Building of Informatics, Technology and Science (BITS)
Volume 4, No 2, September 2022 Page: 956−961
ISSN 2684-8910 (media cetak)
ISSN 2685-3310 (media online)
DOI 10.47065/bits.v4i2.1965

Simulation of Autonomous Vehicle Using ROS2 Based on Convolutional Neural Networks for Object Recognition

Muhammad Miftahudin, Nungki Selviandro*, Muhammad Johan Alibasa

School of Computing, Telkom University, Bandung, Indonesia
Email: [1]miftahudin@student.telkomuniversity.ac.id, [2,*]nselviandro@telkomuniversity.ac.id, [3,*]alibasa@telkomuniversity.ac.id
Email Penulis Korespondensi: nselviandro@telkomuniversity.ac.id
Submitted:**25/07/2022**; Accepted:**18/08/2022**; Published: **30/09/2022**

Abstract−The main justification for implementing an Autonomous Vehicle (AV) system in the real world is the safety aspect of driving, because if there is an error in driving then the error will become a gap that can threaten the safety of the driver himself and other drivers, therefore an AV system is made to reduce driver errors. in driving. The aim of this research is to implement one of the parts of the AV system, that is object recognition, and in this study, we also conduct an experiment with simulating the object recognition feature that has been implemented in order to get more concrete results. Architectural object recognition is designed to extract key features from traffic sign images, the traffic sign detection uses the customized Convolutional Neural Networks (CNNs) architecture. After the architectural has been implemented, training will be carried out using Custom Traffic Sign Dataset and experiments will also be conducted to simulate object recognition by applying ROS2 as a car robotic system that represents a car's functionality system in the real world. the results of this study for the implementation of the modified CNNs architecture is 99.96% and the results of the simulations carried out show that the prototype can detect traffic signs objects with a distance of 10m.

Keywords: Autonomous Vehicle; Object Recognition; Custom Dataset; Convolutional Neural Networks; ROS2

1. INTRODUCTION

Autonomous Vehicle is a technology system in the automotive field that has intelligence in driving a vehicle by recognizing the lane, planning next steps and controlling the steering automatically when interacting with the driver [1]. Attractiveness to AV has grown exponentially in recent years and is expected to engage further in the automotive industry which also involves suppliers, technology providers, academic institutions, municipal governments and regulatory agencies. Considering worldwide every year around 1.35 million people die from traffic accidents, mainly due to wrong driver behavior due to driving over the speed limit and driving under the influence of alcohol or illegal drugs [2]. AV aims to reduce human control of the task of driving, it can be considered a solution to improve safety while driving. Apart from driving safety, AV will also provide several possible benefits such as saving travel time, reducing traffic jams and reducing traffic accident rates [3]. Many studies have been carried out on AV systems, one of which was in 1989, [4] designed a simple structure Neural Network (NN) with three interconnected layers to drive a car on the Carnegie Mellon University campus, in the following year 1993 [5] designed return an AV system using the NN algorithm model.

Object recognition and image classification are some of the problems of machine vision and machine learning [6], [7]. Object classification is a difficult task in image processing because it requires a lot of calculations and classification algorithms that are accurate, consistent and correct. CNNs can overcome this problem by utilizing the correct and simple architecture [8]. Since 2010, this architecture has been widely used thanks to the Computer Vision Competition ILSVRC (ImageNet Large Scale Visual Recognition Challenge), which aims to correctly locate and classify objects and scenes in images [9]. For example, this 2012 contest created the AlexNet architecture, a complex neural network. Since the creation of this architecture, several image detection and classification tasks have been invented, including face detection [10], facial emotion detection [11], dust detection [12], fake image detection and localization [13].

