Convolutional Neural Networks for Autonomous Navigation of the CAT793F in a Mining Environment via Digital Model

Antonio Rola Neto*, Rhuan Lucas Alves Soares†, Giovani Bernardes Vitor‡,
Levi Medeiros Magny†, Denis Silva Loubach*, Bruno Messias Gomes da Silva§
*Computer Science Division, Dept. of Computer Systems, Aeronautics Institute of Technology (ITA), São José dos Campos, Brazil
{antonioarn, dloubach}@ita.br
†Institute of Science and Technology, Federal University of Itajubá (UNIFEI), Itabira, Brazil
{rhuanlas123, lemmagny}@gmail.com
‡ROBSIC Lab, UNIFEI, Federal University of Itajubá (UNIFEI), Itabira, Brazil
giovanibernardes@unifei.edu.br
§Desenho Soluções / MICT / VALE S.A., Belo Horizonte, Brazil
bruno.messias@vale.com

***Abstract*—The mining industry is vast and operates on a global scale, demanding efficiency and, above all, safety in its mining operations. In this context, the development of technologies to ensure the protection of workers and the environment has become increasingly essential, given the presence of heavy vehicles, hazardous materials, and unpredictable conditions. In addition, high fuel consumption, vehicle and equipment maintenance, and the costs associated with specialized labor represent significant challenges for the sector. Thus, this work proposes the study of artificial intelligence tools aimed at learning the behavior of an off-road truck operator—specifically, the CAT 793F model—during the vehicle's driving process. The objective is to develop a convolutional neural network model capable of performing autonomous navigation, trained using a dataset generated from the driving behavior of human operators within a controlled simulation environment. Therefore, a complete methodology was applied for the generation and acquisition of the dataset used to train the model, as well as the adaptation of the intelligent model to consider frames as inputs in order to produce control responses such as steering, braking, and acceleration. As a result, we observed that the trained AI model was able to achieve a Mean Squared Error Loss of around 0.0002, enabling autonomous navigation along a specified trajectory.**

***Index Terms*—Convolutional Neural Network, Artificial Intelligence, End-to-end Navigation.**

I. Introduction

Mining is a relevant industry on a global scale. According to [1], the extraction of mineral resources is an essential activity for the economies of 81 countries, contributing approximately 25% of the world's Gross Domestic Product (GDP). Furthermore, this economic activity forms the foundation of the supply chain for the growing demand for materials used in the manufacturing of products in sectors such as electronics, aerospace, among others [2]. Therefore, due to its importance, improving the efficiency, safety, and sustainability of mining operations has become a key focus of technological innovation.

In this scenario, the operation of off-road trucks becomes an increasing challenge. The shortage of qualified labor, combined with the difficulty and high costs of operator training, along with the hazardous nature of the activity, results in high costs and undesirable accident rates. In this context, autonomous navigation emerges as an alternative, as shown in [3], which reports a 14.4% reduction in the effectiveness of manual driving compared to autonomous driving, highlighting the potential of automation to improve efficiency and safety in operations.

Given the importance of autonomous driving, it is necessary to determine how to implement it, with Artificial Intelligence (AI) being one of the main approaches for this purpose. AI is a technology that enables systems to simulate human cognitive abilities and behavior. Within the field of AI, there is Deep Learning, a technique based on neural networks that simulates the human brain. This approach allows the development of systems capable of identifying patterns, making predictions, and making decisions autonomously, with Convolutional Neural Networks (CNNs) being one of the most effective types of neural networks for image processing, as stated in [4].

In this context, the present work proposes an end-to-end autonomous navigation approach in a simulated environment, illustrated in Figure 1, by training a CNN specialized in driving a vehicle from point A to point B. The developed system is capable of sensing the environment through frames extracted from the simulation and, based on this visual information, controlling the main components of the truck, such as the steering wheel, brake, and accelerator.

This paper is organized as follows. Section II presents the state of the art regarding end-to-end autonomous navigation. The proposed method is described in Section III, detailing the design of the developed system. Section IV presents the qualitative and quantitative results of the navigation performed.

Fig. 1. Simulation environment.

Finally, Section V provides the conclusions and future perspectives.

