

RESEARCH ARTICLE

Hybrid Image Improving and CNN (HIICNN) Stacking Ensemble Method for Traffic Sign Recognition

GÜLCAN YILDIZ¹, AHMET ULU², BEKİR DİZDAROĞLU², AND DOĞAN YILDIZ³

¹Department of Computer Engineering, Ondokuz Mayıs University, 55270 Samsun, Turkey

²Department of Computer Engineering, Karadeniz Technical University, 61080 Trabzon, Turkey

³Department of Electrical-Electronics Engineering, Ondokuz Mayıs University, 55270 Samsun, Turkey


Corresponding author: Gülcan Yıldız (gulcan.ozler@omu.edu.tr)

ABSTRACT Traffic sign recognition techniques aim to reduce the probability of traffic accidents by increasing road and vehicle safety. These systems play an essential role in the development of autonomous vehicles. Autonomous driving is a popular field that is seeing more and more growth. In this study, a new high-performance and robust deep convolutional neural network model is proposed for traffic sign recognition. The stacking ensemble model is presented by combining the trained models by applying improvement methods on the input images. For this, first of all, by performing preprocessing on the data set, more accurate recognition was achieved by preventing adverse weather conditions and shooting errors. In addition, data augmentation was applied to increase the images in the data set due to the uneven distribution of the number of images belonging to the classes. During the model training, the learning rate was adjusted to prevent overfitting. Then, a new stacking ensemble model was created by combining the models trained with the input images that were subjected to different preprocessing. This ensemble model obtained 99.75% test accuracy on the German Traffic Sign Recognition Benchmark (GTSRB) dataset. When compared with other studies in this field in the literature, it is seen that recognition is performed with higher accuracy than these studies. Additively different approaches have been applied for model evaluation. Gradient-weighted Class Activation Mapping (Grad-CAM) was used to make the model explainable. Evidential deep learning approach was applied to measure the uncertainty in classification. Results for safe monitoring are also shared with SafeML-II, which is based on measuring statistical distances. In addition to these, the migration test is applied with BTSC (Belgium Traffic Sign Classification) dataset to test the robustness of the model. With the transfer learning method of the models trained with GTSRB, the parameter weights in the feature extraction stage are preserved, and the training is carried out for the classification stage. Accordingly, with the stacking ensemble model obtained by combining the models trained with transfer learning, a high accuracy of 99.33% is achieved on the BTSC dataset. While the number of parameters the single model is 7.15 M, the number of parameters of the stacking ensemble model with additional layers is 14.34 M. However, the parameters of the models trained on a single preprocessed dataset were not trained, and transfer learning was performed. Thus, the number of trainable parameters in the ensemble model is only 39,643.

INDEX TERMS BTSC, convolutional neural network, deep learning, evidential deep learning, Grad-CAM, GTSRB, image improving, SafeML-II, safety monitoring, traffic sign recognition, transfer learning, uncertainty evaluation.

I. INTRODUCTION

There has been rapid growth in Advanced Driver Assistance Systems (ADAS) worldwide due to advances in sensing, communication, and computing technologies. ADAS is one of the most critical parts of today's smart cars. One of the

The associate editor coordinating the review of this manuscript and approving it for publication was Antonio J. R. Neves .

functions that these systems bring to vehicles is to recognize traffic signs. ADAS can replace some of the drivers' decisions with machine decisions. In this respect, ADAS provides safer and smoother driving by preventing many driver errors that may result in an accident [1].

Needs such as the smooth flow of traffic on the highway and the safe continuation of ordinary life actions within the framework of a particular system and order have led

to the emergence of Traffic Signs (TSs). While designing TSs, care is taken to ensure that they are distinguishable from other objects in the environment and that they are easily understandable regardless of the spoken language. For this purpose, traffic signs are designed using regular and simple geometric shapes such as triangles, circles, rectangles, or polygons. The colors of TSs are usually chosen from primary colors such as blue, yellow, black, green, white, and red. TSs are high-quality designs, each of which has special meanings in the traffic-law order. The pictographs in the center of the TSs reflect these meanings of each TS [2].

TSs are also of great importance in terms of driving safety. While driving, following the TS's carefully and taking the necessary actions according to these signs is essential. However, it can be challenging to detect TSs due to distraction and other environmental factors. The importance of intelligent tools in eliminating such negativities is felt more and more every day [3]. Intelligent vehicles collect real-time data on the road they travel with the help of the sensors they have on them. They can automatically detect TSs by processing these collected data [4].

With traffic sign recognition, TS types are effectively classified. The main difficulty in this classification is that the data quality sensed in different environmental conditions is different. In general, image quality can be affected by several factors. These factors can be listed as the resolution quality of the camera used when collecting road data, the captured image being too bright, too dim, or containing spotlights, the weather conditions at the time the image was taken, motion blur caused by vehicle movement, and objects that hide TSs in the environment where the image was taken. The main source of these difficulties is uncontrollable environmental conditions [5].

Although traffic sign recognition algorithms using deep learning offer high performance in recognition accuracy, these algorithms are highly complex and have high computational costs. Therefore, they face various limitations. Thus, more improvements are needed in traffic sign recognition algorithms. Using different image enhancement techniques may be a better-performing solution to the difficulties. By this way, better accuracy can be achieved while simplifying the network complexity [6].

In this paper, a deep convolutional neural network model with high accuracy for traffic sign classification is presented. To prevent problems such as brightness and distortion in the images, and to provide better extraction of the features, the images were preprocessed. In addition, it is aimed to give faster results by resizing images of different sizes to 60×60 . The main contributions made in this article are as follows:

- The effect of activation function on accuracy is investigated.
- The effect of input images on accuracy is investigated. Here, the images are given to the model after different preprocessing and using them together.
- Preprocessing is done with traditional methods. In addition, feature extraction and classification are per-

formed with deep learning, aiming for better recognition performance. In this way, a hybrid model is created.

- Better performance has been achieved by combining models trained using different input images. Thus, the stacking ensemble model is created.
- By transfer learning, pre-trained models are combined, thus reducing the number of trainable parameters.

The remainder of the study is organized as follows. In Section II, studies on traffic sign classification are given. The details of the proposed method are given in Section III. In addition, image preprocessing steps and data augmentation are presented. Section IV includes the performance evaluation results along with the datasets. Additionally, the stacking ensemble model and misclassified images are shown. In Section V, the conclusion and future work are mentioned.

II. RELATED WORK

The studies for TSR can be divided into traditional and learning methods. In traditional methods, the classification is performed by obtaining the feature extraction by using image processing techniques for features such as shape and color. However, lately, deep learning-based approaches have come to the fore due to their speed and performance superiority. Therefore, deep learning-based methods are examined in this paper.

Sermanet and LeCun [2] proposed a multi-scale ConvNets network producing 98.97% accuracy in GTSRB Competition Phase I. The network architecture consists of a two-stage structure. The features obtained in the first stage are fed directly to the classification layer in addition to the second stage. Thus, it is aimed to achieve higher accuracy. Here, images were resized as 32×32 , and YUV color space was used. After Phase I, the existing network was expanded further, and 99.17% accuracy was achieved by using gray images instead of color.

Cireşan et al. [7] proposed a multicolumn MCDNN approach for the TS classification problem, which includes different deep neural networks trained on preprocessed data. The authors used the GTSRB dataset in this study. In order to improve the disadvantages of this dataset, preprocessing techniques such as image resizing, histogram equalization, adaptive histogram equalization, and contrast normalization were applied. All training images were resized to 48×48 because the MCDNN method demands they all have the same size. Five networks for each of the five datasets comprised a total of 25 trained networks. A 99.46% recognition rate was achieved using the proposed MCDNN and 25 DNN columns. However, training the 25-column MCDNN takes about 37 hours on four GPUs. Due to the large number of factors this strategy requires, its performance is subpar (38.5 M).

In the study [8], a method called hinge loss stochastic gradient descent (HLSGD) was proposed. Four different preprocessing were carried out as original image, histogram

synchronized image, adjust image density values and contrast limited adaptive histogram synchronized image. For each of these preprocesses, 5 CNN networks were trained. In the ensemble method, a total of 20 CNN networks were obtained, with 4 different preprocess and 5 CNN trainings for each. Ensemble method accuracy was calculated as 99.65%. There are 1,162,284 trainable parameters in a single CNN network, and a total of about 23.2 M parameters in the ensemble network.

Aghdam et al. [9] introduced a new ConvNets architecture. In the architecture, the initially given convolution block is divided into two parallel blocks and then concatenated. In addition, a compact version was produced by removing the fully connected layer from this architecture. Finally, an ensemble method is proposed to achieve better performance by using the least number of ConvNets. With the proposed ensemble method, 99.61% accuracy was achieved.

Arcos-García et al. [10] proposed a TS recognition system using a Deep Neural Network with convolutional layers and spatial transformer networks on the GTSRB dataset. With this system, TS images were subjected to a fine-grained classification. CNN with three spatial transformers gradually suppresses background and geometric noise by performing explicit geometric transformations. Also, in this study, the efficacy of momentumless Stochastic Gradient Descent (SGD), SGD with Nesterov's accelerated gradient, RMSprop, and Adam four mini-batch gradient descent optimization algorithms was investigated. The edges of the images are enhanced with global normalization and Gaussian kernels. The network in this study contains three spatial transformer layers with 14 million (M) parameters. This proposed network architecture achieved a recognition performance of 99.71% and 98.87% in GTSRB and BTSC TS datasets, respectively.

Saha et al. [11] proposed a new architecture with high performance for different datasets. A residual CNN network with a hierarchically dilated skip connection is used in the architecture. The performance results for the GTSRB and BTSC benchmarks of the study were calculated as 99.33% and 99.17%, respectively. The number of parameters of the architecture is 6,256 M.

In the study in [12], two lightly deep convolutional neural networks called student and teacher were proposed. The performance of the student model with fewer parameters was measured by distilling information from the teacher model with more parameters. The student model has 0.8 million parameters. Student model performance results for the GTSRB and BTSC datasets are 99.61% and 99.13%, respectively.