Convolutional Neural Networks are a type of Deep Learning Neural Network (NN) algorithm model that has dominated in solving problems in Computer Vision. Convolutional neural networks are designed automatically and adaptively to study the hierarchy of spatial features from low to high level patterns [14]. The term "neural network" refers to a network simulation created computationally that mimics the network of nerve cells (neurons) found in the human brain. The following are some advantages that can be attained by using CNNs to handle computer vision problems [15]: The CNNs' weight sharing feature is the primary consideration. CNNs promote network generalization, limit the amount of network parameters that may be taught, and prevent overfitting. Simultaneously assess the feature extraction layer and the classification layer, which will result in a highly organized model output that is heavily dependent on the derived features. Compared to other neural networks, CNNs make it considerably simpler to implement big networks.

There are several studies that have been done for object recognition using CNNs architecture [16]–[18] and the performance generated by CNNs architecture is very high. We proposed a customized CNNs architecture in this study as method of algorithm, which will subsequently be employed as a simulation model with two convolutional layers and 0.25 dropout. On the dataset created by the ROS2 simulation environment, we also simulated the modified CNNs architecture with finding evaluation. Our simulation is an experiment from this research that tries to build a traffic sign object identification feature with more accurate outcomes. ROS2 will function as a robotic system that represents the actual system functionality of a car so that this research produces a more accurate and real analysis. The second-

Building of Informatics, Technology and Science (BITS)
Volume 4, No 2, September 2022 Page: 956−961
ISSN 2684-8910 (media cetak)
ISSN 2685-3310 (media online)
DOI 10.47065/bits.v4i2.1965

generation robotic operating system ROS2 provides an architectural design similar to vehicle functionality, the steering control interface, acceleration and braking are common functions for vehicles, and the reliability and real-time performance required by ROS2 is provided. Therefore, it was decided later in the simulation process to use the ROS2 robotic system for self-driving cars [19].

The rest of this paper is structured as follows: Section 2 describes in detail the research stages. More information about the training and experimental results is described in Section 3. Section 4 provides an overview of the object recognition methods.

2. RESEARCH METHODOLOGY

2.1 Research Scenario

The traffic sign recognition is a crucial component of the AV system, so we successfully constructed and simulated CNNs architecture for object recognition in this study. Object recognition will be focused on static objects, specifically traffic signs. The CNNs architecture is implemented in a number of phases. We first search and obtain the dataset for the traffic signs. We then modify CNN's architecture and uses it to do object recognition classification. We will then assess the output of the implemented architecture in terms of performance. The research will next use ROS2 to simulate an experiment in object recognition, and in this section will explain how this simulation step works. **Figure 1** illustrates how we conceptualized these steps these steps will be explained in the next sub-bab.

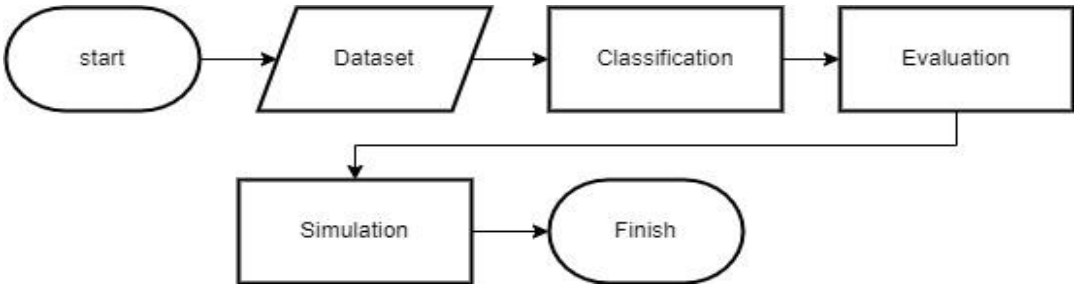

Figure 1. Research Scenario

2.2 Collecting Dataset

The dataset that we use is data that was specifically created by us using the blender application. This was done for reasons other than the fact that this data must be able to be trained by the customized CNN architecture that we proposed. It also had to be able to be simulated on ROS2, which can only accept data /assets that are objects from blender-like applications. To assess the redesigned CNN architecture and test the CNNs model on ROS2, we use six classes of data, including traffic signs. Stop, turn left, 30km/h, 60 km/h, 90 km/h and without a sign giving details. 107 stop data, 149 left turn data, 186 90 km/h data, 171 30 km/h data, and 63 signs without giving details.

