIA is employed in the competition. It provides the full driving structure, including six key duties such as stuck place manager, target rate conclusion, opponent modifier, learning module, and controllers for steering, pedals, and steering. For the opponent modifier that relies on heuristic principles, it provides a robust and straightforward framework [15]. You can utilise a wide variety of learning techniques to figure out the fastest route for your automobile and win the race in record time. In this post, we look at an empirical approach to education. This evolutionary approach is self-adjusting; thus, it may be used to find out what factors influence the success rate. It is simple to use and extrapolate [16]. The driver lacks a rival management system. A second regulator, one that can be used in conjunction with evolutionary learning, has been introduced. This dial uses an evolutionary learning-based method to determine the quickest route to your car. Hyperheuristics have recently been incorporated into a novel learning strategy for optimising the road ahead of cars. The method analyses the effectiveness of different approaches to real-value optimisation for TORCS-based automotive systems [17]. When combining two heuristics, a hyperheuristic framework can be used as a guide. In the TORCS environment, hyperheuristics are effective. Additionally, software engineers will be incorporated into their go-to platform for learning artificial neural networks [18]. The NNs are used to build an iterative control plan that determines the vehicle's course and the rate at which it makes marks. The NNs were educated with player-provided data. Work is gratifying because it encourages the development of critical thinking skills and the selection of novel approaches to problems, but the potential rate is lower than that of those travelling along the same pathways [19]. Here we demonstrate a neural system that has the knowledge of a book and the ability to drive a car without human input. It is also possible to use J-Net to create embedded automobile systems we also describe the results of the reimplementation of Pilot Net and provide details on the unbiased study of J-Net. To get started, J-Net released the framework for a deep neural network. J-Net is now in production. This version has also been trained, so autonomous driving can begin once more [20]. The plan is to give an unbiased opinion on the structure of the network. All the data we have so far are used to train the models. The trained models could be used for autonomous driving in a simulated environment. Three models of complicated brain networks are shown, each of which has the potential to be utilised as a demonstration of autonomous driving. This article draws on videos showing three models of robust neural systems driving autonomously through a realistic route in a driving simulator. Throughout this deduction, quantitative and qualitative assessments of autonomous driving parameters are made, and the outcomes are given [21].

2. Literature Review

Deep learning is not merely a paradigm for machine learning. Profiling learning data is also an integral component of a larger group of machine learning processes. Representations at one layer of a deep neural system are expressed in terms of other, speedier representations at preceding layers. Convolutional networks will constitute the gradients at the center of neural networks [22]. CNN integrates two architectural concepts, namely local representative areas and shared burdens. Plasma or temporal Subsampling contributes to the invariance of scale, shift, and distortion. Convolutional neural networks have been created to process data from a variety of arrays, including colour image, Figure (3), speech, and sound spectrogram (and video). They also benefit from these indicators' characteristics: localized relationships, shared burdens, pooling as well as the use of multiple layers. CNNs are used mainly to evaluate visual vision [23]. Deep instruction for computer vision is critical in many industrial and commercial applications. 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 [24].

