ISSN (Online) 2394-2320

International Journal of Engineering Research in Computer Science and Engineering (IJERCSE)

Volume 11 Issue 6 June 2024

Automatic Steering Wheel Control using CNN

[1] Pranav Karukayil, [2] Himanshu, [3] Dr. R. Usharani

[1] [2] SRMIST KTR, Chennai, Tamil Nadu

Corresponding Author Email: [1] pk8199@srmist.edu.in, [2] hq3498@srmist.edu.in, [3] usharanr2@srmist.edu.in

***Abstract*— The developed Convolutional Neural Network (CNN) for a simulation-based self-driving car project adopts an innovative end-to-end approach, mapping raw pixel data from a single front-facing camera directly to steering commands. This approach has proven highly effective in autonomously navigating through diverse road scenarios, including local roads without lane markings and highways, as well as challenging environments like parking lots and unpaved roads. The CNN autonomously learns internal representations of key processing steps, such as detecting relevant road features, using the human steering angle as the sole training signal. Unlike traditional methods that decompose the problem into explicit components, the end-to-end system optimizes all processing steps concurrently, leading to superior performance and more compact systems. The approach not only enhances overall system efficiency by self-optimizing internal components but also allows for the development of smaller networks, minimizing the number of processing steps needed for effective self-driving functionality.**

***Index Terms*— convolutional neural network, self-driving, autonomous, deep learning, simulation.**

I. INTRODUCTION

This introduction provides an overview of our simulation-based CNN project dedicated to advancing self-driving car technology. The project encompasses three integral components: perception, decision-making, and control. In the upcoming sections, we will delve into the intricate details of each module, elucidating the methodologies, technologies, and innovative approaches employed to simulate a realistic and effective self-driving car system.Our project stands at the intersection of cutting-edge technology and real-world application, aiming not only to contribute to the ongoing technological advancements in autonomous driving but also to address the practical challenges associated with implementing these innovations in real-world scenarios. Rigorous testing and validation processes are central to our approach, ensuring that our simulated self-driving car system meets the highest standards of performance, safety, and reliability. (Gill, Tripat, et.al, 2020).

As navigated, the complexities inherent in developing a simulated self-driving car system, our ultimate aspiration is to make meaningful contributions to the collective effort working towards a future where autonomous vehicles play a pivotal role in fostering safer, more efficient, and intelligent transportation systems. This initiative aligns seamlessly with the broader industry objective of creating intelligent and dependable autonomous vehicles, marking a significant step towards the realization of a transformative and technologically advanced transportation landscape. (Ming Ding, David Smith, et.al, 2019)

II. MOTIVATION

The motivation behind undertaking the Autonomous Horizon self-driving car project stems from a profound belief in the transformative potential of autonomous vehicles to redefine the landscape of transportation.

A. Traffic efficiency and safety enhancement:

The primary motivation is to contribute to the development of self-driving cars to address safety concerns on roads, minimizing accidents caused by human error.

Aiming to alleviate traffic congestion, our project seeks to create a self-driving car system that can optimize routes and traffic flow, enhancing overall transportation efficiency.

B. Accessibility and Mobility:

The project is driven by a commitment to enhancing accessibility and mobility, particularly for individuals with limited transportation options, by creating a more inclusive and efficient transportation system.

C. Human-Machine Collaboration and Environmental Impact:

By reducing traffic inefficiencies and promoting smoother driving patterns, the project aims to contribute to environmental sustainability by minimizing fuel consumption and emissions.

Recognizing the importance of harmonious human machine collaboration, the project is motivated by the potential to create a system that combines the strengths of both automated driving technology and human intuition.

III. OBJECTIVE

To develop a self-driving vehicle utilizing CNN technology, with image data serving as input.

IV. RELATED WORKS

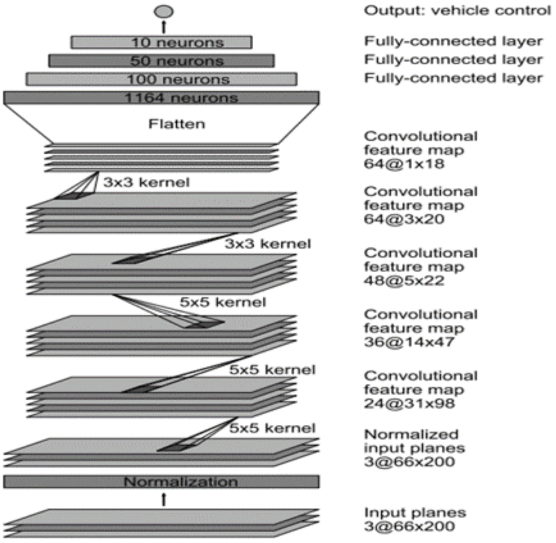

Fig. 1. Network Architecture digram.