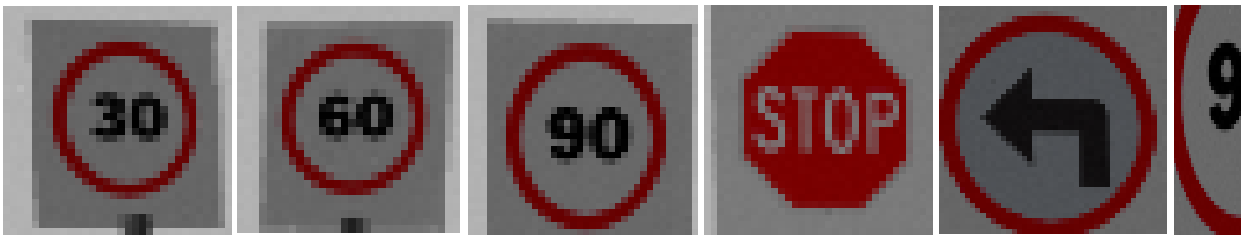

Figure 2. Representation of some samples of data used

2.3 Customization of CNNs for Classification

We used a customized CNNs architecture for this classification stage, as depicted in **Figure 4**. The detailed steps for putting the model into practice are as follows: after initializing the sequential model as a neural network, import conv2D as the first convoluted layer, and then run convolutions on the input in the form of an image using 16 filters, 3x3 kernel sizes, and the Relu activation function. Afterward, import Conv2D more as a second convoluted layer with a doubled filter, 32 filters, 3x3 kernel sizes, and the Relu activation function. Each convoluted layer also has a max-polling layer with a size of 2x2 and a 0.25 dropout to avoid overfitting and to hasten the learning process. After applying 2 layers of convulsions, we import a flatten, and then inputs a 1D vector to the fully connected layer to make the resultant convolution matrix. Last but not least, we employ the softmax activation function to ensure that, even after classifying a large amount of data, the result is still equal to 1, and the class with the highest probability will be chosen.

Building of Informatics, Technology and Science (BITS)
Volume 4, No 2, September 2022 Page: 956−961
ISSN 2684-8910 (media cetak)
ISSN 2685-3310 (media online)
DOI 10.47065/bits.v4i2.1965

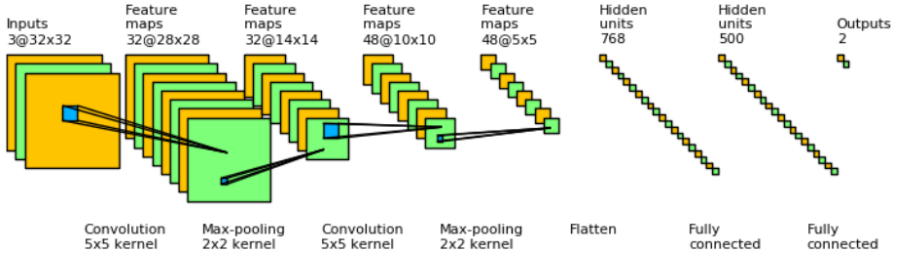

Figure 3. The CNNs Architecture [20]

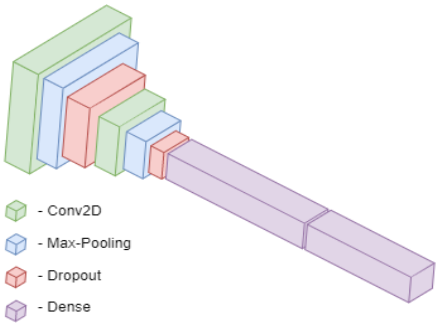

Figure 4. Our Customized CNNs Architecture

Because what was previously input was a multi-dimensional array, it will become a vector after using flatten or input through the flatten layer, and by using this softmax function, it will be normalized into a class between 0 and 1, the softmax is obtained from (1) where K is the number of classes and z_j the value of -j vector.

