Research Article

Detection Of Traffic Signs for Autonomous Driving with The Deep Learning Method

Authors: Hayati AKGÜN, Sinan UĞUZ

To cite to this article: Akgün, H. & Uğuz, S. (2023). DETECTION OF TRAFFIC SIGNS FOR AUTONOMOUS DRIVING WITH THE DEEP LEARNING METHOD . International Journal of Engineering and Innovative Research ,5(3), 259-267. DOI: 10.47933/ijeir.1358959

DOI: 10.47933/ijeir.1358959

To link to this artice: https://dergipark.org.tr/tr/pub/ijeir/archive

International Journal of Engineering and Innovative Research

http://dergipark.gov.tr/ijeir

Detection Of Traffic Signs for Autonomous Driving with The Deep Learning Method

Hayati AKGÜN [1], Sinan UĞUZ [2*]

[1]Isparta University of Applied Sciences, Faculty of Technology, Mechatronics Engineering Department, Isparta, Turkey.
[2]Isparta University of Applied Sciences, Faculty of Technology, Computer Engineering Department, Isparta, Turkey.

[*]Corresponding Author: sinanuguz@isparta.edu.tr
(**Received:** 12.08.2023; **Accepted:** 17.09.2023)

https://doi.org/10.47933/ijeir.1358959

Abstract: Deep learning practices used in many fields, in particular, in health, military, economy, and production industries, are an important area of artificial intelligence in our age. The object classification and object recognition applications, which play a significant role in the development of autonomous vehicle technologies, constitute the focal point of the deep learning studies. It is clear that the recent studies based on the deep learning models show that they are useful and successful performances for safe driving not only for vehicles but also for pedestrians. It is very crucial and significant that the autonomous systems recognize the traffic signs with high accuracy for a safe driving. Especially, the pedestrian crossing, school district, urban speed limits can be regarded among the most critical traffic signs. In this study, we have used the data set including the traffic signs obtained by our own means to carry out trainings by using faster R-CNN which is regarded as one of the most important recognition architectures. Thanks to the hardware module produced as a result of the operation, we have developed a system that warns the driver of the vehicle with audible warning. The developed hardware module can detect not only the speed limits, traffic signs but also pedestrian crossings and school districts and alert the driver in reel-time. The developed hardware module is based on Arduino and because of the GPS sensor, it can also show the speed of the vehicle. Moreover, we have used Python for the developed software and the dataset trainings have been carried out by using the Tensorflow library. We think that the study will contribute a lot to the recognition of traffic signs for the autonomous vehicle applications.

Keywords: CNN, image processing, traffic signs recognition, object detection.

1. INTRODUCTION

Traffic accidents pose significant challenges for countries worldwide. Annually, 1.35 million lives are lost, with 50 million individuals sustaining injuries as a result. Particularly alarming is the fact that traffic-related injuries stand as the leading cause of death for the population aged 5-29, and overall, they rank as the eighth leading cause of death across all age groups. Disturbingly, statistical data indicates an annual increase in these numbers [1]. Beyond the obvious human toll, traffic accidents result in substantial financial burdens. When considering healthcare costs and the economic fallout stemming from damages to vehicles, roads, and various other factors, it is estimated that losses from traffic accidents equate to 2.2% to 2.7% of a country's gross domestic product [2]. It is important to note that the toll extends beyond mere monetary losses, encompassing emotional and psychological hardships. Consequently, attempts to quantify these losses may not fully capture their true extent [3].

Considering the injuries, deaths, and financial damages caused by traffic accidents, serious efforts are being made to enhance traffic safety. Some of these efforts focus on increasing the knowledge and awareness of drivers and pedestrians, while others aim to regulate, improve, and enhance road-related factors. Moreover, there are efforts to improve the functionality and performance of vehicles.

