Hindawi
Journal of Sensors
Volume 2021, Article ID 8870529, 13 pages
https://doi.org/10.1155/2021/8870529

Research Article

A Lightweight Model for Traffic Sign Classification Based on Enhanced LeNet-5 Network

Ameur Zaibi,[1,2] **Anis Ladgham**,[1,3] **and Anis Sakly**[1,2]

[1]*Laboratory of Automation, Electrical Systems and Environment (LAESE), Faculty of Sciences of Monastir, University of Monastir, Tunisia*
[2]*LAESE, National Engineering School of Monastir, University of Monastir, Tunisia*
[3]*Laboratory Electronic and Microelectronic, Faculty of Sciences of Monastir, University of Monastir, Tunisia*

Correspondence should be addressed to Ameur Zaibi; ameurzaibi@yahoo.fr

Received 29 August 2020; Revised 1 April 2021; Accepted 10 April 2021; Published 30 April 2021

Academic Editor: Stelios M. Potirakis

For several years, much research has focused on the importance of traffic sign recognition systems, which have played a very important role in road safety. Researchers have exploited the techniques of machine learning, deep learning, and image processing to carry out their research successfully. The new and recent research on road sign classification and recognition systems is the result of the use of deep learning-based architectures such as the convolutional neural network (CNN) architectures. In this research work, the goal was to achieve a CNN model that is lightweight and easily implemented for an embedded application and with excellent classification accuracy. We choose to work with an improved network LeNet-5 model for the classification of road signs. We trained our model network on the German Traffic Sign Recognition Benchmark (GTSRB) database and also on the Belgian Traffic Sign Data Set (BTSD), and it gave good results compared to other models tested by us and others tested by different researchers. The accuracy was 99.84% on GTSRB and 98.37% on BTSD. The lightness and the reduced number of parameters of our model (0.38 million) based on the enhanced LeNet-5 network pushed us to test our model for an embedded application using a webcam. The results we found are efficient, which emphasize the effectiveness of our method.

1. Introduction

Road safety is attracting the attention of many researchers around the world since it is indispensable in protecting human life. Driver assistance systems have played a very important role. For several years now, systems for the detection, classification, and recognition of road signs have become a very important research topic for researchers. From one research project to another, the authors have tried to improve the accuracy and recognition rate of these systems. To achieve these improvements, some researchers have turned to deep learning models.

Amongst these models that have been successful in the field of object detection [1] and image classification [2] are CNN [3]. CNN's methods are similar to those of traditional supervised learning methods: they receive input images, detect the features of each of them, and then train a classifier on them. However, the features are learned automatically. The CNN do all the tedious work of feature extraction and description themselves: during the training phase, the classification error is minimized in order to optimize the classifier parameters and the features. In addition, the specific architecture of the network makes it possible to extract features of different complexities, from the simplest to the most sophisticated [4].

Object detection and image classification are part of machine vision and machine learning problems [5, 6]. Road sign classification is a difficult task in machine vision since it requires a lot of computational effort and a correct, consistent, and accurate classification algorithm. CNN can solve such problems, thanks to the availability of their precise and simple architecture [7]. Since 2010, these architectures were numerous thanks to the flagship computer vision competition ILSVRC (ImageNet Large Scale Visual Recognition

Challenge) which aimed at correctly locating and classifying objects and scenes in images [8]. For example, during this annual competition in 2012, the AlexNet architecture was created, which is a convolutional neural network. Since the creation of this architecture, it was invented in several image detection and classification tasks: face detection [9], facial emotion detection [10], garbage detection [11], counterfeit image detection, and localization [12].

AlexNet is inspired by LeNet convolutional neural networks. LeNet architectures were produced in 1998 by LeCun et al. [13] for the recognition of digital characters in a document. LeNet is characterized by the simplicity of its architecture, which is small in terms of memory capacity (light) and therefore low in computational complexity, making it excellent for use [14]. LeNet-5 is a very famous architecture in the field of object detection and image classification [15–17]. The traditional LeNet-5 consists of 7 layers including 3 convolutional layers, 3 subsampling layers, and a fully connected layer followed by an output layer. Due to the lightness of this network, the training time is reduced, as well as the number of parameters, which makes the classification task easier for a machine.

A reduced number of parameters and a lightweight model are necessary characteristics for a successful CNN model implementation in an embedded application. A very important keyword in our work that can make it special is a lightweight network. New researches in the fields of object detection and image classification are interested in this parameter for a more flexible and simpler implementation. For example, in this article [18], the authors proposed a lightweight model for license plate detection in complex scenes. In another work [19], the authors of this article improved a lightweight CNN model to solve the problems of facial expression recognition. On the other hand, the detection and classification of players are proposed in this paper [20], and the authors used a lightweight CNN model with a very small number of training parameters to achieve their goal.

As we can see from the order of introduction, our work has several strong points. First of all, we talked about CNN, so we will work with a model based on CNN. In addition, we provide some very famous examples of CNN architectures. We chose to work with an enhanced and modified model of the LeNet-5 (EnLeNet-5) network, and we will mention the improvements and modifications in the following sections of our paper. We insisted on the keyword lightness since after testing our proposed network on 43 classes of road signs in still images, we are going to implement it in an embedded application using the webcam, so this time, we are talking about a detection of the traffic sign in a video, which improves our work and makes it special. Another strong point of our work, by analyzing the modifications on the LeNet-5 architecture in several fields by many researchers around the world, we could notice that our modification and improvement gave excellent and even perfect results depending on the accuracy, the lightness of the model, and the much reduced number of parameters.

The remainder of this paper is organized as follows. Section 2 presents the related work. The proposed method is given in detail in Section 3. Section 4 presents the experimental results. Finally, some conclusions are made in Section 5.