$$Softmax(z_j) = \frac{\sum e^{z_j}}{\sum_{k=1}^{K} e^{z_j}} \quad (1)$$

2.4 CNNs Evaluation (Multi Class-Entrophy)

The method used to calculate the accuracy of the CNNs model that has been set is Multi Class-Entropy (MCE), MCE is a loss function that is obtained when CNNs architecture performs learning. This MCE calculates the probability generated by the softmax activation function against the actual class, or, to put it another way, compares the label data with the data produced by deep learning, and at the end the MCE will adjust the value of the probability or the accuracy of the CNNs model that has been set.

More technically, it is clear from (2) that y will be entered as a probability value from the data label, and $\hat{y}$ will be entered as the outcome of the softmax activation function. Later in the following epoch, $\hat{y}$ will continue to be improved until it approaches y. As a result, deep learning accuracy will increase, and deep learning losses will decrease.

$$CrossEntropy = y\log(\hat{y}) - (1-y)\log(1-\hat{y}) \quad (2)$$

In the evaluation that has been carried out, we also use the metric accuracy method where the correct output of deep learning is divided by the amount of data as in (3). TP is the actual class, and the prediction class is positive, TN is the actual class and the prediction class is negative where TN and TP will be divided by the entire data.

$$Accuracy = \frac{TP+TN}{TP+TN+FP+FN} \quad (3)$$

2.4 Simulated Using Customization of CNN

There are various scenario steps that have been determined for the simulation that we will perform : importing the CNNs model, which has been modified and trained on the prototype, first as a system that would recognize traffic signs in real-time, and secondly using the ROI (Region of Interest) approach for special detection traffic signs when the simulation is run, and the third is to perform localized traffic signs to ascertain the circumstances of the detection, including whether or not traffic signs are present. The third stage can produce commands that will perform modifications to the behavior of the prototype.

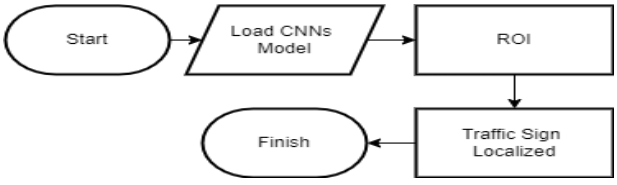

Figure 5. Simulation Scenario

Building of Informatics, Technology and Science (BITS)
Volume 4, No 2, September 2022 Page: 956−961
ISSN 2684-8910 (media cetak)
ISSN 2685-3310 (media online)
DOI 10.47065/bits.v4i2.1965

3. EXPERIMENTS RESULT

This section will discuss the outcomes of the two experiments in this section, starting with the findings of the modification of CNNs with the CME and Adam's optimization methods with a learning rate of 0.005. Second, the outcomes of the ROI approach and real-time localization simulation of the CNNs specialized model applied to the ROS2 system.

3.1 Result

Using a data set taken from a simulated environment with 60 epochs each, we trained a customized CNN, SCNN, and oneCNN using a Ryzen 5 5008H processor with support for a 6GB VRAM RTX 3060, the training resulted in a training run time of each algorithm between 15 -25 minutes. CNNs customization occupies second place between the other two CNN algorithms, as shown in Table 1.