II. Theoretical Foundations

This section addresses four central aspects of the theoretical foundation that supports this research. The first topic explains what Convolutional Neural Networks are and their main applications. The second topic presents the concept of end-to-end learning and its benefits in the field of autonomous vehicles. The third topic discusses the performance of CNN architectures, briefly highlighting their efficiency. Finally, the last topic combines the previous concepts to address related works on the application of CNNs in end-to-end navigation, presenting references that demonstrate their performance.

A. Convolutional Neural Networks

Convolutional Neural Networks are a type of neural network specifically designed to handle data with a grid-like structure, such as images and videos. They use an operation called convolution, which enables the identification of local patterns in the data, such as edges or shapes, with fewer parameters and greater efficiency compared to fully connected networks [5]. CNNs perform well because they leverage features like sparse connections and weight sharing, making them ideal for computer vision tasks.

When compared to other computer vision techniques, CNNs offer significant advantages, as they are capable of automatically learning spatial feature representations during the training process. Furthermore, [6] shows that CNNs are primarily used for processing spatial information such as distance, neighborhood, and direction.

These networks are able to capture increasingly complex patterns as the depth of their layers increases, and they are considered essential for feature extraction in images. This statement is reinforced by [7], which highlights that the algorithms used by autonomous systems to recognize and classify different parts of the road, as well as to make appropriate decisions according to the context, are based on CNNs.

Spatial sampling, weight sharing, and local receptive fields are the main properties of CNNs and, therefore, the reasons why they are considered versatile and fundamental components in the development of autonomous vehicles. These characteristics also help reduce the probability of the model overfitting the training dataset—becoming ineffective at predicting new outcomes—a problem known as overfitting [8].

B. End-to-End Learning

According to Viswanath *et al*. (2018) [9], end-to-end learning is one of the models used for autonomous driving. In this approach, camera images are used as inputs to the neural network, and after collecting samples and training the artificial intelligence, the network outputs the signals that control autonomous vehicle driving, such as steering angle, acceleration, braking, and others depending on the model being trained and implemented in the network.

Based on the results presented by the system, we noted that during training, only human driving data and road features detected by the system were used as inputs. According to Bojarski *et al*. (2016) [10], this type of learning is highly effective, since despite the limited input data used to train the system, an extensive amount of data was not required.

C. Convolutional Neural Network Models

The study conducted in [11] presents a comparison of error predictions made by a set of Convolutional Neural Networks, with the results shown in Figure 2. During the training of the GoogLeNet, VGGNet, and Clarifai networks, we observed that GoogLeNet stood out in terms of training accuracy, showing lower error rates and demonstrating its undeniable viability for application in autonomous vehicle driving.

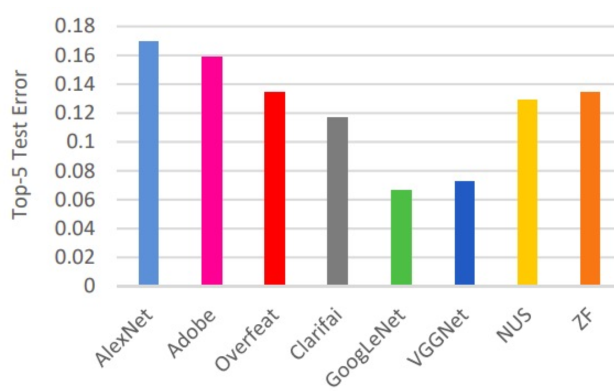


Fig. 2. Comparison of errors between a set of convolutional neural networks. Source: Al-Qizwini *et al*. (2017) [11]

In [12], a comparative analysis of five distinct CNN architectures is presented. GoogLeNet stands out for having a smaller architecture compared to models such as AlexNet and VGGNet, making it a more suitable option in scenarios with computational and memory constraints.

As shown in Figure 3, the GoogLeNet model achieved the second-best performance in terms of accuracy after 30 training epochs, according to the results presented in the study. This performance reinforces the efficiency of the architecture, even with its lower structural complexity.