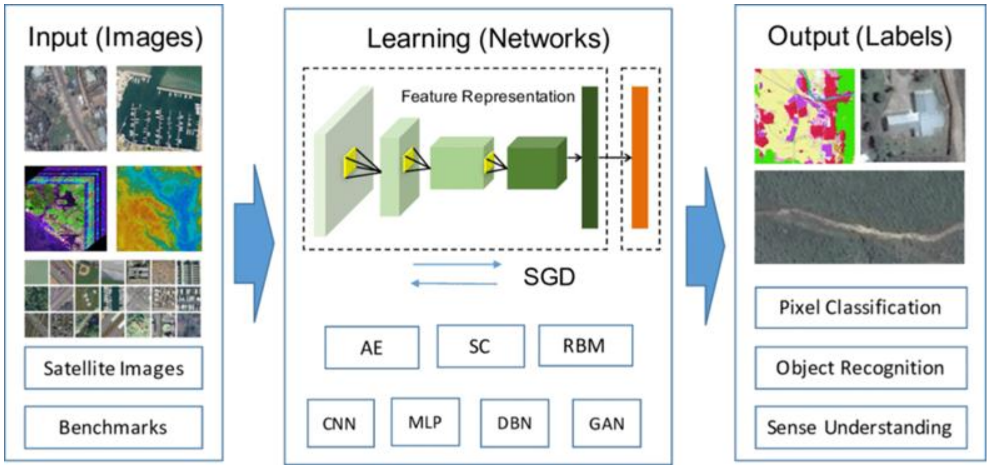

Figure 3: Flow of deep learning development

The Image Net Big Scale Visual Recognition Challenge contributed significantly to the advancement of convolution neural networks. The successful designs in this contest represent the cutting edge of deep learning and neural networks [25]. They provide a springboard for creative thought and a foundation for reconsidering existing problems. Automated machine learning, video-to-video synthesis, deep learning utilising artificial information, and go playing are all seminal works in the field of deep learning. In the 1950s, the first serious attempts to construct autonomous vehicles were made [26]. In 1984, the first fully autonomous automobiles appeared on the road. The defence Advanced Research Projects Agency's battle for autonomous cars, the Grand Challenge competitions held in 2004 and 2005, and the Urban Challenge held in 2007 were all crucial in establishing that robots are capable of performing the challenging individual task of driving. While it is true that autonomous vehicle prototypes can be tested on public roads, a number of issues must be resolved before the technology can be extensively adopted. Currently, several obstacles stand in the way of wholly autonomous vehicles [27]. The 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 systematically compare and contrast the various deep learning architectures utilised in autonomous cars. Sensor and detector combinations used in autonomous cars will also be detailed [28].

3. System for Fully Autonomous Driving

Our method utilised end-to-end learning to develop a strategy for autonomous transportation. Our strategy for autonomous driving was based on the image, raw pixels, and output. The constraint of the vehicle is the steering wheel, Figure (4). In the concluding stage of learning, the system will learn to control an automobile using only real-time camera inputs [29].

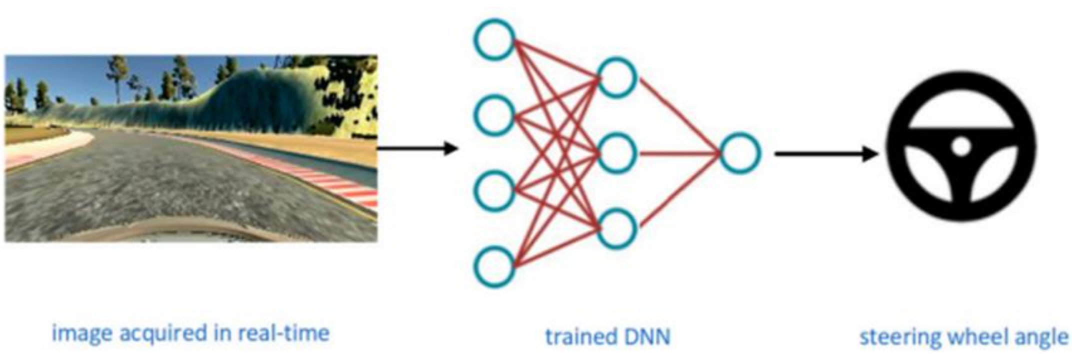

Figure 4: Autonomous driving in real time to do this, the trained deep neural network model is given data from the main camera.

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 utilised an 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. Using the manual driving approach, we recorded camera images and steering values for each chassis. The visuals served as the feature set and guiding parameters for the tag gathering process [30]. The vehicle 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 behaviour 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. During real-time autonomous driving, the same simulated environment was utilised as during training. Success in sovereign driving was determined by the amount of time the vehicle spent on the representative route. Our autonomous driving framework is organised like this [31].

3.1 Design

Pneumatic Circuit: As shown in Figure (5), the pneumatic circuit has three cylinders: brake, accelerator, and clutch. A 5/3 management control valve controls the cylinders. A microcontroller can be used to control the valves. The tanks can be actuated by turning the knobs. These can be activated individually or in combination depending on the need of your truck or car. The input signal is sent through a microcontroller [32].

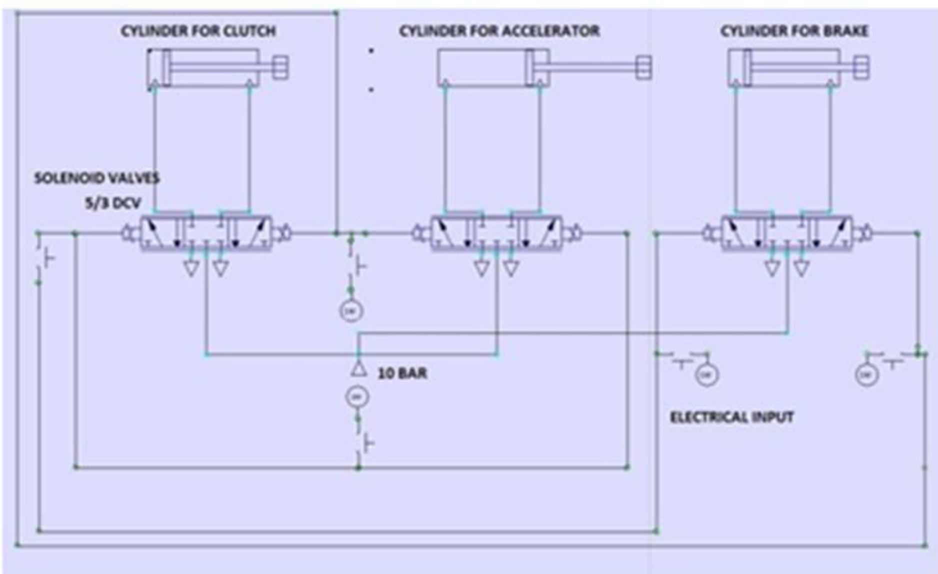

Figure 5: Pneumatic Circuit

The framework, mountings, and arrangement were all included in the CAD drawing for ease of reference. Three tanks were available for each variant. As shown in Figure (6), the framework is bolted to the automobile and electrons are installed.

