Turkish Journal of Electrical Engineering & Computer Sciences

http://journals.tubitak.gov.tr/elektrik/

Research Article

Turk J Elec Eng & Comp Sci
(2022) 30: 79 – 93
© TÜBİTAK
doi:10.3906/elk-2008-147

CNN based sensor fusion method for real-time autonomous robotics systems

Berat YILDIZ[1,*], **Akif DURDU**[2,4], **Ahmet KAYABASI**[1], **Mehmet DURAMAZ**[3]
[1]Electrical Electronics Engineering, Karamanoglu Mehmetbey University, Karaman, Turkey
[2]Electrical Electronics Engineering, Konya Technical University, Konya, Turkey
[3]R&D Robotics Software Engineer at Elfatek Elektronik Ltd. Şti., Konya, Turkey
[4]Robotic Automation Control Laboratory (RAC-LAB), Konya Technical University, Konya, Turkey

Received: 28.08.2020 • **Accepted/Published Online:** 03.09.2021 • **Final Version:** 19.01.2022

Abstract: Autonomous robotic systems (ARS) serve in many areas of daily life. The sensors have critical importance for these systems. The sensor data obtained from the environment should be as accurate and reliable as possible and correctly interpreted by the autonomous robot. Since sensors have advantages and disadvantages over each other they should be used together to reduce errors. In this study, Convolutional Neural Network (CNN) based sensor fusion was applied to ARS to contribute the autonomous driving. In a real-time application, a camera and LIDAR sensor were tested with these networks. The novelty of this work is that the uniquely collected data set was trained in a new CNN network and sensor fusion was performed between CNN layers. The results showed that CNN based sensor fusion process was more effective than the individual usage of the sensors on the ARS.

Key words: Autonomous robotic systems, deep learning, convolutional neural networks, sensor fusion

1. Introduction

The popular autonomous robotic systems (ARS) are capable of decision-making according to the environment. The main principle of autonomous robots is based on the processing and understanding of the data obtained from the sensors in the environment. Therefore, autonomous robots are equipped with sensors that are sensitive to factors such as sound, light, pressure and temperature [1]. The concept of autonomous vehicles emerged with the transfer of ARS to vehicles used in human, animal or freight transportation [2]. Major developments have been made in autonomous vehicle technology in recent years. Studies are carried out for autonomous vehicles such as safety [3], performance enhancement [4], energy and traffic efficiency [5, 6]. Autonomous vehicles can facilitate human life and provide safer transportation, but a simple mistake in autonomous systems can cause tragic accidents. Hence, the data obtained from the sensors should be as accurate and reliable as possible and should be interpreted correctly. The cameras and LIDAR sensors are the leading sensors used in autonomous vehicle technologies [7, 8], but these sensors have advantages and disadvantages compared to each other. The too high or too low light intensity can cause pixel loss in-camera images. Therefore, sometimes negative consequences can occur about obstacle detection [9]. The LIDAR sensor is also sensitive to bad weather [10]. In foggy, rainy and snowy weather, erroneous measurements may be encountered. More positive results can be obtained when data is provided from these sensors in the form of fusion [11, 12]. However, the geometric, temporal or spatial resolutions of the data received from these sensors differ. Chunlei et al. implemented coordinate

[*]Correspondence: beratyildiz@kmu.edu.tr

transformation on camera, LIDAR, and radar data using a multi-sensor fusion and object tracking algorithm (MFOTA) to detect the target object's location and speed accurately. Therefore, geometric, temporal or spatial transformations should be provided between sensor data for a meaningful and common data set [13]. Varuna De Silva et al. suggested spatial alignment using a series of geometric transformation formulas for camera and LIDAR data to avoid obstacles in free space detection. Afterward, Gaussian Process (GP) regression-based resolution matching algorithm was applied to determine lost and missing data [14]. Liang Xiao et al. were advocated a hybrid conditional random field-based (CRF) camera and LIDAR sensor fusion for road detection. Here, the contextual consistency of the sensor data was modeled probabilistically with the constraint of cross-modal consistency and provided appropriate results in urban road detection [15]. Correctly aligning 3D dots with image pixels between the camera and LIDAR is a very difficult process. Xie et al. proposed a new pixel and 3-D point alignment (PPA) method that directly calculates the alignment between the LIDAR point cloud and image pixels, without the need to calibrate the coordinate transformation matrix [16].

In the data fusion methods given above, the main problem is that the number of errors occurring during the coordinate transformation and alignment operations increases with the data density. Although different methods are used to prevent this, deep learning (DL) algorithms are more useful and less complex for path planning and solve many robotic problems [17–19, 33, 34]. It is effective in most areas such as object detection, localization, classification and control. The origin of DL was based on the training of an 8-layer neural network in 1971 using the group data processing algorithm [20]. In the 80s, the concept of DL was seen as a mathematical calculation model [21]. The concept of DL came into existence in the 90s when it was trained to perform visual object recognition [22]. Important developments were made in the DL algorithm in the 2000s that are closely related to computer technologies. Until then, slow computing technology and lack of dataset were negatively affecting DL's performance. More data processing computers were built with NVIDIA's GPU and the parallel computing architecture CUDA [23]. With many advances in technology, DL quickly became popular when the data set problem was overcome with the publication of a data set where 1.2 million images can be categorized into 1000 categories [24]. The success of DL against many problems was also reflected in autonomous robotic systems. Therefore, the use of DL structure can contribute to the development of autonomous systems in ground, sea and air vehicles.

In particular, DL model convolutional neural networks (CNN) are frequently used for object detection [25]. A multimodel fusion detection system (MFDS) between camera and LIDAR to strengthen object detection in autonomous vehicles was developed by Michael et al. Here, objects in images were distinguished using the CNN model and the point cloud consisting of LIDAR data was masked by camera images. The remaining points in the point cloud were divided into clusters according to the euclidean distance. Thus, the length of the object, its distance from the center and its direction were determined [26]. Babak et al. proposed a new multi-sensor fusion pipeline configuration for object detection and tracking. Here, an encoder-decoder-based fully convolutional neural network (FCNx) was used to increase the accuracy of path detection in real-time. Necessary nonlinear state estimates required for object tracking were obtained by making a fusion between camera, LIDAR and radar sensors using extended Kalman filter [27].

Unlike the studies done so far, the raw data obtained from the camera and LIDAR sensors were fused within the CNN network structure. Firstly, the point clouds created with the LIDAR sensor were converted into gray-scale images. Secondly, the LIDAR image and the camera image features were extracted passing through the convolution layer. Then, both data were combined and trained in the fully connected layer (FCL). As a result of the training and testing, the fusion model was compared with the LIDAR and camera model. All results were applied with a mobile autonomous car kit in a real-time environment. In this way, an error from a failed sensor can be compensated by other sensors through data fusion. Besides, applications showed that CNN-based sensor fusion gives better results than an individual sensor application.

2. Generating data sets

2.1. Autonomous vehicle kit

The study was carried out on the mini autonomous vehicle kit supplied by the OPENZEKA company. Also, it had used in the Massachusetts Institute of Technology (MIT) autonomous vehicle racing. The kit includes the NVIDIA Jetson TX2 developer kit processor which was produced for DL algorithms. It has a ZED camera for high-resolution images, Sparkfun Razor inertial measurement unit (IMU) M0 sensor, open power and charger, battery and special vehicle chassis. Besides, the RPLIDAR A2M8 LIDAR was added to the vehicle. The Logitech gamepad was used for manual control of the vehicle. The image of the montaged mini autonomous car kit is given in Figure 1.

