EJONS International Journal on Mathematic, Engineering and Natural ISSN 2602 - 4136

Article Arrival Date 04.03.2021 | Article Type Research Article | Article Published Date 20.06.2022

DEVELOPMENT OF REAL-TIME TRAFFIC SIGN RECOGNITION WITH CONVOLUTIONAL NEURAL NETWORK USING DEEP LEARNING TECHNIQUES

***Batur Alp AKGÜL**
Hasan Kalyoncu University, Engineering Faculty, Electrical & Electronics Engineering, TURKEY
ORCID: 0000-0002-8332-6764
Fatih ALİSİNANOĞLU
Hasan Kalyoncu University, Engineering Faculty, Electrical & Electronics Engineering, TURKEY
ORCID: 0000-0002-4755-4292
Kadir Sercan BAYRAM
Hasan Kalyoncu University, Engineering Faculty, Electrical & Electronics Engineering, TURKEY
ORCID: 0000-0002-8461-2308
Mustafa Sadettin ÖZYAZICI
Hasan Kalyoncu University, Engineering Faculty, Electrical & Electronics Engineering, TURKEY
ORCID: 0000-0003-0637-0996

ABSTRACT

Traffic signs are a mandatory feature of road traffic regulations worldwide. They are responsible for slowing down the speed and carrying out many other valuable duties, notifying drivers about dangerous parts of the route, signaling traffic destination, prohibiting, or allowing passage. In this way, traffic is smoother, it becomes better regulated and drivers understand, mark, and interpret the rules well. For this purpose, Machine Learning (ML) study is carried out with the Deep Learning (DL) approach, the Real-Time (RT) Traffic Signs Recognition (TSR) is successfully developed, and a 99,68% test accuracy is obtained which has been gradually built on autonomous vehicles. It is developed to alert the driver for traffic signs appearing on the road and it is assisted the driver reach the speed limit set in the section of the lane, ride, overtaking, etc. The developed TSR helps to boost safety dramatically on the way to autonomous driving. The system is built on the DL, and it is trained with the Convolutional Neural Network (CNN) model to a classifier and predicts the status which is very effective for image classification purposes, and it is the most common and lovable algorithm for image data processing. It is also visualized how the accuracy and loss rates have changed over time. Later, it is implemented the graphical Unit Interface (GUI) were to show the results and draw the accuracy and the loss graphs. To realize classification as RT, Computer Vision (CV) approach is also included within the developed software to support camera viewing and understanding digital images.

Keywords: Traffic Sign, Recognition, Identification, Classification, Real-Time.

1. INTRODUCTION

The TSR problem is one of the researcher's best-recognized and frequently discussed problems. Low detection accuracy and high demand for performance in hardware computing are therefore the key problems of these systems. In the world of advancement in DL technologies, many researchers and large car companies are working on self-driving autonomous vehicles. These cars to be used in the future need to understand and comply with all traffic rules. Therefore, vehicles should be able to interpret traffic signs and make decisions accordingly. Apart from all this, in traffic accidents, drivers have a role in not obeying or not paying attention to traffic signs.

For this reason, vehicles should recognize the traffic signs and inform or alert the driver to prevent such accidents. ML researchers worked a lot to develop new neural network models and algorithms. So, processing image files RT images to implement DL techniques of CNN together is become possible with huge datasets and computer power. ML has several algorithms that are responsible for bringing the processes knowledge. So, to make machines smarter, developers use ML and DL techniques. Mankind learns by repeating and repeating a task so that he memorizes how to do it. Then the neurons in your brain are triggered automatically and can quickly fulfill the task they learned, and computer-aided DL is very similar to this. DL architectures are used for different types of problems for these operations. For example, object recognition, detection, segmentation, etc., big data plays a major role in CV as huge amounts of data are required for better conclusions. ML-DL advancement has increased the use of these techniques. Therefore, it is used in this study.

There are various signs at the traffic such as no passing, type of speed limits, type of traffic signals, turn left or right, children and bicycle crossing, road work, etc. In this study, a DL model is created to classify and identify traffic signs. Autonomous vehicles and new generation vehicles have acquired the ability with this model to read, recognize, and understand traffic signs, which is a very essential task.

The TSR classifies and predicts the traffic signal displayed from the web camera in RT. It also classifies and estimates the traffic sign from the selected database. Autonomous vehicles, of course, will perform RT recognition. However, testing with the camera is limited. Therefore, offline learning set with thousands of traffic signs has been designed to provide the system with the ability to learn at a high rate and increase the success rate. Besides, the quality and resolution of the camera to be used will also affect the success rate. High-tech cameras are expected to be used for these processes in autonomous vehicles. The developed GUI of the software is given in figure 1., and figure 4 shows the traffic signs stored in the dataset.