In recent years, advancements in autonomous vehicle technologies have reached impressive levels. In particular, the development of deep learning-based models has significantly contributed to rapid and reliable object recognition processes required for autonomous vehicles [4]. By using deep learning models, real-time information about traffic congestion, the number of vehicles, and road conditions can be obtained [5]. As a result, measures can be taken to prevent accidents or risks in traffic. Various datasets from countries such as Sweden [6], Germany [7], China [8], and India [9] have been compiled for deep learning studies based on traffic signs and signals. Factors such as weather conditions, distance, and lighting play crucial roles in preparing these datasets, as improving these factors significantly impacts the accuracy of traffic sign recognition processes [10].

Stallkamp et al. [11] aimed to recognize traffic signs using machine learning algorithms. They used a publicly available dataset containing images of over 50,000 German road signs from 43 different classes. They achieved a classification accuracy of 99.46% using CNN. Kim and Lee [12] worked on lane detection using the Random Sample Consensus (RANSAC) algorithm and CNN networks. They first calculated edges on images and then used RANSAC in conjunction with CNN to detect lanes. CNN was used to detect lanes on complex roads. The CNN architecture consisted of 3 convolutional layers, 2 subsampling layers, and 3 fully connected layers. As a result, lane detection successfully eliminated noisy lines, and performance-wise, it outperformed other lane detection algorithms like RANSAC and Hough Transform. Qian et al. [13] conducted their study by applying the CNN method to achieve high performance in terms of traffic sign detection speed and recognition accuracy. They used Chinese traffic signs and the GTSDB dataset, covering a total of 96 different traffic sign groups. The traffic sign recognition process was conducted in both Chinese and English. The average accuracy rate of the study was reported as 97.56%.

Acros et al. [14] conducted comparative analyses of eight different CNN models trained on the German Traffic Sign Detection Benchmark dataset and the Microsoft COCO dataset. They created three classes for this purpose. R-FCN provided the best results in terms of speed (R-CNN Initial Resnet V2, 95.77%) and accuracy (R-FCN Resnet 101, 95.15%). It was noted that SSD models struggled to detect small traffic signs. Additionally, it was reported that MobileNet-based models were the most suitable architectures for mobile and embedded devices. Hussian et al. [15] utilized the Fast Branch architecture, one of the CNN models, for traffic sign recognition. They completed their study with a 98.52% accuracy rate.

In this study, Object Recognition using Deep Learning was employed to recognize traffic signs and signals. Specifically, certain signs that are particularly risky in urban traffic were selected, prepared with image processing techniques, and trained using the TensorFlow library. The speed of the vehicle was measured through a GPS sensor connected to an Arduino module using a developed Python-based software. The data trained with the TensorFlow library successfully recognized traffic signs with high accuracy. If the vehicle speed exceeded a predefined threshold for the recognized traffic sign, the driver was alerted audibly. The primary goal of this study is to develop a helpful driving assistant system for drivers in urban areas, especially for those with attention deficits or weakened driving reflexes due to age. It is believed that the developed audio assistant system will be beneficial in reducing accidents involving such drivers.

1.1. Convolutional neural networks

As illustrated in Figure 1, a typical CNN model consists of two key stages: feature learning and classification. The convolution and subsequent pooling steps may be repeated multiple times, depending on the specific architecture of the CNN being developed. The convolution process involves breaking the image into smaller parts that can move both vertically and horizontally across the image in square or rectangular shapes. The stride values determine the rate at which these image parts move in these directions. Another crucial element in the convolution process is the kernel (also known as a filter), which represents a set of learnable weights corresponding to the size of the image parts. These weights are used to compute inner products between the image part and the kernel, resulting in a digital value [16]. This process is repeated for each sliding position, generating a filtered image.

In the convolution process, if there are many distinct patterns in the image, the number of parameters increases, making the network more complex and increasing the computational load [17]. Pooling operations are employed to downscale the output of the convolution process. After pooling, and before reaching the fully connected layer stage, the resulting image is transformed into one-dimensional data through a flattening process. Finally, the fully connected layers (also referred to as dense layers) take this processed data as input to make a classification decision. This marks the completion of the classical neural network stages, culminating in the classification process.