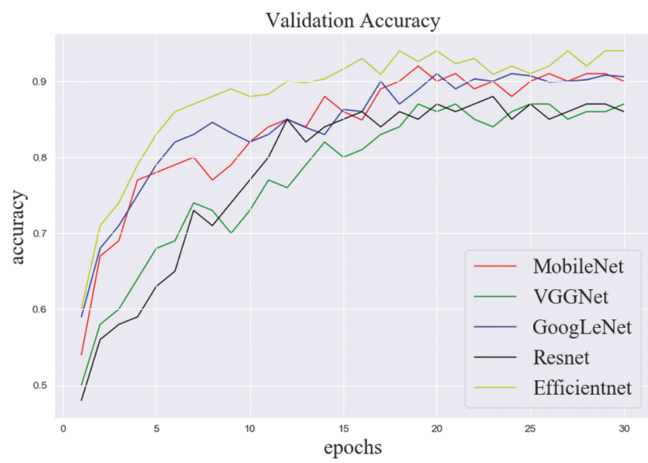


Fig. 3. Comparison of accuracy and epochs between a set of convolutional neural networks. Source: Yanfen Li *et al.* (2020) [12]

D. Related Work

In recent years, end-to-end approaches based on CNNs have proven to be promising in the control of autonomous vehicles, replacing traditional module-based architectures with systems that learn directly from sensory data. In [13], the DAVE-2 system is presented, which uses a CNN to predict the steering angle directly from images captured by strategically positioned cameras on the vehicle. Other works, such as [14], expanded this approach with the DAVE-2SKY model, introducing a closed-loop post-training phase that increases lateral control robustness even under partially observable visual conditions. The end-to-end strategy was further explored in [13], where the GTA V game was used, demonstrating that models composed of a combination of CNNs and recurrent networks are capable of learning driving commands with performance comparable to that of human drivers.

Despite the success of these approaches in simulated or controlled urban environments, the use of CNNs in off-road scenarios presents additional challenges. The proposal in [15] describes a method that replaces the direct prediction of driving commands with a visual prediction of the vehicle's future trajectory, addressing the limitations of traditional end-to-end networks in unstructured terrains. This approach was made feasible through the automatic generation of labeled data using visual odometry, allowing the training of segmentation networks to predict drivable paths in RGB images. The PPC-LSTM architecture, proposed in [16], represents another advancement by integrating spatial, depth, and temporal information, being capable of simultaneously predicting speed and steering angle closely to human behavior.

III. Proposed Approach

In this section, the entire methodology developed in this work will be detailed. As shown in Figure 4, the methodology applied was divided into three phases, with each phase consisting of a set of tasks that were executed to achieve the objective of developing a CNN model trained in a mining environment. So, in Phase 1, the dataset acquisition and generation process was developed using a simulator created by the educational institution itself in partnership with the company Vale S.A. Once the dataset was obtained, the second phase involved the model training process. Finally, in the third phase, the trained model was integrated into the simulation environment to validate its response to input data provided by the simulation system during runtime. The details of each phase will be better described below.

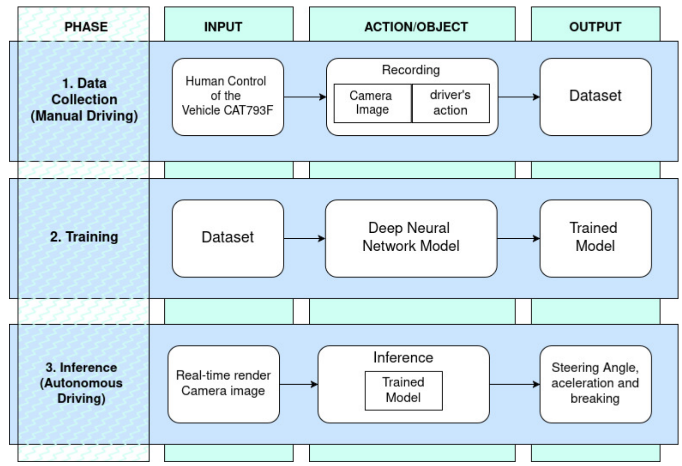


Fig. 4. Development Phases: (1) dataset generation using a simulator developed with Vale S.A.; (2) model training; and (3) integration and validation within the simulation environment.