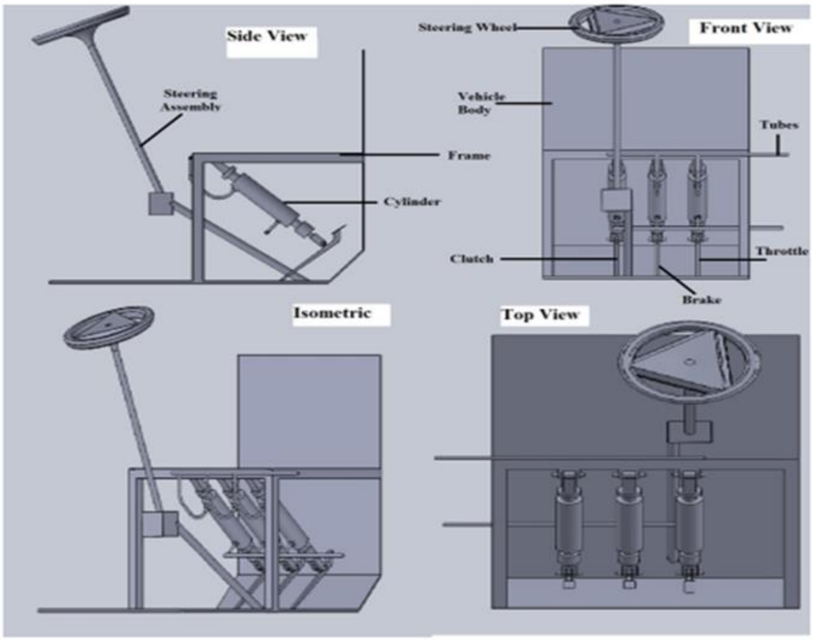

Figure 6: Accelerator, brake, and clutch schematics drawn in cad.

The pneumatic cylinders are connected to the suggested CAD model, which is manufactured in an automobile and can be analyzed with pneumatic inputs, as shown in Figure (7) The management control valves and are connected to the air reservoir.

Figure 7: Fabricated Parts

3.2. Steering system

The CAD version was designed to create the stepper motor and all the equipment pushing the steering group. The engine shaft mounts the driving gear and the powered gear. A framework was used to attach the engine to the car body. The framework does 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 the spinning wheel. This is used to determine the feedback of the wheel and convert it into closed-loop feedback. As shown in Figure (8), the CAD model clearly defines the location of the meeting and drive, as well as the framework in which the engine is mounted. (9) The engineered frame is fabricated and placed within the framework. The engine was operated at different speeds to analyze the frames. The engine can be contained within the framework at high speeds, and the gears may have been engaged throughout. Steering is an essential part of automobile dynamics. It controls the navigation and the direction of the vehicle. Most cars use hydraulic steering, which leaves behind old mechanical systems. A few cars now have electric power-steering systems. Electric power steering offers various benefits, including less maintenance, flexibility, speed, energy conservation, and a favourable environment. Electric steering is used in new technologies such as automated parking and driving. A driverless vehicle can automate its steering wheel [33].