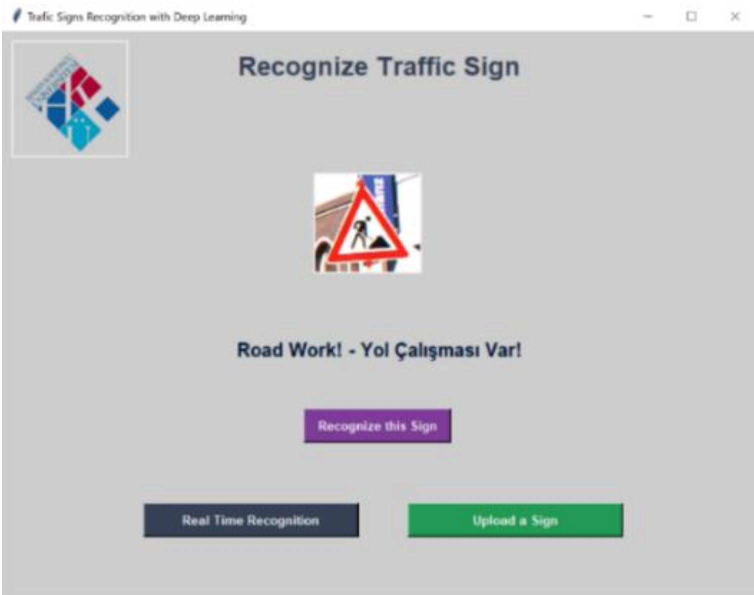

Figure 1: The developed GUI

2. RELATED WORKS

When the literature review is done, it is seen that although a wide variety of methods, techniques, and approaches are used in the field of TSR, satisfactory results have not been achieved yet. Effective applications of real-time image processing and traffic signal localization algorithms were proposed, and a generalized Hough Transforms Algorithm was used in [1-2]. Localization and subsequent classification methods were used in [3-4]. In [5] DL was created for the identification and recognition of traffic signs. Special color barcodes below the road sign to recognize the traffic sign in the vision-based approach were developed in [6]. A genetic-based algorithm is suggested in [7] and color indexing was proposed by [8] to identify the road sign.

CNN design for TSR was developed [9] and TSR for intelligent assistance framework using OpenCV neural network was developed [10]. Moreover, a study for the identification and detection of traffic signs using a fuzzy separation approach was introduced in [11], using fuzzy rules to identify traffic signs. In [12], another algorithm ED-Circles was used to classify traffic signs for RT detection. Studies [13-14] were focused on recognizing traffic signs using the type of algorithm vector machines.

All the steps of the classification process were explained in detail. Another approach is [15] to reconnaissance of traffic signs was developed with the use of CNN. A TSR approach to designing the service network was proposed in[16]. A traffic sign recognition system was developed through the application of a cross-relation technique [17], a Laplace core classifier was used in the decision mechanism for traffic sign classification by [18], and a neural network-based TSR was developed by [19]. Photos are segmented into the HSI color space in [20], and then Matching template methods were studied to identify signs of traffic. In [21], multi-layer CNN was proposed to increase traffic signal recognition. Traffic sign recognition based on CNN was presented in [22] that it was not RT detection. A system for the identification of traffic signs using GPU with CNN was introduced by [23].

3. MATERIAL AND METHODS

3.1. Machine Learning with Deep Learning Approach

ML is based on problem-solving in the real world. It takes just a few artificial intelligence ideas, too. Moreover, ML does through neural networks. That is designed to mimic the ability to make human decisions. That focuses more on DL. It needs to be applied to find a solution for any problem and that calls for human or artificial thought. DL which has recently been a very important approach in ML has been developed with inspiration from the work of the human brain and biological neural networks. It uses types of neural networks, and it consists of multiple layers to pass data before finally producing the output. DL serves to develop artificial intelligence and enable most of its applications. It is applied in many fields such as CV. The basic principle of ML tasks with DL is given in figure 2.

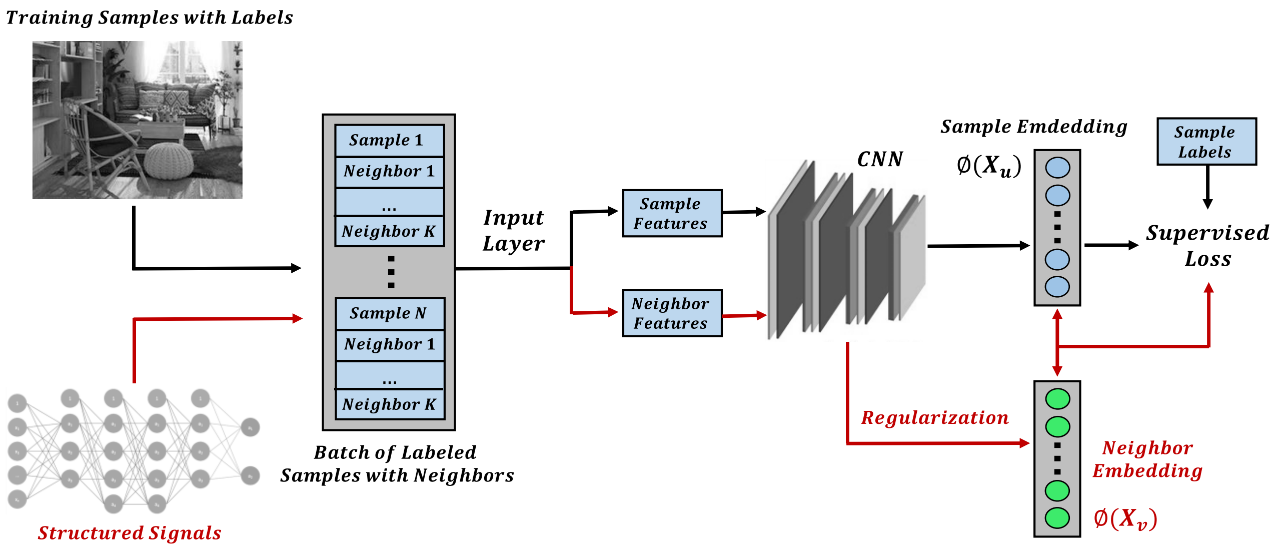

Figure 2: The basic principle of ML tasks with DL

3.2. Using Deep Learning with TensorFlow and Keras

Keras is a Python-based framework that is an open-source neural network. It is a top-level API and can work on other DL frameworks such as TensorFlow. Keras can run smoothly in the Central Processing Unit (CPU) and Graphical Processing Unit (GPU) and provide fast experiments and prototyping. Neural network training is carried out using TensorFlow with multi-threaded programming (CUDA) which is parallel architecture. On the GPU, the entire TSR is executed in real-time. The background of the DL with TensorFlow and Keras is shown in figure 3.