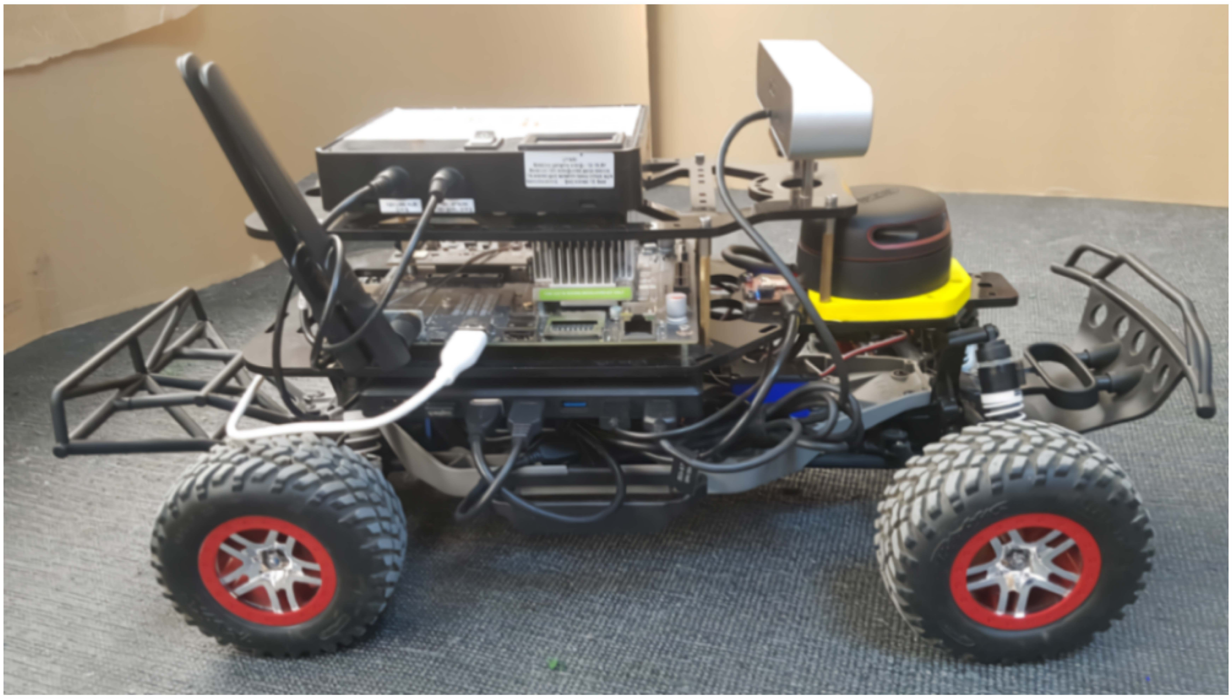

Figure 1. Autonomous vehicle development kit.

2.2. Data set collection

The data set to be used for training and testing operations were collected in the autonomous vehicle parkour (AVP) which was settled in the RACLAB (Robotics-Automation Control Research Laboratory) shown in Figure 2.

The manual controls of the vehicle were made by using a joystick in the AVP which was formed from cardboard and tarpaulin. Meantime the vehicle was in motion, camera images and LIDAR point space data were

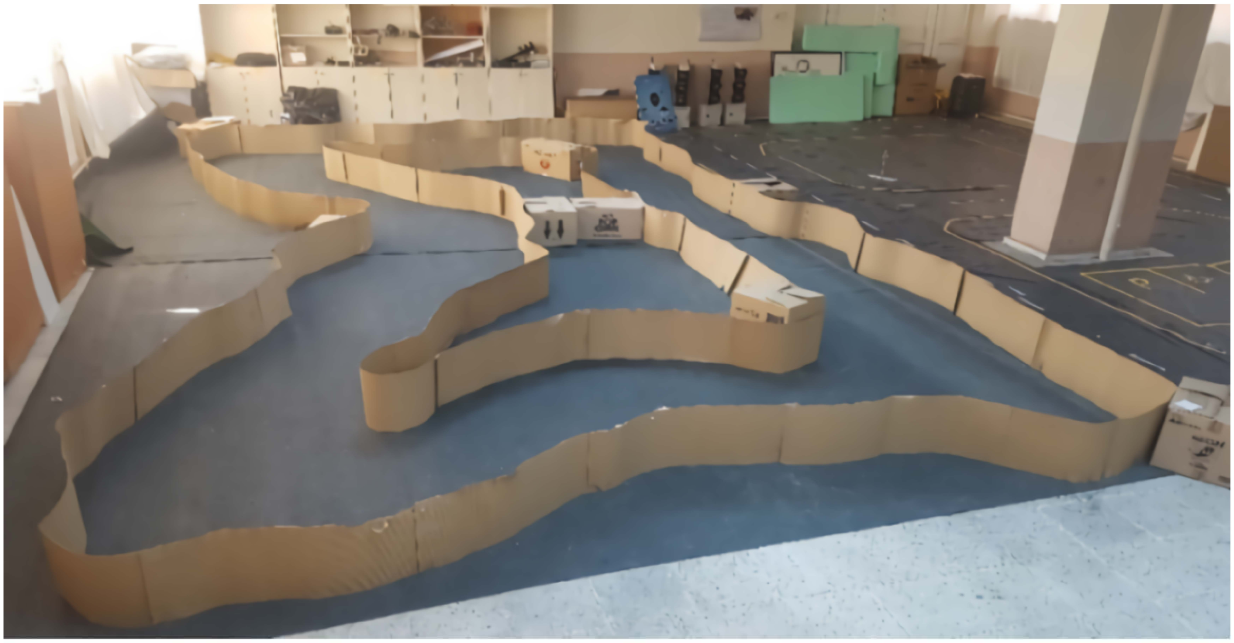

Figure 2. Autonomous vehicle parkour.

collected. The point space formed by the collected LIDAR data was instantly converted to grayscale images during data collection. First, two tours were completed around the AVP without stop. Then, for different scenarios that the vehicle may encounter during autonomous driving, the image and point space data were collected again by changing the position of the vehicle at cardboard edges, corners, narrow and sharp bends. As a result of all the data set studies, 5126 camera images and 5126 grayscale LIDAR images were obtained. The input parameters of the generated data set are camera and LIDAR images. The output parameter is the rotation angle of the vehicle's wheels. The turning capacity of the vehicle in right and left directions is limited to $[-63\ \ 63]$ degrees. This range is divided by π which is equal to 180 degrees and normalized to $[-0.35\ \ 0.35]$.

2.3. Obtaining images from LIDAR point data

The RPLIDAR A2 laser scanner sensor on the car kit is capable of scanning up to 25 meters at a 360-degree angle in 2D, but considering the established AVP characteristics, data from distances of 10 meters is sufficient. In the data set created with LIDAR, 3600 points were detected in 1 s. The point space containing the obstacle distances in the range of 0-10 m was normalized in the range of [0 1]. These normalized values were then converted to grayscale image pixels in the range [0 255] using Equation (1).

$$GIP(x) = \frac{X}{10}255 \tag{1}$$

Here, the grayscale image pixel values corresponding to the point space obtained from the LIDAR sensor are GIP and the distance of the obstacle to the LIDAR sensor is X. After the transformation, 360 points in the range of [0 255] obtained in one round are expressed with a 20×18 grayscale matrix. With this data set, the vehicle was not able to perform autonomous driving efficiently. It was found that the vehicle parts close to the LIDAR were also seen as obstacles. To solve this problem, the pixel values of LIDAR measurements at

15 cm and closer distances were changed to 255. Thus, these parts were shown as a white color in the pictures and it was stated that there was no obstacle. The point space created by LIDAR data with ROS-Rviz and the grayscale image obtained from this point space were given in Figure 3.

