International Journal of Cognitive Informatics and Natural Intelligence
Volume 15 • Issue 4

Efficient Traffic Sign Recognition Using CLAHE-Based Image Enhancement and ResNet CNN Architectures

Utkarsh Dubey, Malaviya National Institute of Technology, Jaipur, India

Rahul Kumar Chaurasiya, Maulana Azad National Institute of Technology, Bhopal, India*

ABSTRACT

Recognition and classification of traffic signs and other numerous displays on the road are very crucial for autonomous driving, navigation, and safety systems on roads. Machine learning or deep learning methods are generally employed to develop a traffic sign recognition (TSR) system. This paper proposes a novel two-step TSR approach consisting of contrast limited adaptive histogram equalization (CLAHE)-based image enhancement and convolutional neural network (CNN) as multiclass classifier. Three CNN architectures (i.e., LeNet, VggNet, and ResNet) were employed for classification. All the methods were tested for classification of German traffic sign recognition benchmark (GTSRB) dataset. The experimental results presented in the paper endorse the capability of the proposed work. Based on experimental results, it has also been illustrated that the proposed novel architecture consisting of CLAHE-based image enhancement, and ResNet-based classifier has helped to obtain better classification accuracy as compared to other similar approaches.

KEYWORDS

Automation, Deep Learning, CLAHE, CNN Architectures, Convolution Neural Network, Histogram-Equalization, Image Equalization, Traffic Sign Recognition

INTRODUCTION

Traffic sign recognition (TSR) has a vital importance in autonomous driver assistant system (Stallkamp, Schlipsing, Salmen, & Igel, 2012). The detection and recognition of traffic signs and symbols from digital images have been an area for research over last few years. Two major tasks are generally involved in a typical TSR system: detecting the approximate sizes and locations of traffic signs on the road (traffic sign detection-TSD) and then categorizing those detected traffic sign's images into their respective classes (TSR). Traffic sign detection and recognition system (TSDR) is a driver supportive system may also be used to inform and warn the driver in difficult circumstances. This system has the ability to detect and recognize all traffic signs, even those signs that may be occluded or somewhat distorted.

TSD is the initial phase of any TSDR system. In this phase, candidate region of interests (ROIs), which is the area with most probability to have the regions of traffic signs, are detected. TSD involves

DOI: 10.4018/IJCINI.295811

*Corresponding Author

locating potential sign image regions from a natural scene image input. This initial stage is called the detection stage, in which a ROI-containing traffic sign is actually localized (Møgelmose, Trivedi & Moeslund, 2012; Gündüz, Kaplan & Günal, 2013). Traffic signs usually are of red, blue or white color and specific shapes like round, square, and triangular. These characteristics differentiate the traffic signs from other outdoor objects, which makes them appropriate to be computed by a computer vision system. This allows the TSDR system to distinguish traffic signs from the background scene (Yuan, Xiong, & Wang, 2017). Thus, the traditional detection method consists of either based on color or shape or a hybrid of both.

Traffic signs are displayed with standard shapes and attractive colors that appeals the human drivers' attention. Various machine learning (ML) and deep learning (DL) methods *viz.* support vector machines (SVM) (Agrawal & Chaurasiya, 2017; Do et al., 2017; Kiran, Prabhu, & Rajeev, 2009; Soni, Chaurasiya, & Agrawal, 2019), k-nearest neighbor (KNN) classifier (Sugiharto & Harjoko, 2016), fully convolutional network (FCN) (Y. Zhu et al., 2016), and convolutional neural network (CNN) (Aghdam, Heravi, & Puig, 2016; Farag & Saleh, 2018; Z. Zhu et al., 2016) have been employed for classification in TSR system.

There has been a lot of research in the field of TSR for the past few years. Jin et al. have designed a CNN-based model achieving high accuracy using a hinge loss stochastic gradient descent method (Jin, Kun Fu, & Changshui, 2014). A multi-column deep neural network for traffic sign classification was proposed by CireşAn et al., where they proclaimed better results than human-recognition-rate (CireşAn et al., 2012),. A fast template matching technique was proposed by Torresen et al. (Torresen, Bakke, & Sekanina, 2004). Li et al. applied adaboost learning with five classical Haar wavelets and four histogram of oriented gradient (HoG) features (Li, Pankanti, & Guan, 2010). Youssef et al. proposed a combination of color segmentation, HoG, and CNN in their TSDR system to achieve improved classification accuracy and computational speed (Youssef et al., 2016). Qian et al. used CNN to learn the discriminative feature of max pooling positions to classify traffic signs and obtained a comparable performance with the-state-of-the-art methods (Qian et al., 2016). Greenhalgh and Mirmehdi depicted a comparison between different classifying techniques namely SVM, MLP, HoG-based classifiers, and decision trees. They found that decision tree with average accuracy of 94.2% had the highest accuracy with lowest computational time (Greenhalgh and Mirmehdi, 2012).

German traffic sign recognition benchmark (GTSRB) dataset has been widely used by researches for testing their proposed approaches (Stallkamp, Schlipsing, Salmen, & Igel, 2011). Most of traffic sign images detected from GTSRB dataset are of very low quality. Hence, pre-processing and normalization of the images has helped to improve the quality of them. Methods used for image processing include gamma correction (Rahman, Rahman, Abdullah-Al-Wadud, Al-Quaderi, & Shoyaib, 2016), histogram equalization (Sepasian, Balachandran, & Mares, 2008), and contrast limited adaptive histogram equalization (CLAHE) (Sepasian et al., 2008).

This paper proposes a novel approach to employ LeNet, VggNet, and ResNet (Arcos-Garcia, Alvarez-Garcia, & Soria-Morillo, 2018; S. Zhou, Liang, Li, & Kim, 2018) architecture-based CNN in TSR. Image classification in the proposed model takes place through a CNN, a state-of-the-art deep learning model. The CNN comprises of an input and consecutively, an output layer, with some multiple concealed layers. There is a relay of convolutional layers convolving with a dot product in the hidden layers. Then comes the rectified linear unit (RELU) layer, which is actually an activation layer to get values only greater than a particular threshold. Some other convolutional layers like fully connected layers, normalization layers, and pooling layers have their input and output masked, thus known as hidden layers.