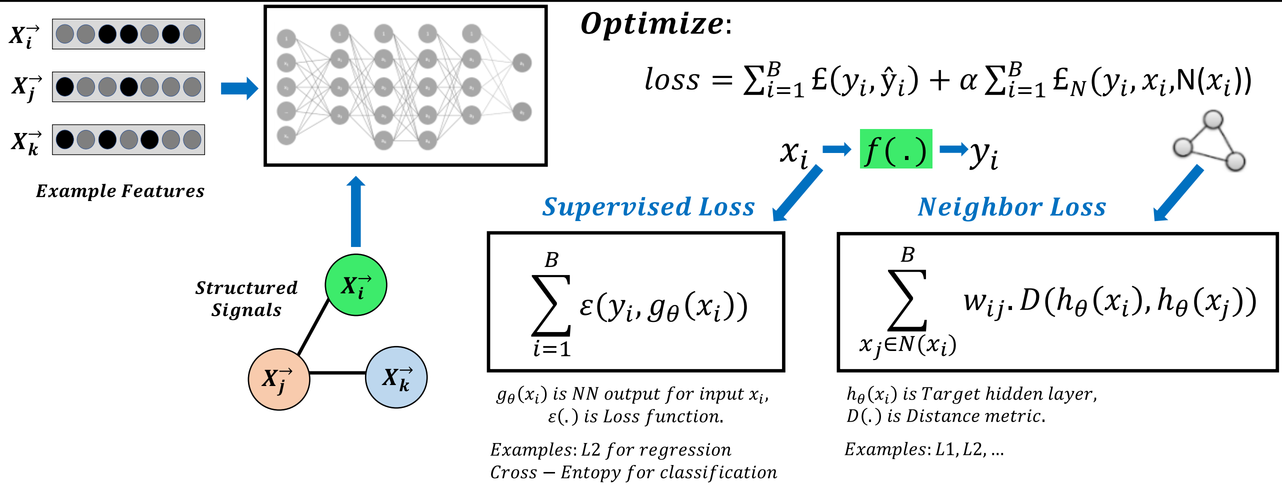

Figure 3: Background of the DL with TensorFlow and Keras

3.3. Computer Vision Architecture with OpenCV

CV is a workspace that allows us to view and define digital sources and it is a multidisciplinary scientific field. The difficulties faced are largely due to biological intuition impaired comprehension. The process usually includes applications such as digital source recognition. OpenCV is a framework that includes numerous highly optimized algorithms that are used in CV tasks. This library also has analytical capabilities when processing RT images, video and library has more than 2800 algorithms. It supports DL frameworks with multiple platforms. It completely scans images for extracting important features out of the image and combines image recognition features.

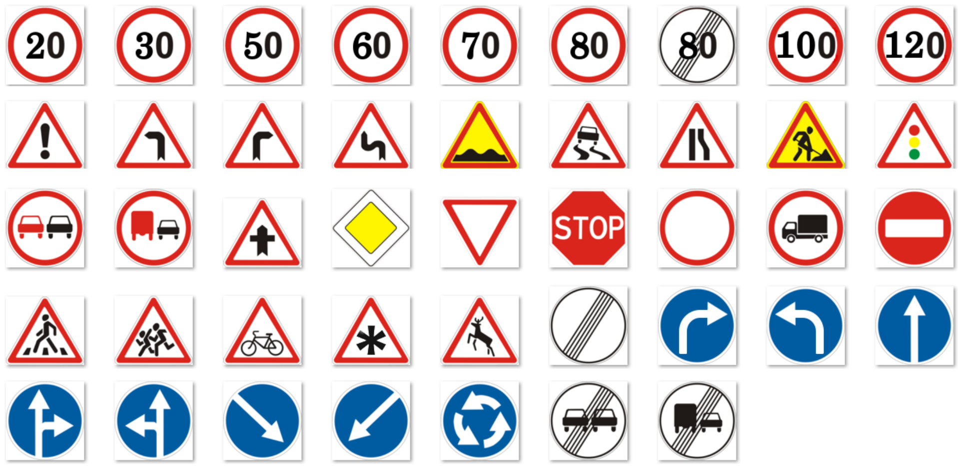

Figure 4: Traffic signs stored in the public dataset.

3.4. Modeling the Architecture with CNN

CV is a workspace that allows us to view and define CNN is a kind of DL algorithm that receives the image as input data and uses filters to learn the different features of the image. This algorithm allows the significant objects presented in the image to be learned. One major feature of CNN which distinguishes it from other ML algorithms is its capability to pre-process the data alone. CNN can adapt to the features learned, improve its filters, it has several hidden layers, and many are convolutionary. In a way, CNN has a multilayer perceptron that is regularized. Photos are conveniently depicted as a 2D matrix and CNN is much helpful in image processing. Using a filter that performs a 2D matrix multiplication on the layer and the filter, a curl is performed on these layers. A simple diagram for CNN to predict traffic signs is given in figure 5. The most essential component of CNN's architecture is a convolutionary network layer and the important part of the convolutional layer contains a learnable filter. It has weights that can learn from the inputs and biases. The network is learned the different filters that come along in the first layer of the visual features and generates a pattern in the network's higher layer.