Trained the weights of our network to minimize the mean squared error between the steering command output by the network and the command of either the human driver, or the adjusted steering command for off-center and rotated images. The network consists of 9 layers, including a normalization layer, 5 convolutional layers and 3 fully connected layers. The input image is split into YUV planes and passed to the network. (Min, Haigen, Yukun, et.al, 2023)

The first layer of the network performs image normalization. The normalizer is hard-coded and is not adjusted in the learning process. Performing normalization in the network allows the normalization scheme to be altered with the network architecture and to be accelerated via GPU processing.

The convolutional layers were designed to perform feature extraction and were chosen empirically through a series of experiments that varied layer configurations. We use strided convolutions in the first three convolutional layers with a 2×2 stride and a 5×5 kernel and a non-strided convolution with a 3×3 kernel size in the last two convolutional layers.

Followed the five convolutional layers with three fully connected layers leading to an output control value which is the inverse turning radius. The fully connected layers are designed to function as a controller for steering, but as noted that by training the system end-to-end, it is not possible to make a clean break between which parts of the network function primarily as feature extractor and which serve as controller. (Bachute, Mrinal, et.al, 2021)

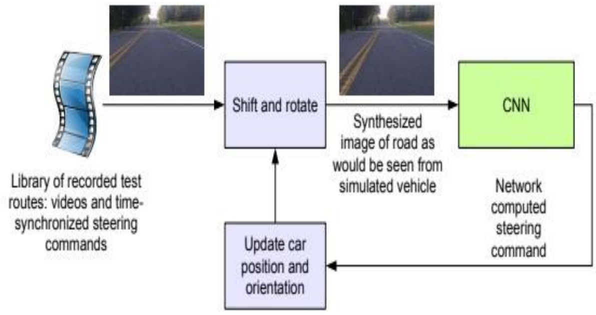

Fig. 2. Block diagram of the drive simulator.

Library of recorded test routes: This includes videos and data from real-world road conditions, likely captured by cameras mounted on vehicles.

Time-synchronized steering commands: This data specifies the steering wheel movements made during each video recording.

Network: This refers to the computer system that processes the input data and generates the synthesized image.

The purpose of this system is to create a realistic simulation of driving on a road. This could be used for a variety of purposes, such as:

Developing and testing self-driving cars: The simulated environment can be used to train and test self-driving car algorithms without the need for real-world testing.

Creating virtual reality experiences: The synthesized images could be used to create realistic driving simulations for virtual reality applications.

Testing autonomous vehicle perception systems: The system can be used to generate different driving scenarios to test how well autonomous vehicles perceive their surroundings.

$$y_j = \sum w_{ij} * x + b_j$$

The equation provided in the image describes the output of a single neuron in the convolutional layer:

Yj: The output value for the j-th neuron in the feature map.

Wij: The weight matrix (filter) connecting this neuron to the input.

x: The local region of the input image.

bj: A bias term added to the output.

*: The convolution operation.

Key Points

A. Important Aspects

- Feature Detection: Filters within a convolutional neural network (CNN) are trained to recognize distinct features such as edges, curves, and textures.
- Multiple Filters: Convolutional layers in a CNN usually consist of numerous filters, enabling the network to extract a diverse range of features from input data.
- Hierarchical Representation: CNNs operate on images in a hierarchical manner. Initial layers focus on identifying basic features, and subsequent layers

ISSN (Online) 2394-2320
International Journal of Engineering Research in Computer Science and Engineering (IJERCSE)
Volume 11 Issue 6 June 2024

integrate these features to recognize progressively complex patterns and objects.

$$\begin{cases} \text{Sigmoid}: R = \frac{1}{1+e^{-y}} \\ \text{Tanh}: R = \frac{e^{y}-e^{-y}}{e^{y}+e^{-y}} \\ \text{ReLU}: R = \max\,(0,y) \end{cases}$$

In a Convolutional Neural Network (CNN), the computations yield numerical outputs that form the network's output. The application of non-linear activation functions to these outputs is crucial for addressing intricate problems. Unlike linear functions that generate straight-line outputs, non-linear functions produce curves or intricate relationships. Given that real-world data often exhibits non-linear patterns, the use of non-linear activation functions is essential in CNNs to capture the complexity and nuances present in the data, enhancing the network's ability to tackle real-world scenarios effectively.