The aforementioned recent CNN-based architectures are preferred over traditional ML and classical DL algorithms in this work. This is motivated by the fact that in the recent research, these architectures have performed better in several similar application areas like natural language processing, image and video recognition, image classification, financial time series, recommender systems,[REMOVED HYPERLINK FIELD] and medical image analysis (Anwar et al., 2018; Sezer

& Ozbayoglu, 2018). Moreover, the TSR task consists of a multiclass classification problem. Hence, a classifier model capable of learning images from several classes should be preferred over traditional ML algorithms. It is also required that the image dataset used to train the model should consist of good quality images. Therefore, the proposed model first pre-processes the traffic sign images of GTSRB dataset using three CLAHE method. In order to show the effectiveness of the proposed model, gamma correction & histogram equalization methods were also employed for image enhancement. The model then classifies the pre-processed images using the three previously mentioned CNN architectures. To sum up, the three major contributions of this paper are as follows:

- First, the paper proposes to use CLAHE-based method for image preprocessing to improve the quality of the input images.
- Second, a novel approach consisting of CLAHE-based image enhancement & recent ResNet CNN architecture is utilized to achieve high recognition rate.
- The effectiveness of the proposed approach has been validated by comparing its performance by implementing other image enhancement (*viz.* gamma correction & histogram equalization) and classification methods (*viz.* LeNet, VggNet).

The paper is structured as follows: *Introduces* section the proposed model with the motivation. Related works are also discussed in the section. The materials and the details of the stages of TSR system are described in the section *Materials and Methods*. Section *Proposed Work* presents the proposed model, while Section *Experimental Results* deals with the experimentations and the corresponding results. *Discussion* section is dedicated to various qualitative and quantitative analysis and discussions. Finally, the summarized contribution and future works are described in the *Conclusion* section.

MATERIAL AND METHOD

We have used the GTSRB dataset to train and test the proposed approach. The GTSRB was introduced as a single-image detection assessment of image-based driver assistance (Stallkamp et al., 2011). It has the following features:

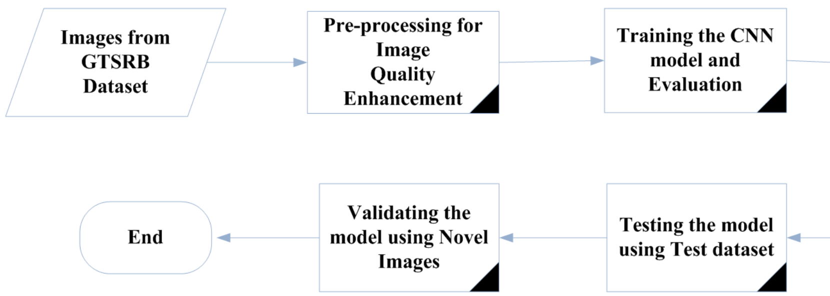
- A single-image detection problem.
- The dataset is divided into 3 categories: training, validation, and testing.
- An online assessment system with quick result and ranking of the results.

The pipeline architecture of the proposed TSR is depicted in Figure 1. The system has been split into several blocks. The input traffic sign images are first pre-processed for quality enhancement. wGamma correction-, histogram equalization-, and CLAHE-based methods were applied for this. The enhanced images were then used to train and test the three CNN models (*viz.* LeNet, VggNet, and ResNet). The performance of the proposed models were then evaluated on the novel images. The detailed descriptions of each block is given in the following sections.

Data Pre-Processing

Most of the images obtained from the dataset have a very low quality and thus we need to pre-process the images for improving their quality for accurate classification. The following operations were involved in the pre-processing step.

Figure 1. The pipeline architecture of the proposed TSR system.



Gray-Scaling

Greyscale images are those in which the value of each pixel is a sample representing only an *amount* of light, i.e., it has information about intensity only. The color images were converted to grayscale at this stage. This step also reduces the dimension as each pixel now has only one intensity level instead of three values for red, green, and blue (R, G, and B).

Normalization

The range of pixel intensity values were normalized using this process. The image data was normalized so that the data has mean zero and equal variance

Equalization

The pixels of the images have to be equalized i.e., the contrast adjustment of the images to increase the quality of images. The equalization process followed by the proposed system is described in detail in section IV.

In general, we shuffle the dataset to increase randomness and variety in training dataset, in order for the model to be more stable and accurate.

Model Training and Evaluation

In this phase, we have designed and implemented a deep learning model that learned to recognize traffic signs from GTSRB dataset. As the CNN were used to classify the images in this dataset, a basic structure and working of CNN is described in this subsection.

Figure 2. (a) The colored image obtained from GTSRB, (b): The corresponding greyscaled image.

CNN

The proposed method adopts an architecture of the CNN model that has been proved to be effective for TSR system (Bouti, Mahraz, Riffi, & Tairi, 2019). A conventional CNN usually comprises of a convolution stages, fully connected layers, and final soft max layers. The CNN model may consist of many convolution layers according to the architecture used. ReLU is then deployed as the activation function for convolutional layers and full connection layers. Finally the multiple layer perceptron (MLP), which is the final pooling layer, containing full connection layers to give the output layer. Schematic illustration of CNN is given in figure 3.

Testing The Model Using The Test Dataset

As the GTSRB dataset was randomly divided into training and testing set, the test images from dataset were used to measure the accuracy of the model. As the TSR is a multiclass classification problem, we have also generated a confusion matrix after testing the model. A confusion matrix allows envision of the efficiency of an algorithm which explains the implementation of a classifier on a data set for which the correct output is given to us pre-hand (Tan, Steinbach, & Kumar, 2016).

Validating the Model Using Novel Images

In this step, we employed the trained and tested TSR system to predict traffic signs of five randomly chosen images of German traffic signs from the web. These completely new images were not used for training or testing of the model.