Each of the CNN filters presents a 2D activation map that is stacked to produce the volume of output. Each connected neuron in the CNN receives input data, executes a dot product and this dot product is presented in a 2D activation map that provides us with a filter response at every spatial position. This is achieved in a non-linear manner, eventually, a single differentiable score function is produced. This function is made up of scores that are provided from the different layers of CNN. Lastly, a loss function for evaluating model performance is generated at the end of the process.

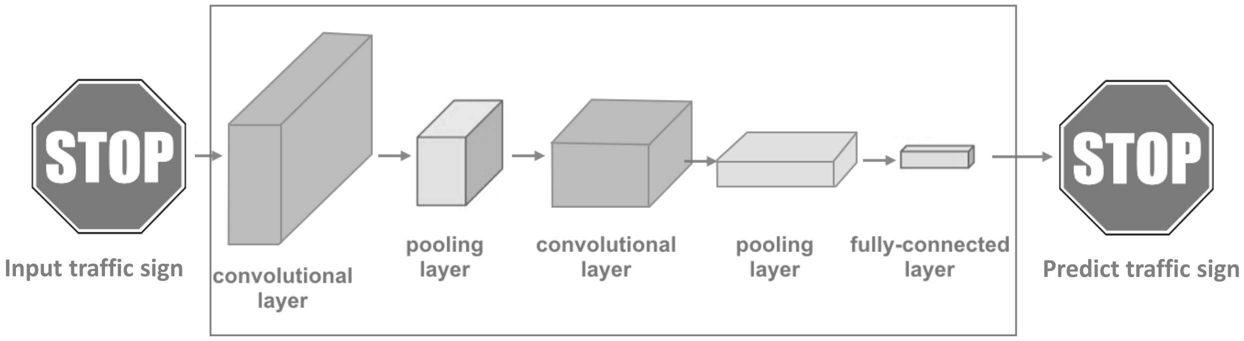

Figure 5: Simple diagram of CNN for the predicted traffic sign.

4. DEVELOPMENT STEPS OF THE TSR

4.1. Dataset of the TSR

The development of datasets continues on public platforms, in this study public datasets are used. The dataset consists of over 22,000 images of various traffic signs and it is also divided into 43 different traffic sign classes from numbers 0 to 42 and each class has different traffic signs. The dataset is reasonably variable, there are many images in some classes, while there are very few images in some classes. The dataset consists of a training folder and a test folder including the images in each class used to test the developed model. Low resolution and blurred traffic signs taken from different angles and different distances are also included in the dataset. A large dataset is used as much as possible to increase the accuracy rate and minimize the margin of error. The traffic sign distribution of the training dataset is given in figure 6.

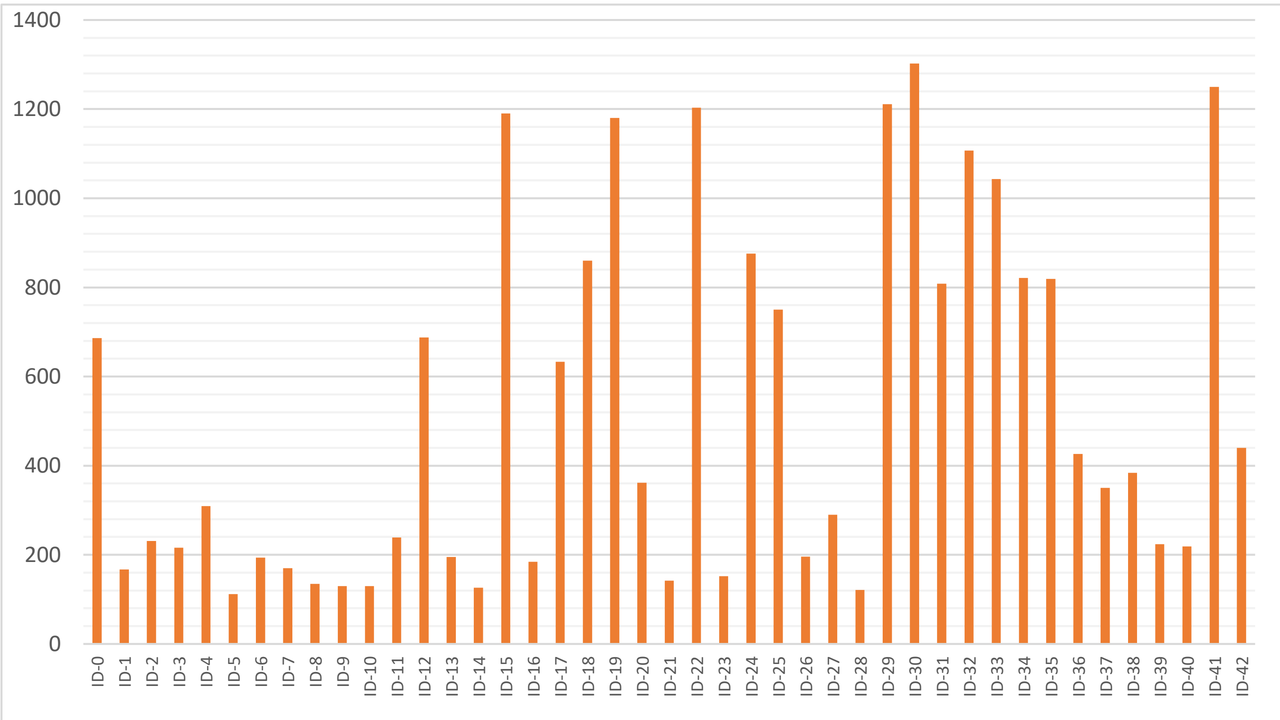

Figure 6: Distribution of the training dataset

4.2. Steps to Build Software Architecture

The method of creating this model for the classification of traffic signs is completed in four steps. The first step is to explore the dataset. The developed software automatically detects how many classes there are and imports each class one by one in one matrix. With the aid of the operating system module, all classes are repeated, displayed in the list of data and tags, and related tags are added. Finally, all images and tags are stored in lists and the shape of the data is created. The second step is to build a CNN model. It is for classifying the images into their respective categories. It has been developed a CNN model for image classification purposes. Moreover, the model is compiled with Adam optimizer because there are multiple classes to categorize, and it is well-performing.