2. Related Work

Recent research on road sign recognition systems uses deep learning models based on CNN. In order to recognize many classes of road signs, as, for example, in the GTSRB database which contains 43 classes, it is necessary to extract as many important features as possible from a road sign. CNN has the advantage of hidden feature extraction processing, allowing parallel processing through a parallel structure and real-time operation [21]. For example, in this article [22], the traffic sign recognition system is composed of two parts, a detection block based on color information to filter out the most nonimage-related parts. Then, they used BCT (Bilateral Chinese Transform) and VBT (Vertex and Bisector Transform) for the detailed areas to filter and position the region of interest (ROI). Another block of road sign classification and type recognition is through the CNN. It is used to judge whether the candidate region is a traffic sign and what type of sign it is. In another work [23], the use of CNN eliminated the manual work of feature extraction and provided resistance to spatial variations, the system was tested on GTSRB, and the accuracy rate using CNN reached 97.6%. In this paper [24], the authors started by reprocessing a traffic sign image to improve quality and contrast using the histogram equalization method, which shows the importance of the image processing part of an image. Then, the images are recognized by a CNN, and the large-scale structure of the information contained in traffic sign images is obtained using a hierarchical meaning detection method based on graphic models. On the other hand, two CNN architectures are presented in this work [25]; the first one contains 8 layers, and after an enhancement, the second one contains 6 layers; it is a lightweight architecture. Both models are tested on a database of road signs in Saudi Arabia. The performances of the proposed architectures were remarkable. From one model to another CNN architecture change, this work [26] proposes a model called "Improved VGG" (IVGG) inspired from the VGG model. The IVGG model includes 9 layers, compared to the original VGG model which contains 16 layers; it is added a max pooling operation and a dropout operation after several convolutional layers, to capture the main features and save training time. According to this work, we can notice the importance of a light model in terms of the number of trained parameters, training time, and speed of convergence. Also in this paper [27], the model is considerably improved on the basis of the classical LeNet-5 convolutional neural network model using the Gabor layers and selecting the Adam method as the optimization algorithm. Finally, for the classification and recognition of road signs, the experiments are conducted on the German database with an excellent accuracy of 99.75% and an average processing time per frame of 5.4 ms. According to the results of this last article, we can notice the lightness of a model based on LeNet-5 which is a strong point of our work.

Another real-time embedded traffic sign recognition [28] uses an efficient convolutional neural network for

classification and a multiscale, deep-separable convolution operation for detection. This model has only 0.9 million parameters while achieving 98.6% of accuracy on GTSRB. What is more, in the latter paper, the authors tested their method using both traditional VGG 16 and LeNet architectures. Based on the results (test time and test accuracy), they found that LeNet was faster and more accurate than VGG 16 in this case. It was more appropriate to choose a network similar to LeNet. Still, our vision is based on models based on the LeNet network; in this document [29], the editors of this paper were impressed by the accuracy of a Korean sign recognition system that used LeNet-5 to classify 6 types of Korean road signs achieving 100% accuracy by correctly recognizing 16 signs while driving on the Korea Advanced Institute of Science and Technology (KAIST) campus road; the authors of this work decided to test the model proposed by the Korean researchers on the GTSRB database; they obtained an accuracy of 89%, and the number of trained parameters was 0.13 million. For the European database (comprised of a set of road signs from 6 different European countries: Belgium, Croatia, France, Germany, the Netherlands, and Sweden), they obtained an accuracy of 89.8% and a number of parameters equal to 0.35 million with a very reduced processing time of a single image for both databases which is equal to 0.0067 ms. The processing time for each model depends on the number of parameters and the frame used. All this new research shows the importance and efficiency of models based on CNN architectures in the implementation of a road sign recognition system. In the next section, we will explain our method described in Figure 1 while showing the strong points of our work. We have created an improved architecture inspired by the LeNet-5 network. We achieved a remarkable test accuracy of 99.84% compared to other works. Moreover, our model is light in terms of the number of parameters and processing time. On the other hand, we managed to implement our model in an embedded application using the webcam and the results were excellent. The improvements of the model are indicated in the following sections.

3. Proposed Method

Recent research on traffic sign recognition systems has shown great interest in image quality and contrast. The use of image processing techniques improves the task of classification and the accuracy of the system. For example, in this work [30] that uses the Yolov3 technique, when preprocessing incoming road images, color contrasts are enhanced and edges are sharpened for easier detection of small traffic signs. This new technique, developed by the authors of this latest paper, improves the edge characteristics of these small traffic signs, making them easier to detect. What is more, their method improves the contrast and sharpness of the edges, allowing these low-quality traffic signs to become sensitive for high probability detection. The mean average precision (mAP) on the Korean Traffic Sign Data Set (KTSD) without the pretreatments was 88.24% and with the pretreatments 98.15% which shows the importance of this task.

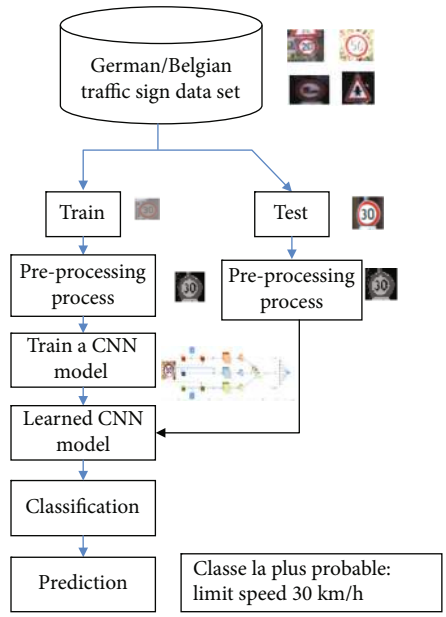

Figure 1: Our proposed method.

Figure 2: Preprocessing process.

3.1. Image Processing and Enhancement. In this paper, we seek to improve the image quality before being trained and tested by our model to achieve excellent accuracy in classifying road signs. As we already said, we are looking for the lightness of the model for the embedded implementation using the webcam, so we convert the RGB space to grayscale.

The reasons for this conversion are first, it has been shown that the color-coded information is sensitive to noise, lighting conditions, and quality of the capture equipment [31]. In addition, the number of trained parameters and training time will be reduced compared to the case of using an RGB image.

We will explain how the number of channels in an image will affect the number of parameters. In a convolution layer, the number of parameters is equal to the sum of the number of weights of this layer Wc and the number of bias Bc. $Wc = (\text{size (width) of kernels used in the convolution layer})^2 * (\text{number of channels of the input image}) * (\text{number of kernels})$ and Bc = number of kernels. In the case of an RGB image, the number of channels of the input image is equal to 3 but in the case of a grayscale image is equal to 1.

Histogram equalization is used afterwards to improve the image contrast [32]; for example, in this new research

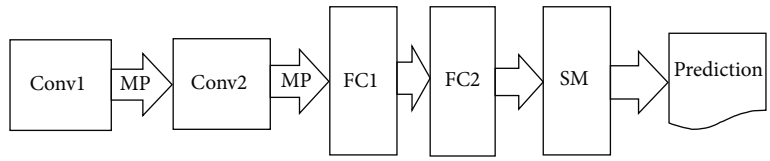

Figure 3: Classic LeNet-5 model network.

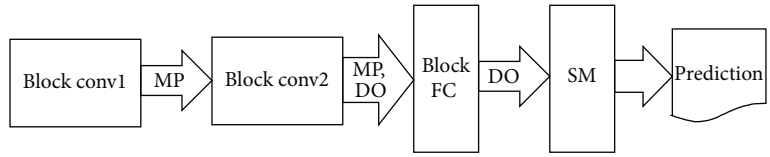

Figure 4: Enhanced LeNet-5 model network.