Nartey et al. [13] presented a new semi-supervised method for datasets with classes with small and variable numbers of data. The integration of weakly-supervised learning and self-training with self-paced learning is employed within the framework to produce attention maps for enhancing the

training set. A novel algorithm is also utilized to generate and select pseudo-labeled samples through a pseudo-label generation and selection process. When the rate of pseudo-labeled samples was selected as 20% with self-training, 99.27% accuracy was reached on the GTSRB dataset. Similarly, 98.97% accuracy was obtained on BTSC when the sample rate was 40%.

Bayouhd et al. [14] proposed a new model using transfer learning with 2D to 3D transformation. Here, the deep-network model that processes 2D features was pre-trained. Then, the obtained features were 3D transformed and given to a shallower model. With this hybrid model, it was aimed at learning quickly with fewer parameters. The total parameter number of the model, which achieved 99.28% accuracy on the GTSRB benchmark, is 4 M.

Bi et al. [15] presented an enhanced VGG-16 network for traffic sign recognition. Convolutional layers that appear redundant in the VGG-16 network are not included in the designed network, thus reducing the number of parameters. In addition, it was planned to achieve higher accuracy with lower parameters by adding the GAP (global average pooling) and the BN (batch normalization) layer. As such, the total number of parameters in the network is 1.15 million. The accuracy value in the GTSRB data set is 99.21%. In the migration test of the study, 99.02% accuracy was obtained with the preprocessed BTSC dataset. As a result, although the number of parameters of the method is low, the accuracy performance is lower than most similar studies in the literature.

In the study in [16], a new thin - single row but deep architecture called Deepthin was proposed. Each convolution layer contains less than 50 trainable features in the low parameter number model. Both RGB and gray colors are used separately for the model input image. The accuracy results for the GTSRB dataset were calculated as 99.42% and 99.40% for the color and gray input images, respectively. In addition, three models (6 in total) for gray and color images were combined and trained, and an ensemble model was created. The accuracy result of the ensemble model with 2.69×10^5 parameters is 99.72%. In addition, model performance with learning from scratch and transfer learning for the BTSC dataset is investigated. While 98.97% test accuracy is achieved with scratch-learning, 99.29% accuracy is achieved with transfer learning.

Fang et al. advanced the MicronNet architecture by applying batch normalization and factorization [17]. Thus, according to MicronNet on the GTSRB dataset, it reduced the number of parameters and increased the accuracy value. With the number of parameters 0.44 M, the MicronNet-BF model reached 99.38% accuracy in GTSRB.

Youssef [18] proposed new methods for traffic sign detection and classification. The CNN network presented in the classification section has 0.8 parameters. With the CNN network, a 99.20% accuracy value was obtained in the GTSRB dataset.

Patel et al. studied the effect of regularization processes on the traffic sign recognition system [19]. The processes used are L1, L2, dropout, and batch normalization. When the results are examined, it is seen that 99.56% accuracy has been reached on the GTSRB dataset by using batch normalization and dropout together.

Zhang and Huang proposed a fast and highly accurate traffic sign recognition model [20]. The developed Inception module and multi-scale feature fusion were used in the feature extraction phase of the proposed model. Global average pooling and batch normalization were used for fast training by reducing the number of model parameters. With this model, 99.6% accuracy was achieved in the GTSRB dataset.

In Zheng and Jiang's study [21], a performance comparison of convolutional neural networks and transformers on traffic sign recognition was made. For this, seven convolutional neural networks and five vision transformers models were tested. In the study on three different datasets, it was seen that lower accuracy was obtained with vision transformers (ViTs) compared to the convolutional neural networks. For the GTSRB dataset, the best test result with ViTs was achieved at 86.03%. The highest accuracy-producing CNN network in the study was VGG16, with 98.84%. When a comparison is made over the accuracy metric, it is seen that there is a very high-performance difference between the two models. In short, ViTs performed lower than CNN networks in traffic sign recognition. In addition, compared to CNN models, more datasets are needed to train ViTs models, and these models are also sensitive to translation invariance.

The primary objective of [22] is to explore the application of convolutional neural networks (CNNs) in the classification of traffic signs, with a particular focus on leveraging pre-trained models such as ResNet50, DenseNet121, and VGG16. In order to improve the accuracy and resilience of the model, the authors adopt an ensemble learning approach that employs majority voting. This approach enhances the model's performance by aggregating the outputs of multiple CNNs. In the study of three different datasets, 98.84% recognition accuracy on GTSRB, 98.33% on BTSD and 94.55% on TSRD are achieved with the ensemble approach.

When the studies are examined, primarily, the proposed model has either achieved high accuracy in a single dataset or has a low number of parameters. It was observed that lower accuracy results were obtained in studies with different datasets. In our study, an efficient, powerful, and highly accurate model was proposed for different datasets.

III. METHODOLOGY

The proposed classification approach consists of three stages: the preprocessing stage used to make the images in the data set more meaningful, the data augmentation stage to increase and stabilize the data number, and the deep learning stage for classification.

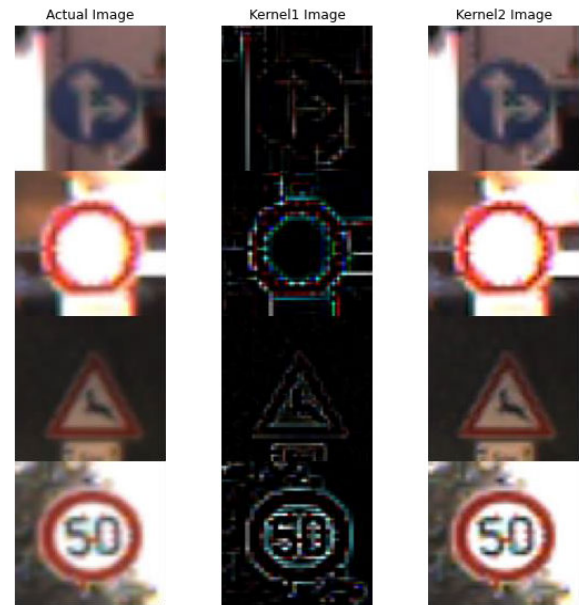


FIGURE 1. Left: original sample images in GTSRB dataset. Middle: The images on the left are *kernel 1* filtered images. Right: The images on the left are *kernel 2* filtered images.

TABLE 1. Data augmentation arguments.

Symbol	Range Value
Width shifting	0.1
Height shifting	0.1
Zooming	0.2
Shearing	0.15
Rotation	10°

A. PREPROCESSING

When the images in the data set are examined, it is seen that they are obtained in different environments, such as illumination, brightness, and blurring. Under these conditions, more parameters for feature extraction will be needed when creating a deep-learning model. It will be beneficial to apply preprocessing and image enhancement methods to the dataset to provide better learning and obtain a higher accuracy result with low parameters. Although there are effective methods based on deep learning developed for image enhancement, simple classical image enhancement methods are used here. Because deep learning models contain much more complexity.

First, edge detection and unsharp masking were applied to the dataset for image enhancement processes in this study. For this, two different kernel matrices are used. To filter the image, the image is convoluted with the kernel values. The kernel matrices used are given with (1) and (2). The edges are highlighted with *Kernel 1* [23], [24]. Since the shapes of the traffic signs are of a particular structure, the edge information carries essential information. In addition, the image has been improved with *Kernel 2* to contribute to

TABLE 2. Details and layer information of the proposed architecture in this study.

Layer	Kernel	Stride	Channels	Output size	Parameters
Convolution layer	5x5	1	100	60x60	7600
Batch Normalization					400
ReLU					
Convolution layer	5x5	1	100	60x60	250100
Batch Normalization					400
ReLU					
Max-pooling layer	2x2	2	100	30x30	
	7x7				367575
Convolution layer	5x5	1	75	30x30	187575
	3x3				67575
Batch Normalization					300
Activation Function					
Max-pooling layer	2x2	2	75	15x15	
	7x7				919000
Convolution layer	5x5	1	250	15x15	469000
	3x3				169000
Batch Normalization					1000
Activation Function					
Max-pooling layer	2x2	2	250	7x7	
Concatenate			250	21x7	
Flatten					
Fully-Connected			128		4704128
Batch Normalization					512
Dropout(0.5)					
Fully-Connected				43	5547
Activation (Softmax)					

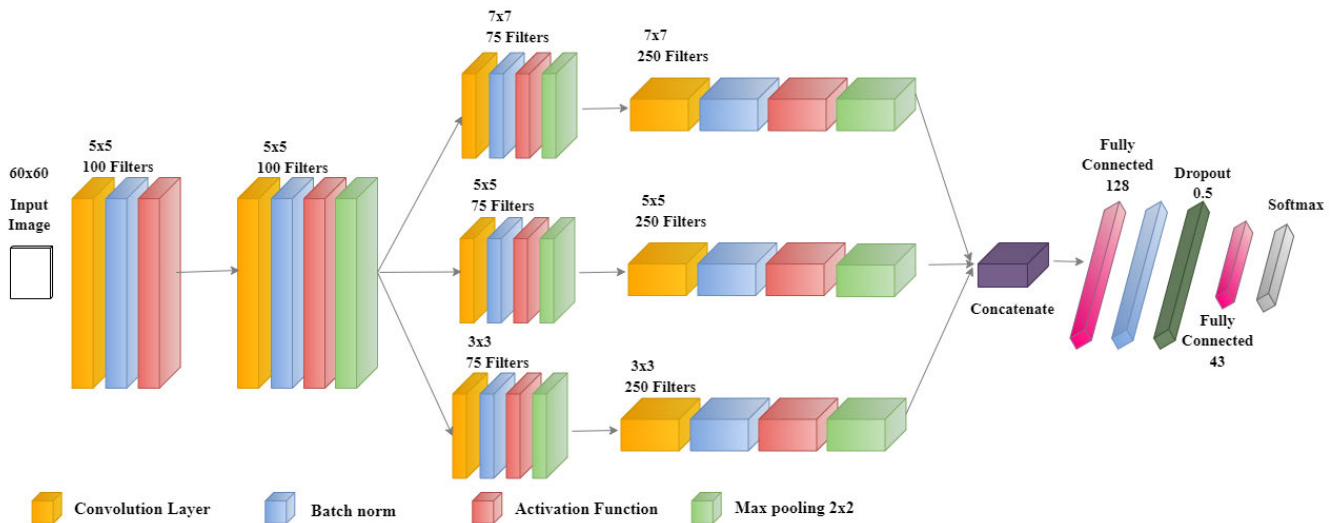


FIGURE 2. Layer information of the proposed model architecture.