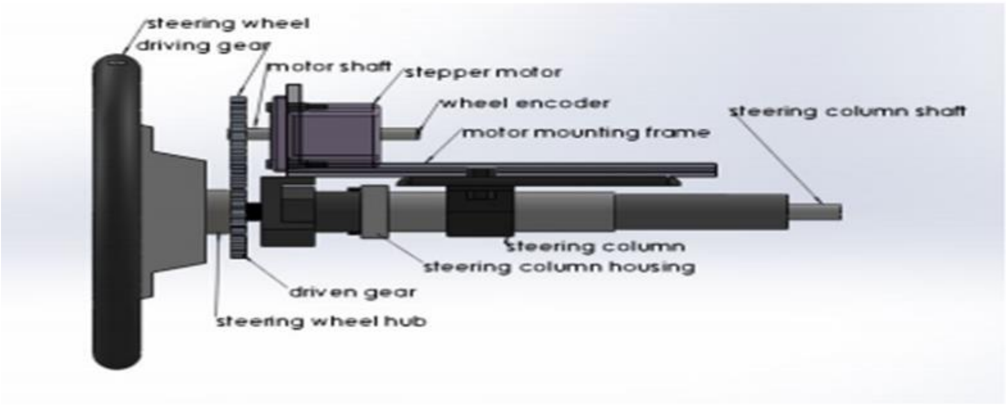

Figure 8: Cad model

3.3 The J-Net Architecture

The dimensions of the very first convolution were 320x65x3, then we picked three Normalizations and scattering. We then applied the kernel size of 5x5 using 16 feature maps. We have chosen three convolutional to extract more feature information. The primary convolutional coating had been 12 16. Parameter efficacy was improved by presenting more layers to a deep neural system. Simply by moving deeper than you would with other parameters, it is more likely to achieve greater performance. Graphics have also seen profound neural networks become more efficient. This is because graphics can capture hierarchical arrangements that deep models cannot capture. Simple features such as borders or lines are captured by the lower layers of a heavy neural network. Additional layers can extract more complex features, such as geometric contours. The layers are also used to yank items. The job required features that could be pulled. These were not the most basic capabilities, or geometric contours, since the automobile was to be inducted on a representative trail. We have chosen to use three convolutional layers in the final version. This is followed closely by one flattened layer and two fully connected layers [34].

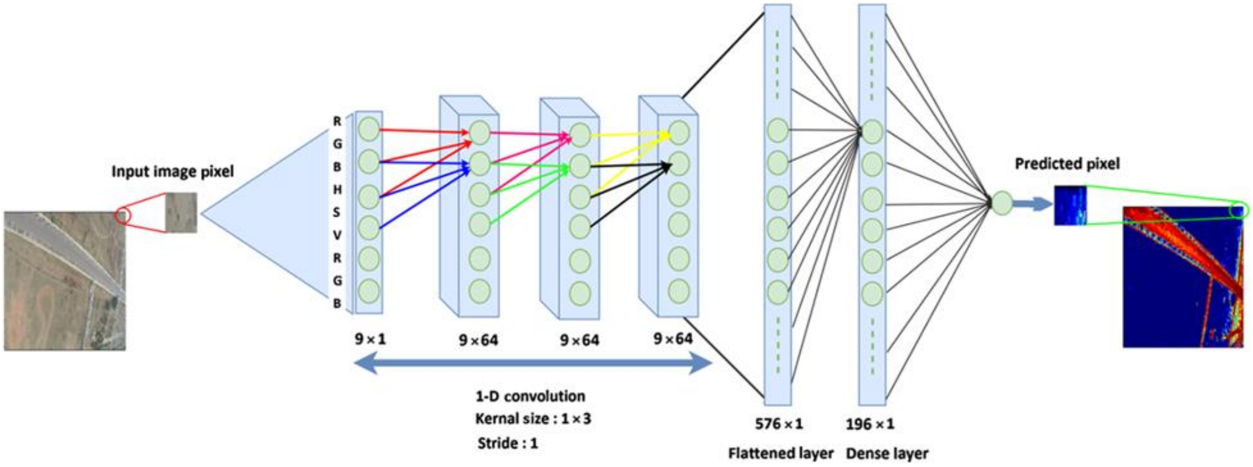

Figure 9: Architecture of a proposed deep neural network J-Net that can be used to drive autonomously.

The primary advantage of upper pooling is that it is independent of the parameters. This reduces the likelihood of over-matching. Maximum pooling frequently results in a more accurate version. If the convolutions move at a slower rate, it becomes more difficult to compute the version. A new layer is also conceivable because it adds additional hyperparameters to the songs, such as pooling area dimensions and rotational stride. The pooling coating operates separately on each thick segment of the input signal and spatially resizes it. MaxPooling with dimensions of 2 × 2 was chosen. This reduces the sample size of each thickness segment in the input 2 image, Figure (9). Additionally, it reduced the activation by 75 %. The thickness measurement remained unchanged. This case allowed us to reduce the number of trainable network parameters even though the feature map did not significantly interfere. After the initial convolution, the subsequent convolution coating had the same MaxPooling as that of the initial convolution. The size of the third kernel after the second convolution was 5 × 5 pixels. This resulted in the creation of 18,496 trainable parameters. The final solution of the J-Net was composed of three convolutional layers. Each stratum includes the ReLU activation function described in the previous subsection. MaxPooling's coating magnitude was 2 × 2. Following the final convolutions, convolution was applied and MaxPooling surgery was performed on 32,544 trainable parameters. As we developed the agent-study system, we had to connect these nodes to the final node. However, the compacted layer lacks new parameters. It only made a single correction to the existing conditions. The final two layers of the DNNs consisted of only two connected layers. This means that the first layer utilised ten output nodes, whereas the second layer utilized only one. We re-implemented three neural network units, LeNet-5, Alex Net, and Pilot Net, to assist with autonomous driving learning. This allowed us to evaluate the performance of our goal. It is conceivable to evaluate J-Net objectively in comparison to other network architectures. All models were trained using the same created dataset, and their exact conditions were described in the simulator [35]. Unfortunately, the LeNet-5 model, autonomous navigation was weak. The vehicle was excluded from further inspection because it could not follow the trace. According to the Inference with Pilot Net, the autonomous driving capabilities of Alex Net and Pilot Net were remarkable throughout the course. The system architectures of J-Net Alex Net and Pilot Net are compared below, Figure (10). Convolution is an expensive process because an illumination solution is desired. It adds a substantial number of system nodes and weights associated with the down sampling of each node. Convolution is a possible solution to this problem. This allows filter adjustments of a few pixels at a time and reduces the size of the feature map. However, reducing the sampling rate of a graphic can lead to the loss of some essential features and a reduction in data. The proposed pooling operation is the second method of graph sampling. Instead of bypassing any convolutions, we took a stride with all the adjacent convolutions and merged them. Following each convolutional layer, we implemented maximal rotational operations to reduce the size of the dense-neural-network layers. Maximum coating depth: Every point on this feature map was compared to a small community and all answers were provided.