Sigmoid
Formula: R = 1 / (1 + e^-y)
Smooth curve that squashes input values between 0 and 1
Can lead to vanishing gradient problems (slow/stalled learning)
Tanh
Formula: = (e^y - e^-y) / (e^y + e^-y)
Similar to Sigmoid, but outputs range between -1 and 1
Also can have vanishing gradient issues
ReLU (Rectified Linear Unit)
Formula: R = max (0, y)
Very simple: if the input is negative, output is 0; if positive, the output is the input value itself
Currently preferred due to its efficiency and improved training

V. METHODOLOGY

A. Data Collection:

The simulation gathers data through three cameras mounted on the car: a left camera, a right camera, and a center camera. Alongside these images, the simulation also supplies a CSV file containing information about the steering angle and speed associated with each captured frame. This dataset is valuable for training and analyzing the performance of autonomous driving algorithms. (Li, Qing, et.al, 2020)

B. Data preprocessing:

Data pre-processing is an essential step in readying raw data for analysis or machine learning applications, and the combined use of NumPy, Pandas, and OpenCV streamlines this process. NumPy facilitates numerical operations on arrays, enabling efficient manipulation and transformation of numerical data, including tasks like normalization and reshaping. Pandas, built on NumPy, aids in data manipulation through functions like reading data from various sources, handling missing values, and facilitating sorting and grouping. OpenCV, a computer vision library, proves invaluable for image and video analysis, offering tools for tasks such as resizing, filtering, and feature extraction. This comprehensive workflow involves loading data with Pandas, cleaning and transforming it using both Pandas and NumPy, employing OpenCV for image-specific processing, and finally combining and exporting the pre-processed data for subsequent analysis or machine learning endeavors. The seamless integration of these libraries enhances efficiency and effectiveness in preparing diverse datasets for further exploration and modeling. (Bianchini, Monica, et.al, 2021)

C. Image augmentation:

Image augmentation, a crucial technique in computer vision, is efficiently implemented using Keras and TensorFlow. Keras' `ImageDataGenerator` simplifies the process by providing a high-level interface for dynamically augmenting training images during model training. This tool offers a range of transformations, such as rotation, zooming, and flipping, fostering model generalization by exposing it to diverse variations in the dataset. The seamless integration of Keras with TensorFlow ensures optimized operations and facilitates the creation of more robust models, as the augmented data helps prevent overfitting and improves the model's ability to handle diverse real-world scenarios. (Fujiyoshi, Hironobu, et.al, 2019)

TensorFlow, serving as the backend for Keras, enhances the image augmentation process by offering a versatile set of tools for customization. Developers can create custom data augmentation layers or pipelines tailored to specific project requirements, enabling more advanced and specialized transformations. The performance benefits of TensorFlow contribute to the efficiency of image augmentation, making it suitable for handling large-scale datasets. Overall, the combined power of Keras and TensorFlow streamlines the implementation of image augmentation techniques, making them accessible to both developers and researchers engaged in computer vision tasks, ultimately leading to more robust and accurate deep learning models.

D. Model Training:

Model training using Keras and scikit-learn combines the strengths of deep learning and traditional machine learning, offering a versatile and comprehensive approach to building effective models. In Keras, with its high-level neural network interface built on TensorFlow, deep learning tasks are streamlined, involving the definition of model architecture, compilation with optimizers and loss functions, and iterative parameter adjustments during training. On the other hand, scikit-learn focuses on classical machine learning, providing a user-friendly API for various algorithms in classification, regression, clustering, and more. Its versatility extends to feature extraction, preprocessing, and model evaluation, making it a valuable tool for a broad range of machine

IFERP
ISSN (Online) 2394-2320
International Journal of Engineering Research in Computer Science and Engineering (IJERCSE)
Volume 11 Issue 6 June 2024

learning applications. (Bachute, Mrinal R, et.al,2021)

The integration of Keras and scikit-learn allows practitioners to leverage the strengths of both libraries within a unified workflow. Keras excels in complex tasks such as image classification and natural language processing, while scikit-learn offers a wide array of algorithms for traditional machine learning scenarios. This combination proves beneficial for hybrid models or situations where the integration of deep learning and classical machine learning approaches is necessary. Overall, the collaboration between Keras and scikit-learn provides a powerful and flexible toolkit for model training, accommodating a diverse set of machine learning challenges. (Lee, Der-Hau, Kuan-Lin Chen, et.al)

VI. INNOVATION IDEA OF THE PROJECT

A. Dynamic Learning Architecture:

Implementing a dynamic learning architecture that allows the self-driving car to continuously adapt and improve its performance based on real-time data and user interactions. This iterative learning process enhances the system's ability to handle diverse and evolving driving scenarios.