TABLE 3. Test accuracy result obtained by using Kernel 1.

Block 1	Block 2	Block 3	Accuracy (%)	F1-Score	Precision	Recall
ReLU	ReLU	ReLU	99.43	99.43	99.45	99.43
ReLU	PReLU	LeakyReLU (0.01)	99.52	99.53	99.54	99.52
ReLU	PReLU	LeakyReLU (0.1)	99.47	99.47	99.48	99.47
ReLU	LeakyReLU (0.01)	LeakyReLU (0.1)	99.62	99.62	99.63	99.62

TABLE 4. GTSRB test results by data augmentation and preprocessing.

Preprocess	Data augmentation	Top-1 Accuracy (%)	F1-Score	Precision	Recall
Yes (Kernel 1)	Yes	99.62	99.62	99.63	99.62
Yes (Kernel 1)	No	99.13	99.13	99.16	99.13
No	Yes	99.51	99.51	99.52	99.51
No	No	99.28	99.28	99.30	99.28

TABLE 5. Test accuracy results for Kernels.

Kernel	Top-1 Accuracy (%)	F1-Score	Precision	Recall
Kernel 1	99.62	99.62	99.63	99.62
Kernel 2	99.65	99.65	99.66	99.65
Kernel 1 & Kernel 2	99.39	99.39	99.41	99.39

the classification stage. The images because of the filtering performed on the sample images from the data set are shown in Figure 1.

$$Kernel\ 1 = \begin{bmatrix} -1 & -1 & -1 \\ -1 & 8 & -1 \\ -1 & -1 & -1 \end{bmatrix} \quad (1)$$

$$Kernel\ 2 = \frac{-1}{256}x \begin{bmatrix} 1 & 4 & 6 & 4 & 1 \\ 4 & 16 & 24 & 16 & 4 \\ 6 & 24 & -476 & 24 & 6 \\ 1 & 16 & 24 & 16 & 4 \\ 1 & 4 & 6 & 4 & 1 \end{bmatrix} \quad (2)$$

It has been seen in the literature that image dimensions affect model performance [16], [25]. Therefore, images were resized as 60 × 60 after image enhancement processes. The images used are in RGB color space.

B. DATA AUGMENTATION

The images of the classes in the used dataset do not have an equal distribution. To prevent this, data augmentation values are used for the training healthier. The data augmentation operations and values used are given in Table 1.

C. PROPOSED DEEP LEARNING MODEL

The network architecture proposed in this article is designed based on [9]. The proposed network architecture consists of



FIGURE 3. Example images for each class of the dataset.

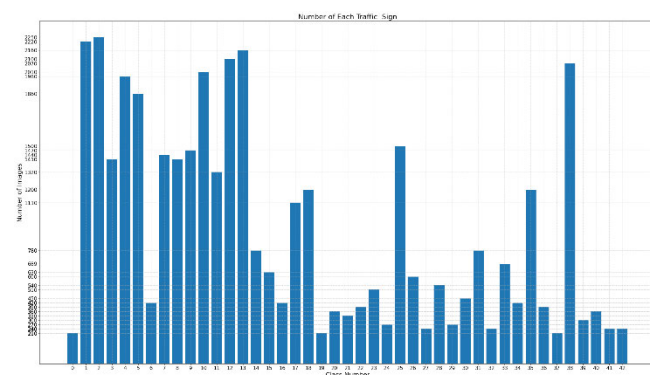


FIGURE 4. Number of each traffic sign in GTSRB.

8 convolution blocks. Convolution blocks have a convolution layer, batch normalization layer, activation function, and max pooling layers (excluding first block), respectively. In the first two block, the convolution layer has a 5 × 5 kernel filter size of 100 channels and one stride parameter value. Then this second block is connected to 3 separate parallel convolution blocks, and two serial intermediate

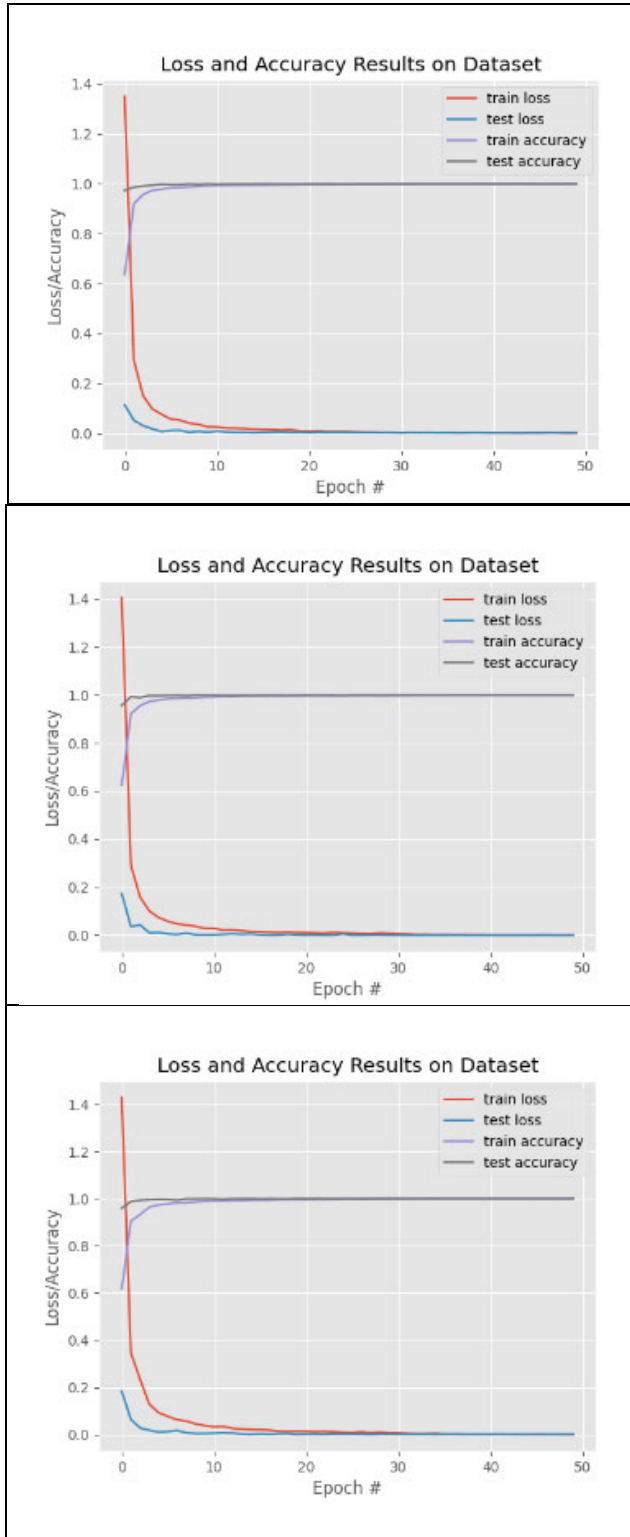


FIGURE 5. Loss and accuracy results during training top) kernel 1, middle) kernel 2, bottom) kernel 1 and kernel 2.

blocks are created. In 2-serial convolution blocks, there are 75 channels filters in the first convolution and 250 channels in the second convolution. The kernel size is 7×7 , 5×5 and 3×3 respectively in the three parallel blocks.

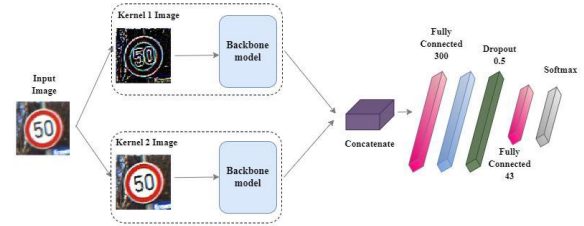


FIGURE 6. Flow diagram of stacking ensemble model.

Activation functions were tested in different combinations for three separate code blocks. The activation functions used in this study are ReLU (Rectified Linear Unit), Leaky ReLU (LReLU), and Parametric ReLU (PReLU) [26]. Three separate outputs of two convolution blocks were combined with the concatenate layer and flattened with a flatten layer. Then, the hidden layer, batch normalization, and dropout layers with 128 neurons were added. The dropout rate is 0.5. The last layer for classification has 43 neurons, representing the number of classes. For all blocks, the max-pooling layer kernel is chosen as 2×2 . Details and layer information of the proposed architecture in this study are given in Table 2. In addition, the layer information image is shown in Figure 2. The detailed model with parameter information is given in Appendix A.

IV. EXPERIMENTAL RESULTS

This section will give information about the datasets used and the experimental results.

A. DATASET

The dataset used for training and testing the proposed model is GTSRB [27]. The data set consists of traffic sign images belonging to 43 classes. There are 39,209 images in the training set and 12,630 images in the test set. Image sizes vary from 15×15 to 250×250 . Example images for each class of the dataset are shown in Figure 3. When the images are examined, it is seen that the images were taken in different lighting, brightness, blurring, and angle conditions.

The number of images belonging to the classes in the data set does not show an equal distribution. The histogram representing the number of images belonging to each class is given in Figure 4.

B. PERFORMANCE EVALUATION

Python and Tensorflow framework were used for the experimental study. Operations were performed on Google Colab. First, preprocessing was applied to the data set. The dataset, which includes images with resolutions ranging from 15×15 to 250×250 , was resized to 60×60 . Image sharpening and local contrast enhancement were then applied. Data augmentation was applied to eliminate the difference in the number of images between the classes. The training dataset was split into 90% training set and 10% validation set. Adam optimization algorithm was used in the training phase

TABLE 6. GTSRB test results of different architectures.