A. Environment and Setup

As previously mentioned, the data collection and system testing environment was implemented in a virtual environment developed using Unreal Engine, and the simulation setting takes place at the Conceição Mine in Itabira, Minas Gerais (MG), result of a 3D reconstruction of the area to create a digital model. The cockpit available for interaction with the virtual environment is from the CAT 793F truck, an off-road truck used in mining operations for transporting ore. The cockpit and the simulation environment exchange and process information through the Robot Operating System 2 (ROS 2) Distro Foxy, thus enabling integration between hardware and software. Figure 5 presents the above-mentioned entities as well as the supervisory system, which is capable of configuring and debugging the simulation environment.

B. Data Collection

In the beginning of the data collection phase, illustrated in Figure 4, the first step was to define the starting and ending points of the route, as shown in Figure 6, with point A marking the beginning of the trajectory and point B as the destination. Based on this definition, the vehicle was repeatedly driven between the two points in order to generate a significant volume of data, which is essential for building the training dataset.

In addition, we decided to record exclusively the values corresponding to braking, steering, and acceleration, as these

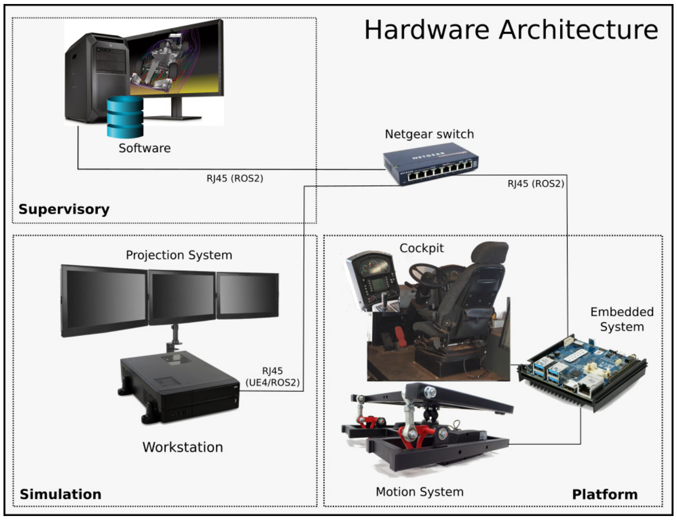


Fig. 5. Simulator Structure.

are considered the fundamental commands for the truck's movement along the route. This choice aimed to simplify the complexity of the task by focusing data collection on the variables directly related to the dynamic control of the vehicle.

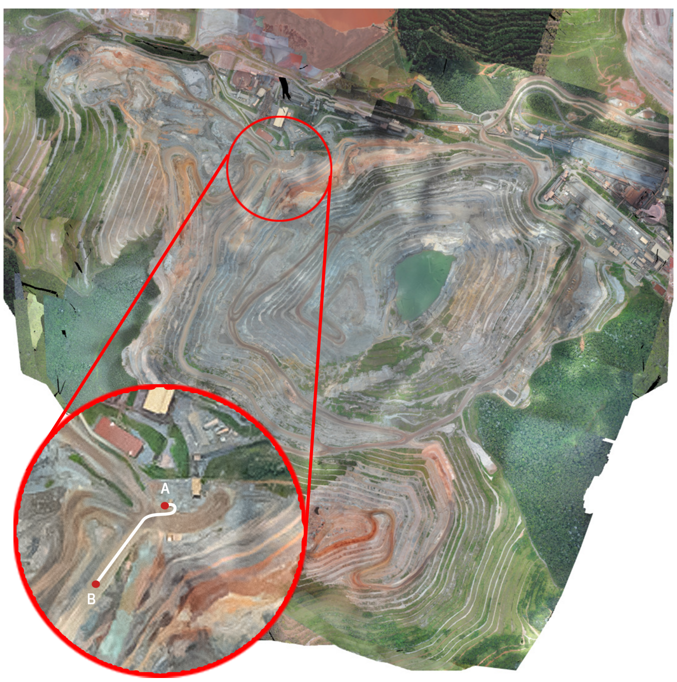


Fig. 6. Route for data collection.

Based on the initial definitions, a ROS node was developed to read data from the simulator's distributed network. This node was programmed to operate at a frequency of 10 Hz, capturing in each cycle the values of the target components and associating them with the corresponding frame of the simulation at that specific moment in time.