B. Predictive Traffic Analysis:

Introducing an innovative predictive traffic analysis module that leverages historical data, machine learning algorithms, and real-time traffic updates to anticipate and proactively navigate through potential congestion, optimizing the vehicle's route for efficiency.

C. Ethical Decision-Making Framework:

Developing an ethical decision-making framework that integrates ethical principles into the autonomous driving system. This innovation ensures that the self-driving car makes decisions aligned with societal values and adheres to ethical considerations, especially in ambiguous or critical situations.

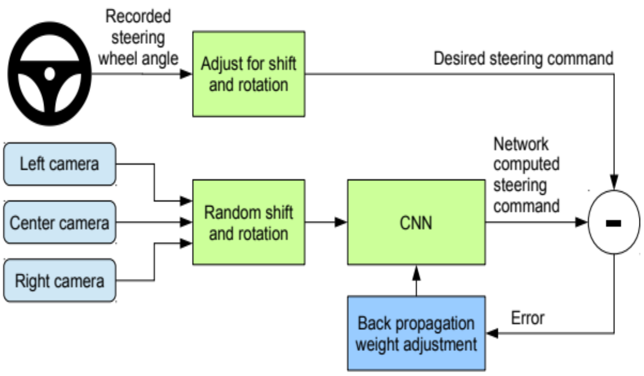

Fig. 3. Training the neural network

VII. RESULTS AND DISCUSSION

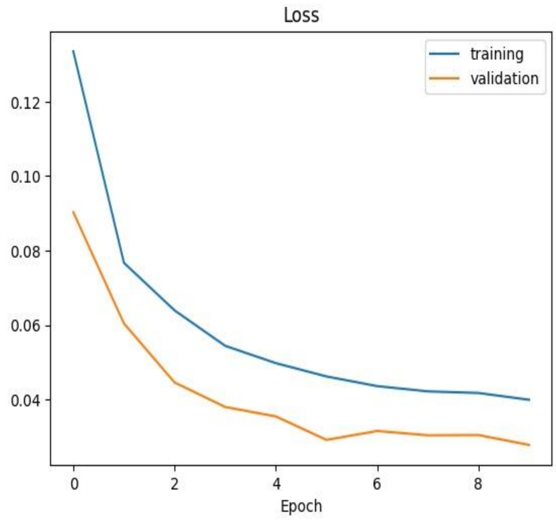

Fig. 4. Training v/s validation loss

Training loss and validation loss are metrics used in machine learning to measure how well a model fits training data and new data, respectively.

Training loss is used to optimize the model's parameters during training. Validation loss helps monitor the model's performance during training and detect overfitting.

Training and validation loss values provide important information about how learning performance changes over time. They can help diagnose problems with learning that can lead to an underfit or an overfit model. (Lee, Der-Hau et.al, 2021)

Validation loss is calculated on a separate validation dataset that the model has not seen during training. If the validation loss starts increasing while the training loss decreases, it may signal overfitting.

Validation data sets are an important part of AI, machine learning, and deep learning models. These models use these data sets to identify and learn from data such as text images. Li, Qing, et.al,2020)

The y-axis of the graph shows the loss, which is a measure sthe model is making better predictions. The x-axis shows the number of epochs, which is one complete pass through the training data.

As you can see from the graph, the training loss (blue line) is lower than the validation loss (orange line) for most of the epochs. This is a good sign, because it means that the model is learning from the training data.

Upon completing the model training, it is linked to the Udacity simulation. Once a successful connection is established, the trained model is incorporated into the car and tested within an unfamiliar environment. During testing, it is observed that the car operates seamlessly in the new environment, demonstrating effective functionality without the need for human intervention.

IFERP connecting engineers... developing research

ISSN (Online) 2394-2320

International Journal of Engineering Research in Computer Science and Engineering (IJERCSE)

Volume 11 Issue 6 June 2024

VIII. CHALLENGES AND LIMITATIONS

A. Environmental factors:

Rapid changes in lighting conditions and rain during sunrise or sunset can impact the visibility of objects and affect the performance of vision-based systems.

B. Creating self-driving car datasets

It involves challenges like annotating multi-modal data, handling imbalances, ensuring privacy, and addressing simulation-realism discrepancies.

C. Limited Critical Events:

Rarity of accidents and extreme scenarios hinders robust model training. Sensor Variability.