[33], the authors use contrast limited adaptive histogram equalization (CLAHE) to equalize the brightness distribution of the image, and the uniform pixel distribution of the gray image is modified to make the image details clearer so that the image contrast can be improved. Adjusting the histogram of the training images effectively helps to make the training process more dynamic. The collected traffic sign images are susceptible to the shooting angle, shooting distance, and other factors, which will lead to different sizes and seriously affect the feature extraction and classification tasks [34]. The size of the traffic sign image is adjusted to 32×32 to speed up the training process compared to a large image, and in order to ensure equal representation of all features, data normalization was used, which is a simple process applied to obtain the same data scale of all examples. Summarizing our preprocessing process, we will find data loading, data resizing, grayscale conversion, application of equalization histograms, and finally data normalization. This process is applied in our work on training and test images. Data augmentation is applied on training images. Figure 2 shows the sequence of the preprocessing process.

3.2. Enhanced LeNet-5 Architecture. Once the data augmentation technique is used to increase the number of training traffic sign images, then after enhancement using the preprocessing process, we will train the image through an improved model network. As already mentioned, our architecture is inspired by the famous LeNet-5. The traditional LeNet-5 architecture is composed of 7 layers taking into account the softmax output layer. The distribution of the layers in this traditional architecture is as presented in Figure 3 and as follows: a first convolution layer (Conv1), a first max pooling layer (MP), a second convolution layer (Conv2), a second max pooling layer (MP), and 2 fully connected layers (FC1 and FC2) and the softmax output layer (SM).

In our work, we are going to modify this distribution by putting, each time, two successive convolution layers: the first features extracted during convolution are very important and are low-level features, so by putting two successive convolution layers, it is possible to extract high-level features (not only edges and corners) from the input images. On the other hand, the first convolution layer of a CNN is essentially a standard image filter (+ a Rectified Linear Unit, ReLU). Its purpose is to take a raw image and extract the basic features (e.g., edges and corners) from it. These features are called low-level features. For example, in this paper [35], the authors made a modification on the first layer of the fully connected network by adding the results of the first convolution operation since the first convolution characteristics can contain elements as important as those injected in the fully connected network. According to our method, we choose to have two successive convolutional layers before the pooling layers, in order to be able to build better data representations without quickly losing all your spatial information. This is the pooling convolution method; it was used in AlexNet, VGG, Inception, and ResNet.

In fact, compared to other layers, a fully connected (FC) layer has the largest number of parameters because each neuron is connected to all other neurons. Moreover, fully connected layers are incredibly expensive in terms of calculation. In some cases (AlexNet and LeNet), more than half of the total calculation costs, in terms of the number of parameters, comes only from these fully connected layers. Compared to the traditional LeNet-5 model, our model contains only one fully connected layer, which is fixed between the last convolution and the output layer. This choice allowed the reduction of the number of parameters, which affects the lightness of the model and reduces the complexity of the calculation.

The formation of convolutional neural networks is complicated by the fact that the distribution of inputs in each layer changes during training as the parameters of the previous layers change. It slows down training by requiring lower learning rates and careful parameter initialization and makes it notoriously difficult to train models with saturating nonlinearities [26]. The batch normalization (BN) layers can solve this problem; by normalizing the input of each layer, it ensures that the distribution of input data in each layer is stable, thus achieving the goal of accelerated training. So there, in our model, we added BN after each convolution layer and also after the fully connected layer. Moreover, to reduce overfitting and improve the road sign classification effect, we added a dropout's regularization rate of 0.5 after the last convolution operation (convolution+subsampling) and also after

Figure 5: German database classes.

Figure 6: Distribution of the training data set.

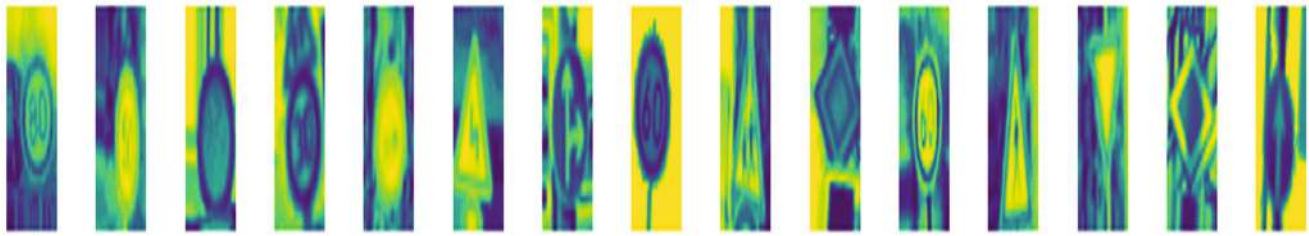

Figure 7: Images after preprocessing process.

Table 1: Performance ([training accuracy/validation accuracy]; [loss/validation loss]) comparison of our 2 model networks for 20 epochs: for our first network, we used Adadelta as the loss function optimizer and LeakyReLU as the activation function; for our second network, we used Adam and ReLU.

Performances	Adadelta+LeakyReLU(our first network)	Adam+ReLU(our second network)
Epoch 1 [train/val]; [loss/val loss]	[0.597/0.958]; [1.47/0.157]	[0.635/0.9766]; [1.325/0.071]
Epoch 5 [train/val]; [loss/val loss]	[0.962/0.992]; [0.128/0.027]	[0.9696/0.9904]; [0.0992/0.034]
Epoch 10 [train/val]; [loss/val loss]	[0.980/0.996]; [0.065/0.013]	[0.983/0.9933]; [0.056/0.021]
Epoch 15 [train/val]; [loss/val loss]	[0.985/0.998]; [0.047/0.004]	[0.9876/0.9986]; [0.039/0.0039]
Epoch 20 [train/val]; [loss/val loss]	[0.987/0.998]; [0.039/0.004]	[0.9897/0.9982]; [0.0328/0.0051]
Training time (h)	0.86	0.91
Test accuracy (%)	99.84	99.78
Test score	0.004	0.009

Table 2: CNN-training parameters.

CNN-training parameters	Value
Batch size training	20
Batch size validation	50
Steps per epoch	2000
Optimizer	Adadelta
Learning rate	1.0
Activation function	LeakyReLU
Input image size	(32, 32, 1)

Table 3: The output data format of the input layer, every intermediate layer, and the output layer in our final model network.