As a result of this first phase, illustrated in Figure 4, a dataset of 5,082 samples was obtained. The instantaneous values of the vehicle control commands were stored in a CSV file [17], while the frames from the simulation were saved as images in JPG format. With this data, it was possible to compose the dataset required for training the convolutional neural network model, whose architecture will be detailed in Subsection III-C.

C. Convolution Neural Network Architecture

In addition to the definitions made during the Data Collection stage, some decisions also had to be made to define the architecture of the CNN. The first important point is that this is not a classification problem, but rather a regression task. Another key aspect is that the project requires the network to be both efficient—so it can be integrated into the simulator—and powerful enough to extract sufficient features to better identify changes in the frames.

The GoogLeNet architecture [18] was chosen for being a computationally efficient Deep Learning solution and, as shown in [11], it shows good results for autonomous vehicle problems. This network represents a milestone in the evolution of convolutional neural networks, standing out for its efficiency and depth. Its main innovation lies in the Inception modules, which combine, in parallel, convolutions with filters of different sizes (1×1, 3×3, and 5×5) and pooling operations, enabling multi-scale feature extraction with reduced computational cost. The architecture was carefully designed to maintain a feasible computational budget, making it suitable for practical applications—even on devices with limited resources. Composed of 22 layers with trainable parameters, GoogLeNet also incorporates auxiliary classifiers in intermediate layers, which help gradient backpropagation and act as a form of regularization, mitigating issues such as vanishing gradients during training [18].

In this work, the architecture was adapted for a multivariate regression task with the goal of predicting, from simulation frames, the continuous control commands of an autonomous off-road truck: brake, steering wheel and throttle. The final layers of the network were modified to produce three continuous outputs.

D. Training

The proposed model was trained in a hardware setup composed of an Intel Core i7-8565U, 12 GB of RAM (11.9 GB usable), Ubuntu 20.04 64-bit operating system, and a x64-based processor architecture. The loop began with a relatively low initial learning rate, set at 1×10^{-6}, which decreases over the training according to Eq. 1. This choice is justified by the nature of the task, which involves predicting continuous variables from visual data—characterizing a regression problem that is sensitive to abrupt variations in the network's weights. A low learning rate helps ensure greater stability during the optimization process, preventing undesirable oscillations in the loss function and promoting smoother convergence, especially during the initial training epochs.

$$LR = \frac{10^{-6}}{0.01 \cdot epoch + 1} \tag{1}$$

The loss function used was the Mean Squared Error (MSE), optimized with the Adam algorithm. The dataset was structured using a custom class that associates RGB images with their corresponding control values, and was divided into training and testing subsets. During training, the accuracy of the network was evaluated using the coefficient of determination (R-Squared Score), measured individually for each of the three predicted variables, allowing the quality of the network's fit to be evaluated for each output.

It is important to note that some libraries, such as PyTorch—used in this project—may return negative values for the coefficient of determination. This result indicates that the model performs worse than a simple average of the target values, meaning it is unable to capture the patterns present in the data. This situation will be discussed in more details in Section IV.

E. Inference

In the final phase, defined in Figure 4, the trained model described in Subsection III-D was integrated into a ROS node capable of capturing a frame from the simulation and performing inference using the CNN. As output, it is possible to observe the accelerator, brake, and steering wheel values changing continuously as the off-road truck moves through the simulated environment, thus enabling end-to-end autonomous navigation.

IV. Analysis and Discussion

This section aims to present two approaches: the first is related to the quantitative analysis of the proposed system, discussed in Subsection IV-A, and the second, addressed in Subsection IV-B, focuses on the qualitative analysis of the experiments.

A. Quantitative Results of the Model

As already stated, the main objective is to demonstrate the effectiveness of using machine learning in the development of an autonomous vehicle. One of the initial concerns was that, due to the highly irregular environment of the mine, the neural network would struggle to adapt. However, the work of [6] is right in asserting that perception is one of the areas where CNNs lead and impress.

It is possible to observe in Figure 7 that GoogLeNet was able to adequately minimize the MSE. In 100 epochs and approximately 10^4 iterations, it was able to reduce the error to a range around 0.02.