Tabel 1. Results from training conducted with other CNN models

Method	Accuracy
Traditional CNN [20]	98.64%
SCNN [21]	99.97%
OneCNN [22]	99.87%
Our model	99.96%

3.2 Analysis

We show the output from the camera that is applied to the car prototype while simulating the custom CNNs architecture in real-time, along with information on camera angle, speed, and detecting conditions. The prototype will operate under the condition that, if it sees a traffic sign through the camera, it will stop at a speed of 0 km/h for a stop sign, run at a speed of 30 km/h for a sign that indicates a speed limit of 30 km/h, go at a speed of 60 km/h for a sign that indicates a speed limit of 60 km/h, run at a speed of 90 km/h for a sign that indicates a speed limit of 90 km/h and if it notices a left turn signal, it makes a 50 km/h left turn. Figures 6 to 8 show illustrations of these circumstances.

Based on the simulation scenario in Figure 5, the analysis of the simulation performed on various traffic signs is conducted. In order to determine how near the detection distance from the car camera is to the detected sign, we draw a blue line with a length of 10m, as can be seen in Figures 6 to 8.In the first case, a left turn traffic sign is used as the basis for the simulation, which results in a detection with a range that fits the end of the blue line and is roughly 10 meters on a straight road and then, at a speed of 50 km/h, the car moves to the left.

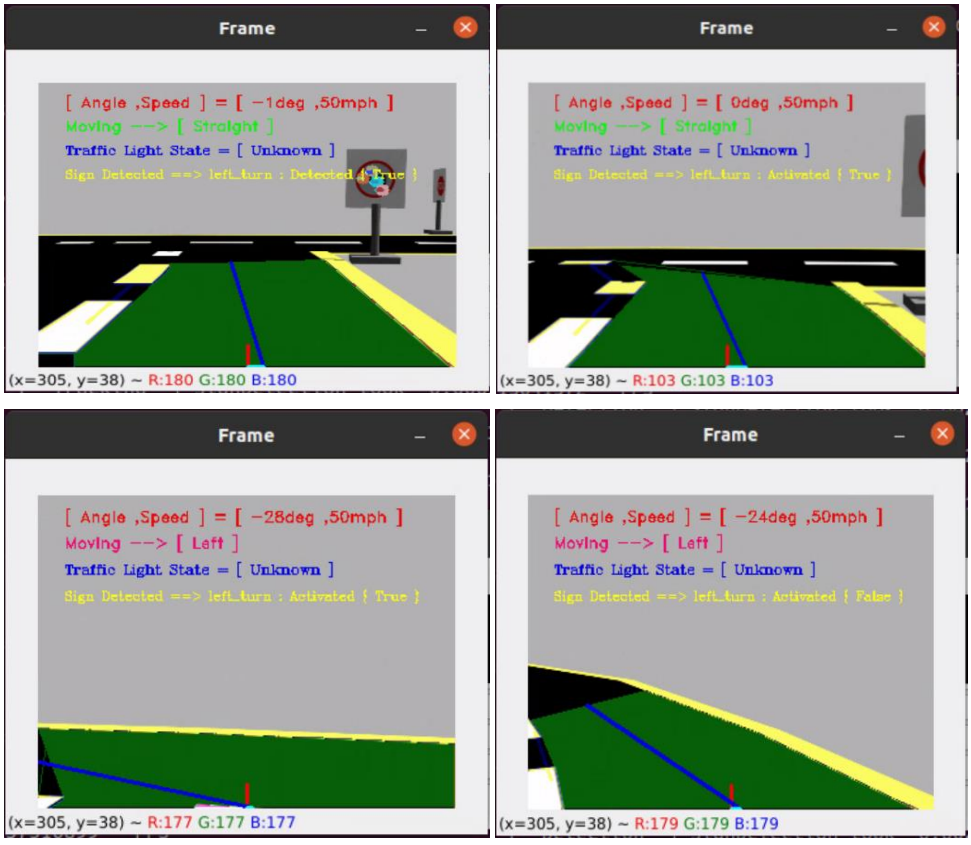

Figure 6. Results for the turn left sign when simulating

Building of Informatics, Technology and Science (BITS)
Volume 4, No 2, September 2022 Page: 956−961
ISSN 2684-8910 (media cetak)
ISSN 2685-3310 (media online)
DOI 10.47065/bits.v4i2.1965

in the same simulation scenario, a simulation is carried out to detect a 60km/h traffic sign placed next to a straight road and as seen in Figure 7 it turns out that the detection distance is the same as the distance when detecting a left turn traffic sign in the first simulation, which is right at the end of the blue line.