PROPOSED WORK

In this work, we have proposed a two-step TSR approach. In first step, images are preprocessed to enhance the quality. This step is followed by CNN-based multiclass classification system. As described earlier, the pre-processing step of the proposed system consists of several operations *viz.* grey-scaling, normalization, and equalization. The first novelty of the proposed model lies with the fact that we have employed CLAHE-based image equalization method before applying CNN on the traffic sign images. However, the other commonly used gamma correction and histogram equalization methods were also applied. These two methods were applied to show that the former approach helps to get better classification accuracy. The equalization methods are described in section 3.1.

Figure 3. Schematic diagram of CNN Architecture.

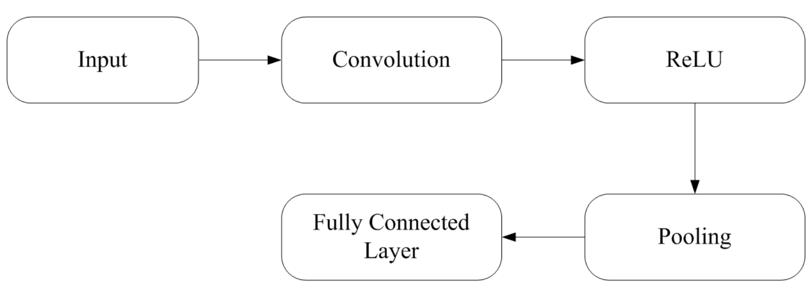


Image Equalization

Gamma Correction

Usually known as gamma, is a non-linear method to encode or decode pixel values or luminance values in digital images. The mathematical formulation as a power-law expression is given in the Equation (1):

$$V_{out} = A V_{in}^{\gamma} \tag{1}$$

where, the input intensity value V_{in} is raised to the power γ and multiplied with a constant A to give output values V_{out}. If $\gamma < 1$, it is called an *encoding gamma,* in this case the encoding with contracting power-non linearity is known as gamma compression. Conversely if $\gamma > 1$, it is known as *decoding gamma,* also called gamma expansion.

In gamma correction, γ is the measure of slope of the transformation function. With increase in value of γ, the transformation curve becomes steeper. As the curve becomes steeper, the corresponding intensities values are more spread, increasing the contrast. In adaptive gamma correction, the γ for low-contrast images is calculated as in Equation (2):

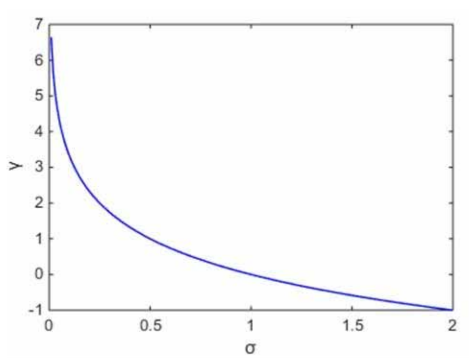
$$\gamma = -log_2\left(\sigma\right) \tag{2}$$

Histogram Equalization

Some images have a very poor contrast value, i.e., the objects in the image are less distinguishable from the background. Histogram equalization is a method to increase the contrast value, and make the details in the image more identifiable thus increasing the contrast value. Histogram equalization help to redistribute the intensities. It makes the increment of image's contrast possible by efficiently laying out the most frequently used intensity values and allows areas with lesser contrast values to get a higher one.

Histogram equalization is applied in image's backgrounds and foregrounds which can be either dark or bright or a combination of the two. Theoretically, the original histogram can be easily recovered from the histogram equalization function by a non-computationally intensive calculation. Drawback of this method is that sometimes it increases the background noise too, reducing the actual signal.

Figure 4. Plot of γ with respect to σ using the formula of Equation (2)



Suppose the given image is f, shown as matrix $m_r \times m_c$ representing the integer pixel intensities with values between 0 to $L-1$. Where, L is the number of possible intensity values, which is usually 256 for 8 bit images. Let p denote the normalized histogram of f, where each value denotes possible integer pixel intensities. We find the value of as in Equation(3):

$$p_n = \frac{number\ of\ pixels\ with\ intensity\ n}{Total\ number\ of\ pixels} \tag{3}$$

where, $n=0,1 \ldots L-1$. Further, the histogram equalized image g is described as in Equation (4):

$$g_{i,j} = floor(\left(L-1\right)\sum_{n=0}^{f_{i,j}} p_n) \tag{4}$$

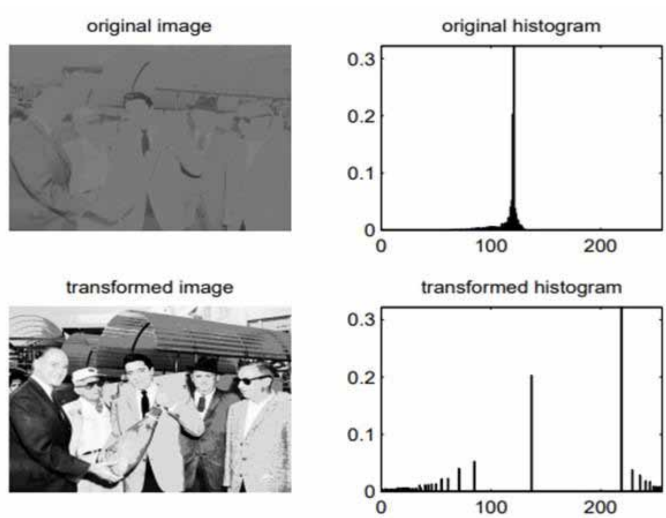
If p_x is the probability density function (PDF) of f and T is the cumulative distributive function (CDF) of X. Then output Y is defined by $T\left(X\right)$ and is uniformly distributed over $\left[0, L-1\right]$. The mathematical formulation for Y is given in Equation (5):

$$Y = T\left(X\right) = \left(L-1\right)\int_0^X p_X\left(x\right)dx \tag{5}$$

CLAHE

It is a technique for enhancing the quality of image by improving its contrast and limiting the amplification. It is different from ordinary histogram equalization as in the adaptive method of histogram equalization, several histograms are computed for distinct sections of the image, and uses these histograms to redistribute the lightness values of the image.