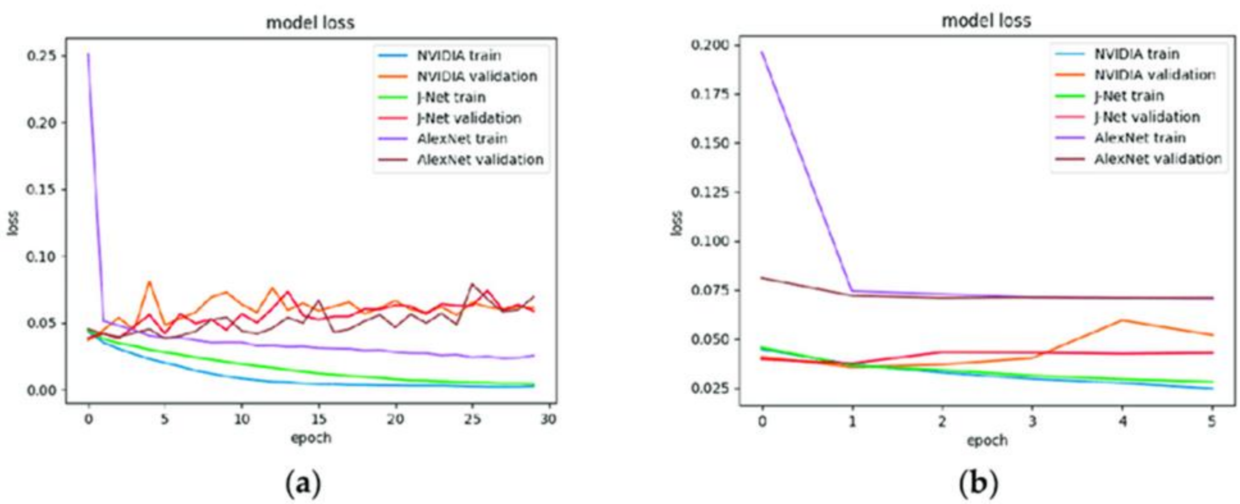

Figure 10: Here, we compare the various architectures of deep neural networks that we deployed and utilised for full-stack autonomous driving: AlexNet, PilotNet, and J-Net.

This section provides details regarding the re-implementations of AlexNet and PilotNet. During the course of our research, we undertook the task of re-implementing the AlexNet architecture with the aim of gaining expertise in the field of autonomous driving. The original AlexNet architecture consists of a limited number of two parallel computation pipelines. Due to the limited availability of two graphics processing units (GPUs) in the original AlexNet architecture, the convolutional operations were partitioned into two distinct sections. Our available hardware is adequate for training AlexNet using a solitary GPU. Hence, we conducted a replication of the AlexNet architecture and utilised a solitary convolutional pathway. Furthermore, we employed various filters that were compatible with the initial solution. The difference in size between the input in the initial design and our implementation is another distinguishing factor. The dimensions of the AlexNet architecture were 224 × 224 pixels. Nevertheless, the dimensions of our instance are 320 × 160 pixels. Furthermore, extraneous information originating from the border pixels was eliminated. The capabilities of our implementations are limited to handling images that possess a resolution no greater than 320 × 65 pixels. During the implementation of AlexNet, it was observed that there were two layers that overlapped with each other. This discrepancy in the input can be attributed to the notable disparity. While the inclusion of three initial pooling layers substantially augmented the pixel count for convolutions, the simplicity of our input necessitates consideration of whether there exists an adequate number of pixels to accommodate multiple convolutional layers with initial kernels. The system was streamlined and modified in order to align with our specific requirements. In order to achieve this, the initial two aggregating layers of the original system are removed. After the forward convolutional layer, a maximum pooling layer is introduced, followed by the addition of 384 features. The complete range of input parameters was ascertained by analyzing photographs captured both before the construction of the original structure and during our subsequent redesign. The quantity of trainable parameters has been diminished in comparison to the initial configuration. The sole discrepancy between the original structure and our implementation of Alex Net may have been the preceding coating. The original Alex Net architecture comprised of a total of 1,000 nodes, corresponding to the 1,000 classes in the Image Net competition output. The application necessitates the utilisation of a solitary output node, specifically designated for the purpose of steering prediction in the vehicle controller. Pilot Net, alternatively referred to as NVIDIA CNN, was developed by investigators from NVIDIA Corporation. The utilisation of deep neural networks enables the achievement of the utmost degree of autonomous driving. The objective is to improve the DARPA Autonomous Vehicles (DAVE) platform. The architecture of Pilot Net comprises a normalization layer, followed by five convolutional layers, and concludes with four fully connected layers. The key differentiation between the Pilot Net architecture and our implementation of Pilot Net lies in the utilisation of a reduced number of layers and a broader set of parameters as inputs to our system. The order of the remaining convolutional, flattened, and fully connected layers remained unchanged. Every characteristic was allocated to a convolutional layer. The variation in the quantity and types of trainable parameters is attributed to the smaller size of the original Pilot Net input. The entire Pilot Net system underwent a reimplementation process subsequent to its initial implementation. Pilot Net can be accessed during the 30th and 6th epochs. Starting from the fifth epoch, there was a culmination of validation loss. There is a notable temporal gap between the period of training and the occurrence of childbirth. The implementation of Pilot Net necessitated the adoption of a premature termination procedure, along with the selection of four training epochs. The model was validated through four epochs of sovereign driving, which revealed that it performed well as a coach. The empirical option for your J-Net version training was to add six epochs. The J-Net version was trained with fewer epochs than the original. This could range from four to ten epochs with equivalent results. Six epochs were selected, and their identification of this version enabled robust autonomous driving. Each utilised hyperparameter pruning techniques to facilitate the training of neural networks and to facilitate the autonomous driving task on a representative route in a simulation. On the basis of the trainable parameters enumerated in the preceding section, it was anticipated that J-Net, which had only 1.8 million B, would be the best version .