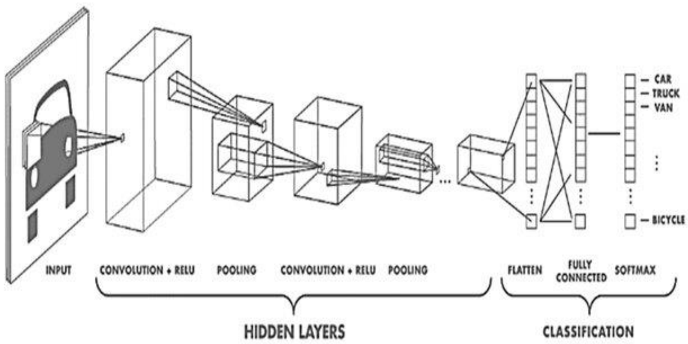

Figure 1. Classical CNN architecture [21]

CNN architectures emerged in 1998 with LeNet-5 [18]. Subsequently, in 2012, AlexNet was introduced [19]. These developments paved the way for the creation of various other CNN architectures that provided effective solutions to object classification and object detection problems.

These architectures experimented with numerous convolutional operations, pooling operations, and various types of activation functions. The ImageNet Large Scale Visual Recognition Challenge competition (LLSVRC), also known as the ImageNet Challenge, played a significant role in this progress. This competition involved the classification of millions of training images across various class labels.

The VGG net architecture, which includes two versions, VGG-16 and VGG-19 [20], was developed by the Visual Geometry Group at the University of Oxford and emerged as the winner of the 2014 ImageNet Challenge.

2. METHODS

In the study, the first step involved obtaining images required for image processing. These images were categorized into 5 groups, specifically set as speed limit, school zone, pedestrian crossing, traffic lights, and radar. After organizing the images in this manner, 20% of them were allocated for testing, while the remaining 80% were used for training. Following this process, labeling was performed using the LabelImg software. With LabelImg, a .xml file is generated for each image in which the file contains the class name of the labeled object and the coordinates of the object's location.

Subsequently, the .xml files were converted to the CSV format, which is compatible with the TensorFlow library. In the next step, for the images categorized into 5 groups, each class was assigned an ID number along with class names. Then, the deep learning model, specifically Faster R-CNN, was utilized, and necessary adjustments were made to the model.

In this study, Python programming language was used in conjunction with libraries such as TensorFlow, Keras, NumPy, Pillow, lxml, Cython, Matplotlib, Pandas, and OpenCV. The model training concluded after 8 hours and 94,932 iterations.

2.1. Dataset

In this study, the images necessary for traffic safety were obtained within the boundaries of Isparta, a city in Turkey. The dataset consists of 2,565 images. These traffic signs are categorized into five different groups, with a focus on signs deemed to be of greater importance. The resolution of the images was set to an average of 2448x3264 pixels.

In the study, the total number of images used for speed limits is approximately 930. As shown in Figure 2, when these sample images are processed by the system, audible warnings are provided with the voice prompt "speed limit."

Figure 2. Speed limit signs

Figure 3 provides the pedestrian crossing signs and signals used in the model. A total of 615 of these traffic signs were utilized. When these traffic signs are detected by the system, it provides an audible warning with the voice prompt "caution, pedestrian crossing." Additionally, if a driver exceeds the speed limit at pedestrian crossings, they will receive the same type of audible warning.

Figure 3. School zone signs

Figure 4 displays the traffic lights and signals used in the study. A total of 690 images were obtained from these signals. When the system detects these images, it provides an audible warning by saying "lights".

Figure 4. Traffic lights signs

In Figure 5, various traffic radar warning signs used in the application are provided. There are a total of 150 images related to radar signs in the study. When these signs are detected by the model, the system provides an audible warning by saying "radar".