The third step is to train and validate the developed model. After creating the model architecture, it should be trained. The model is trained by the file that has all the information for the classification. Once trained in the model, a model file is included testing code and run as a model. Besides, its graphics are drawn for accuracy and loss directly. The fourth step is to test the model with the test dataset. The data set contains a test folder and contains details about the display path and related class labels in a test file. The image path and tags are extracted using pandas. Then, to estimate the model, images are resized, and a numerical array is made containing all image data. The accuracy score is observed in how the model predicts real tags. 99,68% of test accuracy is achieved in this model. In the end, the trained model is saved.

The fifth step is to create the traffic signs classifier GUI. A GUI has been created for the software developed to classify and predict traffic signs. To do this, the first trained model is loaded, then the GUI is created to load the image and a button is used to classify the traffic sign. The resulting image is then converted into shape size. The same dimension used when creating the model must be provided to estimate the traffic sign. After this, the class is estimated and returned a number representing the class to which the traffic sign belongs. A dictionary is used to get information about the class.

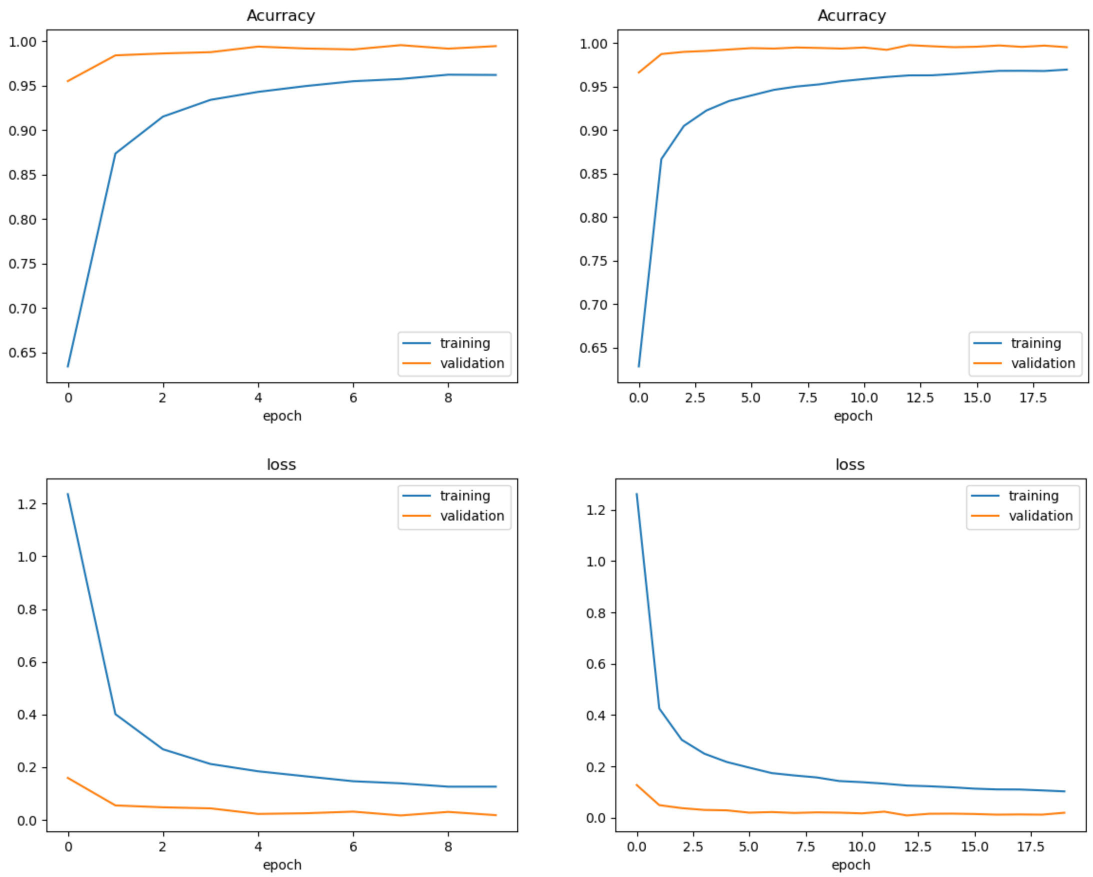

Figure 7: Accuracy and loss charts of the model

4.3. Steps for Real-Time Detection

In this study, the OpenCV framework is used for gathering the images from the webcam and feeding them into the developed DL CNN model to classify and identify traffic signs as RT. The approach is completed in three steps. The first step is to take the traffic sign image as input from a camera. Traffic signs on the images are processed as inputs with a webcam. To access the webcam, an endless loop has been made to capture every frame in the image. The methods provided by OpenCV are used to access the camera and set the capture objects. The webcam reads every frame in the image, and it is stored in a frame variable file. The second step is to detect and feed the traffic sign on the image to the classifier. The Haar Cascade which is the one of best kinds of classifier methods is used to find the traffic sign in the image. Returns a series of detections with x, y coordinates, and height, which are the width of the object's bounding box, and border boxes are drawn for the image. Thus, this is achieved by extracting the boundary box of the image, and then it is pulled out the traffic sign image from the frame with this code. The third step is to calculate the score to check accuracy for a detected sign. The score is a value to determine the accuracy percentage of the traffic sign estimation. Finally, results are plotted for the RT recognition process of the traffic sign.