Layers (type)	Output shape
Conv2d_1 (Conv2D)	(none, 28, 28, 60)
Conv2d_2 (Conv2D)	(none, 24, 24, 60)
Max_pooling2d_1 (MaxPooling1)	(none, 12, 12, 60)
Conv2d_3 (Conv2D)	(none, 10, 10, 30)
Conv2d_4 (Conv2D)	(none, 8, 8, 30)
Max_pooling2d_2 (MaxPooling2)	(none, 4, 4, 30)
Dropout_1 (dropout) (50%)	(none, 4, 4, 30)
Flatten_1 (flatten)	(none, 480)
Dense_1 (dense)	(none, 500)
Dropout_2 (dropout) (50%)	(none, 500)
Output layer	(none, 43)

the fully connected layer. We added BN and dropout after the fully connected layer to speed up the model convergence and thus the training speed.

Table 4: Number of trainable parameters of our proposed network compared with that of previous state-of-the-art approaches.

Method	Number of trainable parameters (million)
Faster R-CNN [41]	2.92
Multiscale CNN [42]	1.4
Lightweight deep network (student model) [43]	0.8
Lightweight deep network (teacher model) [43]	7.9
Our first network (our final network)	0.38

Table 5: Training time of our proposed network compared with that of previous state-of-the-art approaches.

Network	Training time (h)
LeNet-5 [34]	0.92
Optimized LeNet-5 [34]	0.68
Classical LeNet-5 [44]	0.87
Our second network	0.91
Our first network (our final network)	0.86

Another strength of our model is that instead of using the tanh (hyperbolic tangent), ReLU (Rectified Linear Unit) and sigmoid activation functions as in the traditional LeNet-5 model, we used LeakyReLU [36]. ReLU allows the network to converge quickly, and although it looks like a linear function, ReLU has a derived function and allows backpropagation. But when the inputs approach zero, or are negative, the gradient of the function becomes zero; the network cannot perform backpropagation and cannot learn what

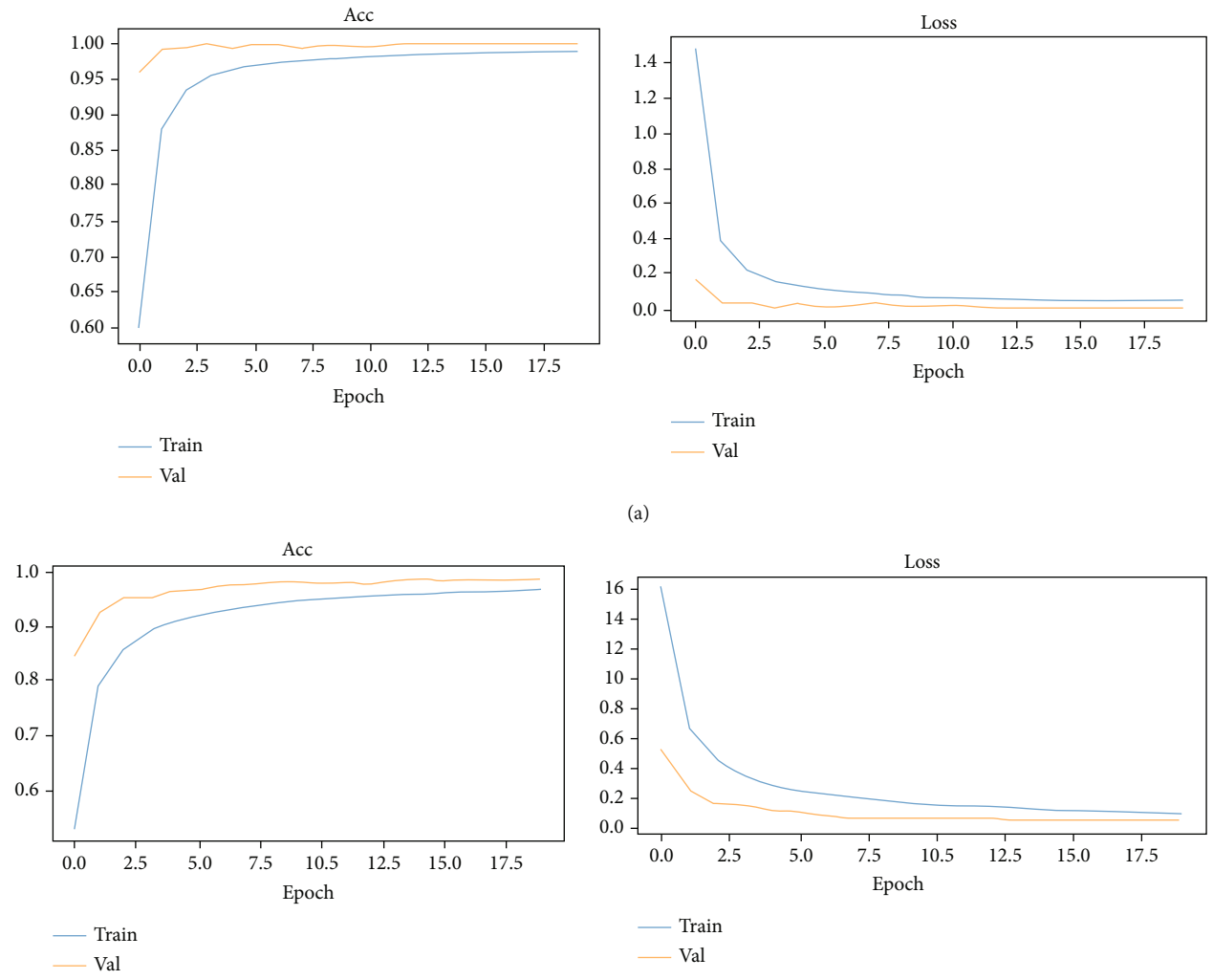

Figure 8: Performance comparison between our final model network and the classic LeNet-5 network: (a) with our final model network; (b) with the classic LeNet-5 network.

decreases the model's ability to fit or train properly from the data. For this reason, the ReLU function is chosen which prevents the problem of the ReLU dying: this variation of the ReLU has a slight positive slope in the negative area, which allows backward propagation even with negative input values. For example, in this work [37], the authors compared ReLU and LeakyReLU according to the classification accuracy on the GTSRB database. The effect is excellent when using LeakyReLU. The LeakyReLU function has as equation

$$f(x) = \begin{cases} 0.01x, & \text{for } x < 0, \\ x, & \text{for } x \geq 0. \end{cases} \tag{1}$$

The function ReLU has as equation

$$f(x) = \begin{cases} 0, & \text{for } x < 0, \\ 1, & \text{for } x \geq 0. \end{cases} \tag{2}$$

Table 6: Accuracy of our proposed network compared with that of previous state-of-the-art approaches on German traffic sign data set.