Figure 5. Radar signs

3. EXPERIMENTAL

In this study, model training was conducted using the Faster R-CNN architecture, which is a region proposal network, through the transfer learning technique. Firstly, cropping operations were applied to remove unwanted portions from the images. Subsequently, these images were fed into the Faster R-CNN model. For the study, 542 images were reserved for testing, while 2,023 images were allocated for training, and the training was carried out.
For the model testing phase, as shown in Figure 6, a USB camera connected to a computer inside the vehicle and a GPS module connected to an Arduino board were used to obtain real-

time speed data of the vehicle. This real-time speed data was transferred to the system, and the necessary warning systems related to traffic signs were provided to the driver.

Figure 6. Testing the model

In Figure 7, the test results of the system are presented. It can be observed that traffic signs were detected with an overall success rate of over 90%.

Figure 7. Results of the model

4. RESULTS AND DISCUSSION

The numbers of traffic signs and signals read by the model from a moving vehicle during the testing process are provided in Table 1 for different speed limits. The vehicle speed was set at 30, 40, 50, and 60 km/h, taking into account urban speed limits. It can be observed that the best results were obtained at a speed of 30 km/h when the system encountered pedestrian crossings, school zones, speed limits, traffic lights, and radar signs. The primary reasons for this can be attributed to the processing speeds of the camera and graphics card used.

Table 1. The rate of detecting traffic signs of the system at different speeds

	60 km/h		**50 km/h**		**40 km/h**		**30 km/h**	
Traffic signs	**Sign**	**Read Sign**	**Sign**	**Read Sign**	**Sign**	**Read Sign**	**Sign**	**Read Sign**
Crosswalk	210	118	210	142	210	181	210	197
School area	122	60	122	82	122	88	122	102
Speet Limit	184	92	184	108	184	145	184	175
Lights	192	98	192	106	192	124	192	152
Radar	35	16	35	19	35	24	35	31
Total	743	384	743	457	743	562	743	657
Total Rate	51.75 %		64.02 %		75.74 %		88.54 %	

The computer used in the study had the following specifications: Nvidia 1060 GTX graphics card, 32 GB RAM, and an Intel Core i7 4210U processor. Image processing using TensorFlow was performed, and the training process took eight hours. Figure 8 presents the Tensorboard graph during the training phase. Model training was completed in 94,932 steps.