5. RESULTS AND DISCUSSIONS

5.1. Evaluations of Dataset

The distribution of the training dataset was given in figure 6. When it is analyzed, distribution is not even. So, it is important to find how much dataset is required for good classification. It is seen that dataset class ID is 30 with minimum image and minimum class ID is 5. While there are 112 images in class 5, there are 1302 images in class 30. In the RT tests with a web camera, the accuracy rate of class 30 is 100%, while the accuracy of class 5 is under 95%. The accuracy rate of all classes with over 500 images results in over 97%. Since the success rate target in this study is a minimum of 97%, all classes in the dataset should have 500 images. Therefore, the threshold value of dataset classes has been determined as 500. The dataset should be developed with a minimum number of 500 images of all classes with less than 500 images in the dataset.

Table 1: Accuracy table of the trained model.

Epochs	ETA /SC	Loss	Accuracy	Val Loss	Val Accuracy
Epoch 1/30	612	1.2131	0.6403	0.1281	0.9639
Epoch 5/30	640	0.1963	0.9395	0.0259	0.9931
Epoch 10/30	606	0.1311	0.9598	0.0262	0.9921
Epoch 15/30	628	1.1126	0.9664	0.0171	0.9955
Epoch 20/30	611	0.1044	0.9695	0.0211	0.9937
Epoch 25/30	587	0.0991	0.9715	0.0178	0.9944
Epoch 30/30	587	0.0946	0.9728	0.0209	0.9946
Test Score			**0.0137**		
Test Accuracy			**0.9968**		

5.2. Evaluations of the Epochs and Accuracy Rates

Epochs value indicates how many iterations go through to reach the best accuracy. It is important to find how much accuracy and how much loss is received for each iteration. It is seen from the charts in figure 7, produced by the developed software for epochs 10 and epochs 20, between 10 and 15 epochs value is good estimation value to train network model.

It is computed with 30 epochs and after about 15 epochs it is going to the same level. Therefore, the epochs threshold value is set to 15 for this study. Accuracy and loss charts for different epochs are given in figure 7 to compare and find the good threshold value for the estimation. Moreover, it is seen that from table 1, as the number of epochs rises, the accuracy rate rises, and the loss rate reduces. Also, figure 8 shows the charts of the accuracy and loss values of all 30 epochs. This chart supports the decision to set the epoch threshold value to 15. It is seen that from the results in Table 1, the developed model is trained very well.