Figure 5. The application of histogram equalization



In ordinary adaptive histogram equalization (AHE), the contrast in the near-constant regions of the image is over-amplified, amplifying this noise in the near-contrast regions. CLAHE is a variant of AHE reducing the problem of noise amplification by limiting contrast amplification.

In CLAHE, the transformation function's slope is given by the contrast amplification near the given pixel value proportional the histogram's value at that pixel value and to the slope of the neighborhood (CDF). The histogram is clipped at a certain value that is predefined before the computation of the CDF restricting the amplification, limiting slope of CDF and transformation function and thus reduce noise. The clip-limit depends on the normalization of the histogram and thereby on the size of the neighborhood region. The amplification factor is usually between 3 and 4. The extra clipped part is not removed but redistributed uniformly among the histogram bins.

The redistribution of the clipped area pushes some area again over the clip limit (as shown in the shaded region), which makes the clip limit effectively larger than the predefined one. Until the excess area is limited to be negligible, this redistribution keeps on repeating. The aforementioned image equalization techniques were used to improve the contrast of the images obtained from the GTSRB. The three different architectures of CNN on which our model has been trained is described in the subsection 3.2.

CNN Architectures

LeNet Architecture

The LeNet architecture is a famous neural network architecture for CNN (especially when the model is trained on the MNIST dataset) (Bouti et al., 2019). LeNet is relatively small and easily understandable. Nonetheless, the numbers of nodes can be altered. As shown in figure 7(b), a receptive field of size 5×5 is used tiling the input image with stride I in 2 convolutional layers. 2×2 kernels with stride 2 arrangements resample the data and decreases size of the data after each convolutional layer, max pooling layer. In addition with the SDD meta-architecture, a feature extractor is used, so that the network architecture does not cleave. Larger and more convolutional layers are required to process high-resolution images, thus this method requires large amount of resources. The error rate achieved on the MNIST data set is below 1%.

VggNet Architecture

There are many types of VGG architecture like the VGG-16 & VGG-19. We have employed VGG-16 in this work. This architecture's name VGG-16 comes from the 16 layers it is made of shown in (S. Zhou et al., 2018). VGG was the winner of the 2014 ImageNet competition. The input is usually of size 224×224 that is passed through a relay of convolutional layers with filters of layer 3×3, sometimes it also uses 1×1 linear filters that transforms the channel linearly. The convolutional stride is fixed at 1 pixel and the spatial padding is also fixed after convolution. Then the input is passed through the

Figure 6. Limiting amplification in CLAHE.

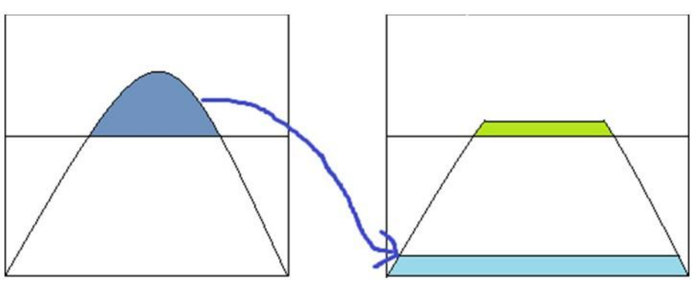
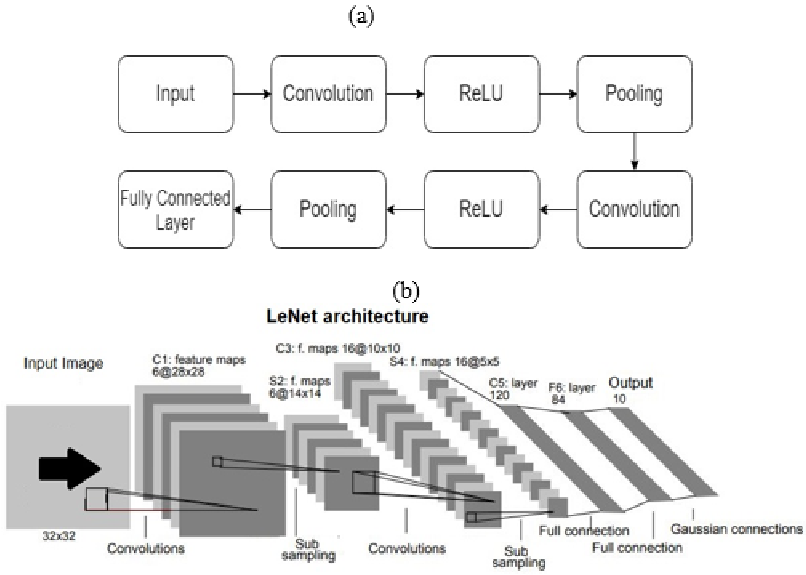


Figure 7. (a) Layers of LeNet architecture. (b) The original architecture of LeNet.



max-pooling layer that consists of 5 pooling layers. Finally, there is the soft-max layer. The hidden layers are occupied with the activation function, the ReLU. The VGG model has two characteristics: The first property being that the convolution cores are small, most cores are of size 3×3, and few of them are 1×1. The activation function follows the convolution operation; the second characteristic is the small pooling kernel, the pooling kernel is only of size 2×2, compared to the LeNet's 3×3 pooling kernel making the layers deeper. Convolution core expands the number of channels, pooling layers narrows the height and width, which makes the model structure deeper and wider. There are 13 convolutional layers, 5 max pooling layers and 3 dense layers summing up to 21 layers and only 16 weight layers. Conv 1 has number of filters as 64 while Conv 2 has 128 filters, Conv 3 has 256 filters while Conv 4 and Conv 5 has 512 filters.

ResNet Architecture

ResNets have different sizes, depending on the size of each layer and the number of layers of the model (Das, 2017). The baseline architectures are the same as VggNet, except there is a shortcut connection to each 3×3 filters pair. In the first comparison, there is an identity mapping for all the shortcuts and zero-padding for incrementing dimensions. ResNet architecture has only one convolution and pooling step then 4 layers of almost same functionality. Every layer almost performs the same function as they perform 3×3 convolution with a definite feature map dimension (F) [64, 128, 256, 512] respectively, avoiding the input after every 2 convolutions. The height (H) and width (W) dimensions remain same in the whole layer.