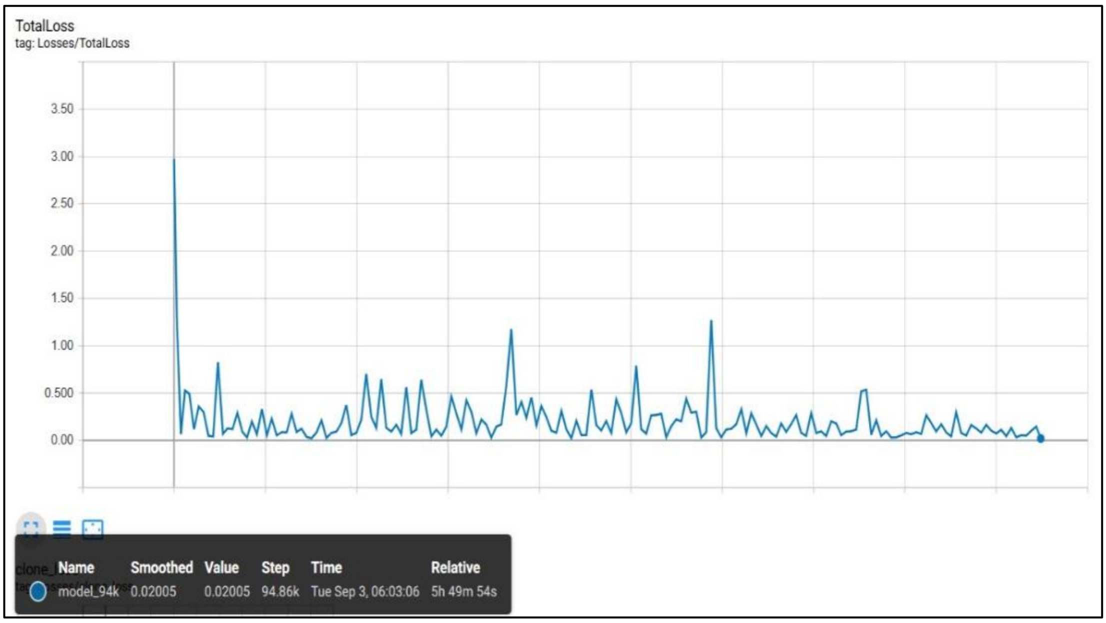

Figure 8. The loss value of the model

Certainly, recommendations can be made for researchers planning to undertake further advanced versions of this study focused on autonomous driving research. The use of the Faster R-CNN architecture in the study, which is based on region proposal, can lead to delays in real-time applications due to its computational complexity. Therefore, it may be advantageous to explore deep learning architectures like SSD (Single Shot MultiBox Detector) or YOLO (You Only Look Once) that offer faster real-time processing times in object detection.

After conducting training sessions in the study, the resulting model can be deployed on Nvidia Jetson-based boards. This facilitates the development of a compact module suitable for commercial applications without the necessity of having a laptop computer within the vehicle. In this study, an Nvidia Jetson Nano board was utilized; however, it was observed that the 4 GB RAM available on the board proved inadequate for the proper functioning of the application. Data augmentation contributes significantly to the performance improvement in many deep learning studies. Comparisons can be made by augmenting data to assess performance outcomes.

5. CONCLUSION

The success of deep learning architectures in achieving high scores in object recognition has paved the way for more ambitious and successful applications in the field of autonomous driving. However, in this complex process with many variables, more work is needed for some crucial elements. One of these is the process of recognizing traffic signs. Different countries have their own unique traffic signs, and as a result, deep learning-based models developed need to be customized for the specific country in which the application will be used.

Within traffic signs, some are particularly vital in urban traffic. For instance, the recognition of signs such as speed limits, school zones, and pedestrian crossings, and warning drivers about them, will be valuable in preventing adverse outcomes. Therefore, the primary aim of this study is to detect these traffic signs in real-time, provide audible warnings to the driver based on the vehicle's current speed condition, and alert them accordingly. We trained the Faster R-CNN architecture using a dataset we created with our own resources, primarily leveraging Python-based libraries, including the TensorFlow library. After approximately 95,000 iterations, the network's error has decreased to 0.174.

A module has been developed using Arduino and GPS to obtain the real-time speed of the vehicle. When a deep learning model detects a traffic sign, the driver is audibly warned based on the measured speed information using the module.

7. ACKNOWLEDGEMENTS

This work was supported by Research Fund of Isparta University of Applied Sciences. Project Number: 2019-YL1-0004

REFERENCES

[1] World Health Organization. (2018). Global Status Report On Road Safety 2018. https://www.who.int/publications/i/item/9789241565684

[2] Wijnen, W., & Stipdonk, H. (2016). Social costs of road crashes: An international analysis. Accident Analysis & Prevention, 94, 97-106. [DOI: 10.1016/j.aap.2016.05.006]

[3] Uğuz, S., & Büyükgökoğlan, E. (2022). A hybrid CNN-LSTM model for traffic accident frequency forecasting during the tourist season. Tehnički vjesnik, 29(6), 2083-2089. [DOI: 10.17559/TV-20220506004223]

[4] Fujiyoshi, H., Hirakawa, T., & Yamashita, T. (2019). Deep learning-based image recognition for autonomous driving. IATSS Research. [DOI: 10.1016/j.iatssr.2019.04.004]

[5] Nidhal, A., Ngah, U. K., & Ismail, W. (2014). Real-time traffic congestion detection system. In 2014 5th International Conference on Intelligent and Advanced Systems (ICIAS) (pp. 1-5). IEEE. [DOI: 10.1109/ICIAS.2014.7033872]

[6] Zhu, Y., Zhang, C., Zhou, D., Wang, X., Bai, X., & Liu, W. (2016). Traffic sign detection and recognition using fully convolutional network guided proposals. Neurocomputing, 214, 758-766. [DOI: 10.1016/j.neucom.2016.08.059]