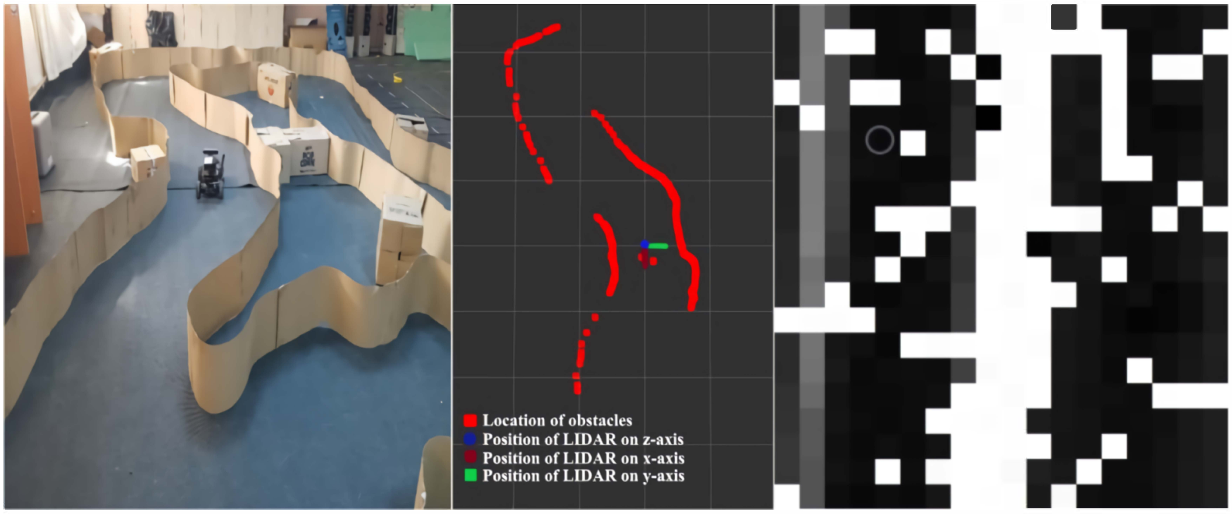
🟥 Location of obstacles
🟦 Position of LIDAR on z-axis
🟥 Position of LIDAR on x-axis
🟩 Position of LIDAR on y-axis

Figure 3. Grayscale transformation.

2.4. Data augmentation

The fact that the camera images were always the same type when creating the data set made it difficult for the vehicle to learn healthily. Therefore, to increase the diversity of data and to expand the data set, images taken from two cameras in the ZED camera were presented in a single frame. Thus, by combining the images into a single frame, an image capable of displaying a wider angle was obtained. In AVP, when the vehicle was in motion, the histogram curve of the camera images and images obtained from the LIDAR data were shown in Figure 4. Here, the intensity of the snapshots corresponding to the wheel angles was expressed. Negative $(-)$ values on the horizontal axis of the graph indicate left turn angles, while positive $(+)$ values indicate the right turn angles. On the other hand, it was observed that the number of images corresponding to the angle values close to 0 was higher in the training performed with these collected images. That is why, during autonomous driving tests, it was observed that the vehicle tended to go straight and that the wheels had vibrations. In sharp bends, despite the end of the bend, it continued its rotation and hit the barriers. To solve this problem, a new data set was created by increasing the number of images labeled with angles greater than 0.15 and less than -0.15. The histogram curve of the new data set created after all these data augmentation processes are given in Figure 4.

In the experiments with this new data set, it was found that the vehicle memorized the AVP and was sensitive to the minor changes made to the AVP. To avoid this problem, to increase the diversity in the data set and to prevent memorization, the brightness of the randomly selected pictures was changed. Thus, differences were created between similar or consecutive images. Thanks to this brightness difference, it was provided to learn about the small differences that may occur in the same images during the training and testing periods.

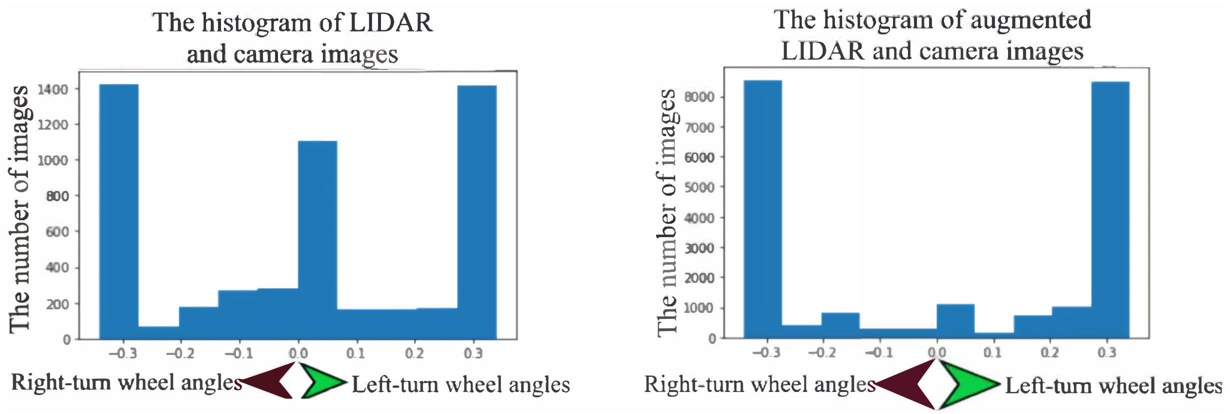

Figure 4. The histograms of Camera and LIDAR images

3. Convolutional neural networks

According to the nature of the problems to be solved, the DL algorithms can be examined in 3 groups, as in Figure 5, deep neural networks (DNN), convolutional neural networks (CNN) and recurrent neural networks (RNN).

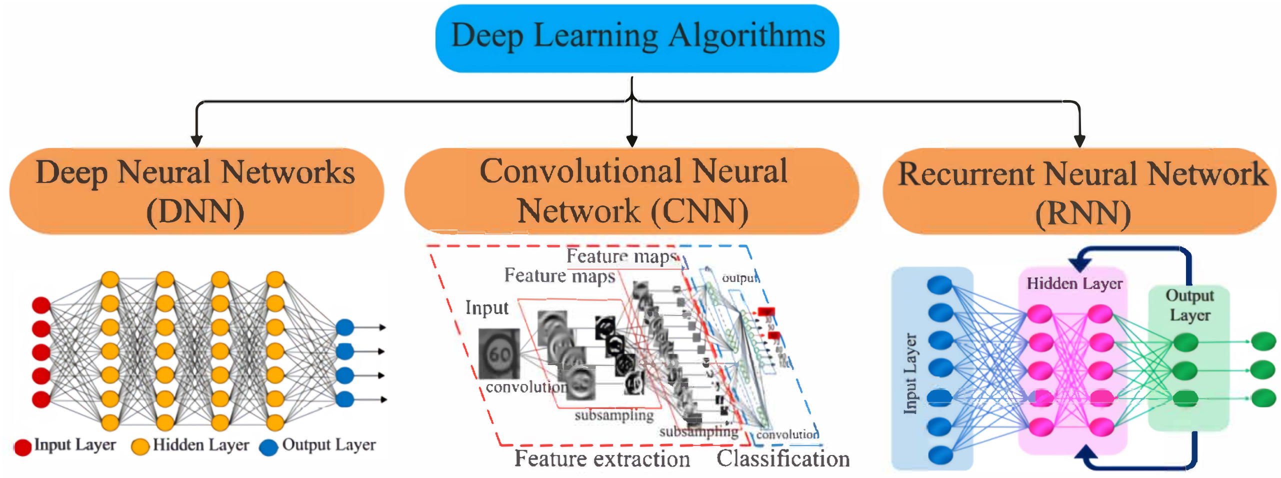

Figure 5. Deep learning algorithm [28].