4. Results and Discussions

To examine how it stacks up against other deep neural networks such as 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 an identical initial set of data. For the purpose of autonomous driving, these models were utilised to draw conclusions from the simulation, Figure (11). 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 connexions between neurones 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 hidden layers. This is true for 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 42,452,305 and PilotNet's 328,219 trainable parameters and J-150,965 Net's trainable parameters, the numbers are strikingly different in the table below. The system described in this study, called J-Net, contained around half as many trainable parameters as PilotNet and roughly 280 times less than AlexNet. Throughout the process, we also performed several floating-point calculations. The following numerical criteria led to this conclusion: 42.45 million multiplications, the same number of AlexNet improvement operations; 347.82 million multiplication operations; inclusion in PilotNet, which also contained around 150.84 million multiplication operations; inclusion in its J-Net.

These models were also compared in terms of their scale: The memory requirements for PilotNet and J-Net variants were 4.2 and 1.5 MB, respectively, while AlexNet needed 509.5 MB. Each model was trained with the same optimiser and loss feature as well as the same data set. There were differences in the version overfitting. This will be shown by the training loss and identification ratios. Each version required a different number of epochs. The size of the trained version has an impact on the inference due to memory limits for embedded hardware components. These factors directly influence the system structure, which includes differences in form, dimensions, and interactions between layers

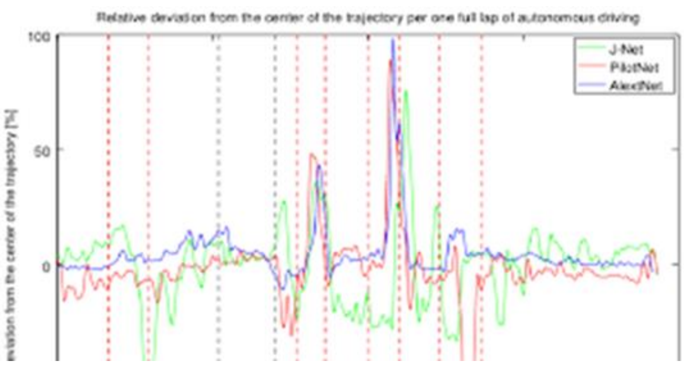

Figure 11: Percentage of a lap driven autonomously that deviates from the centered trajectory.

Histograms may be used for 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 Figure (12). 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. Histograms show a 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 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.

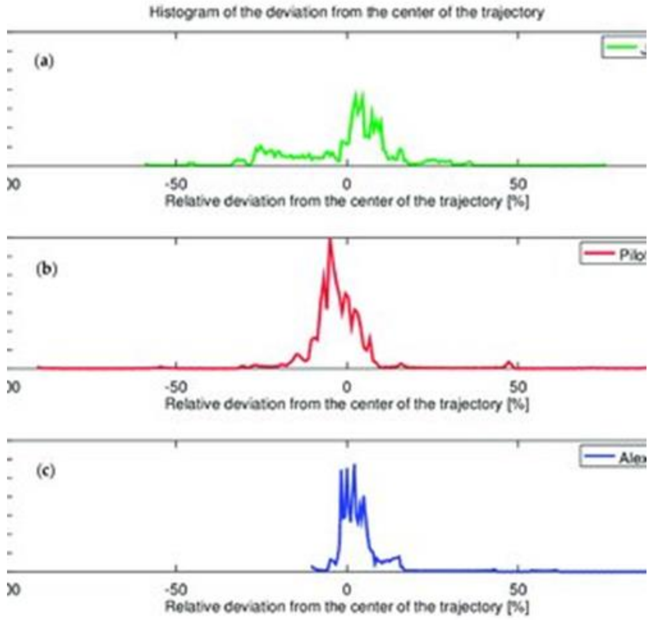

Figure 12: Histograms showing the amount of time an autonomous vehicle strays from the center of its trajectory over the course of a single lap. (a) J-Net; (b) Pilot Net; (c) AlexNet.