[7] Shustanov, A., & Yakimov, P. (2017). CNN design for real-time traffic sign recognition. Procedia engineering, 201, 718-725. [DOI: 10.1016/j.proeng.2017.09.416]

[8] Changzhen, X., Cong, W., Weixin, M., & Yanmei, S. (2016). A traffic sign detection algorithm based on deep convolutional neural network. In 2016 IEEE International Conference on Signal and Image Processing (ICSIP) (pp. 676-679). IEEE. [DOI: 10.1109/SIPROCESS.2016.7919591]

[9] Kulkarni, R., Dhavalikar, S., & Bangar, S. (2018). Traffic Light Detection and Recognition for Self Driving Cars Using Deep Learning. In 2018 Fourth International Conference on Computing Communication Control and Automation (ICCUBEA) (pp. 1-4). IEEE. [DOI: 10.1109/ICCUBEA.2018.8617440]

[10] Stallkamp, J., Schlipsing, M., Salmen, J., & Igel, C. (2011). The German traffic sign recognition benchmark: a multi-class classification competition. Neural networks. [DOI: 10.1016/j.neunet.2012.02.016]

[11] Stallkamp, J., Schlipsing, M., Salmen, J., & Igel, C. (2012). Man vs. computer: Benchmarking machine learning algorithms for traffic sign recognition. Neural networks, 32, 323-332. [DOI: 10.1016/j.neunet.2012.02.016]

[12] Kim, J., & Lee, M. (2014). Robust Lane Detection Based On Convolutional Neural Network and Random Sample Consensus. In Loo C.K., Yap K.S., Wong K.W., Teoh A., Huang K. (eds) Neural Information Processing. ICONIP 2014. [DOI: 10.1007/978-3-319-12637-1_3]

[13] Qian, R., Zhang, B., Yue, Y., Wang, Z., & Coenen, F. (2015). Robust Chinese traffic sign detection and recognition with deep convolutional neural network. In 2015 11th International Conference on Natural Computation (ICNC) (pp. 791-796). IEEE. [DOI: 10.1109/ICNC.2015.7378104]

[14] Arcos-Garcia, A., Alvarez-Garcia, J. A., & Soria-Morillo, L. M. (2018). Evaluation of deep neural networks for traffic sign detection systems. Neurocomputing, 316, 332-344. [DOI: 10.1016/j.neucom.2018.07.072]

[15] Hussain, S., Abualkibash, M., & Tout, S. (2018). A survey of traffic sign recognition systems based on convolutional neural networks. In 2018 IEEE International Conference on Electro/Information Technology (EIT) (pp. 0570-0573). IEEE. [DOI: 10.1109/EIT.2018.8399772]

[16] Vasilev, I., Slater, D., Spacagna, G., Roelants, P., & Zocca, V. (2019). Python Deep Learning: Exploring deep learning techniques and neural network architectures with Pytorch, Keras, and TensorFlow. Packt Publishing Ltd. Birmingham.

[17] Mueller, J. P., & Massaron, L. (2019). Deep learning for dummies. John Wiley & Sons, Canada.

[18] LeCun, Y., Bottou, L., Bengio, Y., & Haffner, P. (1998). Gradient-based learning applied to document recognition. Proceedings of the IEEE, 86(11), 2278-2324.

[19] Krizhevsky, A., Sutskever, I., & Hinton, G. E. (2012). Imagenet classification with deep convolutional neural networks. In Advances in neural information processing systems. [DOI: 10.1016/j.neunet.2012.02.016]

[20] Simonyan, K., & Zisserman, A. (2014). Very deep convolutional networks for large-scale image recognition. arXiv preprint arXiv:1409.1556.

[21] Singh, A., Patil, D., Reddy, G. M., & Omkar, S. (2017). Disguised Face identification (DFI) with facial Key Points using Spatial Fusion Convolutional Network. In IEEE International Conference on Computer Vision Workshops