The simplest of the DL algorithms is the DNN model. The DNN algorithm including neural network structures consists of three layers which are called the input layer, hidden layer and output layer. The main factor in calling this algorithm deep is that the number of layers in its structure is more than two [29]. DNN algorithm is an algorithm that includes classification and prediction models in which mathematical calculations are concentrated in parallel layers. RNN algorithm is used to train and learn the events in sequential relations, language models or time-series events by the machines. RNN is a DL algorithm formed by recording the output of data passing through a layer and estimating the new output of the layer by applying feedback to the input data. The DL algorithm, which is slightly more advanced than the DNN algorithm, is the CNN algorithm. The main difference of this algorithm compared to the DNN algorithm is feature extraction that is performed automatically and randomly in convolution layers. Thanks to a large number of random filters/kernels that can be applied to the input data in the convolution layer, the user does not need to perform feature extraction as in the DNN algorithm. Thus, the image features that will be transmitted to the neural network structure of very large data can be obtained in more detail. CNN algorithms are used in many applications such as object detection, obstacle recognition and object tracking where images are used as input. As in Figure 6, the pixel of the image overlapped by the pixel of the applied kernel is multiplied and the pixel value of the new image is generated.

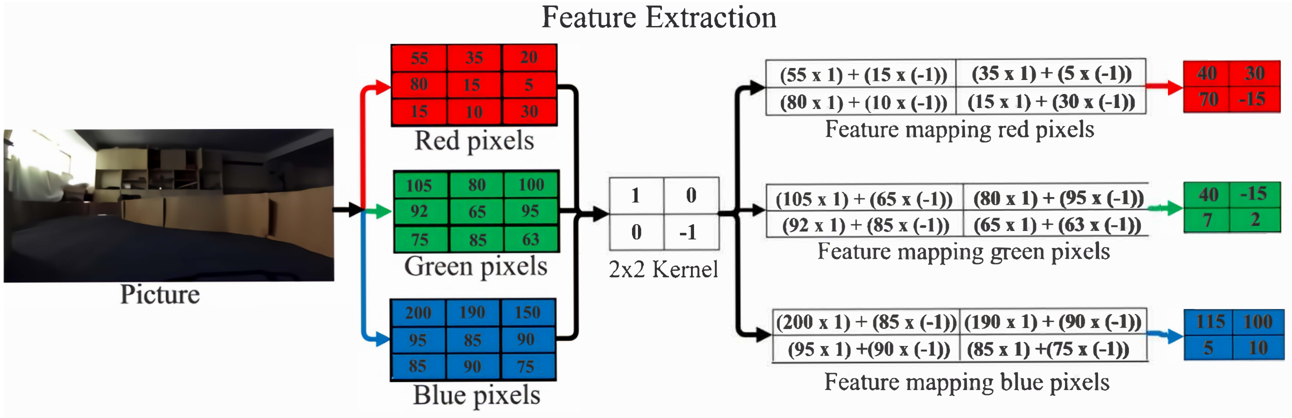
Feature Extraction
Picture
55 35 20
80 15 5
15 10 30
Red pixels
105 80 100
92 65 95
75 85 63
Green pixels
200 190 150
95 85 90
85 90 75
Blue pixels
1 0
0 -1
2x2 Kernel
(55 x 1) + (15 x (-1)) (35 x 1) + (5 x (-1))
(80 x 1) + (10 x (-1)) (15 x 1) + (30 x (-1))
Feature mapping red pixels
40 30
70 -15
(105 x 1) + (65 x (-1)) (80 x 1) + (95 x (-1))
(92 x 1) + (85 x (-1)) (65 x 1) + (63 x (-1))
Feature mapping green pixels
40 -15
7 2
(200 x 1) + (85 x (-1)) (190 x 1) + (90 x (-1))
(95 x 1) +(90 x (-1)) (85 x 1) +(75 x (-1))
Feature mapping blue pixels
115 100
5 10

Figure 6. Convolutional layer feature extraction.

When this operation is performed over all image pixels, a new filtered image is created. Then, one-dimensional matrices containing the features of the new image form the input parameters of the neural network structure given in Figure 7. After applying a series of mathematical calculations, the relationship between the input and the output parameter is determined with neural network structure Equation (2).

$$f\left(\sum_{i=1}^{n} x_i w_i + b\right) = y_i \tag{2}$$

Here, input parameters presented to the network x_i, weight values produced by the network according to the output value w_i, bias value b, activation function $f(.)$ and the output value is y_i. In this study, CNN algorithms are applied on camera and LIDAR images to provide obstacle avoidance.

3.1. Training of models

The concept of machine learning and artificial intelligence began to emerge by using a system that takes into account the structure of neurons in the human brain similar to the structure of computers [30]. Instead of complete information flow, the networks formed with the structure similar to neurons created the machine learning system by extracting the different features of some data and learning them. On the other hand, DL is a system that trains each of the data trained in networks, which is one of the subbranches of machine learning, in separate layers in advance [31]. In our study for autonomous vehicle navigation, the features of the dataset obtained from the camera and LIDAR images were first extracted separately in the CNN layers and then fused in the FCL. Eighty percent of all images formed after data duplication were used for training and the remaining 20% was used for testing. To use the output values in the network more efficiently in training and testing operations, the wheel angles labeled with the images [-0.35 0.35] were multiplied by 3 and spread to a wider

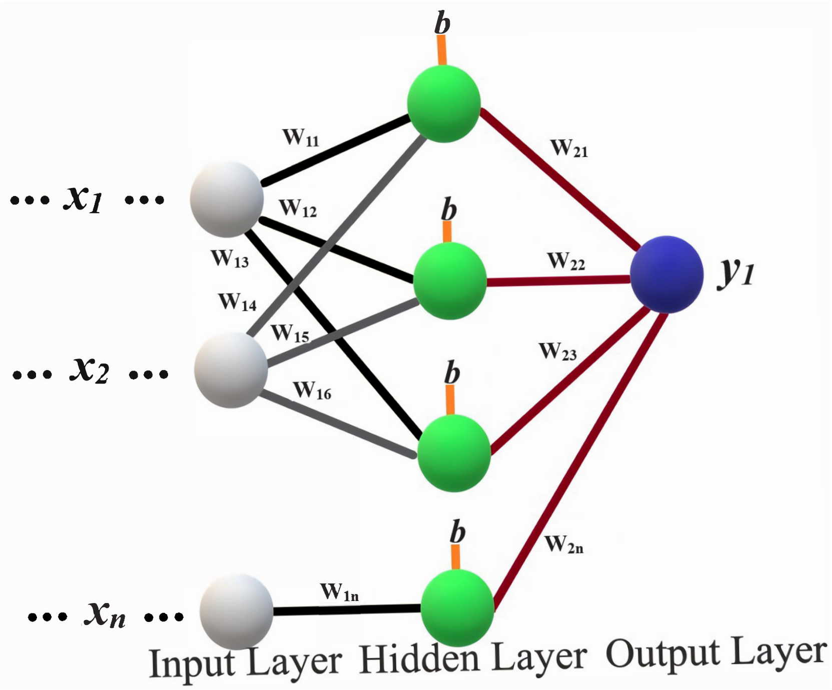

Figure 7. Neural network structure.

matrix [-1.05 1.05]. The main objective here was to make a comparison between the input data and the output data easier.