Methods	Accuracy (%)
Optimized LeNet-5 [34]	Between 93 and 96
Modified LeNet-5 network [35]	95.2
Improved LeNet-5 [27]	99.75
LeNet architecture [45]	96.23
Small CNN [46]	97.4
Improved LeNet-5 [44]	98.12
Lightweight deep network [43]	99.61
Deep CNN [37]	98.96
Efficient CNN [41]	99.66
Ours	*99.84*

According to our model, after each convolution layer, we add the LeakyReLU activation function and also add the fully connected layer before the output layer.

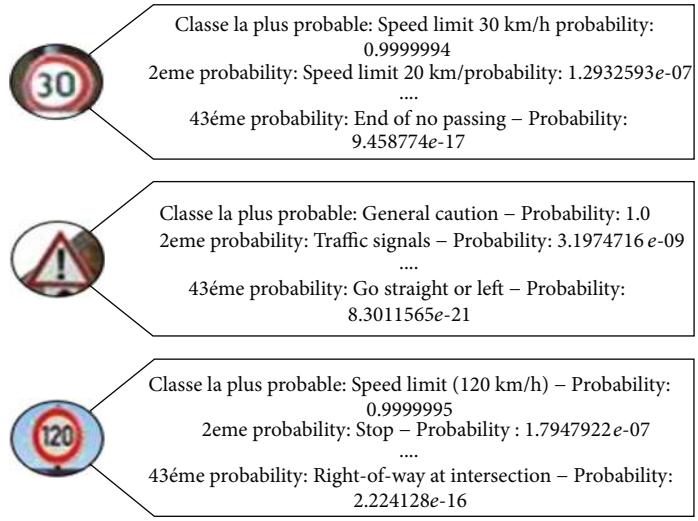

Figure 9: Prediction results on GTSRB.

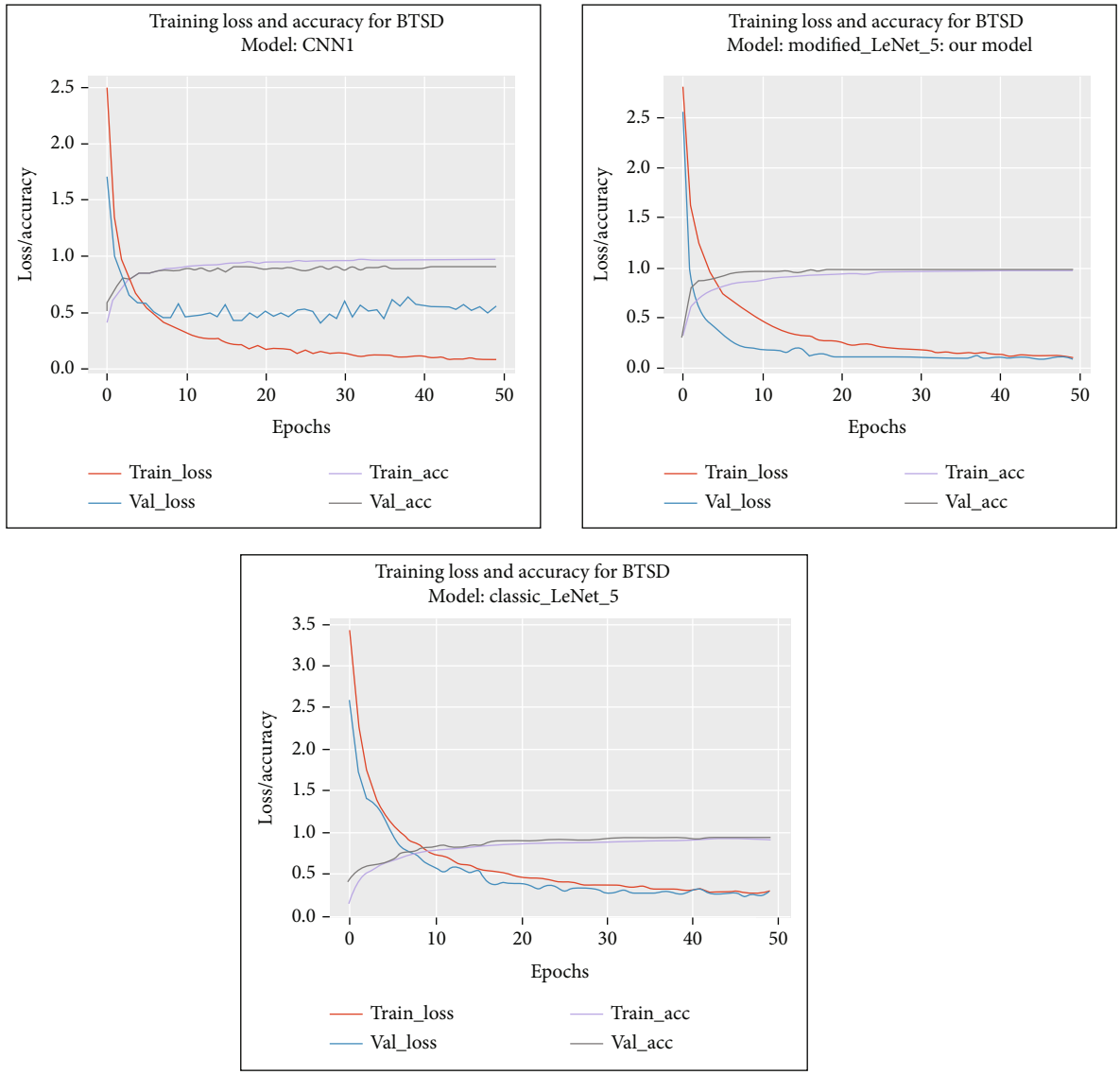

Figure 10: Performance comparison between our final model network, the classic LeNet-5 network, and the other CNN model "CNN_1."

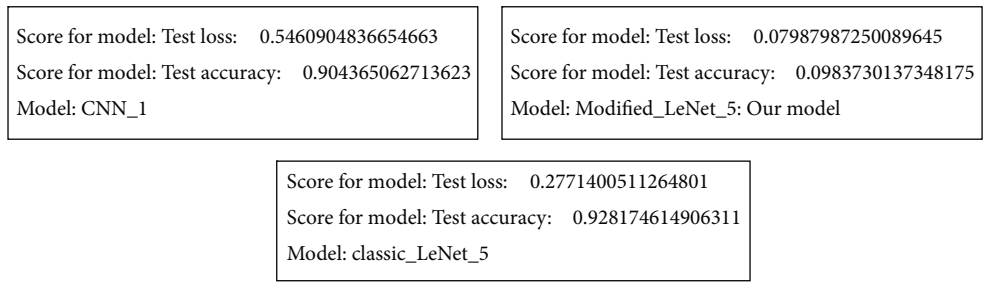

Figure 11: Test loss and test accuracy results for the different models we tested.