Model name	Top-1 Accuracy (%)	Parameters	F1-Score	Precision	Recall
Proposed (Ensemble)	99.75	14.34 M	99.75	99.76	99.75
DeepThin [16] (Ensemble)	99.72	2.69×10^5	-	-	-
Student model [12]	99.61	7.32×10^5	-	-	-
[9]*	99.61	5.6 M	99.4	99.37	99.63
Single CNN with 3 STNs [10]	99.71	14.6 M	99.71	99.71	99.71
Human (average) [29]	98.84	-	-	-	-
Human (best individual) [29]	99.22	-	-	-	-
MSCNN [2]	99.17	1.43 M	-	-	-
MCDNN [7]	99.46	38.5 M	-	-	-
DHCNN [11]	99.33	6.26 M	-	-	-
[14]	99.21	1.15 M	-	-	-
HLSGD [8]	99.65	23.2 M	-	-	-
Hybrid-TSR [13]*	99.28	4 M	99.33	99.24	99.36
[16]	99.56	-	-	-	-
[17]	99.60	-	-	-	-
[18]	98.84	-	-	-	-
[17]	99.38	0.44 M	-	-	-
[18]	99.20	0.8 M	-	-	-
[13]	99.27	-	99.94	99.97	99.93
[22]	98.84	-	98.84	98.84	98.84
Proposed (Kernel 1)	99.62	7.15 M	99.62	99.63	99.62
Proposed (Kernel 2)	99.65	7.15 M	99.65	99.66	99.65

*F₁ score, precision, and recall values are calculated from the table in the study.

of our deep learning model and *categorical_crossentropy* is selected as the loss function. The batch size is 32, and the learning rate is 0.001. The ReduceLROnPlateau class in the Keras library was used to reduce the learning rate. The parameters used here are *factor* = 0.02, *patience* = 10, and *min_lr* = 0.0001. In other words, the learning rate is reduced when the validation loss stops decreasing for ten epochs during training. The new learning rate is determined by (3).

$$new_lr = lr \times factor \quad (3)$$

where *lr* denotes the current learning rate, and *new_lr* denotes the learning rate to be used from now on. The factor specifies the learning rate update rate parameter of the ReduceLROnPlateau class used. For the proposed model, the GTSRB dataset is trained for 50 epochs. The number of parameters is 7,152,312.

1) ACTIVATION FUNCTION SELECTION

Initially, training and testing were conducted according to different activation functions for three hidden convolution layer blocks. A data set was created using *Kernel 1* in

the preprocessing stage. The test accuracy results from the training on this data set are given in Table 3. When the results were examined, it was seen that the highest test top-1 accuracy value was 99.62%, obtained by using ReLU and Leaky ReLU activation functions with 0.1 and 0.01 parameter values together.

2) DATA AUGMENTATION AND PREPROCESS SELECTION

Data augmentation was done with the ImageDataGenerator class in the Keras library. The test results were compared with and without data augmentation to the data set preprocessed with *Kernel 1*. In addition, training was conducted with the same parameters to understand whether preprocessing affects accuracy and the results were examined. Comparisons of test accuracy results according to preprocessing and data augmentation are given in Table 4. When the results are examined, it is seen that the accuracy increases with data augmentation and preprocessing, and thus a more accurate classification is made. In the next step, the data augmentation process was applied, and the processes were continued.

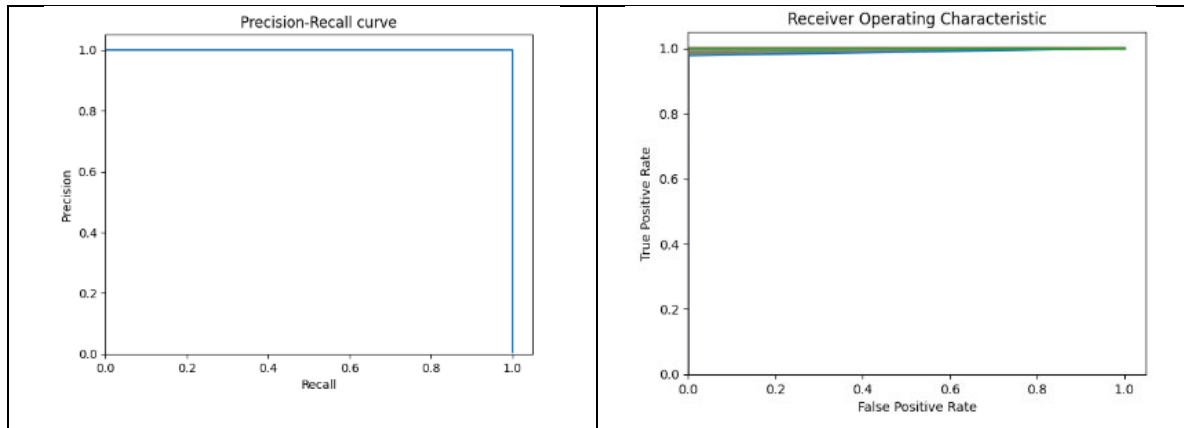


FIGURE 7. Left) Precision-recall curve Right) ROC curve.

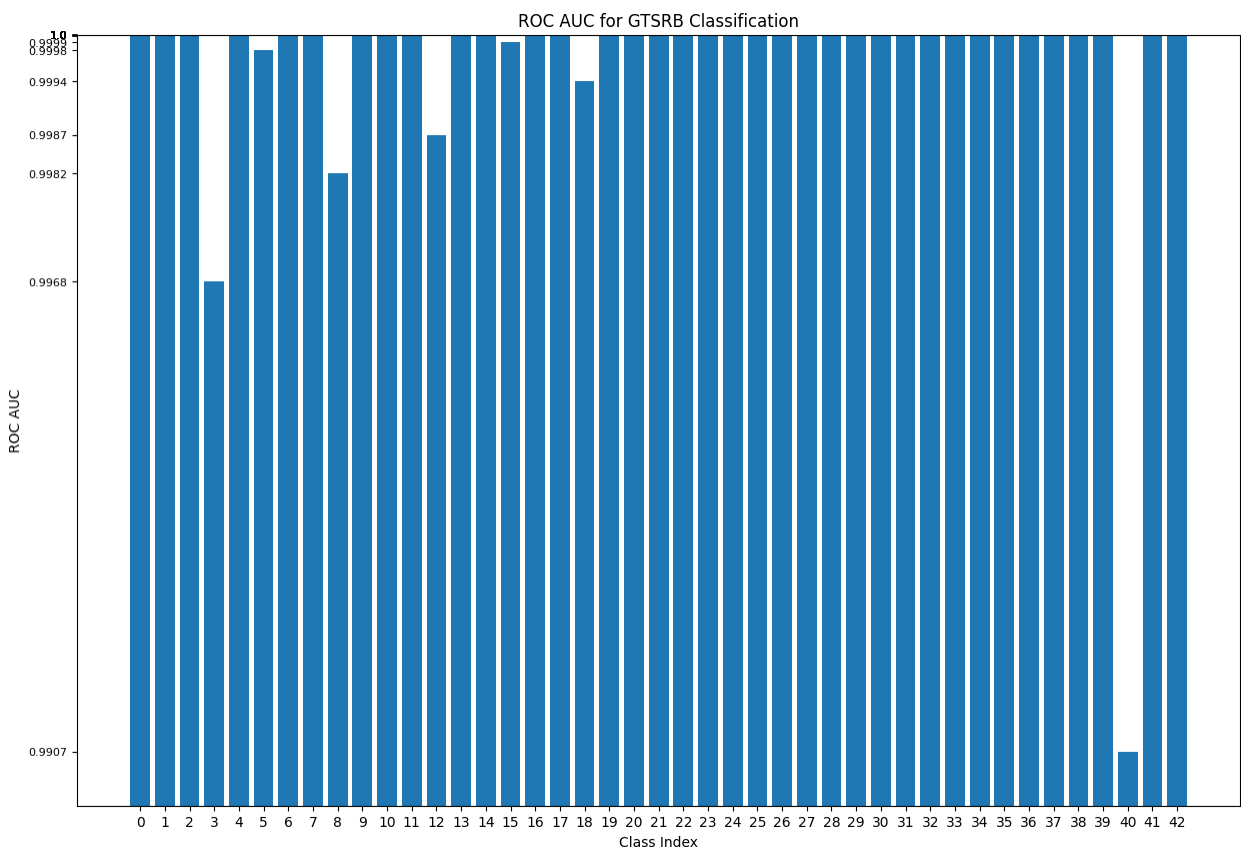


FIGURE 8. ROC AUC for GTSRB Classification (In order to see the difference between the fields, the y-axis is displayed between 0.99-1.0).

3) KERNEL SELECTION

After the activation function and data augmentation parameters were selected, training was also carried out with *Kernel 2*. In addition, after the preprocessing process with the use of *Kernel 1* and *Kernel 2* together, training was carried out. The test accuracy results after the procedures applied during the preprocessing are given in Table 5, and the loss and accuracy results obtained during the training are given in Figure 5. When the graphs are examined, learning decreases

after about 10 epochs. Although the accuracy values given in the graph are not seen because the range is limited, the learning continues up to 50 epochs. In cases where learning is slow, the learning rate is reduced so that learning continues quickly and effectively.

When the results were examined, it was seen that the highest test accuracy was obtained by giving the images obtained by filtering with *Kernel 1* to the existing model input.

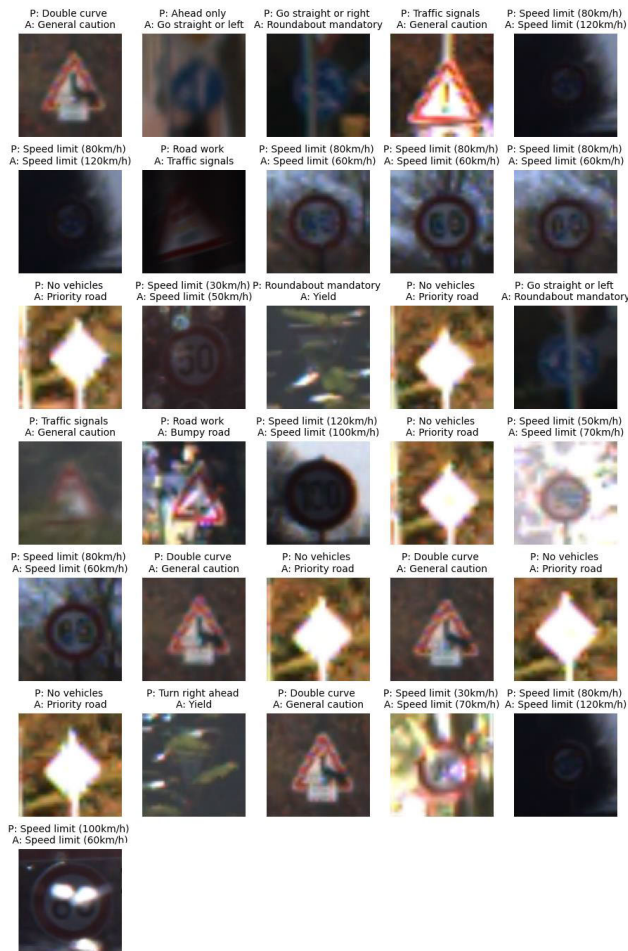


FIGURE 9. Misclassified images on GTSRB, A: Actual, P: Predict.