5. Conclusion

Tablets that can infer from and train system learning units have allowed 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 can 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 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 automobiles during the later phases of production. It will 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 the 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 rotation due to the engine driveway. In the future, vehicle equipment systems might be automated to enable faster speeds. The use of stepper motors and pneumatic cylinders would allow this to be accomplished.

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 the 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, low cost, and small footprint that are all essential in industrial settings.

References

[1] X. Han, H. Zheng, and M. Zhou, "Card: Classification and regression diffusion models", *Advances in Neural Information Processing Systems*, vol. 35, pp.18100-18115, 2022.

[2] a. Khan, A. Sohail, U. Zahoora, and A. S. Qureshi, "A survey of the recent architectures of deep convolutional neural networks", *Artificial intelligence review*, vol. 53, pp.5455-5516, 2020.

[3] J. Qi, X. Liu, K. Liu, F. Xu, H. Guo, X. Tian, ... and Y. Li, "An improved YOLOv5 model based on visual attention mechanism: Application to recognition of tomato virus disease", *Computers and electronics in agriculture*, vol.194, pp.106780, 2022.

[4] K. He, X. Chen, S. Xie, Y. Li, P. Dollár, and R. Girshick, "Masked autoencoders are scalable vision learners". In Proceedings of the IEEE/CVF conference on computer vision and pattern recognition, pp. 16000-16009. 2022.

[5] L. M. Dang, K. Min, H. Wang, M. J. Piran, C. H. Lee, and H. Moon, "Sensor-based and vision-based human activity recognition: A comprehensive survey", *Pattern Recognition*, vol. 108, pp. 107561, 2020.

[6] G. Zheng, X. Li, R. H. Zhang, and B. Liu, "Purely satellite data–driven deep learning forecast of complicated tropical instability waves", *Science advances*, vol. 6 no.29, pp. 1482, 2020.

[7] W. Wang, E. Xie, X. Li, D. P. Fan, K. Song, D. Liang, ... and L. Shao, "Pyramid vision transformer: A versatile backbone for dense prediction without convolutions". In Proceedings of the IEEE/CVF international conference on computer vision, pp. 568-578, 2021.

[8] J. Van der Laak, G. Litjens, & F. Ciompi, "Deep learning in histopathology: the path to the clinic". *Nature medicine*, vol. 27, no.5, pp.775-784, 2021.

[9] M. H. Guo, T. X. Xu, J. J. Liu, Z. N. Liu, P. T. Jiang, T. J. Mu, ... & S. M. Hu, "Attention mechanisms in computer vision: A survey". *Computational visual media*, vol. 8, no.3, pp. 331-368, 2022.

[10] C. Y. Wang, A. Bochkovskiy, & H. Y. M. Liao, "YOLOv7: Trainable bag-of-freebies sets new state-of-the-art for real-time object detectors". In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pp. 7464-7475. 2023.

[11] S. Minaee, Y. Boykov, F. Porikli, A. Plaza, N. Kehtarnavaz, & D. Terzopoulos, "Image segmentation using deep learning: A survey". *IEEE transactions on pattern analysis and machine intelligence*, vol. 44, no. 7, pp. 3523-3542, 2021.

[12] Y. Li, Y. Zhang, R. Timofte, L. Van Gool, L. Yu, Y. Li, ... & X. Wang, "NTIRE 2023 challenge on efficient super-resolution: Methods and results". In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pp. 1921-1959. 2023.

[13] D. Acuna, H. Ling, A. Kar, & S. Fidler, "Efficient interactive annotation of segmentation datasets with polygon-rnn++". In Proceedings of the IEEE conference on Computer Vision and Pattern Recognition, pp. 859-868, 2018.

[14] T. C. Wang, M. Y. Liu, J. Y. Zhu, G. Liu, A. Tao, G. Kautz, & B. Catanzaro, "Video-to-video synthesis", arXiv preprint arXiv:1808.06601, 2018.

[15] M. Bojarski, D. Del Testa, D. Dworakowski, B. Firner, B. Flepp, P. Goyal, ... & K. Zieba, "End to end learning for self-driving cars", arXiv preprint arXiv:1604.07316, 2016.

[16] M. Bojarski, P. Yeres, A. Choromanska, K. Choromanski, B. Firner, L. Jackel, & U. Muller, "Explaining how a deep neural network trained with end-to-end learning steers a car". arXiv preprint arXiv:1704.07911, 2017.