The R2 Score data were also collected and can be seen in Figure 8. It is impressive that the network was able to achieve satisfactory values for steering and braking. However, in Figure 8(c), difficulties in achieving better prediction and convergence can be observed. For this issue, possible causes that hindered the network from reaching an ideal learning curve were investigated. These include: the high sensitivity of the simulator's modeling regarding the accelerator pedal, the unpredictability of driver behavior, and inconsistencies in the truck's physics within the simulation environment, including modeling flaws.

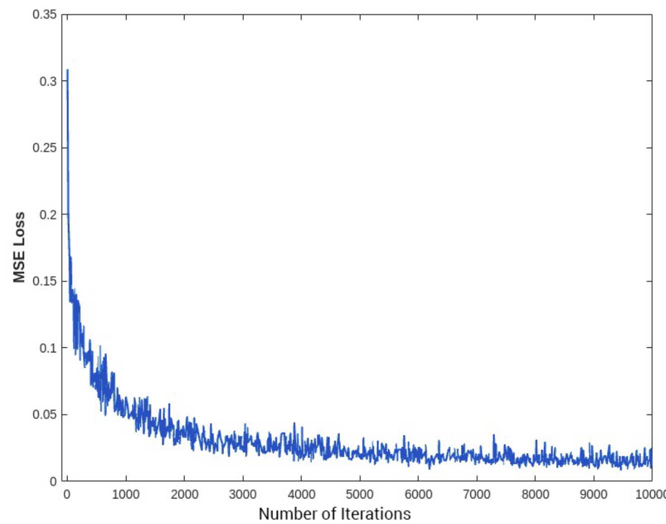
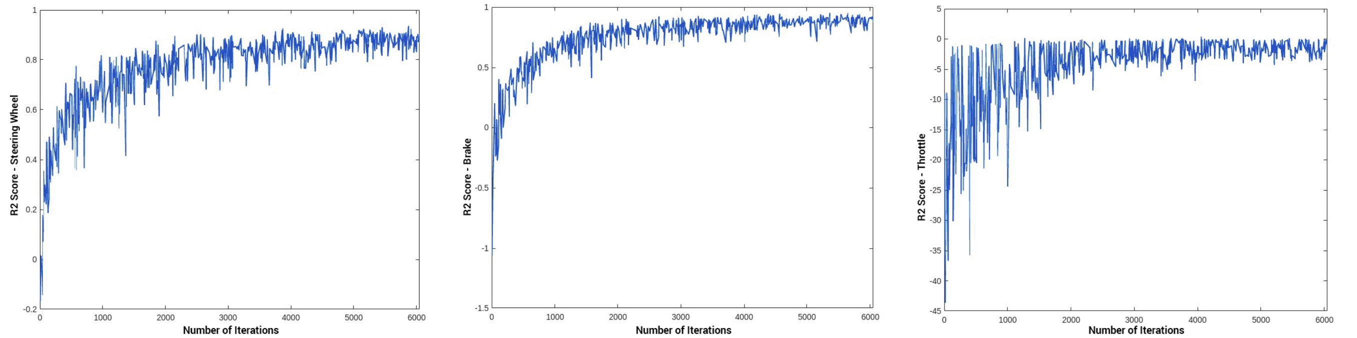


Fig. 7. Convergence curve of the Mean Squared Error (MSE Loss) obtained during the network training.

In order to mitigate the limitation related to the accelerator, a second training cycle of the same model was executed, maintaining the same network architecture. The resulting model achieved an average MSE of around 0.0002, with evidence of specific improvement in the accelerator prediction. The Table I presents a comparison between the target values and the predicted values for 10 samples from the test set. In this specific sample, the R2 coefficients were 0.999 for the brake, 0.999 for the steering wheel, and 0.910 for the accelerator, indicating a strong correlation between predicted and actual values. However, it is important to note that these values do not represent the overall performance of the network, as they were obtained from a limited data subset. Even so, the results suggest that the second training contributed to reducing the variance of the predictions and to a more accurate fit to the reference values.

TABLE I
Comparison between Target and Prediction for 10 samples.