Figure 8. Schematic diagram of VGG 16 Model.

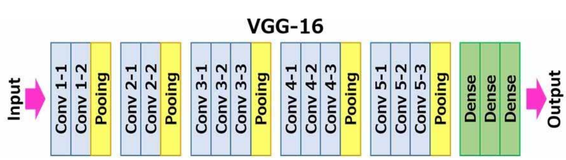
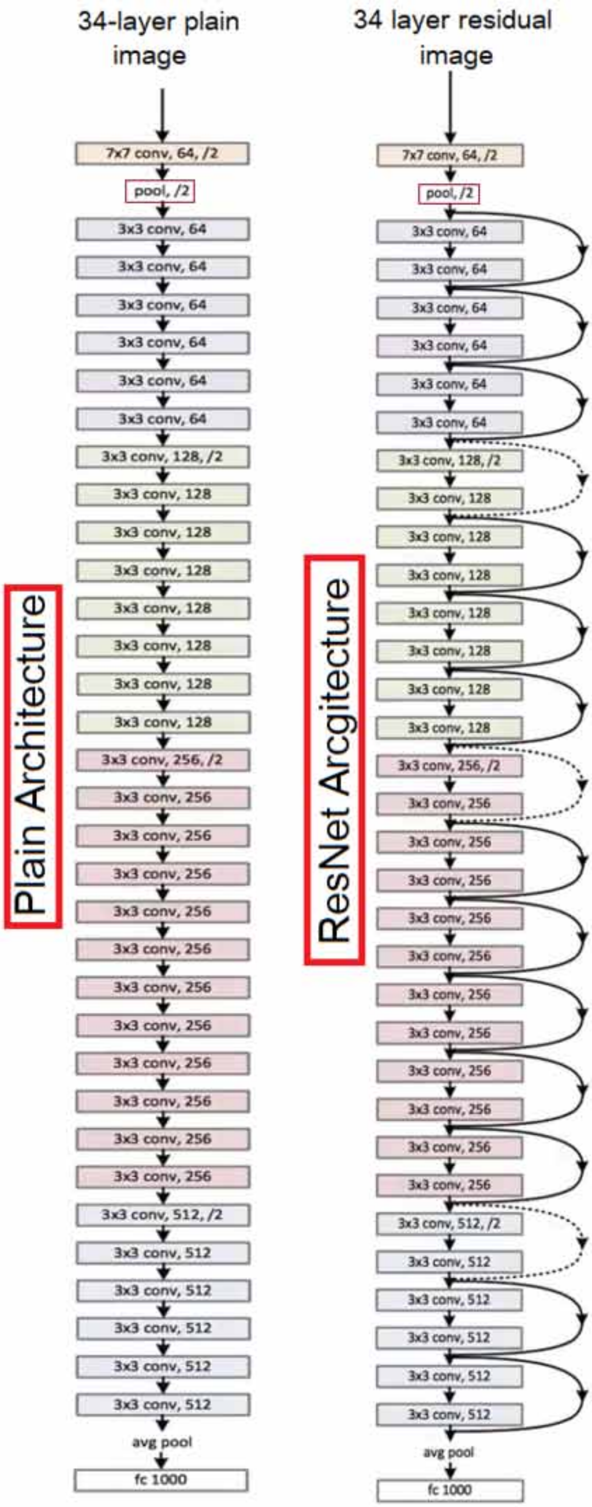


Figure 9. Comparison of plain Residual layer with ResNet Architecture.



As the number of stride increases from 1 to 2, there is a reduction between the layers at the first convolution of each layer. Let us see the working of ResNet architecture. Consider a plain neural network (A) without residual network as shown in figure 10. So in the network (A) the input X is passed to this neural network (NN) to give the activation $A1$.

Now, consider a deeper network (B) in which a residual block (with 2 extra layers and a skip connection) is added in the previous network. So now, the activation A1 is being passed to residual block that in turns gives new activation A3. If there were no skip connection, then A3 would have been as in Equation (6):

$$A3 = ReLU\left(W2.\ A2 + b2\right) \tag{6}$$

where, $W2$ and $b2$ are weights and bias associated with layer $L2$. However, with skip connection another term A1 will be passed to $L2$. Therefore, the formation of $A3$ will be modified as in Equation (7):

$$A3 = ReLU\left(W2.\ A2 + b2 + A1\right) \tag{7}$$

If we use $L2$ regularization or the weight decay methods, they will force $W2$ and $b2$ to become close to zero. In the worst case, if these become zero, then $A3=ReLU\left(A1\right)$. As ReLU function will map negative values to 0 and positive values to $A1$. Also, as it knows that $A1$ being output of previous activation function and is positive, thus, $A3=A1$. This means that identity function is easy for residual blocks to learn. By addition of residual blocks, model complexity was not increased. As this is only copying the previous activation to the next layers. However, this is only the worst-case situation, but it may turn out that these additional layers learn something useful. In that case, the network performance will improve. Hence, adding the residual blocks/skip connections does not reduce the network performance but in fact increases the chances that new layers will learn something useful.

In this paper, we have used 152 layers in ResNet architecture, thus called ResNet-152 as it gave the highest accuracy. We compare the discussed architectures in tabular form in table 1. The top-5 error rates correspond to the percentage of test examples for which the correct class was not in the top 5 predicted classes. The tabulated top-5 error rates are obtained in the ILSVRC competition in the respective years these architectures competed as shown in the Table 1. The number of parameters tabulated is the sum of weights and biases.

Figure 10. Diagrammatic comparison of Neural Network architecture without and with Residual Network layers.