4) STACKING ENSEMBLE

The stacking ensemble model is designed to obtain better results by using previously trained subnets together. It is possible to combine different models with the stacking ensemble, as well as combine the same model for different inputs [28]. In this study, an ensemble model was created by combining the model results obtained because of filtering with two different kernel values used in the preprocessing stage. That is, the model parameters and hyper-parameters used are the same, but the input images are different. All layers in sub-models are set as non-trainable to preserve the parameter values in the existing sub-models. After adding existing models, a hidden layer for the meta-learner to interpret the inputs, batch normalization, dropout, and a final output layer for classification have been added. The fully connected layer has 300 neurons. The dropout rate is 0.5. The created stacking ensemble model details are given in Figure 6.

Although the total number of parameters in the model is 14,344,867, the number of trainable parameters is only 39,643 because we made the parameters in the sub-models non-trainable. The model was trained with 20 epochs.

Compile options are the same as in the sub-models. Accordingly, the model test accuracy result was 99.75%.

The current classification studies for the GTSRB dataset and the top-1 accuracy and number of parameter comparisons of the method proposed in this paper are given in Table 6. When the results are examined, it is seen that higher performance is achieved with the proposed stacking ensemble model. The accuracy value of the proposed model is higher when compared to studies with a low number of parameters. In addition, when compared with [7], [8], and [11] studies, it is seen that both the number of parameters and the accuracy value are higher, even when the results of the proposed method are compared on a single kernel basis.

Precision – recall curve graph is given in Figure 7-left to evaluate model results over precision - recall metrics. In addition, results are provided with the Receiver Operating Characteristic (ROC) curve to assess the classification performance of the model. The ROC curve are given in Figure 8-right. In addition, the Area Under the Curve (AUC) of the ROC on GTSRB Figure 8. ROC AUC average score is 0.9996.

5) MISCLASSIFIED IMAGES

Misclassified images were examined with the stacking ensemble model. Of the 12,630 images, only 31 of them could not predict the correct class. These images are shown in Figure 9, with the actual and predicted class names indicated at the top. When the images are examined, there are images that are indistinguishable even with the human eye, highly degraded, and have illumination problems. It can already be seen in the comparison chart that it produces higher accuracy when compared to human (best individual) results.

It is also possible to see the results numerically with the confusion matrix (Figure 10). When the confusion matrix is examined, it is seen that the images labeled 3 and 18 are more misclassified than the others. However, misclassified images were often incorrectly predicted to the same class. This may be because the predicted and real images are similar in shape. In the preprocessing and feature extraction phase, the shape can be considered to come to the forefront. In addition, precision, recall, and F1-score results were calculated on the test images for each class. The results are given in Table 7. The support column represents the total number of test images for each class.

6) EXPLAINABILITY

Post-hoc explanatory approaches are used to examine and illustrate the outputs of CNN layers that try to boost comprehensibility in order to get rid of the inherent lack of explainability in CNNs. Grad-CAM produces heatmaps that show the areas that have a favorable influence on expected output. Grad-CAM locates regions that contribute to the estimated output using spatial data from convolutional layers. Following the flattening layer and completely connected layers, this spatial information is lost. As a result, these crucial regions are defined using gradients in the convolution

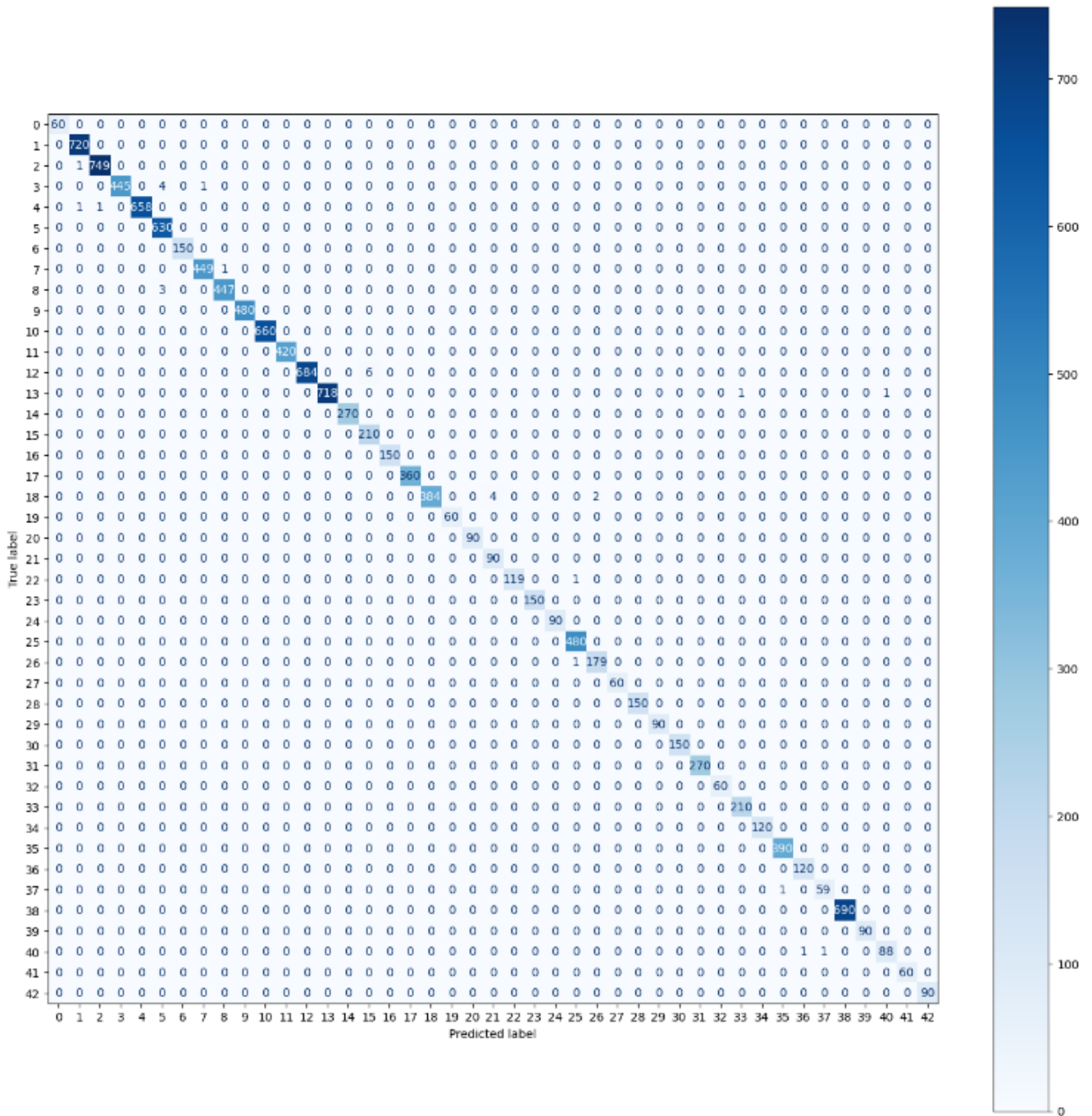


FIGURE 10. Confusion matrix of proposed ensemble model on GTSRB.

layer [30]. Grad-CAM was selected as the method in this investigation to interpret the CNN model’s predictions. Grad-CAM’s efficacy was evaluated using a proposed model and enhanced visualization outcomes.

Figure 11 shows the visualization outcomes produced by Grad-CAM. The original photos, the Grad-CAM method’s heatmaps, and the visualization outcomes derived by superimposing the original image onto the heatmap are all shown in the figure. Different colors are used to indicate the importance

of individual pixels in the classification results, representing how much consideration each pixel received from the CNN-based classification model.

7) UNCERTAINTY EVALUATION

In various machine learning problems, deterministic neural networks have demonstrated their ability to learn effective predictors. However, the traditional approach to minimizing prediction loss during network training misses the model’s

TABLE 7. GTSRB test results by data augmentation and preprocessing.

Class name	Precision	Recall	F1-Score	Support	Class name	Precision	Recall	F1-Score	Support
Speed limit (20km/h)	1.0000	1.0000	1.0000	60	Bumpy road	1.0000	0.9917	0.9958	120
Speed limit (30km/h)	0.9972	1.0000	0.9986	720	Slippery road	1.0000	1.0000	1.0000	150
Speed limit (50km/h)	0.9987	0.9987	0.9987	750	Road narrows on the right	1.0000	1.0000	1.0000	90
Speed limit (60km/h)	1.0000	0.9889	0.9944	450	Road work	0.9959	1.0000	0.9979	480
Speed limit (70km/h)	1.0000	0.9970	0.9985	660	Traffic signals	0.9890	0.9944	0.9917	180
Speed limit (80km/h)	0.9890	1.0000	0.9945	630	Pedestrians	1.0000	1.0000	1.0000	60
End of speed limit (80km/h)	1.0000	1.0000	1.0000	150	Children crossing	1.0000	1.0000	1.0000	150
Speed limit (100km/h)	0.9978	0.9978	0.9978	450	Bicycles crossing	1.0000	1.0000	1.0000	90
Speed limit (120km/h)	0.9978	0.9933	0.9955	450	Beware of ice/snow	1.0000	1.0000	1.0000	150
No passing	1.0000	1.0000	1.0000	480	Wild animals crossing	1.0000	1.0000	1.0000	270
No passing for vehicles over 3.5 metric tons	1.0000	1.0000	1.0000	660	End of all speed and passing limits	1.0000	1.0000	1.0000	60
Right-of-way at the next intersection	1.0000	1.0000	1.0000	420	Turn right ahead	0.9953	1.0000	0.9976	210
Priority road	1.0000	0.9913	0.9956	690	Turn left ahead	1.0000	1.0000	1.0000	120
Yield	1.0000	0.9972	0.9986	720	Ahead only	0.9974	1.0000	0.9987	390
Stop	1.0000	1.0000	1.0000	270	Go straight or right	0.9917	1.0000	0.9959	120
No vehicles	0.9722	1.0000	0.9859	210	Go straight or left	0.9833	0.9833	0.9833	60
Vehicles over 3.5 metric tons prohibited	1.0000	1.0000	1.0000	150	Keep right	1.0000	1.0000	1.0000	690
No entry	1.0000	1.0000	1.0000	360	Keep left	1.0000	1.0000	1.0000	90
General caution	1.0000	0.9846	0.9922	390	Roundabout mandatory	0.9888	0.9778	0.9832	90
Dangerous curve to the left	1.0000	1.0000	1.0000	60	End of no passing	1.0000	1.0000	1.0000	60
Dangerous curve to the right	1.0000	1.0000	1.0000	90	End of no passing by vehicles over 3.5 metric tons	1.0000	1.0000	1.0000	90
Double curve	0.9574	1.0000	0.9783	90					
					accuracy			0.9975	12630
					macro avg	0.9965	0.9976	0.9970	12630
					weighted avg	0.9976	0.9975	0.9975	12630