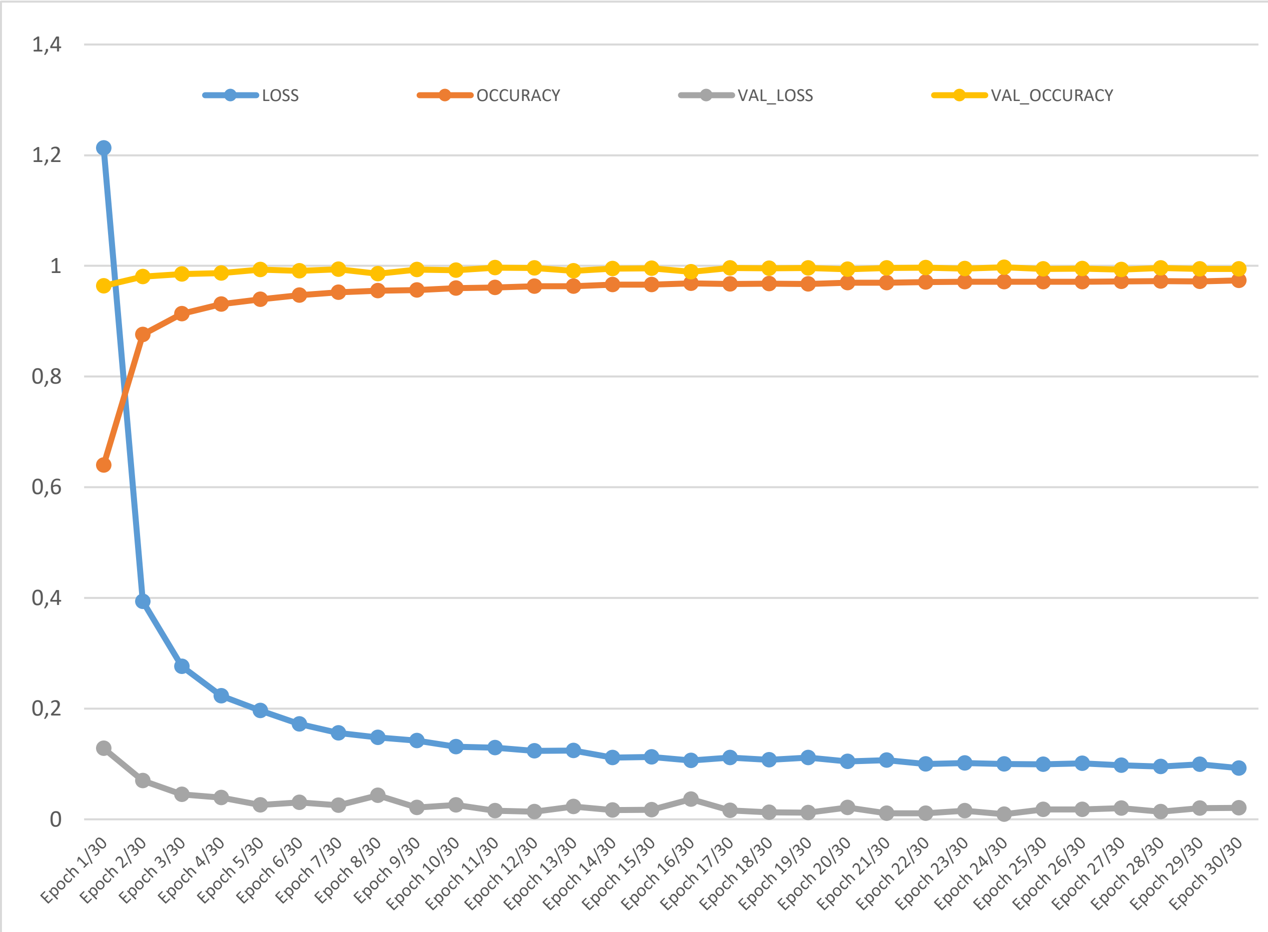

Figure 8: The accuracy and loss charts for all 30 epochs

373

5.3. Evaluations of TSR tests with the Local Database

In such profound (DL) projects, testing in the laboratory environment is the first step towards success. The success rate for the classification of traffic sign pictures stored on the computer is 100%. Although there are no time and distance parameters in this kind of identification system, the recognition of the traffic signs of the trained model is very positive. Figure 9 shows the local TSR tests with some traffic signs. It is seen that from figure 9, the fact that the signs are static and immobile increases the success rate.

5.4. Evaluations of TSR in the Real Environment

In such In TSR tests performed with some real traffic signs on the road, a minimum 99,45% probability rate has been reached and the test results are given in figure 10. For testing, an A4 Tech brand, PK-910H model standard webcam was used. The technical features of the camera are very important, capturing high-resolution image quality is possible with the use of such cameras. Obtaining a clean and clear image will increase the working speed of the algorithms used in the developed software as well as increase the success rate. Considering that a high-tech camera capable of automatically focusing on objects with a zoom option will be integrated into vehicles, probability rates seem acceptable.

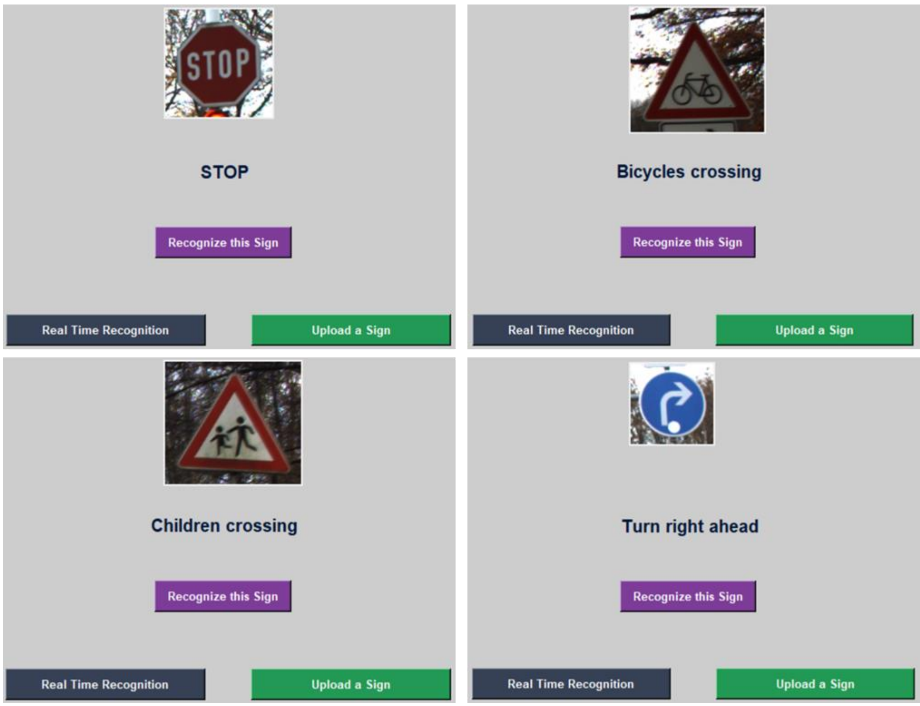

Figure 9: TSR tests are done with the local database.

Class (25) Road Work, Probability (100.0%)

Class (9) No Passing, Probability (99.45%)

Class (13) Yield, Probability (100.0%)

Class (1) Speed Limit, Probability (100.0%)

Class (36) Go Straight or Right, Probability (99.98%)

Class (2) Speed Limit, Probability (99.88%)

Figure 10: TSR tests are done in the real environment.

5.5. Evaluations of TSR Results with Future Work

The success and performance of the software developed in embedded systems are extremely important. Thus, in future studies, a device will be designed by integrating the developed TSR into embedded systems. The performance of TSR in embedded systems will be examined. Dataset optimization is another important task because of course, the accuracy percent can be increased with the large-scale dataset but extending the dataset can increase recognition time. For a vehicle going on the highway, 1 second can be quite a long time in terms of identifying the traffic sign. Another important point is how many meters away the vehicles can be able to detect the traffic sign. This distance is also very important. Therefore, in the next study, the relationships between dataset optimization, recognition time of traffic sign, and sign recognition distance will be discussed and evaluated deeply.