3.2. Training of camera image

Pictures taken from the camera with a size of 500 × 1470 pixels were cropped to a size of 376 × 1344. In this way, redundancies in the images required for training-testing procedures were eliminated and the time spent was saved. As a result of many designed network models, the most efficient CNN network model used for camera images was given in Figure 8. In this network model, the images were first filtered through certain sizes of Kernel and its features were extracted. Then, these data were trained in neural networks and the wheel angles were estimated at the output of the network. In the designed network structure, the pictures were transmitted to the FCL after passing through 7 convolution layers. In the convolution layers, 80 kernels in 3 × 3 dimension and 64, 32, 32, 16, 8 kernels in 2 × 2 dimension were applied to the images, respectively. Also, a 2 × 2 MaxPooling layer was applied in the 3rd convolution layer to speed up the training process and reduce the picture size without pixel loss.

The rectified linear unit (Relu) activation function given in Figure 9 was used to normalize the values in each convolution layer to prevent the parameter values to reach high steps and slow down the network while updating the weights of the training sequence. Multidimensional matrices passing through the activation functions were converted to one-dimensional matrices in the flatten layer and transferred to the FCL layers. The one-dimensional matrices were trained in 4 FCL layers with 1024, 256, 64 and 16 neurons, respectively. Also, following the FCL layer with 1024 neurons, 0.2 dropouts were applied to avoid network memorizing. Finally, before the output layer, the hyperbolic tangent (tanh) activation function given in Figure 9 was used. It normalized more efficiently than ReLu because the wheel angles were between [-1.05 1.05].

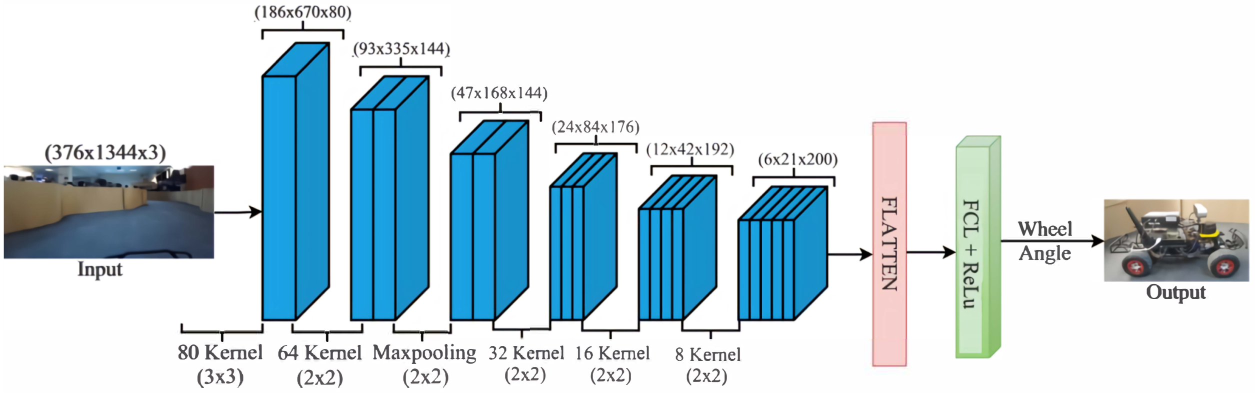

Figure 8. Designed CNN model for training of the camera images.

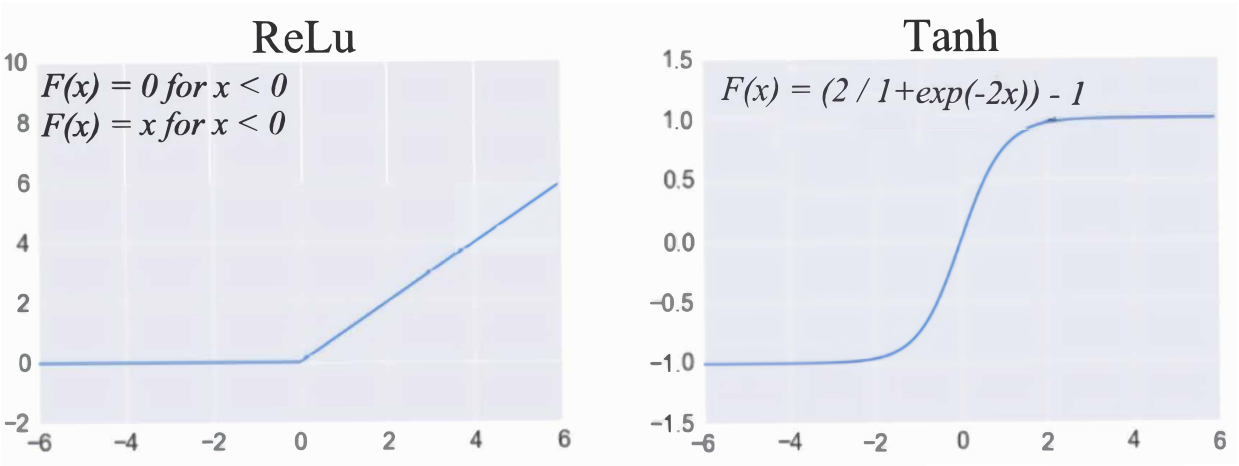

Figure 9. Rectified linear unit (ReLu) and hyperbolic tangent (Tanh) activation functions

During the training, the mean squared error (MSE) given in (3) was used as the loss function.

$$MSE = \frac{1}{n}\sum_{i=1}^{n}(y_i - \tilde{y_i})^2 \tag{3}$$

Here, the average of the square of the difference between the actual wheel angle in the dataset y_i and the wheel angle estimated by the designed CNN network model $\tilde{y_i}$ gives the MSE. The parameters used for training and testing in the CNN layer were given in Table 1.

3.3. Training of LIDAR image

Grayscale 20×18 images generated by transforming from LIDAR point cloud data were trained using the network structure given in Figure 10. In the designed network structure, the pictures are transmitted to the FCL layer after passing through 6 convolution layers. Sixty-four, 32, 32, 16, 16 and 8 kernels in 2×2 dimensions were applied to the images in the convolution layers, respectively. Relu and Tanh functions were also used as activation functions and were trained in 4 FCL layers with 1024, 256, 64 and 16 neurons, respectively, as in the

Table 1. Parameter values used in the CNN network model.

Parameters	
Optimization function	Adadelta
Loss	MSE
Activation function	ReLu,TanH
Learning rate (LR)	1.0
Epoch	30
Batch size	2
Validation split	0.2
Dropout camera/fusion model	0.2/0.3

training of camera images. The parameter values used to train the LIDAR images are the same as those given in Table 1.