Target			Predicted		
Brake	**Steering**	**Accel.**	**Brake**	**Steering**	**Accel.**
0.000	0.026	0.179	0.007	0.033	0.146
0.995	0.036	0.002	1.007	0.036	0.003
0.000	0.000	0.069	-0.012	-0.007	0.082
0.000	0.990	0.050	-0.002	0.993	0.041
0.000	-0.183	0.105	0.000	-0.181	0.105
0.989	-0.003	0.002	0.969	-0.009	0.007
0.000	-0.034	0.068	-0.004	-0.037	0.089
0.000	0.997	0.005	-0.003	0.995	0.030
0.000	-0.563	0.005	0.000	-0.549	0.023
0.000	0.993	0.005	0.000	0.997	0.010
	R2 Score:		0.999	0.999	0.910



(a) Steering Wheel Curve (b) Brake Curve (c) Throttle Curve

Fig. 8. Convergence curve of the R2 Score for each of the variables. (a) shows the curve for steering, that is, the values of the steering wheel. (b) shows the curve for braking, and (c) shows the curve for acceleration.

B. Qualitative Results of the Experiment

From a qualitative perspective, the model demonstrated consistent and functional behavior during the autonomous navigation simulation, as can be publicly seen[1]. After training, we observed that the system was able to drive the truck smoothly and coherently along the predefined route, performing smooth turns, maintaining a stable trajectory, and responding appropriately to simple variations in the environment. These observations indicate that the neural network was able to extract relevant representations from the input images and translate them into plausible control commands.

Although the prediction of the accelerator was mitigated, the model still showed difficulty in the gradient descent of this variable. However, it is possible to assert that the model was already effective in identifying the visual patterns of the mine and was able to steer the wheel correctly along the path, compensating for the inaccuracy of the accelerator. Similarly, the brake operated at the appropriate moments, which ensured a certain level of safety throughout the route, even if not ideally.

V. Conclusion

Based on the results obtained, it can be concluded that the chosen methodology effectively enabled end-to-end autonomous navigation using convolutional neural networks. The network demonstrated the ability to significantly reduce the loss function and achieve consistent performance across training cycles, indicating strong generalization capabilities. Furthermore, improvements were observed in the prediction of braking, steering, and particularly the accelerator components, reinforcing the effectiveness of the approach.

Despite the promising results, it is plausible to consider the possibility of model overfitting, as the images used to build the dataset were captured under the same simulation conditions — that is, without variations in lighting or weather.

Also, the system did not operate within an ideal frequency range due to limitations in the simulator's frame capture strategy. This restriction compromised an accurate analysis of the computational cost during the use of the GoogLeNet architecture.

Thus, future work includes improving the process of acquiring input frames for the CNN, as well as defining more complex trajectories within the simulated environment, aiming to further validate the proposed approach and eventually enable its application in a real-world scenario. Additionally, efforts will focus on generating a more robust dataset by introducing variations in lighting and weather conditions, in order to reduce the risk of model overfitting.

Acknowledgment

The authors would like to express their sincere gratitude to Vale S.A for providing access to the simulation platform developed in collaboration with Laboratory of Robotics, Intelligent and Complex Systems - ROBSIC. The use of this simulator was fundamental to the development and validation of the proposed methodology.

References

[1] U. N. E. O. of the Secretary-General (EOSG), "Transforming extractive industries for sustainable development," *UN Executive Office of the Secretary-General (EOSG) Policy Briefs and Papers*, p. 18, 2021, accessed: May 13, 2025. [Online]. Available: https://doi.org/10.18356/27082245-22

[2] C. Escamilla. (2024) The importance of mining in modern society. [Online]. Available: https://mining.arizona.edu/news/importance-mining-modern-society

[3] G. E. P. Cerna, J. R. C. Hernández, J. C. M. Herazo, and A. P. Castillo, "Evaluation of the overall effectiveness (oee) of autonomous transportation system (ahs) equipment and its impact on mine design. open pit mine case study," *Procedia Computer Science*, vol. 224, pp. 468–473, 2023, 18th International Conference on Future Networks and Communications / 20th International Conference on Mobile Systems and Pervasive Computing / 13th International Conference on Sustainable Energy Information Technology. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S1877050923011134

[4] Y. Guo, Y. Liu, A. Oerlemans, S. Lao, S. Wu, and M. S. Lew, "Deep learning for visual understanding: A review," *Neurocomputing*, vol. 187, pp. 27–48, 2016, recent Developments on Deep Big Vision. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0925231215017634

[1] Demonstration of autonomous navigation in the simulator: https://drive.google.com/file/d/1pr7HhIWVHl9RVYB_AzvZjwgbzr0Yr6gC. Accessed on May 27, 2025.