reliance on its predictions. Unlike Bayesian neural networks that indirectly infer prediction uncertainty through weight uncertainties, our study used the method that explicitly models prediction reliability using the subjective logic theory proposed in the [31] study. Here, a Dirichlet distribution is presented to represent class probabilities, which treat the predictions of the neural network as subjective opinions.

The resulting estimator for multiclass classification is represented by another Dirichlet distribution with parameters determined by the continuous output of the neural network.

In our study, a sample classroom image was rotated 180 degrees in total with a 10-degree angle to determine which classroom image the model predicted. The model

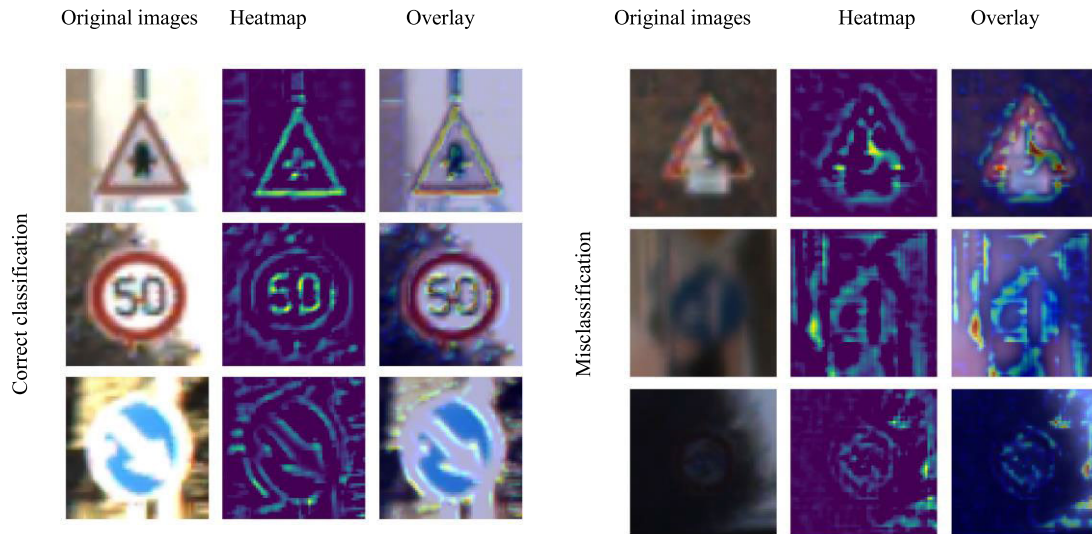


FIGURE 11. Sample image results obtained with Grad-CAM.

TABLE 8. BTSC test results of different architectures.

Model name	Top-1 Accuracy (%)	Parameters	F1-Score	Precision	Recall
Proposed (Ensemble)	99.33	14.34 M	99.38	99.48	99.33
DeepThin [16] (Ensemble)	99.29	2.69×10^5	-	-	-
[9]	92.12	5.6 M	-	-	-
[10]	98.87	14.6 M	98.86	98.95	98.87
[11]	99.17	6.26 M	-	-	-
[12]	99.13	0.8 M	-	-	-
[15]	99.02	1.15 M	-	-	-
[13]	98.97	-	-	-	-
Proposed (Kernel 1)	99.21	7.15 M	99.29	99.40	99.21
Proposed (Kernel 2)	99.17	7.15 M	99.20	99.31	99.17

used here is the one in which a single kernel is given to the input while the image is rotated. The model is retrained with *Kernel 1* to estimate the uncertainty with the presented loss function. The class label, probability, and uncertainty value estimated by the model at each angle change on the sample images are given in Figure 12. When the results are examined, it is seen that the uncertainty value increases when an incorrect estimation is made due to rotation.

8) SAFETY MONITORING

Unique difficulties are presented by the expanding use of artificial intelligence (AI) and data-driven decision-making systems in autonomous cars. Operating in dynamic environments, autonomous vehicles are prone to encountering unknown observations due to factors such as limited training data, distributional shifts, and cyber-security threats. To ensure safer autonomous driving and prevent

catastrophic accidents, it is crucial for AI algorithms to make dependable decisions by improving their understanding of the environment. In this study, SafeML-II, an approach designed in [32], is used to increase the security of machine learning. SafeML-II provides a model-agnostic solution adaptable to multiple machine learning and deep learning classifiers by leveraging empirical cumulative distribution function (ECDF)-based statistical distance measures and incorporating a human-in-the-loop technique. By using the GTSRB dataset, the approach’s performance is proved, highlighting its potential to increase the security and dependability of machine learning-based categorization systems in autonomous cars.

The concept of SafeML revolves around measuring statistical distances and predicting model accuracy in the absence of available tags. Having real-time prediction of accuracy can be critical for applications with security concerns. Statistical differences are discussed in this section. To do this, test data is

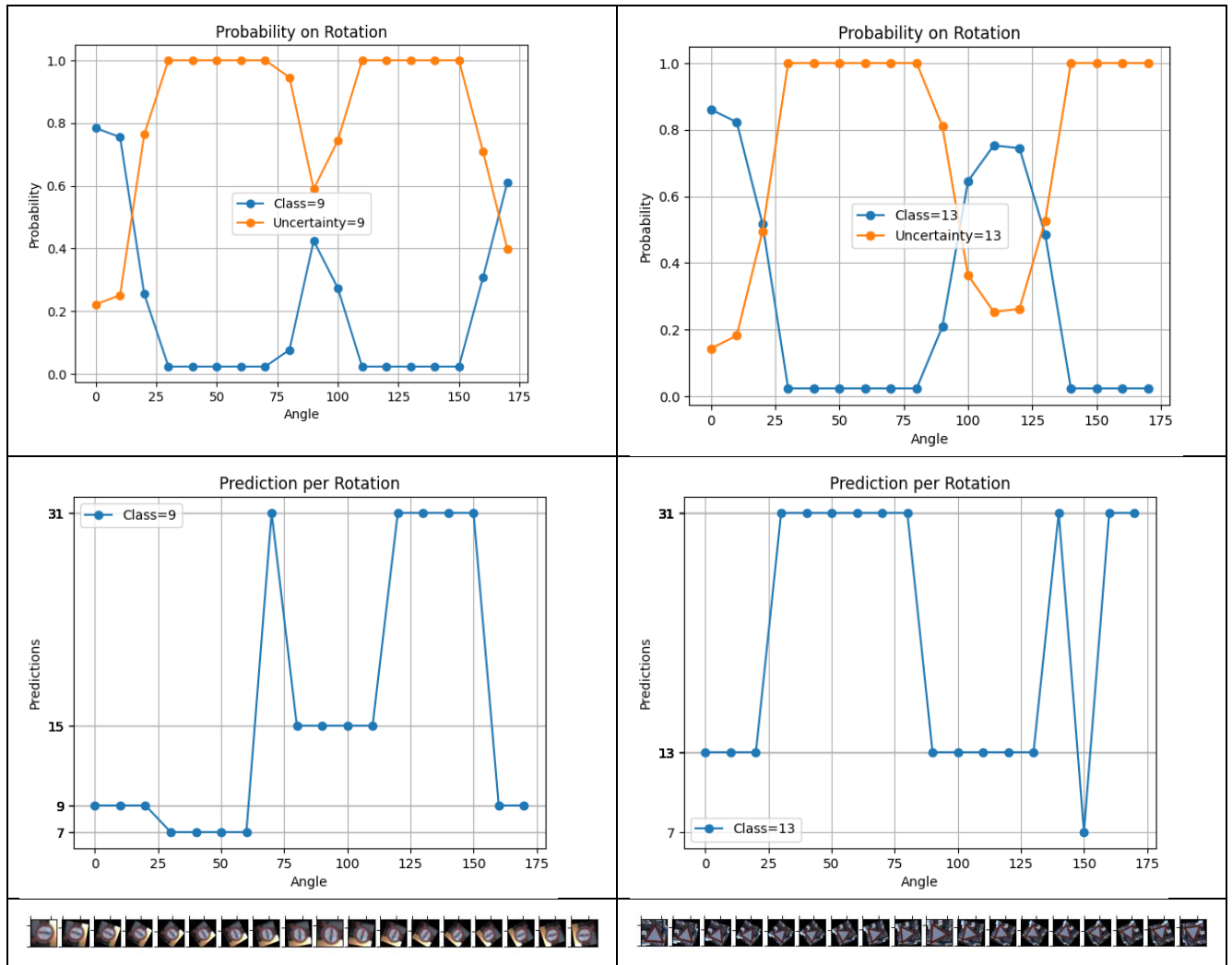


FIGURE 12. Results of classes 9 and 13 according to different angles between 0 and 180 degrees. Top: probability and uncertainty on rotation, Middle: class prediction per rotation, Bottom: Example image of rotation.

determined where the actual labels do not match the predicted labels. For example, when dealing with label = 3, reliable training images (actually labeled 3 and estimated as 3) are compared to images that were misclassified (for example, actually labeled 3 but classified as 5) when they should have labeled 3. Statistical parametric mapping attempts to identify statistical factors that contribute to our CNN model’s inaccurate decisions. An example application is given in Figure 13. Here, the first line shows the Statistical Parametric Mapping between the two 8 VOLUME XX, 2017 images mentioned, the second line shows the correctly labeled image and the third line shows the wrong-labeled image according to the R, G, and B color channels, respectively. In the last line, a Wasserstein distance (WD) measure is presented for the real image with label 3. Here, an ECDF-based distance measurement is employed, considering the p-value, to ensure that the results are more reliable and less noisy when used for the purpose of statistical interpretability. It is evident that SafeML-II exhibits a statistically sound distance

representation and effectively captures the foreground areas without including background signals.