Diverse sensor setups pose challenges in generalization across different self-driving car configurations.

D. Privacy Compliance:

Balancing useful data with privacy concerns, especially with license plates and personal information.

IX. SCOPE AND APPLICATION

- Decision Making: CNNs can be trained to make autonomous driving decisions based on the perceived environment, such as: Steering control: Adjusting the steering wheel to safely navigate through traffic.
- Real-time Processing: CNNs are efficient at processing large amounts of data in real-time, enabling self-driving cars to react quickly to changing situations on the road
- Scalability: CNNs can be further developed and improved without changing the fundamental architecture, allowing for continuous advancements in self-driving car capabilities.
- Integration with Other AI Techniques: CNNs can be combined with other AI techniques, like reinforcement learning, to create a more robust and adaptable self-driving system.
- Data Efficiency: Some CNN architectures are designed to require less data for training, which can be beneficial in situations where large datasets are not readily available.

X. CONCLUSION

This project empirically demonstrated that CNNs are able to learn the entire task of lane and road following without manual decomposition into road or lane marking detection, semantic abstraction, path planning, and control. A small amount of training data from less than a hundred hours of driving was sufficient to train the car to operate in diverse conditions, on highways, local and residential roads in sunny, cloudy, and rainy conditions. The CNN is able to learn meaningful road features from a very sparse training signal (steering alone).

This iterative learning process, coupled with the continuous evaluation of performance metrics, ensures the model's adaptability and robust decision-making capabilities. The project's commitment to user engagement, as evidenced by interactive canvas interactions, enriches the training dataset and contributes to a more holistic understanding of the self driving car's behavior. The inclusion of functionalities for saving and loading model states enhances the utility of the developed model, allowing for the preservation of learned behaviors and providing flexibility for future testing and training scenarios. Overall, this research contributes to advancing the field of autonomous systems, emphasizing the significance of user-driven adaptability in shaping the trajectory of self-driving vehicles.

REFERENCES

[1] Min, Haigen, Yukun Fang, Xia Wu, Xiaoping Lei, Shixiang Chen, Rui Teixeira, Bing Zhu, Xiangmo Zhao, and Zhigang Xu. "A fault diagnosis framework for autonomous vehicles with sensor self- diagnosis." Expert Systems with Applications 224 (2023), pp.120002.

[2] ang, Shizhe, Ming Ding, David Smith, Paul Tyler, Thierry Rakotoarivelo, and Mohamed Ali Kaafar. "The impact of adverse weather conditions on autonomous vehicles: how rain, snow, fog, and hail affect the performance of a self-driving car." IEEE vehicular technology magazine 14, no. 2 (2019), pp.103-111.

[3] Li, Qing, and Feixiong Liao. " The impact of adverse weather conditions on autonomous vehicles: how rain, snow, fog, and hail affect the performance of a self-driving car Transportation Research Part B: Methodological 140 (2020), pp.151-175.

[4] Gill, Tripat. "Blame it on the self-driving car: how autonomous vehicles can alter consumer morality." Journal of Consumer Research 47, no. 2 (2020), pp. 272-291.

[5] Bianchini, Monica, Milan Simic, Ankush Ghosh, and Rabindra Nath Shaw, eds. Machine Learning for Robotics Applications. Springer, 2021.

[6] Lee, Der-Hau, Kuan-Lin Chen, Kuan-Han Liou, Chang-Lun Liu, and Jinn-Liang Liu. "Deep learning and control algorithms of direct perception for autonomous driving." Applied Intelligence 51, no. 1 (2021), pp.237-247.

[7] Ramos, Sebastian, Stefan Gehrig, Peter Pinggera, Uwe Franke, and Carsten Rother. "Detecting unexpected obstacles for selfdriving cars: Fusing deep learning and geometric modeling." In 2017 IEEE Intelligent Vehicles Symposium (IV), pp. 10251032. IEEE, 2017.

[8] Bachute, Mrinal R., and Javed M. Subhedar. "Autonomous driving architectures: insights of machine learning and deep learning algorithms." Machine Learning with Applications 6 (2021): 100164.

[9] Fujiyoshi, Hironobu, Tsubasa Hirakawa, and Takayoshi Yamashita. "Deep learning-based image recognition for autonomous driving." IATSS research 43, no. 4 (2019), pp.244-252.

[10] C. Chen, A. Seff, A. Kornhauser, and J. Xiao Deepdriving: Learning affordance for direct perception in autonomous driving. IEEE International Conference on Computer Vision, pp. 2722-2730. 2015.