[5] I. Goodfellow, Y. Bengio, and A. Courville, *Deep Learning*. MIT Press, 2016, http://www.deeplearningbook.org.

[6] S. Grigorescu, B. Trasnea, T. Cocias, and G. Macesanu, "A survey of deep learning techniques for autonomous driving," *arXiv preprint arXiv:1910.07738*, 2020, https://arxiv.org/pdf/1910.07738.pdf.

[7] N. Barla. (2022) Self-driving cars with convolutional neural networks (cnn). https://neptune.ai/blog/self-driving-cars-with-convolutional-neural-networks-cnn.

[8] C. F. G. D. Santos and J. P. Papa, "Avoiding overfitting: A survey on regularization methods for convolutional neural networks," *ACM Comput. Surv.*, vol. 54, no. 10s, Sep. 2022. [Online]. Available: https://doi.org/10.1145/3510413

[9] P. Viswanath *et al.*, "End to end learning based self-driving using jacintonet," in *2018 IEEE International Conference on Intelligent Transportation Systems (ITSC)*. IEEE, 2018, https://ieeexplore.ieee.org/document/8576190.

[10] M. Bojarski, D. Del Testa, D. Dworakowski, B. Firner, B. Flepp, P. Goyal, L. Jackel, M. Monfort, U. Muller, J. Zhang, X. Zhang, J. Zhao, and K. Zieba, "End to end learning for self-driving cars," *arXiv preprint arXiv:1604.07316*, 2016, https://arxiv.org/pdf/1604.07316.pdf.

[11] M. Al-Qizwini, I. Barjasteh, H. Al-Qassab, and H. Radha, "Deep learning algorithm for autonomous driving using googlenet," in *2017 IEEE Intelligent Vehicles Symposium (IV)*. IEEE, 2017, https://sci-hub.se/10.1109/IVS.2017.7995703.

[12] Y. Li, H. Wang, L. M. Dang, T. N. Nguyen, D. Han, A. Lee, I. Jang, and H. Moon, "A deep learning-based hybrid framework for object detection and recognition in autonomous driving," *IEEE Access*, vol. 8, pp. 194 228–194 239, 2020. [Online]. Available: https://ieeexplore.ieee.org/stamp/stamp.jsp?tp=&arnumber=9238023

[13] G. A. M. Novello, H. Y. Yamamoto, and E. L. L. Cabral, "An end-to-end approach to autonomous vehicle control using deep learning," *Revista Brasileira de Computação Aplicada*, vol. 13, no. 3, pp. 32–41, 2021.

[14] J. Jhung, I. Bae, J. Moon, T. Kim, J. Kim, and S. Kim, "End-to-end steering controller with cnn-based closed-loop feedback for autonomous vehicles," in *2018 IEEE Intelligent Vehicles Symposium (IV)*, 2018, pp. 617–622.

[15] C. J. Holder and T. P. Breckon, "Learning to drive: End-to-end off-road path prediction," *IEEE Intelligent Transportation Systems Magazine*, vol. 13, no. 2, pp. 217–221, 2021.

[16] S. L. X. L. X. Q. P. G. Guofa Li, Liang Yang, "Human-like decision making of artificial drivers in intelligent transportation systems: An end-to-end driving behavior prediction approach," *IEEE Electronic Library (IEL) Journals*, vol. 14, no. 6, pp. 188–205, 2021.

[17] K. Jamsa, *Introduction to Data Mining and Analytics*. Jones & Bartlett Learning, 2020. [Online]. Available: https://books.google.com.br/books?id=1VPODwAAQBAJ

[18] C. Szegedy, W. Liu, Y. Jia, P. Sermanet, S. Reed, D. Anguelov, D. Erhan, V. Vanhoucke, and A. Rabinovich, "Going deeper with convolutions," *arXiv preprint arXiv:1409.4842*, 2014, https://arxiv.org/pdf/1409.4842.pdf.