9) MIGRATION TEST

A performance comparison was made using the BTSC [33] data set, which is also extensively used in the literature, in addition to the GTSRB data set to assess the model’s resilience. There are 2534 traffic sign images in the test, and 4591 traffic sign images in the train of the BTSC dataset used for classification. Image resolution values range from 11×10 to 562×438 [11]. There are 62 different classes of traffic signs. The photos in the collection also have other flaws, including occlusion and illumination. Due to the high number of classes and the few images in the dataset, training is, therefore, more challenging than the GTSRB dataset. Figure 14 contains sample photos for each class in the dataset.

With the proposed method trained with the GTSRB dataset, training was carried out by applying transfer learning. Here,

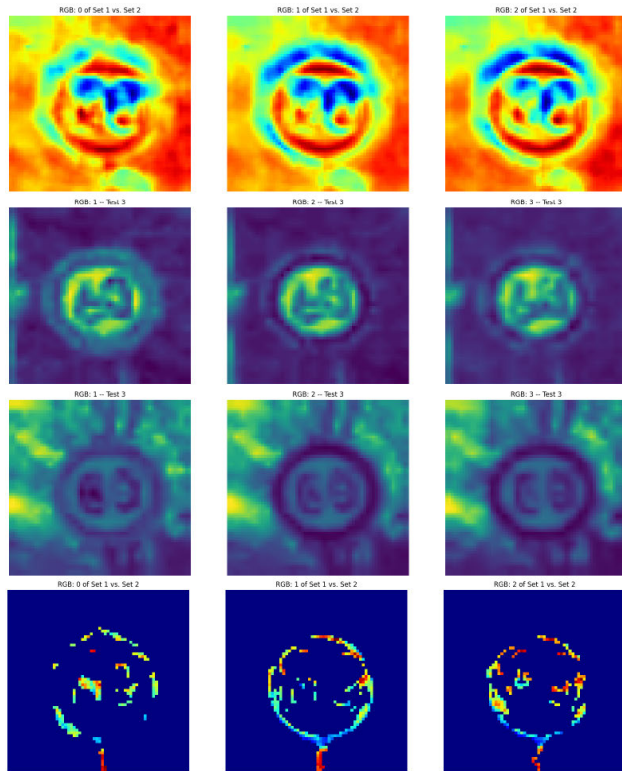


FIGURE 13. Line 1: Statistical Parametric Mapping, line 2: correctly labeled image Line 3: incorrectly labeled image, Line 4: Wasserstein distance measurement. (Columns represent R, G, B color channels in order.)

the layers up to the classification stage are non-trainable, and pre-trained and obtained weights are used. The last six layers, including classification as a similar model, were trained, and only the last class layer was modified to represent 62 classes. With the single model, training was carried out with 100 epochs for *Kernel 1* and *Kernel 2* inputs. Other parameters remained the same as those used in the GTSRB dataset. Then, the models trained with *Kernel 1* and *Kernel 2* were trained by combining them with the stacking ensemble model, similar to GTSRB. The number of epochs for the stacking ensemble model is 50. The results of all test operations performed with the proposed models and the results of different architectures on the BTSC dataset are shown in Table 8. When the results are examined, it is seen that the recommended stacking ensemble model has reached a high accuracy of 99.33%. Thus, it has been proven that the proposed model has high performance with different datasets, and its robustness test has been carried out.

In the test dataset, a total of 17 images were misclassified with the ensemble method. Misclassified images are given in Figure 15. When these images are examined, it is seen that there are slight differences in similar shape and appearance between the similarly predicted class and the real labeled class. For example, classroom images with labels between 45-50 often express similar parking situations.



FIGURE 14. Example images for each class of the dataset.



FIGURE 15. Misclassified images on BTSC, A: Actual, P: Predict.

V. CONCLUSION

In this study, a new deep and wide neural network model for traffic sign recognition is proposed. Input images given to the model are preprocessed using edge detection and image enhancement. Thus, more features are obtained with fewer parameters. It is aimed to provide faster training by resizing the images as 60×60 . It is ensured that the activation function used in the model is chosen to increase the accuracy. In addition, the learning rate was adjusted

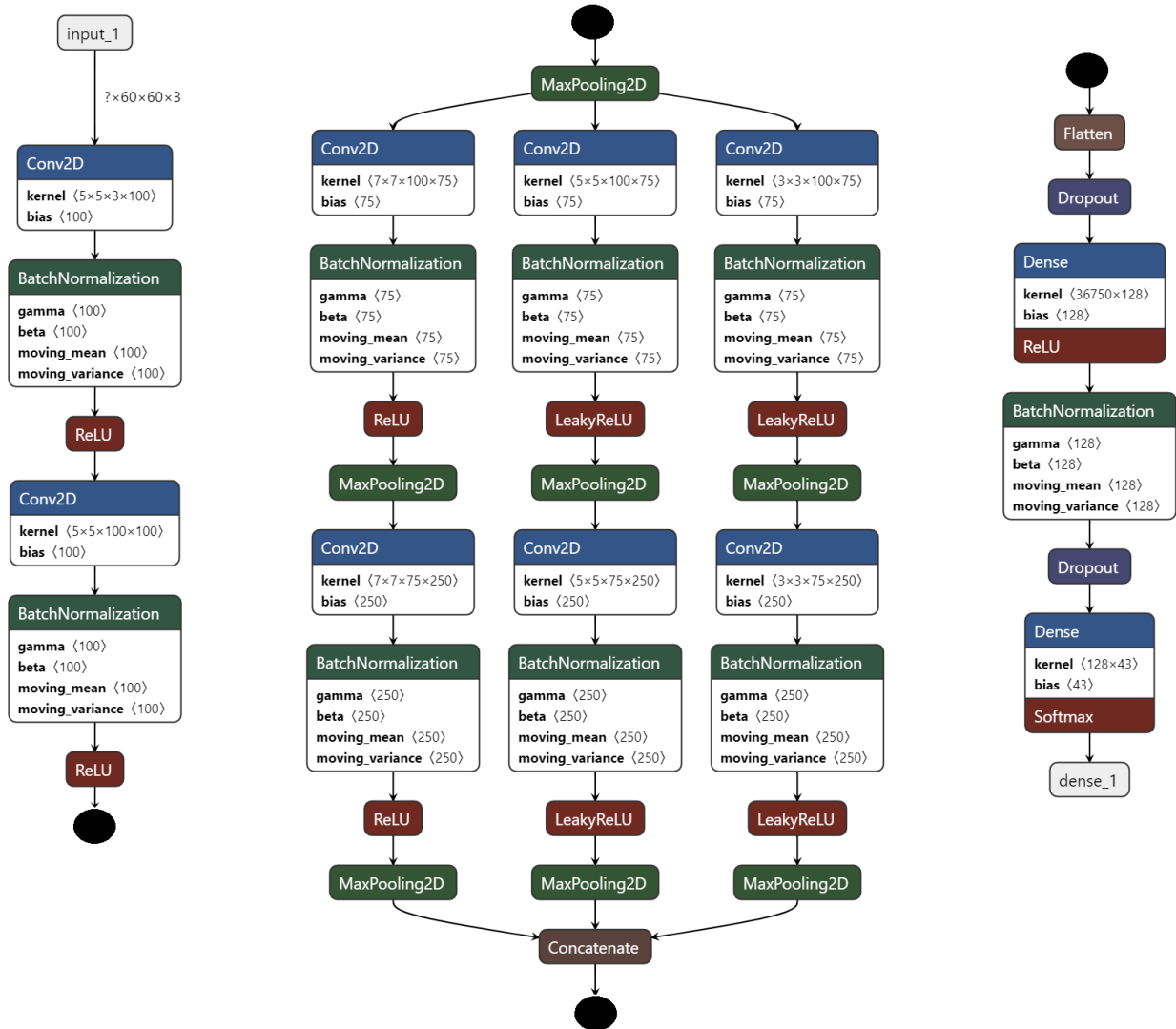


FIGURE 16. The proposed detailed model with parameter information.

during the training so that the model could learn better. Additively, the models trained with images whose input image is subjected to different preprocessing processes are turned into stacking ensemble models, resulting in better performance. The stacking ensemble model achieves a very high performance of 99.75% on the GTSRB benchmark. The number of parameters of the model trained on the dataset with a single preprocessing step is 7.15 M. In the stacking ensemble model, the models trained on the dataset obtained with two different preprocessing stages were combined and additional layers were added for classification. The total number of parameters in this model is 14.34 M. However, the number of trainable parameters in the ensemble model is only 39,643, since the parameters in the previously trained models with transfer learning are not trainable. As a result, a new hybrid network providing high performance was designed by using both image enhancement methods and a deep learning approach. Future studies will focus on how to reduce the number of parameters without reducing the accuracy value. Different new generation learning approaches such as

attention and residual network structure will be tried. Metrics such as F1-score and accuracy can be updated to achieve higher performance. In addition, the robustness of the model will be tested by training on different datasets. Preprocessing steps that will produce better results for different weather conditions and distorted images will be investigated.

APPENDIX A

See Fig 16.