Figure 7. Results for the 60 km/h sign when simulating

However, different from the detection distance for stop traffic signs where we placed the stop traffic signs on the road which is a bit complicated to be detected by the car camera, precisely on the turning road, and for the distance it is 5m and almost approaching the car. So, it can be concluded that the results of these various simulations are different when faced with several straight road conditions or non-straight roads. Maybe something else will become a more complex problem when the simulation is carried out on a more real conditions where there will be conditions of rain, snow, fog, night or even a half-broken traffic sign. Therefore, the results of this analysis can be used as the basis that any conditions that occur when the detection is carried out will cause the results of the detection to be good or bad in real-time.

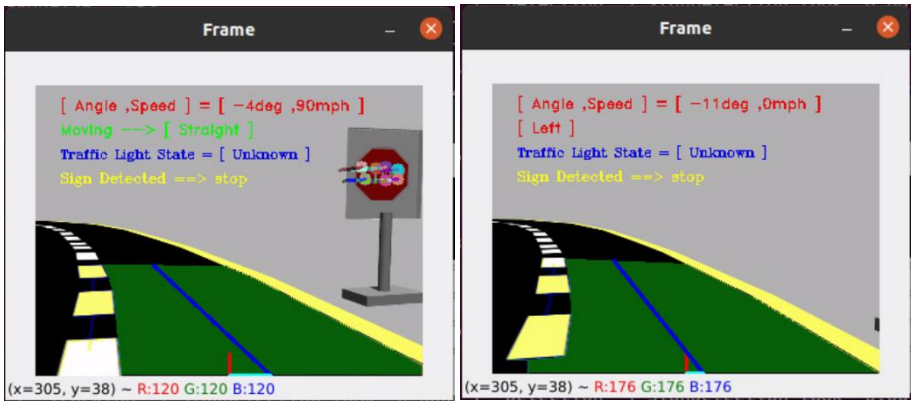

Figure 8. Results for the stop sign when simulating

4. CONCLUSION

Based on the analysis, implementation, training and simulation that has been done, we have customized the architecture of CNNs for object recognition features in AV by producing 99.96% accuracy with 60 epochs. Additionally, by running simulations with the ROS2 robotic system with 10m detection findings in real-time, we have improved upon earlier relevant studies. More thorough results and analysis of this simulation are provided in section 3. We hope that the system for car functionality in the real world may be applied to the findings of this investigation and the implemented model. Other research may also use this study finding as a foundation for future work, specifically the implementation of the lane detection model. Future studies may also use the model that will be simulated with ROS2, ensuring that the study's findings are high-performing and applicable.

REFERENCES

[1] S. Kato, E. Takeuchi, Y. Ishiguro, Y. Ninomiya, K. Takeda, and T. Hamada, “An open approach to autonomous vehicles,” IEEE Micro, vol. 35, no. 6, pp. 60–68, 2015.
[2] W. H. Organization, “Global status report on road safety 2018: Summary (No. WHO/NMH/NVI/18.20),” World Health Organization, 2018.
[3] N. Lang et al., “Self-driving vehicles, robo-taxis, and the urban mobility revolution,” The Boston Consulting Group, vol. 7, p. 2016, 2016.