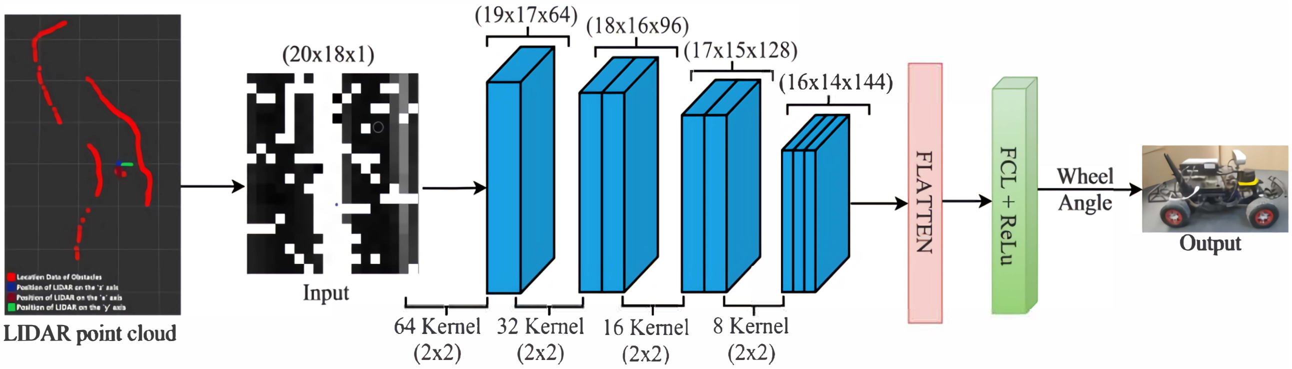

Figure 10. Designed CNN model for training of the LIDAR images.

3.4. Training of fusion data

The images obtained from the camera and the LIDAR were separately passed through the convolution layers of the CNN network model and their features were extracted. The extracted features were passed through the flatten layer and transformed into one-dimensional matrices. The feature element number of the one-dimensional matrix obtained from the camera images was considerably higher than the LIDAR images. When the data were fused in this state before the training was performed the results from the camera data predominate than LIDAR during autonomous driving. To avoid this, the data was first passed through an FCL with 1024 neurons and then fused. With this process, the features extracted from the camera and LIDAR were transferred to an equal number of neurons and represented by an equal number of weights at the output of these neurons. Thus, the effects of camera and LIDAR features on the decision-making mechanism were equalized before data fusion. After data fusion was achieved, training of these data continued in 4 FCL layers with 1024, 256, 64 and 16 neurons, respectively. As in the other models, the Relu activation function was used at each layer's output in the fusion model and the tanh activation function was also used in the final output layer. The CNN network model designed for combining camera and LIDAR data is given in Figure 11.

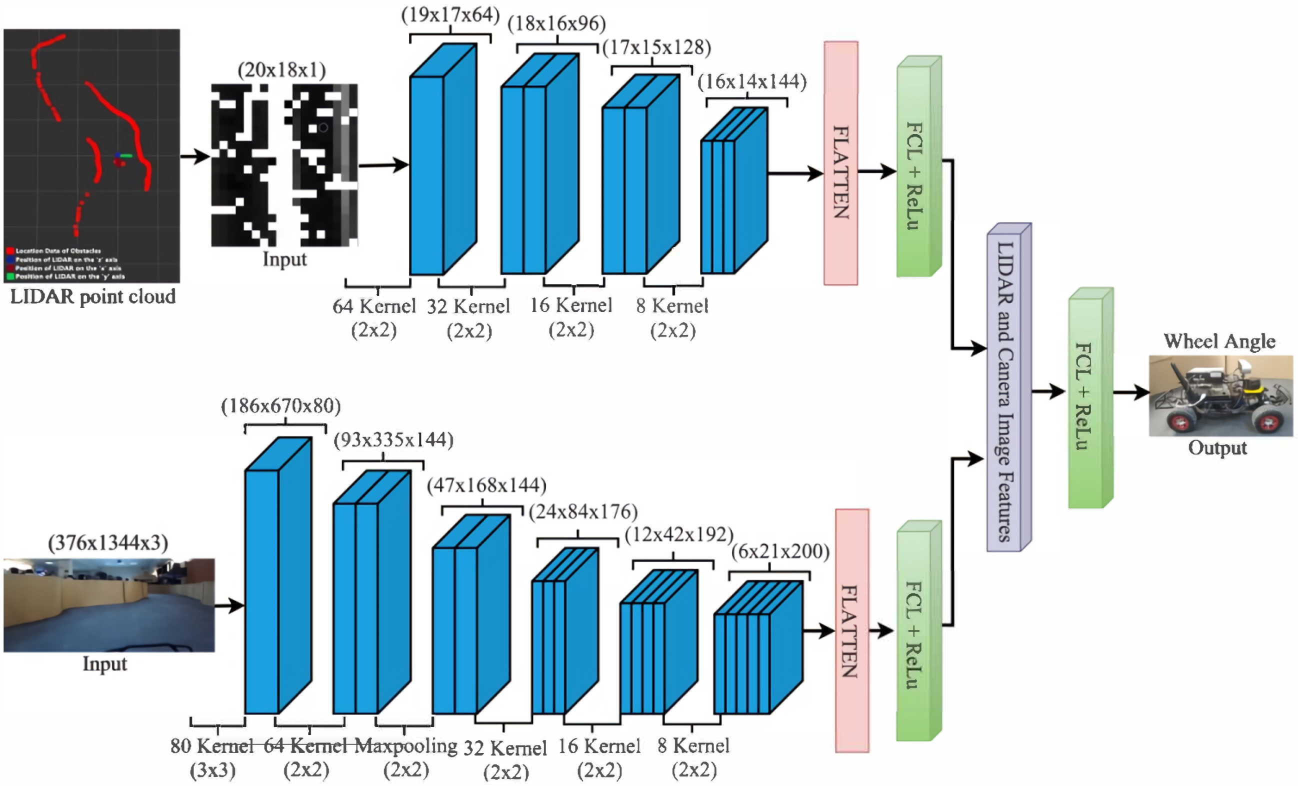

Figure 11. Designed CNN model for training of the fusion data.

4. Result and discussion

MSE states how close a regression curve is to a series of points [32]. In machine learning models, it is generally preferred to measure the performance of predicted values resulting from learning. The fact that the MSE value is close to 0 as a performance criterion shows that the predictions produced by the learning are close to the real value and that the training process is efficient. However, this value is not desirable to be zero in machine learning applications. This is considered as an indication that the network is memorized. Training error function value (Loss), training accuracy value (Accuracy), test error function value (Validation Loss) and test accuracy value (Validation Accuracy) obtained as a result of training and testing performed with network models are given in Table 2. Here, Loss is the MSE value between input and output during training. Accuracy is the output rate that the network predicts correctly during training. Validation Loss is the MSE value between input and output during testing. Validation Accuracy is the output rate that the network estimates correctly during the test.

The type of problem is regression, therefore, the accuracy metric can be different for training and validation. That is why the vehicle completed the AVP autonomously without any problems. Besides, changes in error functions during training and testing with camera, LIDAR and combined data are shown in Figure 12 as another evaluation criterion. The fact that the curves of the training and testing error functions are close to each other is an indication that the training process is efficient.

As seen in Figure 12, the closeness of the results of the training and test error functions performed with the fused data is an indication that the fusion model provides better learning than the camera and LIDAR model. In line with these data obtained from the training and test results, autonomous driving tests were

Table 2. Performance evaluation criteria of the CNN network model.