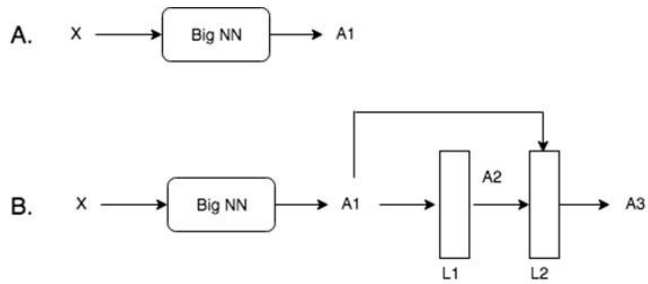


Table 1. Comparison of the CNN architectures

CNN architecture	Year	Developed By	Top-5 error rates (%)	No. of parameter
LeNet	**1998**	**YannLeCun et al (LeCun, Bottou, Bengio, & Haffner, 1998)**	**15.3**	**60,000**
VggNet	2014	Simonyan, Zisserman (Simonyan & Zisserman, 2014)	7.3	4,000,000
ResNet	2015	Kaiming He (He, Zhang, Ren, & Sun, 2016)	3.6	138,000,000

EXPERIMENTAL RESULTS

Experiments were performed on GTSRB dataset to validate the performances of the proposed models. A brief introduction of the experimental setups followed by the quantitative description of the dataset is given first. Then, the classification results are reported.

The Dataset

GTSRB was used to evaluate the performance of the proposed TSR system. GTSRB is a huge database with 51,840 color images (34,800 for training, 12,630 for testing, and 4410 for validation). The sizes of these images are (32×32×3). There are 43 classes in GTSRB and the physical traffic sign examples are unique in this dataset.

The proposed method was implemented in python with the Tensorflow environment enabled. The implementations of gray-scaling the images are done using OpenCV library, histogram equalization of images from the dataset is done using the skimage library. The following results are all obtained on a mainstream PC with an 8-core 2.0 GHz CPU and 8GB RAM.

Image Pre-Processing

The images were preprocessed following the steps of grey-scaling, normalization and equalization. Gamma correction, histogram equalization, and CLAHE methods were employed for the equalization. Figure 12 depicts the images after preprocessing using the given methods.

Classification Performance

We have used three architectures namely LeNet, VggNet, and ResNet architectures to train the CNN model. We have trained the CNN architectures with the input image is of size 32×32×3. According

Figure 11. Sample traffic sign images from the GTSRB.

Figure 12a. Images pre-processed after contrast improvement using (a) gamma transformation

Figure 12b. Images pre-processed after contrast improvement using (b) histogram equalization

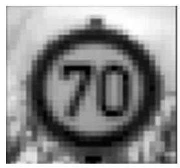


Figure 12c. Images pre-processed after contrast improvement using CLAHE method of Figure 2(b)

to pipeline architecture of the model discussed in section II, the images were first pre-processed to improve their quality, followed by the training of the CNN models. The CNN model, in their respective architectures, were designed with parameters described in Table 2.

We had three datasets *viz.* training, testing, and validation dataset. The models were trained on the training dataset and the test dataset were utilized for testing. Further, the test dataset also helped to determine those parameters of the model that help to avoiding the overfitting. A number of pilot runs over the different values of parameters were performed for selection of parameters and eliminating the problem of overfitting. On the final models, we have obtained the test and validation accuracy. Consequently, we compared the classification accuracy values to find the best image processing technique and the best CNN architecture to train the model for most efficient TSR system. Table 3 and 4 presents the values of classification accuracies obtained on test and validation set, respectively.

Table 2. Parameters set for designing of CNN model on different architectures

Architecture	Total number of layers	Number of Convolutional Layers	Filter Size of convolution layer	Number of strides in convolution layer	Number of pooling Layers	Filter Size of pooling layer	Number of strides in pooling layer	Output layers
LeNet	7	3	5×5	1	2	2×2	1	10
VggNet	16	8	3×3	2	5	2×2	2	43
ResNet	152	150	7×7	2	2	3×3	2	1000

Table 3. Comparison of testing accuracy obtained after training the model on the training dataset.

Architecture	Gamma Correction	Histogram Equalization	CLAHE
LeNet	89.3%	90.9%	92.451%
VggNet	96.8%	97.1%	97.5%
ResNet	96.9%	97.4%	98.1%

Table 4. Comparison of validation accuracy obtained after training the model on the training dataset.

Architecture	Gamma Correction	Histogram Equalization	CLAHE
LeNet	91.519%	93.719%	95.669%
VggNet	97.829%	98.619%	98.884%
ResNet	96.325%	98.417%	99.425%

A confusion matrix is a matrix summarizing the results of a classification problem. The count values of the results whether they are correct or incorrect are summarized in a matrix columned into their respective class (Tan et al., 2016). The confusion matrix gets its name from its task of finding the errors in the model as it is a measure of how confused our classification model is. It is a vital tool in order to not only study the errors, but also the type of errors and thus rectify the problems in the classification model efficiently. In a normalized confusion matrix, a row represents an instance of the actual class (i.e. an actual surgical step), whereas a column represents an instance of the predicted class (i.e. the predicted surgical step). The diagonal elements of the confusion matrix represent the degree of correctly predicted classes. For a better representation, we have taken the logarithm of the values. We depict the performance of different models using the normalized confusion matrix as in Figure 13.