Building of Informatics, Technology and Science (BITS)
Volume 4, No 2, September 2022 Page: 956−961
ISSN 2684-8910 (media cetak)
ISSN 2685-3310 (media online)
DOI 10.47065/bits.v4i2.1965

[4] D. Pomerleau, "An autonomous land vehicle in a neural network," Advances in Neural Information Processing Systems, vol. 1, 1998.
[5] D. A. Pomerleau, "Knowledge-based training of artificial neural networks for autonomous robot driving," in Robot learning, Springer, 1993, pp. 19–43.
[6] L. Anis, T. Ghada, S. Anis, and M. Abdellatif, "Optimal feature selection based on hybridization of MSFLA and Gabor filters for enhanced MR brain image recognition using SVM," International Journal of Tomography & Simulation, vol. 27, no. 3, pp. 3–5, 2014.
[7] A. Ladgham, A. Sakly, and A. Mtibaa, "MRI brain tumor recognition using modified shuffled frog leaping algorithm," in 2014 15th International Conference on Sciences and Techniques of Automatic Control and Computer Engineering (STA), 2014, pp. 504–507.
[8] K. O'Shea and R. Nash, "An introduction to convolutional neural networks," arXiv preprint arXiv:1511.08458, 2015.
[9] O. Russakovsky et al., "Imagenet large scale visual recognition challenge," Int J Comput Vis, vol. 115, no. 3, pp. 211–252, 2015.
[10] S. S. Farfade, M. J. Saberian, and L.-J. Li, "Multi-view face detection using deep convolutional neural networks," in Proceedings of the 5th ACM on International Conference on Multimedia Retrieval, 2015, pp. 643–650.
[11] E. Dandıl and R. Özdemir, "Real-time facial emotion classification using deep learning," Data Science and Applications, vol. 2, no. 1, pp. 13–17, 2019.
[12] S.-H. Lee, C.-H. Yeh, T.-W. Hou, and C.-S. Yang, "A lightweight neural network based on AlexNet-SSD model for garbage detection," in Proceedings of the 2019 3rd High Performance Computing and Cluster Technologies Conference, 2019, pp. 274–278.
[13] S. Samir, E. Emary, K. El-Sayed, and H. Onsi, "Optimization of a pre-trained AlexNet model for detecting and localizing image forgeries," Information, vol. 11, no. 5, p. 275, 2020.
[14] R. Yamashita, M. Nishio, R. K. G. Do, and K. Togashi, "Convolutional neural networks: an overview and application in radiology," Insights Imaging, vol. 9, no. 4, pp. 611–629, 2018.
[15] L. Alzubaidi et al., "Review of deep learning: Concepts, CNN architectures, challenges, applications, future directions," Journal of big Data, vol. 8, no. 1, pp. 1–74, 2021.
[16] Á. Arcos-García, J. A. Alvarez-Garcia, and L. M. Soria-Morillo, "Deep neural network for traffic sign recognition systems: An analysis of spatial transformers and stochastic optimisation methods," Neural Networks, vol. 99, pp. 158–165, 2018.
[17] D. Tabernik and D. Skočaj, "Deep learning for large-scale traffic-sign detection and recognition," IEEE transactions on intelligent transportation systems, vol. 21, no. 4, pp. 1427–1440, 2019.
[18] A. Arcos-Garcia, J. A. Alvarez-Garcia, and L. M. Soria-Morillo, "Evaluation of deep neural networks for traffic sign detection systems," Neurocomputing, vol. 316, pp. 332–344, 2018.
[19] M. Reke et al., "A self-driving car architecture in ROS2," in 2020 International SAUPEC/RobMech/PRASA Conference, 2020, pp. 1–6.
[20] J. Murphy, "An overview of convolutional neural network architectures for deep learning," Microway Inc, pp. 1–22, 2016.
[21] P. Sermanet and Y. LeCun, "Traffic sign recognition with multi-scale convolutional networks," in The 2011 international joint conference on neural networks, 2011, pp. 2809–2813.
[22] F. Jurišić, I. Filković, and Z. Kalafatić, "Multiple-dataset traffic sign classification with OneCNN," in 2015 3rd IAPR Asian Conference on Pattern Recognition (ACPR), 2015, pp. 614–618.