6. CONCLUSIONS

This article addresses implementing the classification and identification with DL using CNN for TSR tasks. The developed system for traffic sign recognition has yielded very good results with an accuracy rate of 99,68% in the real environment. The developed system carries out a classification for different traffic signs in less than half a second. Therefore, the model presented in this study may be one of the best in terms of precision or optimization. To find the best network architecture, the network system is trained with the training data set and validated with the dataset. Experiments show consistent findings with the correct classification of traffic sign patterns for the complex background images. The model is created with extremely low application time using the TensorFlow and Keras library with the CV approach and the Python language.

REFERENCES

[1] Fursov, V., Bibkov, S., Yakimov, P.: "Localization of objects contours with different scales in images using Hough transform", Computer Optics, pp. 500-510, (2013).

[2] Yakimov, P.: "Tracking traffic signs in video sequences based on a vehicle velocity", Computer Optics, pp. 790-810, (2015).

[3] Ruta, A., Porikli, F., Li, Y., Watanabe, S., Kage H., Sumi K.: "A New Approach for In-Vehicle Camera Traffic Sign Detection and Recognition", IAPR Conference on Machine Vision Applications (MVA), Machine Vision for Transportation, pp. 1-7, (2009).

[4] Nikonorov A., Yakimov P., Petrov M.: "Traffic sign detection on GPU using color shape regular expressions", IMTA-4, pp. 1-8, (2013).

[5] Tabernik, D., Skocaj, D.: "Deep Learning for Large-Scale Traffic-Sign Detection and Recognition", pp. 1-14, (2019).

[6] Pacheco, L., Batlle, J., Cufi, X.: "A new approach to real-time traffic sign recognition based on color information", Proceedings of the Intelligent Vehicles Symposium, pp. 335–350, (1994).

[7] Aoyagi, Y., Asakura, T.: "A study on traffic sign recognition in scene image using genetic algorithms and neural networks", Proceedings of the IEEE IECON Int. Conf. on Industrial Electronics, Control, and Instrumentation, pp. 1835–1845, (1996).

[8] Lalonde, M., Li, Y.: "Detection of Road Signs Using Color Indexing", Technical Report CRIM-IT-95/12-49, Centre de Recherche Informatique de Montreal, (1995).

[9] Shustanova, A., Yakimova, P.: "CNN Design for Real-Time Traffic Sign Recognition", International Conference "Information Technology and Nanotechnology", ITNT-2017, pp. 718–725, (2017).

[10] Lorsakul, A., Suthakorn, J.: "Traffic Sign Recognition for Intelligent Vehicle/Driver Assistance System Using Neural Network on OpenCV", The International Conference on Ubiquitous Robots and Ambient Intelligence (URAI), pp. 1-6, (2007).

[11] Abedin, Z., Dhar, P., Hossein, M., K.: "Traffic sign detection and recognition using fuzzy segmentation approach and artificial neural network classifier respectively", International Conference on Electrical, Computer, and Communication Eng. (ECCE), pp. 520–530, (2017).

[12] Berkaya S., K., Gunduz, H., Ozsen, O., Akinlar, C., Gunal, S.: "On circular traffic sign detection and recognition", Expert Systems with Applications, pp. 67–75, (2016).

[13] Shi, M., Wu, H., Fleyeh H.: "Support vector machines for traffic signs recognition", IEEE Int. Joint Conference on Neural Networks, pp. 3810–3830, (2008).

[14] Shustanov, A., Yakimov P.: "CNN design for real-time traffic sign recognition", Procedia Engineering, pp. 715–730, (2017).

[15] Sermanet, P., LeCun, Y.: "Traffic sign recognition with multi-scale Convolutional Networks", International Joint Conference on Neural Networks, pp. 2805–2815, (2011).

[16] Kryvinska, N., Poniszewska-Marandac, A., Gregus, M.: "An Approach towards Service System Building for Road Traffic Signs Detection Recognition", Procedia Computer Science, pp. 64–71, (2018).

[17] Piccioli, G., Micheli, E., Parodi, P.: Campani, M.: "Robust Method for Road Sign Detection and Recognition", Image and Vision Computing, pp.205-225, (1996).

[18] Novovicova, J., Paclik, P., Pudil, P., Somol, P.: "Road Sign Classification Using Laplace

Kernel Classifier", Pattern Recognition Letters 21, pp. 1164-1175, (2000).

[19] Saha, S., K., Chakraborty, D., Bhuiyan, A.: “Neural Network-based Road Sign Recognition” International Journal of Computer Applications, pp. 35-41, (2012).

[20] Paclk, P., Novovicov, J., Pudil, P., Somol, P.: “Road sign classification using Laplace kernel classifier”, Pattern Recognition Letters, pp. 1164–1175, (2000).

[21] Sermanet, P., LeCun, Y.: “Traffic sign recognition with multi-scale convolutional networks”, in Neural Networks (IJCNN), International Joint Conference, pp. 2809 – 2813, (2011).

[22] Wu, Y., Liu, Y., Li, J., Liu, H., Hu X.: “Traffic Sign Detection based on Convolutional Neural Networks”, Proceedings of Int. Joint Conf. on Neural Networks, pp. 747-753, (2013).

[23] Shustanov, A., Yakimov, P.: “A Method for Traffic Sign Recognition with CNN using GPU”, International Joint Conference on e-Business and Telecommunications (ICETE), pp. 41-48, (2017).