As shown in Figure 4, our model is similar to the traditional LeNet-5 model. For our model, we will start with block conv1: this block contains two successive convolution layers (C1, C2); each convolution layer is followed by a BN layer with a LeakyReLU activation function. Correspondingly for the second block Conv2: two successive convolution layers (C3, C4) each convolution layer is followed by a layer of BN with a Leaky activation function. Each block is followed by an MP layer. The second convolution block is followed by an MP layer and a DO (DropOut) regularization. As shown in Figure 4, our model contains a single FC block. This block contains the number of hidden layer nodes equal to 500, followed by the BN layer with LeakyReLU function and DO (DropOut). SM is the classification layer of the image features; they have a total of 43 layers since we have 43 classes of road signs.

4. Experimental Results

GTSRB [38] is a very famous international database of road signs. In order to train and test the model, some research projects [38] used this database. GTSRB contains 43 kinds of road signs, training and test images taken under real conditions, as shown in Figure 5, a total of more than 50,000 images.

The database contains road sign images of different sizes, and the number of images in different categories is different, which will lead to the imbalance of the data set, thus affecting the accuracy of classification, as shown in Figure 6.

In our work, to avoid this problem, we use the data augmentation technique [39] to increase the number of training images since the CNN becomes more efficient with a huge database. Also, by increasing the number of data, we will get a variation of exposures and more points of view for the same image which ensures better prediction. The data augmentation technique is applied on the training images.

In order to train and evaluate our improved model, we made this distribution of data: 20% for the test and 80% for the training, and from this 80% of training, we chose 20% for the validation. The training data is used to train the model, the validation data allows us to supervise the performance of the model while training it (a reduced version of the test data), and the test data is used to evaluate the model. On the other hand, in order to build our model, the TensorFlow frameworks are used.

Table 7: Accuracy of our proposed network compared with that of previous state-of-the-art approaches and 2 CNN models tested by us (model: CNN_1, model: classic_LeNet_5) on Belgian Traffic Sign Data Set (BTSD).

Methods	Accuracy (%)
Single CNN with 3 STNs [47]	98.87
One CNN [48]	98.17
INNLP+SRC (PI) [49]	97.83
Small CNN [46]	98.1
Model: CNN_1	90.43
Model: classic_LeNet_5	92.81
Model: modified_LeNet_5 (our proposed network)	*98.37*

After applying processing and data enhancement techniques to the data to be driven from the GTSRB database as shown in Figure 7.

We will build our model inspired by LeNet-5. First, in the traditional LeNet-5 model, the number of filters in C1 equals 6 and in C2 equals 16. In our improved model, we increased the number of filters as follows: for the first block (C1 and C2), the number equals 60, and for the second block (C3 and C4), the number equals 30. In the traditional model, the size of the filters in C1 and C2 is $5 * 5$. For the first block (C1, C2), a filter size of $5 * 5$ is chosen, and for the second block (C3, C4), a filter size of $3 * 3$ is chosen. Secondly, the traditional model contains two fully connected layers FC1 and FC1 with a number of nodes equal to 84 and 120, respectively. In our model, we have reduced the number of fully connected layers by increasing the number of nodes to 500. In this work, the LeakyReLU activation function was chosen instead of the tanh, sigmoid, and ReLU functions that are used in the traditional LeNet-5 model or in other modified models. On the other hand, in order to optimize the parameters, a loss function had to be set. These losses often measure the quadratic or absolute error between the output generated by the model and the desired output. In our work, we use cross-entropy which is designed specifically for classification problems. It minimizes the distance between two probability distributions: predicted and actual. As a glide slope technique, we use the Adadelta optimizer [40] because of the good performance found when used with the LeakyReLU activation function (Table 1). The CNN training parameters are set in Table 2.

Figure 12: Predict results on BTSD.

When training 20 epochs, we can notice that the second model using Adam as the optimizer and ReLU as the activation function is better trained than the first model using Adadelta and LeakyReLU. On the other hand, when comparing based on training time, our first model consumes less time than the second model. Moreover, we reach an accuracy of 99.84% thanks to the first model. In addition, the test score was 0.004 in the case of the 1st model; on the other hand, it was 0.009 in the case of the 2nd model. According to these comparisons we chose to work with the 1st model which architecture's presented in Table 3 as the final model after all the improvements made.

Our improved model contains 8 layers between convolution, MaxPooling, and fully connected. Taking into account the BN layers, our model contains 12 layers, so it is deeper than the traditional LeNet-5 model. Usually, a deep model that contains many layers costs in terms of the number of parameters as shows in Table 4. An advantage of our model is that it is a bit deep but light; moreover, the training time is reduced compared to several other architectures used as shown in Table 5.

From Figure 8, we can see that the training of our final model network is better than that of the traditional LeNet-5 architecture. With LeNet-5 at epoch = 20, the training accuracy is equal to 0.9609, validation accuracy is equal to 0.9882, loss is equal to 0.106, and validation loss is equal to 0.043. With our improved model at epoch = 20, training accuracy = 0.9879, validation accuracy = 0.9989, loss = 0.039, and validation loss = 0.0042. On the other hand, with LeNet-5, test accuracy = 0.9887 (98.87%), but with our model, test accuracy = 0.9984 (99.84%). We can conclude that our improved model is very powerful and precise; moreover, our improvements were successful.

In order to build a successful traffic sign classification system, many researchers used the LeNet-5 model in their work [27, 34, 35, 43, 44]. Our improvements were very successful, and the accuracy of our model was the best as shown in Table 6.

In order to demonstrate the efficiency of the improved architecture, we try to predict images that have never been seen by the trained model architecture. The results were excellent as shown in Figure 9.

We also use the Belgian database which contains 62 classes, in order to show the efficiency and the good accuracy of our model on several road sign databases. We also trained other models among which the classic LeNet-5 in order to make comparisons according to the accuracy of training and testing with our model. The found results show the effectiveness of our method as illustrated in Figure 10.

From the curves in Figure 10 and from the results shown in Figure 11, it can be seen that the best performance is that of our model; we were able to achieve an excellent test accuracy of 98.37%. But, with the model CNN_1, the accuracy was 90.43%, and in the case of classic_LeNet_5, it was 92.81%.

Our model was able to recognize the traffic signs of the Belgian database correctly with an excellent accuracy compared to other works as shown in Table 7, which confirms the effectiveness of our method as shown in Figure 12.

The good performance of our model's architecture pushed us towards an implementation in an embedded application using the webcam. First, we start by opening the webcam. We have used the model architecture that has been trained and registered. The road sign image captured by the webcam goes through the preprocessing process (resizing, grayscaling, and normalization). The model we have trained contains the output of labels and 43 types of signs. Our model analyzes this frame and generates its feature vectors. Finally, it will decide which class this frame belongs to (the prediction). The flowchart of our application using webcam is presented in Figure 13.