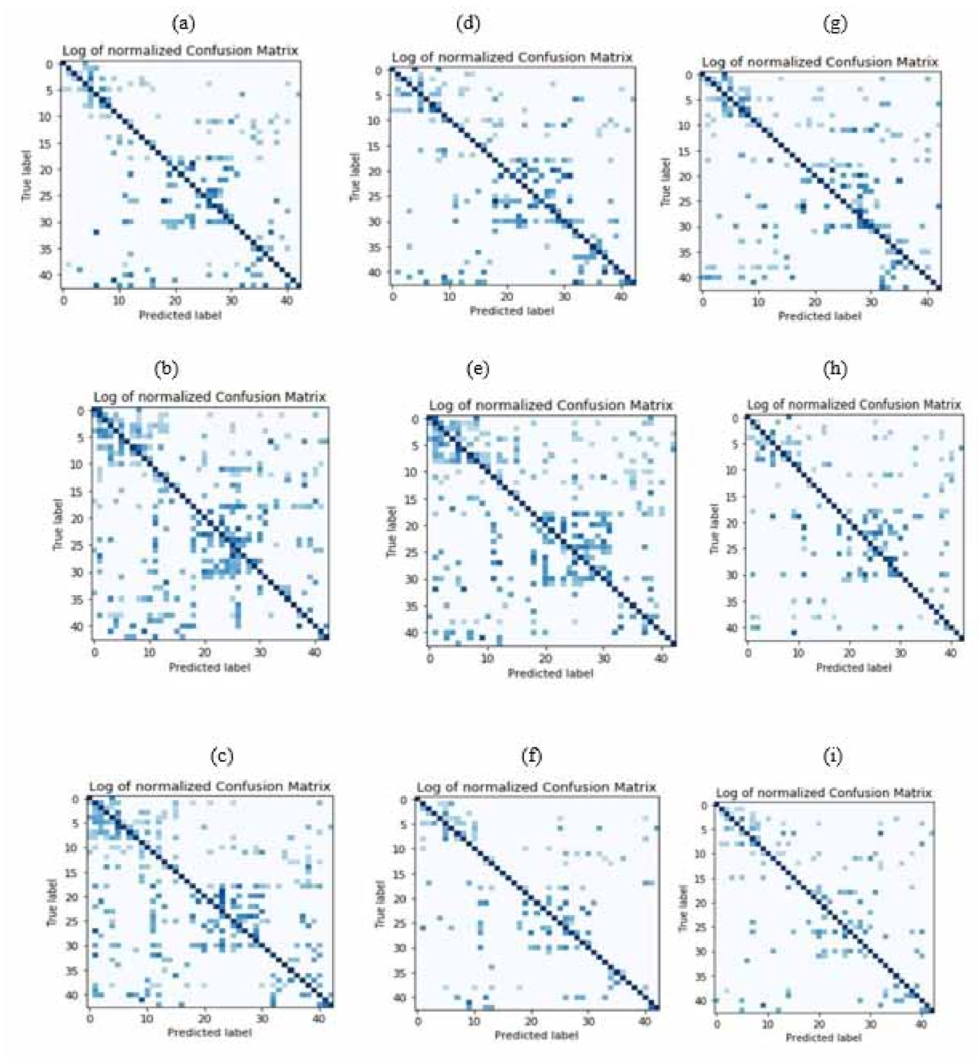
In order to show the effectiveness of the proposed model, its performance is compared with other state-of-the-art methods that have been used on same data set for TSR. Table 5 presents results of classification accuracies obtained by distinct methods. It is evident from the Table 5 that the proposed method was able to achieve better results than most of the existing methods. Only the CNN with colored probability model achieved comparable results, the method uses three dimensional color images.

DISCUSSION

We have analyzed the image processing methods to pre-process traffic sign images obtained from GTSRB and trained the CNN model on different architectures. After experimental analysis, it was evident that CLAHE is the best method to improve contrast of the images for the TSR system. The proposed ResNet152 architecture produced better testing accuracy in comparison to most of the other state-of-the-art approaches. The proposed ResNet architecture consisted of 152 convolutional layers and was comparatively better than the tradition ResNet technique that generally uses only 34 convolutional layers. Furthermore, the highest accuracy obtained by us, i.e., 98.6% is much higher than the other state-of-the-art approaches because we have applied one of the best CNN methods in a combination with suitable image pre-processing method to enhance image quality. The proposed CLAHE-based method adjusts the pixel values adequately to restore the quality of images when the traffic signs may be distorted due to weather or some other real-time reasons.

Though the CNN with colored probability model achieved equal results, the method uses three dimensional color images. Consequently, the process is very complex, time consuming, and energy inefficient. This kind of process may not be recommended for real time detection in TSR systems.

Figure 13. Confusion matrix for (a) LeNet architecture with gamma transformation (b) LeNet architecture with histogram equalization (c) LeNet architecture with CLAHE (d) VggNet architecture with gamma transformation (e) VggNet architecture with histogram equalization (f) VggNet architecture CLAHE (g) ResNet architecture with gamma transformation (h) ResNet architecture with histogram equalization (i) ResNet architecture CLAHE.



Hence, if compare the performance based on the optimum balancing between the classification accuracy and computational cost, the presented approach is certainly the best among all the existing methods for an efficient and affordable TSR system.

The enhancement in the results of the proposed process is achieved due to two important factors: first, we have properly pre-processed images using a suitable methodology, and second, the ResNet-152 architecture involves deep learning using extensive convolutional layers increasing the accuracy of image recognition. The best testing accuracy obtained in our model is 98.6% and the best validation accuracy obtained is 99.425%. The validation accuracy obtained on training the model is high because we have trained our model in such a way that it has properly evicted the problem of overfitting.

Table 5. Comparison of testing accuracy obtained with the state-of-art

Approach	Accuracy (in %)
Proposed Model	98.6
Multi-task CNN model(Qian, Zhang, Yue, Wang, & Coenen, 2015)	90.2
Fully Convolution Network and Edgebox(W. Zhou, Li, Lu, & Tian, 2012)	93.2
CNN with feature extraction methods like GABOR, HOG and LBP(Berkaya, Gunduz, Ozsen, Akinlar, & Gunal, 2016)	97.04
Fully Convolution Network (Y. Zhu et al., 2016)	97.69
CNN on colored images using Color Probability Model (Yang, Luo, Xu, & Wu, 2015)	98.6

CONCLUSION

In this paper, a novel method for designing an efficient TSR is proposed. CLAHE-based image equalization was applied during the image pre-processing step. Further, different CNN architectures namely LeNet, VggNet and ResNet were employed to solve the multi-class classification problem. The novelty of the proposed model lies in using a combination of CLAHE and ResNet, for image enhancement and classification, respectively. The effectiveness of the proposed model was experimentally validated, where the combination obtained 98.6% classification accuracy and 99.425% validation accuracy. This is so far the highest reported classification accuracy on GTSRB dataset (both test and validation). The future work aims at improving the testing accuracy of the model through efficient training and better image processing technique.