[17] A. Mehta, A. Subramanian, & A. Subramanian, "Learning end-to-end autonomous driving using guided auxiliary supervision". In Proceedings of the 11th Indian Conference on Computer Vision, Graphics and Image Processing, pp. 1-8, 2018.

[18] Y. Chen, J. Wang, J. Li, C. Lu, Z. Luo, H. Xue, & C. Wang, "Lidar-video driving dataset: Learning driving policies effectively". In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 5870-5878. 2018.

[19] A. Ramezani Dooraki, & D. J. Lee, "An end-to-end deep reinforcement learning-based intelligent agent capable of autonomous exploration in unknown environments", *Sensors*, vol.18, no. 10, pp. 3575, 2018.

[20] Y. Zhang, J. Gao, & H. Zhou, "Breeds classification with deep convolutional neural network". In Proceedings of the 2020 12th International Conference on Machine Learning and Computing, pp. 145-151, 2020.

[21] K. S. George, A. Abhiram, A. Jose, A. S. Madhav, N. Najiya, & S. Aswin, "Steering Angle Estimation for an Autonomous Car". In 2022 IEEE International Conference on Signal Processing, Informatics, Communication and Energy Systems (SPICES), Vol. 1, pp. 168-173, IEEE. 2022.

[22] K. Yao, & Y. Zheng, "Fundamentals of Machine Learning. In Nanophotonics and Machine Learning: Concepts, Fundamentals, and Applications" *Cham: Springer International Publishing*, pp. 77-112, 2023.

[23] M. Trimboli, L. Avila, & M. Rahmani-Andebili, "Reinforcement learning techniques for MPPT control of PV system under climatic changes. In Applications of Artificial Intelligence in Planning and Operation of Smart Grids", *Cham: Springer International Publishing*, pp. 31-73, 2022.

[24] M. Alami Machichi, L. El Mansouri,Y. Imani, O. Bourja, R. Hadria, O. Lahlou, ... & F. Bourzeix, "CerealNet: A Hybrid Deep Learning Architecture for Cereal Crop Mapping Using Sentinel-2 Time-Series". *In Informatics*, Vol. 9, no. 4, p. 96, MDPI,2022.

[25] S. A. Fayaz, S. Jahangeer Sidiq, M. Zaman, & M. A. Butt, "Machine learning: An introduction to reinforcement learning". *Machine Learning and Data Science: Fundamentals and Applications*, PP.1-22, 2022.

[26] S. S. A. Zaidi, M. S. Ansari, A. Aslam, N. Kanwal, M. Asghar, & B. Lee, "A survey of modern deep learning based object detection models", *Digital Signal Processing*, vol. 126, pp. 103514, 2022.

[27] H. K. Chen, & D. W. Yan, "Interrelationships between influential factors and behavioral intention with regard to autonomous vehicles". *International journal of sustainable transportation*, vol. 13, no.7, pp.511-527, 2019.

[28] E. A. Oyekanlu, A. C. Smith, W. P. Thomas, G. Mulroy, D. Hitesh, M. Ramsey, ... & D. Sun, "A review of recent advances in automated guided vehicle technologies: Integration challenges and research areas for 5G-based smart manufacturing applications". *IEEE access*, vol. 8, pp. 202312-202353, 2020.

[29] E. Jones, P. R. Devaraj, J. J. Smart, & D. K. Pagoti, "Automation In Long-Haul Trucking-Current State Overview, Limitations And Disruptive Effects". *International Supply Chain Technology Journal*, vol. 6, no. 03. 2020.

[30] J. P. Prasad, "AI Based Wireless Sensor Networks in Real Time Traffic Monitoring using Spherical Grid Routing Protocol". 2021.

[31] M. A. Khan, H. El Sayed, S. Malik, M. T. Zia, N. Alkaabi, & J. Khan, "A journey towards fully autonomous driving-fueled by a smart communication system". *Vehicular Communications*, vol. 36, pp.100476. 2022.

[32] Q. Lu, L. Chen, S. Li, & M. Pitt, "Semi-automatic geometric digital twinning for existing buildings based on images and CAD drawings". *Automation in Construction*, vol. 115, pp. 103183. 2020.

[33] C. Zhou, X. H. Liu, & F. X. Xu, "Intervention criterion and control strategy of active front steering system for emergency rescue vehicle". *Mechanical Systems and Signal Processing*, vol. 148, pp.107160. 2021.

[34] L. Zhang, X. Zhang, C. Bao, & K. Ma, "Wavelet J-Net: A Frequency Perspective on Convolutional Neural Networks". In 2021 International Joint Conference on Neural Networks (IJCNN), pp. 1-8. IEEE. 2021.

[35] Y. Shi, & P. Sheng, "J-Net: asymmetric encoder-decoder for medical semantic segmentation". *Security and Communication Networks*, vol. 2021, pp. 1-8. 2021.