We downloaded some road sign images from Google using a smartphone. Then, we will show the traffic sign image to the webcam to make the prediction. Even with the vibration of the hand picking up the phone, the classification was perfect. In addition, the same road sign (speed limit 50 km/h) is displayed at different viewing positions to the

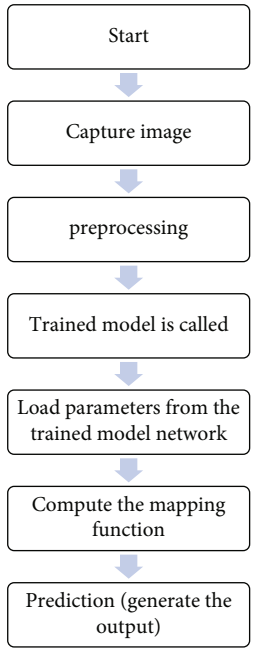

Figure 13: Flowchart of our application using webcam.

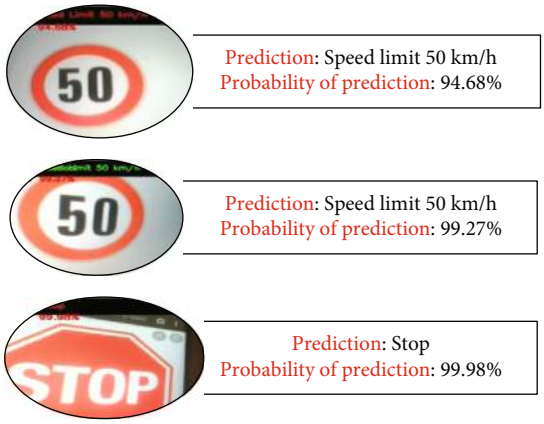

Figure 14: Real-time road sign classification.

webcam, and the prediction is still correct and perfect, as shown in Figure 14.

5. Conclusion

The objective of road sign classification is to develop a system capable of automatically assigning a class to a road sign image. The applications of automatic classification of road signs are numerous, but the accuracy of our system was remarkable and among the best when compared with other works. We have modified and improved the architecture inspired by the famous LeNet-5. Our improvements allow us to obtain an accuracy of 99.84% and a reduced number of trained parameters compared to the depth of our model. Lightness allows us to try our model with an embedded application that uses the webcam. In this case, the classification is also very accurate.

Data Availability

No data were used to support this study.

Conflicts of Interest

The authors declare that they have no conflicts of interest.

References

[1] Y. Zhang, K. Sohn, R. Villegas, G. Pan, and H. Lee, "Improving object detection with deep convolutional networks via Bayesian optimization and structured prediction," in *2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 249–258, Boston, MA, USA, June 2015.

[2] F. Özyurt, T. Tuncer, E. Avci, M. Koç, and İ. Serhatlioğlu, "A novel liver image classification method using perceptual hash-based convolutional neural network," *Arabian Journal for Science and Engineering*, vol. 44, no. 4, pp. 3173–3182, 2019.

[3] J. Wu, *Introduction to convolutional neural networks*, Jianxin Wu, 31.

[4] G. Yao, T. Lei, and J. Zhong, "A review of convolutional-neural-network-based action recognition," *Pattern Recognition Letters*, vol. 118, pp. 14–22, 2019.

[5] A. Ladgham, A. Sakly, and A. Mtibaa, *Optimal feature selection based on hybridization of MSFLA and Gabor filters for enhanced MR brain image recognition using SVM*, Anis Ladgham, Anis Sakly, Abdellatif Mtibaa, 17.

[6] 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)*, pp. 504–507, Hammamet, Tunisia, December 2014.

[7] K. O'Shea and R. Nash, "An introduction to convolutional neural networks," 2015, July 2020, http://arxiv.org/abs/1511.08458.

[8] O. Russakovsky, J. Deng, H. Su et al., "ImageNet large scale visual recognition challenge," *International Journal of Computer Vision*, vol. 115, no. 3, pp. 211–252, 2015.

[9] 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-ICMR'15*, pp. 643–650, Shanghai, China, 2015.

[10] E. Dandıl and R. Özdemir, "Real-time facial emotion classification using deep learning," vol. 2, no. 1, p. 5, 2019.

[11] 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 on - HPCCT 2019*, pp. 274–278, Guangzhou, China, 2019.

[12] 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.

[13] Y. Lecun, L. Bottou, Y. Bengio, and P. Haffner, "Gradient-based learning applied to document recognition," *Proceedings of the IEEE*, vol. 86, no. 11, pp. 2278–2324, 1998.

[14] T.-B. Xu, P. Yang, X.-Y. Zhang, and C.-L. Liu, "LightweightNet: toward fast and lightweight convolutional neural networks via architecture distillation," *Pattern Recognition*, vol. 88, pp. 272–284, 2019.

[15] M. Kayed, A. Anter, and H. Mohamed, "Classification of garments from fashion MNIST dataset using CNN LeNet-5 architecture," in *2020 International Conference on Innovative Trends in Communication and Computer Engineering (ITCE)*, pp. 238–243, Aswan, Egypt, February 2020.

[16] L. Chao, W. Changyuan, L. Zhi, and H. Wenbo, "Blink fatigue detection algorithm based on improved Lenet-5," in *Proceedings of the 2019 International Conference on Precision Machining, Non-Traditional Machining and Intelligent Manufacturing (PNTIM 2019)*, Xi'an, China, 2019.

[17] T. Li, D. Jin, C. du et al., "The image-based analysis and classification of urine sediments using a LeNet-5 neural network," *Computer Methods in Biomechanics and Biomedical Engineering: Imaging & Visualization*, vol. 8, no. 1, pp. 109–114, 2020.

[18] H. Xiang, Y. Zhao, Y. Yuan, G. Zhang, and X. Hu, "Lightweight fully convolutional network for license plate detection," *Optik*, vol. 178, pp. 1185–1194, 2019.

[19] A. Wikanningrum, R. F. Rachmadi, and K. Ogata, "Improving lightweight convolutional neural network for facial expression recognition via transfer learning," in *2019 International Conference on Computer Engineering, Network, and Intelligent Multimedia (CENIM)*, pp. 1–6, Surabaya, Indonesia, November 2019.

[20] K. Lu, J. Chen, J. J. Little, and H. He, "Lightweight convolutional neural networks for player detection and classification," *Computer Vision and Image Understanding*, vol. 172, pp. 77–87, 2018.

[21] E. Cengil, A. Cinar, and Z. Guler, "A GPU-based convolutional neural network approach for image classification," in *2017 International Artificial Intelligence and Data Processing Symposium (IDAP)*, pp. 1–6, Malatya, September 2017.