Error functions	**Camera training error rate**	**LIDAR training error rate**	**Fusion training error rate**
Loss	0.0020	0.0048	0.0009
Accuracy	0.0388	0.0386	0.039
Validation Loss	0.0037	0.0116	0.0041
Validation Accuracy	0.0361	0.0347	0.0376

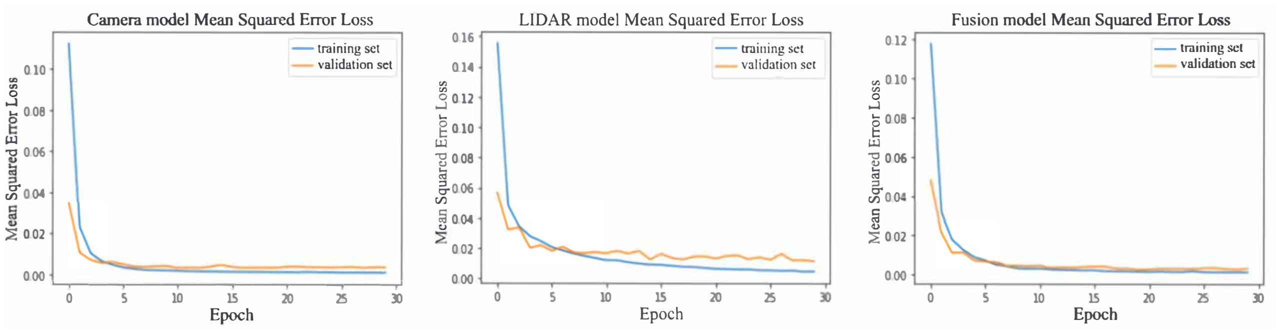

Figure 12. Loss graph of camera, LIDAR and fusion data, respectively.

carried out on the vehicle. According to test drives, the vehicle completed AVP laps autonomously without hitting obstacles. Moreover, changes were made on the AVP to see if the vehicle had memorized the AVP or learned, and autonomous driving was retried on the new AVP with these changes. As a result of the trials, it was observed that the vehicle could complete the new AVP where the vehicle was driving autonomously without hitting the obstacles.

As another comparison criterion, the wheel angle values were recorded separately for the camera, LIDAR and combined models when the vehicle was driving autonomously on the AVP shown in Figure 13. The recorded values are taken into account, it is seen that the fusion model performs closer to manual driving (ground truth) than other models. According to the wheel angle graph obtained from the camera model, the frequent change of wheel angles in the positive or negative direction shows that the vehicle was wobbled in the AVP. The very rapid change in angles of the LIDAR model has also shown that there were vibrations in the wheels of the vehicle during autonomous driving. On the other hand, wobble and vibration problems are less frequent in the fusion model. From this, it can be concluded that less energy was consumed and a more efficient automatic driving was achieved with the sensor fusion technique. In addition, while AVP was completed in the range of 34-40 s with the camera and LIDAR model, this time decreased to 28-33 s with the fusion algorithm.

5. Conclusion

The results obtained on the computer were tested on the autonomous car kit in real-time. According to the results of the training and testing performed only with the camera, the autonomous vehicle completes the AVP with small errors, but changes in the light intensity that may occur in the environment adversely affect the vehicle and cause the vehicle to hit the edges of the AVP. Excessive wobbling and vibrations are observed on the wheels when driving with images obtained only from LIDAR data. Therefore, it can be stated that when

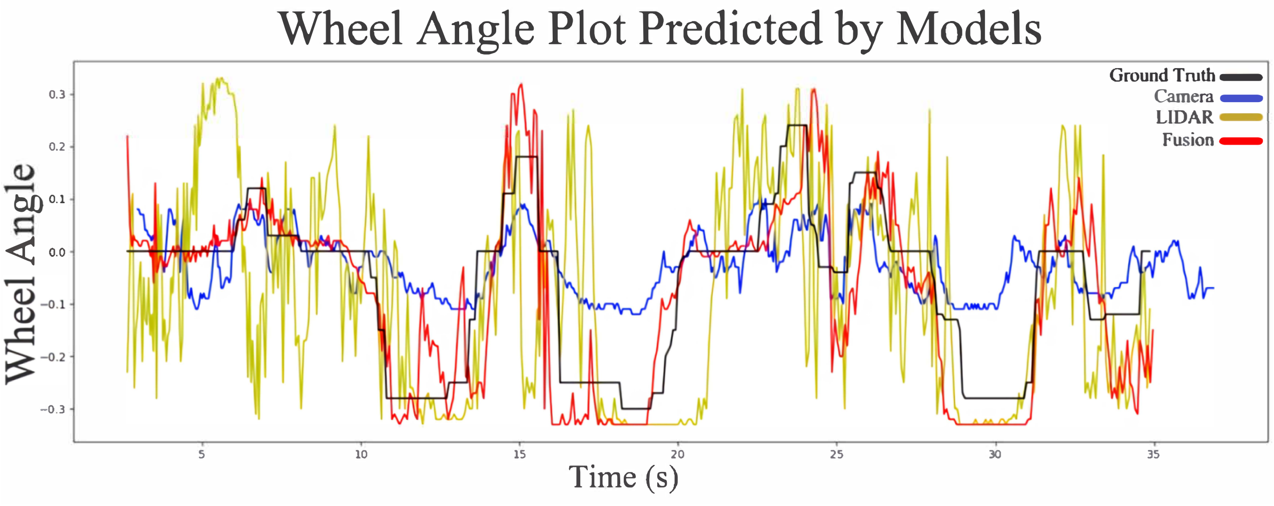

Figure 13. Wheel angle graphs of camera, LIDAR and fusion models according to ground truth.

the camera and LIDAR sensors are used alone, positive results can be obtained in limited areas. During the test drives performed after the fusion process with camera and LIDAR data, the problem of wobbling and shaking caused by the LIDAR was reduced significantly. This allowed the vehicle to complete the AVP autonomously and more quickly. It was also observed that during the test drives the problems associated with the camera caused by the increase or decrease of brightness in ambient lighting are greatly overcome by the fusion system. With this developed sensor fusion system, LIDAR will be able to perform the camera function in case of insufficient camera vision or failure in autonomous driving at night. In bad weather conditions such as fogy or rainy, deviations in LIDAR data can be corrected by camera data. In this way, more efficient autonomous driving will be realized and the problems caused by sensor losses will be reduced.

In the later stages of our study, different fusion techniques will be tried for autonomous driving. Apart from the camera and the LIDAR sensor, different sensor data, such as the IMU, can be fused in different ways and cause reducing system errors. Moreover, depth data of stereo cameras and LIDAR data can be processed together and more efficient results can be obtained in obstacle detection.

Acknowledgment

The authors are thankful to RAC-LAB (www.rac-lab.com) for providing the trial version of their commercial software for this study.

Conflict of Interest: The authors declare that they have no conflict of interest.