REFERENCES

- [1] J. Piao and M. McDonald, “Advanced driver assistance systems from autonomous to cooperative approach,” *Transp. Rev.*, vol. 28, no. 5, pp. 659–684, Sep. 2008.
- [2] P. Sermanet and Y. LeCun, “Traffic sign recognition with multi-scale convolutional networks,” in *Proc. Int. Joint Conf. Neural Netw.*, Jul. 2011, pp. 2809–2813.
- [3] M.-Y. Fu and Y.-S. Huang, “A survey of traffic sign recognition,” in *Proc. Int. Conf. Wavelet Anal. Pattern Recognit.*, Jul. 2010, pp. 119–124.
- [4] S. B. Wali, M. A. Abdullah, M. A. Hannan, A. Hussain, S. A. Samad, P. J. Ker, and M. B. Mansor, “Vision-based traffic sign detection and recognition systems: Current trends and challenges,” *Sensors*, vol. 19, no. 9, p. 2093, May 2019.

- [5] Y. Li, L. Ma, Y. Huang, and J. Li, "Segment-based traffic sign detection from mobile laser scanning data," in *Proc. IEEE Int. Geosci. Remote Sens. Symp.*, Jul. 2018, pp. 4607–4610.
- [6] A. Shustanov and P. Yakimov, "CNN design for real-time traffic sign recognition," *Proc. Eng.*, vol. 201, pp. 718–725, Jan. 2017.
- [7] D. Cireşan, U. Meier, J. Masci, and J. Schmidhuber, "Multi-column deep neural network for traffic sign classification," *Neural Netw.*, vol. 32, pp. 333–338, Aug. 2012.
- [8] J. Jin, K. Fu, and C. Zhang, "Traffic sign recognition with Hinge loss trained convolutional neural networks," *IEEE Trans. Intell. Transp. Syst.*, vol. 15, no. 5, pp. 1991–2000, Oct. 2014.
- [9] H. H. Aghdam, E. J. Heravi, and D. Puig, "A practical and highly optimized convolutional neural network for classifying traffic signs in real-time," *Int. J. Comput. Vis.*, vol. 122, no. 2, pp. 246–269, Apr. 2017.
- [10] Á. Arcos-García, J. A. Álvarez-García, and L. M. Soria-Morillo, "Deep neural network for traffic sign recognition systems: An analysis of spatial transformers and stochastic optimisation methods," *Neural Netw.*, vol. 99, pp. 158–165, Mar. 2018.
- [11] S. Saha, S. A. Kamran, and A. S. Sabbir, "Total recall: Understanding traffic signs using deep convolutional neural network," in *Proc. 21st Int. Conf. Comput. Inf. Technol. (ICCIT)*, Dec. 2018, pp. 1–6.
- [12] J. Zhang, W. Wang, C. Lu, J. Wang, and A. K. Sangaiah, "Lightweight deep network for traffic sign classification," *Ann. Telecommun.*, vol. 75, nos. 7–8, pp. 369–379, Aug. 2020.
- [13] O. T. Nartey, G. Yang, S. K. Asare, J. Wu, and L. N. Frempong, "Robust semi-supervised traffic sign recognition via self-training and weakly-supervised learning," *Sensors*, vol. 20, no. 9, p. 2684, May 2020.
- [14] K. Bayouh, F. Hamdaoui, and A. Mtibaa, "Transfer learning based hybrid 2D-3D CNN for traffic sign recognition and semantic road detection applied in advanced driver assistance systems," *Appl. Intell.*, vol. 51, pp. 124–142, Aug. 2021.
- [15] Z. Bi, L. Yu, H. Gao, P. Zhou, and H. Yao, "Improved VGG model-based efficient traffic sign recognition for safe driving in 5G scenarios," *Int. J. Mach. Learn. Cybern.*, vol. 12, no. 11, pp. 3069–3080, Nov. 2021.
- [16] W. A. Haque, S. Arefin, A. Shihavuddin, and M. A. Hasan, "DeepThin: A novel lightweight CNN architecture for traffic sign recognition without GPU requirements," *Exp. Syst. Appl.*, vol. 168, Apr. 2021, Art. no. 114481.
- [17] H.-F. Fang, J. Cao, and Z.-Y. Li, "A small network MiconNet-BF of traffic sign classification," *Comput. Intell. Neurosci.*, vol. 2022, pp. 1–10, Mar. 2022.
- [18] N. Youssouf, "Traffic sign classification using CNN and detection using faster-RCNN and YOLOV4," *Heliyon*, vol. 8, no. 12, Dec. 2022, Art. no. e11792.
- [19] V. Patel, S. Shukla, S. Shrivastava, and M. Gyanchandani, "Regularized CNN for traffic sign recognition," in *Proc. Int. Conf. Smart Technol. Syst. Next Gener. Comput. (ICSTSN)*, Mar. 2022, pp. 1–5.
- [20] X. Zhang and W. Huang, "Traffic sign recognition algorithm based on convolutional neural network," in *Proc. 3rd Int. Conf. Comput. Vis., Image Deep Learn. Int. Conf. Comput. Eng. Appl. (CVIDL ICCEA)*, May 2022, pp. 1117–1119.
- [21] Y. Zheng and W. Jiang, "Evaluation of vision transformers for traffic sign classification," *Wireless Commun. Mobile Comput.*, vol. 2022, pp. 1–14, Jun. 2022.
- [22] X. R. Lim, C. P. Lee, K. M. Lim, and T. S. Ong, "Enhanced traffic sign recognition with ensemble learning," *J. Sensor Actuator Netw.*, vol. 12, no. 2, p. 33, Apr. 2023.
- [23] K. Kaur, N. Jindal, and K. Singh, "Fractional derivative based unsharp masking approach for enhancement of digital images," *Multimedia Tools Appl.*, vol. 80, pp. 3645–3679, Sep. 2021.
- [24] A. Polesel, G. Ramponi, and V. J. Mathews, "Image enhancement via adaptive unsharp masking," *IEEE Trans. Image Process.*, vol. 9, no. 3, pp. 505–510, Mar. 2000.
- [25] G. Yildiz and B. Dızdaroglu, "Convolutional neural network for traffic sign recognition based on color space," in *Proc. 2nd Int. Informat. Softw. Eng. Conf. (IISEC)*, Dec. 2021, pp. 1–5.
- [26] K. He, X. Zhang, S. Ren, and J. Sun, "Delving deep into rectifiers: Surpassing human-level performance on ImageNet classification," in *Proc. IEEE Int. Conf. Comput. Vis. (ICCV)*, Dec. 2015, pp. 1026–1034.
- [27] J. Stallkamp, M. Schlipsing, J. Salmen, and C. Igel, "The German traffic sign recognition benchmark: A multi-class classification competition," in *Proc. Int. Joint Conf. Neural Netw.*, Jul. 2011, pp. 1453–1460.
- [28] M. Kim, M. Lee, M. An, and H. Lee, "Effective automatic defect classification process based on CNN with stacking ensemble model for TFT-LCD panel," *J. Intell. Manuf.*, vol. 31, no. 5, pp. 1165–1174, Jun. 2020.
- [29] J. Stallkamp, M. Schlipsing, J. Salmen, and C. Igel, "Man vs. computer: Benchmarking machine learning algorithms for traffic sign recognition," *Neural Netw.*, vol. 32, pp. 323–332, Aug. 2012.
- [30] N. I. Papandrianos, A. Feleki, S. Moustakidis, E. I. Papageorgiou, I. D. Apostolopoulos, and D. J. Apostolopoulos, "An explainable classification method of SPECT myocardial perfusion images in nuclear cardiology using deep learning and grad-CAM," *Appl. Sci.*, vol. 12, no. 15, p. 7592, Jul. 2022.
- [31] M. Sensoy, L. Kaplan, and M. Kandemir, "Evidential deep learning to quantify classification uncertainty," in *Proc. Adv. Neural Inf. Process. Syst.*, vol. 31, 2018, pp. 1–11.
- [32] K. Aslansefat, S. Kabir, A. Abdullatif, V. Vasudevan, and Y. Papadopoulos, "Toward improving confidence in autonomous vehicle software: A study on traffic sign recognition systems," *Computer*, vol. 54, no. 8, pp. 66–76, Aug. 2021.
- [33] M. Mathias, R. Timofte, R. Benenson, and L. Van Gool, "Traffic sign recognition—How far are we from the solution?" in *Proc. Int. Joint Conf. Neural Netw. (IJCNN)*, Aug. 2013, pp. 1–8.



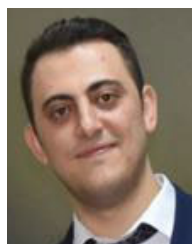
GÜLCAN YILDIZ received the B.S. and M.S. degrees in computer engineering from Ondokuz Mayıs University, Samsun, Turkey, in 2013 and 2017, respectively. She is currently pursuing the Ph.D. degree with the Department of Computer Engineering, Karadeniz Technical University, Trabzon, Turkey. She has been a Research Assistant with the Department of Computer Engineering, Ondokuz Mayıs University, since 2014. Her research interests include computer vision, deep learning, and image processing.



AHMET ULU received the B.S. degree in computer engineering from Ondokuz Mayıs University, Samsun, Turkey, in 2013, and the M.S. degree in computer engineering from Karadeniz Technical University, Trabzon, Turkey, in 2018, where he is currently pursuing the Ph.D. degree in computer engineering. He is currently a Research Assistant with the Department of Computer Engineering, Karadeniz Technical University. His research interests include computer vision, machine learning, and deep learning.



BEKİR DİZDAROĞLU received the bachelor's, master's, and Ph.D. degrees from the Department of Electrical and Electronics Engineering, Karadeniz Technical University, in 1994, 1998, and 2007, respectively. Since 2007, he has been a Faculty Member with the Department of Computer Engineering, Karadeniz Technical University. His research interests include image processing, computer vision, and deep learning.



DOĞAN YILDIZ received the B.S. degree in electrical–electronics engineering and mathematics from Fatih University, Istanbul, in 2013, and the M.S. and Ph.D. degrees in electrical–electronics engineering from Ondokuz Mayıs University, Samsun, in 2016 and 2021, respectively.

Since 2013, he has been a Research Assistant with the Electrical–Electronics Engineering Department, Ondokuz Mayıs University. His research interests include wireless sensor networks, localization, signal processing, telecommunication and applications, electromagnetic waves, and electromagnetic radiation measurements.

• • •