[22] S.-C. Huang, C.-Y. Li, H.-Y. Lin, and W.-L. Tai, *Traffic sign detection and recognition using image features and convolutional neural network*, Shu-Chun Huang, Chih-Yi Li, Huei-Yung Lin, Wen-Lung Tai, p. 6.

[23] A. D. Kumar, R. Karthika, and L. Parameswaran, *Novel deep learning model for traffic sign detection using capsule networks*, Amara Dinesh Kumar, R.Karthika, Latha Parameswaran, p. 5.

[24] H. Xu and G. Srivastava, "Automatic recognition algorithm of traffic signs based on convolution neural network," *Multimedia Tools and Applications*, vol. 79, no. 17-18, pp. 11551–11565, 2020.

[25] D. A. Alghmgham, G. Latif, J. Alghazo, and L. Alzubaidi, "Autonomous traffic sign (ATSR) detection and recognition using deep CNN," *Procedia Computer Science*, vol. 163, pp. 266–274, 2019.

[26] S. Zhou, W. Liang, J. Li, and J.-U. Kim, "Improved VGG model for road traffic sign recognition," *Computers, Materials & Continua*, vol. 57, no. 1, pp. 11–24, 2018.

[27] J. Cao, C. Song, S. Peng, F. Xiao, and S. Song, "Improved traffic sign detection and recognition algorithm for intelligent vehicles," *Sensors*, vol. 19, no. 18, p. 4021, 2019.

[28] X. Bangquan and W. Xiao Xiong, "Real-time embedded traffic sign recognition using efficient convolutional neural network," *IEEE Access*, vol. 7, pp. 53330–53346, 2019.

[29] C. Gamez Serna and Y. Ruichek, "Classification of traffic signs: the European dataset," *IEEE Access*, vol. 6, pp. 78136–78148, 2018.

[30] J. A. Khan, Y. Chen, Y. Rehman, and H. Shin, "Performance enhancement techniques for traffic sign recognition using a deep neural network," *Multimedia Tools and Applications*, vol. 79, no. 29-30, pp. 20545–20560, 2020.

[31] H. M. Bui, M. Lech, E. Cheng, K. Neville, and I. S. Burnett, "Using grayscale images for object recognition with convolutional-recursive neural network," in *2016 IEEE Sixth International Conference on Communications and Electronics (ICCE)*, pp. 321–325, Ha-Long City, Quang Ninh Province, Vietnam, July 2016.

[32] K. G. Dhal, A. Das, S. Ray, J. Gálvez, and S. Das, "Histogram equalization variants as optimization problems: a review," *Archives of Computational Methods in Engineering*, pp. 1–26, 2020.

[33] Y. Pan, V. Kadappa, and S. Guggari, "Identification of road signs using a novel convolutional neural network," in *Cognitive Informatics, Computer Modelling, and Cognitive Science*, pp. 319–337, Elsevier, 2020.

[34] C. Zhang, X. Yue, R. Wang, N. Li, and Y. Ding, "Study on traffic sign recognition by optimized Lenet-5 algorithm," *International Journal of Pattern Recognition and Artificial Intelligence*, vol. 34, no. 1, article 2055003, 2020.

[35] D. Yasmina, R. Karima, and A. Ouahiba, "Traffic signs recognition with deep learning," in *2018 International Conference on Applied Smart Systems (ICASS)*, pp. 1–5, Medea, Algeria, November 2018.

[36] A. K. Dubey and V. Jain, "Comparative study of convolution neural network's Relu and leaky-Relu activation functions," in *Applications of Computing, Automation and Wireless Systems in Electrical Engineering*, S. Mishra, Y. R. Sood, and A. Tomar, Eds., vol. 553, pp. 873–880, Springer Singapore, Singapore, 2019.

[37] S. Yin, J. Deng, D. Zhang, and J. Du, "Traffic sign recognition based on deep convolutional neural network," in *Computer Vision*, J. Yang, Q. Hu, M.-M. Cheng, L. Wang, Q. Liu, X. Bai, and D. Meng, Eds., vol. 771, pp. 685–695, Springer Singapore, Singapore, 2017.

[38] Y. Saadna and A. Behloul, "An overview of traffic sign detection and classification methods," *International Journal of Multimedia Information Retrieval*, vol. 6, no. 3, pp. 193–210, 2017.

[39] L. Perez and J. Wang, "The effectiveness of data augmentation in image classification using deep learning," 2017, July 2020, http://arxiv.org/abs/1712.04621.

[40] M. D. Zeiler, "ADADELTA: an adaptive learning rate," 2012, July 2020, http://arxiv.org/abs/1212.5701.

[41] J. Li and Z. Wang, "Real-time traffic sign recognition based on efficient CNNs in the wild," *IEEE Transactions on Intelligent Transportation Systems*, vol. 20, no. 3, pp. 975–984, 2019.

[42] P. Sermanet and Y. LeCun, "Traffic sign recognition with multi-scale convolutional networks," in *The 2011 International Joint Conference on Neural Networks*, pp. 2809–2813, San Jose, CA, USA, July 2011.

[43] J. Zhang, W. Wang, C. Lu, J. Wang, and A. K. SangaiahLightweight deep network for traffic sign classification," *Annales des Telecommunications*, vol. 75, no. 7-8, pp. 369–379, 2020.

[44] W. Li, X. Li, Y. Qin, W. Song, and W. Cui, "Application of improved LeNet-5 network in traffic sign recognition," in *Proceedings of the 3rd International Conference on Video and Image Processing*, pp. 13–18, Shanghai China, December. 2019.

[45] A. Bouti, M. A. Mahraz, J. Riffi, and H. Tairi, "A robust system for road sign detection and classification using LeNet architecture based on convolutional neural network," *Soft Computing*, vol. 24, no. 9, pp. 6721–6733, 2020.

[46] W. Li, D. Li, and S. Zeng, "Traffic sign recognition with a small convolutional neural network," *IOP Conference Series: Materials Science and Engineering*, vol. 688, article 044034, 2019.

[47] 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.

[48] F. Jurisic, I. Filkovic, and Z. Kalafatic, "Multiple-dataset traffic sign classification with one CNN," in *2015 3rd IAPR Asian Conference on Pattern Recognition (ACPR)*, pp. 614–618, Kuala Lumpur, Malaysia, November 2015.

[49] M. Mathias, R. Timofte, R. Benenson, and L. Van Gool, "Traffic sign recognition – how far are we from the solution?," in *The 2013 International Joint Conference on Neural Networks (IJCNN)*, pp. 1–8, Dallas, TX, USA, August 2013.