References

[1] Asama H, Fukuda T, Arai T, Endo I. Distributed autonomous robotic systems 2. Springer Science and Business Media, 2013

[2] Fraedrich E, Beiker S, Lenz B. Transition pathways to fully automated driving and its implications for the sociotechnical system of automobility. European Journal of Futures Research 2015; 3 (1): 11. doi:10.1007/s40309-015-0067-8

[3] Virdi N, Grzybowska H, Waller ST, Dixit V. A safety assessment of mixed fleets with Connected and Autonomous Vehicles using the Surrogate Safety Assessment Module. Accident Analysis and Prevention 2019; 131: 95-111. doi:10.1016/j.aap.2019.06.001

[4] Loeb B, Kockelman KM. Fleet performance and cost evaluation of a shared autonomous electric vehicle (SAEV) fleet: A case study for Austin, Texas. Transportation Research Part A: Policy and Practice 2019; 121: 374-385. doi:10.1016/j.tra.2019.01.025

[5] Waseem M, Sherwani AF, Suhaib M. Integration of solar energy in electrical, hybrid, autonomous vehicles: a technological review. SN Applied Sciences 2019; 1 (11): 1459. doi:10.1007/s42452-019-1458-4

[6] Gao Z, Wang J, Zhang X, Dong J, Chen L et al. Traffic oscillations mitigation in vehicle platoon using a car-following control model for connected and autonomous vehicle. Journal of Advanced Transportation 2019. doi:10.1155/2019/3067291

[7] Xu Q, Yang Y, Zhang C, Zhang L. Deep convolutional neural network-based autonomous marine vehicle maneuver. International Journal of Fuzzy Systems 2018; 20 (2): 687-699. doi:10.1007/s40815-017-0393-z

[8] Tahboub K, Güera D, Reibman AR, Delp EJ. Quality-adaptive deep learning for pedestrian detection. In 2017 IEEE International Conference on Image Processing (ICIP), 2017. pp. 4187-4191.

[9] Schlosser J, Chow CK, Kira Z. Fusing lidar and images for pedestrian detection using convolutional neural networks. In 2016 IEEE International Conference on Robotics and Automation (ICRA), Stockholm, Sweden; 2016. pp. 2198-2205. doi:10.1109/ICRA.2016.7487370

[10] Premebida C, Carreira J, Batista J, Nunes U. Pedestrian detection combining RGB and dense LIDAR data. In 2014 IEEE/RSJ International Conference on Intelligent Robots and Systems, Chicago, USA; 2014. pp. 4112-4117. doi:10.1109/IROS.2014.6943141

[11] Adarsh S, Ramachandran KI. Design of sensor data fusion algorithm for mobile robot navigation using ANFIS and its analysis across the membership functions. Automatic Control and Computer Sciences 2018; 52 (5): 382-391. doi:10.3103/S0146411618050036

[12] Bostanci E, Bostanci B, Kanwal N, Clark AF. Sensor fusion of camera, GPS and IMU using fuzzy adaptive multiple motion models. Soft Computing 2018; 22 (8): 2619-2632. doi:10.1007/s00500-017-2516-8

[13] Yi C, Zhang K, Peng N. A multi-sensor fusion and object tracking algorithm for self-driving vehicles. Proceedings of the Institution of Mechanical Engineers, Part D: Journal of automobile engineering 2019; 233 (9): 2293-2300. doi:10.1177/0954407019867492

[14] De Silva V, Roche J, Kondoz A. Robust fusion of LiDAR and wide-angle camera data for autonomous mobile robots. Sensors 2018; 18 (8): 2730. doi:10.3390/s18082730

[15] Xiao L, Wang R, Dai B, Fang Y, Liu D et al. Hybrid conditional random field based camera-LIDAR fusion for road detection. Information Sciences 2018; 432: 543-558. doi:10.1016/j.ins.2017.04.048

[16] Xie S, Yang D, Jiang K, Zhong Y. Pixels and 3-D points alignment method for the fusion of camera and LiDAR data. IEEE Transactions on Instrumentation and Measurement 2018; 68 (10): 3661-3676. doi:10.1109/TIM.2018.2879705

[17] Graves A, Mohamed A.R, Hinton G. Speech recognition with deep recurrent neural networks. In 2013 IEEE international conference on acoustics, speech and signal processing, Vancouver, Canada; 2013. pp. 6645-6649. doi:10.1109/ICASSP.2013.6638947

[18] Lenz I, Lee H, Saxena A. Deep learning for detecting robotic grasps. The International Journal of Robotics Research 2015; 34 (4-5): 705-724. doi:10.1177/0278364914549607

[19] Yang P.C, Sasaki K, Suzuki K, Kase K, Sugano S et al. Repeatable folding task by humanoid robot worker using deep learning. IEEE Robotics and Automation Letters 2016; 2 (2): 397-403. doi:10.1109/LRA.2016.2633383

[20] Ivakhnenko AG. Polynomial theory of complex systems. IEEE transactions on Systems, Man, and Cybernetics 1971; (4): 364-378. doi:10.1109/TSMC.1971.4308320

[21] Fukushima K. Neocognitron: A hierarchical neural network capable of visual pattern recognition. Neural networks 1988; 1 (2): 119-130. doi:10.1016/0893-6080(88)90014-7

[22] LeCun Y, Bottou L, Bengio Y, Haffner P. Gradient-based learning applied to document recognition. Proceedings of the IEEE 1998; 86 (11): 2278-2324. doi:10.1109/5.726791

[23] NVIDIACorporation. what is CUDA. 2019.

[24] Deng J, Dong W, Socher R, Li LJ, Li K et al. Imagenet: A large-scale hierarchical image database. In 2009 IEEE conference on computer vision and pattern recognition, Miami, FL; 2009. pp. 248-255. doi:10.1109/CVPR.2009.5206848

[25] Dhillon A, Verma GK. Convolutional neural network: a review of models, methodologies and applications to object detection. Progress in Artificial Intelligence 2020; 9 (2): 85-112. doi:10.1007/s13748-019-00203-0

[26] Person M, Jensen M, Smith AO, Gutierrez H. Multimodal Fusion Object Detection System for Autonomous Vehicles. Journal of Dynamic Systems, Measurement, and Control 2019; 141 (7): 071017. doi.org/10.1115/1.4043222

[27] Shahian Jahromi B, Tulabandhula T, Cetin S. Real-Time Hybrid Multi-Sensor Fusion Framework for Perception in Autonomous Vehicles. Sensors 2019; 19 (20): 4357. doi:10.3390/s19204357

[28] NVIDIACorporation. Convolutional Neural Network (CNN). 2019.

[29] Hinton GE. Learning multiple layers of representation. Trends in cognitive sciences 2007; 11 (10): 428-434. doi:10.1016/j.tics.2007.09.004

[30] Göranzon B, Florin M. Artifical Intelligence, Culture and Language: On Education and Work. Springer Science and Business Media, 2012.

[31] LeCun Y, Bengio Y, Hinton G. Deep learning. Nature 2015; 521 (7553): 436-444. doi:10.1038/nature14539

[32] Willmott CJ, Matsuura K. Advantages of the mean absolute error (MAE) over the root mean square error (RMSE) in assessing average model performance. Climate research 2005; 30 (1): 79-82. doi:10.3354/cr03007915

[33] Durdu A, Bol N, Öztürk E, Duramaz M, Korkmaz M et al. Convolutional Neural Networks Based Active SLAM and Exploration. Avrupa Bilim ve Teknoloji Dergisi 2021; (22): 342-346. doi:10.31590/ejosat.862953

[34] Durdu A, Korkmaz M. A novel map-merging technique for occupancy grid-based maps using multiple robots: a semantic approach. Turkish Journal of Electrical Engineering and Computer Sciences 2019; 27 (5): 3980-3993. doi:10.3906/elk-1807-335