REFERENCES

Aghdam, H. H., Heravi, E. J., & Puig, D. (2016). A practical approach for detection and classification of traffic signs using convolutional neural networks. *Robotics and Autonomous Systems*, *84*, 97–112. doi:10.1016/j.robot.2016.07.003

Agrawal, S., & Chaurasiya, R. K. (2017). *Automatic traffic sign detection and recognition using moment invariants and support vector machine.* Paper presented at the 2017 International Conference on Recent Innovations in Signal processing and Embedded Systems (RISE). doi:10.1109/RISE.2017.8378169

Anwar, S. M., Majid, M., Qayyum, A., Awais, M., Alnowami, M., & Khan, M. K. (2018). Medical image analysis using convolutional neural networks: A review. *Journal of Medical Systems*, *42*(11), 226. doi:10.1007/s10916-018-1088-1 PMID:30298337

Arcos-Garcia, A., Alvarez-Garcia, J. A., & Soria-Morillo, L. M. (2018). Evaluation of deep neural networks for traffic sign detection systems. *Neurocomputing*, *316*, 332–344. doi:10.1016/j.neucom.2018.08.009

Berkaya, S. K., Gunduz, H., Ozsen, O., Akinlar, C., & Gunal, S. (2016). On circular traffic sign detection and recognition. *Expert Systems with Applications*, *48*, 67–75. doi:10.1016/j.eswa.2015.11.018

Bouti, A., Mahraz, M. A., Riffi, J., & Tairi, H. (2019). A robust system for road sign detection and classification using LeNet architecture based on convolutional neural network. *Soft Computing*, 1–13. doi:10.1007/s00500-019-04307-6

Cireşan, D., Meier, U., Masci, J., & Schmidhuber, J. (2012). Multi-column deep neural network for traffic sign classification. *Neural Networks*, *32*, 333–338. doi:10.1016/j.neunet.2012.02.023 PMID:22386783

Das, S. (2017, Nov. 16). CNN Architectures: LeNet, AlexNet, VGG, GoogLeNet, ResNet and more…. *Medium*.

Do, H. N., Vo, M.-T., Luong, H. Q., Nguyen, A. H., Trang, K., & Vu, L. T. (2017). *Speed limit traffic sign detection and recognition based on support vector machines.* Paper presented at the 2017 International Conference on Advanced Technologies for Communications (ATC).

Farag, W., & Saleh, Z. (2018). *Traffic signs identification by deep learning for autonomous driving*. Academic Press.

Greenhalgh, J., & Mirmehdi, M. (2012, August). Traffic sign recognition using MSER and random forests. In *2012 Proceedings of the 20th European Signal Processing Conference (EUSIPCO)* (pp. 1935-1939). IEEE.

Gündüz, H., Kaplan, S., Günal, S., & Akınlar, C. (2013). Circular traffic sign recognition empowered by circle detection algorithm. *Proceedings of the 2013 21st Signal Processing and Communications Applications Conference (SIU)*, 1–4. doi:10.1109/SIU.2013.6531432

He, K., Zhang, X., Ren, S., & Sun, J. (2016). *Identity mappings in deep residual networks.* Paper presented at the European conference on computer vision.

Jin, J., Kun, F., & Changshui, Z. (2014). Traffic sign recognition with hinge loss trained convolutional neural networks. *IEEE Transactions on Intelligent Transportation Systems*, *15*(5), 1991–2000. doi:10.1109/TITS.2014.2308281

Kiran, C., Prabhu, L. V., & Rajeev, K. (2009). *Traffic sign detection and pattern recognition using support vector machine.* Paper presented at the 2009 Seventh International Conference on Advances in Pattern Recognition. doi:10.1109/ICAPR.2009.58

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

Li, Y., Pankanti, S., & Guan, W. (2010, August). Real-time traffic sign detection: an evaluation study. In *2010 20th International Conference on Pattern Recognition* (pp. 3033-3036). IEEE. doi:10.1109/ICPR.2010.743

Møgelmose, A., Trivedi, M. M., & Moeslund, T. B. (2012). Vision-Based Traffic Sign Detection and Analysis for Intelligent Driver Assistance Systems: Perspectives and Survey. *IEEE Transactions on Intelligent Transportation Systems*, *13*(4), 1484–1497. doi:10.1109/TITS.2012.2209421

Qian, R., Yue, Y., Coenen, F., & Zhang, B. (2016, August). Traffic sign recognition with convolutional neural network based on max pooling positions. In *2016 12th International Conference on Natural Computation, Fuzzy Systems and Knowledge Discovery (ICNC-FSKD)* (pp. 578-582). IEEE. doi:10.1109/FSKD.2016.7603237

Qian, R., Zhang, B., Yue, Y., Wang, Z., & Coenen, F. (2015). *Robust Chinese traffic sign detection and recognition with deep convolutional neural network.* Paper presented at the 2015 11th International Conference on Natural Computation (ICNC).

Rahman, S., Rahman, M. M., Abdullah-Al-Wadud, M., Al-Quaderi, G. D., & Shoyaib, M. (2016). An adaptive gamma correction for image enhancement. *EURASIP Journal on Image and Video Processing*, *2016*(1), 1–13. doi:10.1186/s13640-016-0138-1

Sepasian, M., Balachandran, W., & Mares, C. (2008). Image enhancement for fingerprint minutiae-based algorithms using CLAHE, standard deviation analysis and sliding neighborhood. *Proceedings of the World congress on Engineering and Computer Science*.

Sezer, O. B., & Ozbayoglu, A. M. (2018). Algorithmic financial trading with deep convolutional neural networks: Time series to image conversion approach. *Applied Soft Computing*, *70*, 525–538. doi:10.1016/j.asoc.2018.04.024

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

SoniD.ChaurasiyaR. K.AgrawalS. (2019). Improving the Classification Accuracy of Accurate Traffic Sign Detection and Recognition System Using HOG and LBP Features and PCA-Based Dimension Reduction. Available at SSRN 3358756. 10.2139/ssrn.3358756

Stallkamp, J., Schlipsing, M., Salmen, J., & Igel, C. (2011). *The German traffic sign recognition benchmark: a multi-class classification competition.* Paper presented at the 2011 international joint conference on neural networks. doi:10.1109/IJCNN.2011.6033395

Stallkamp, J., Schlipsing, M., Salmen, J., & Igel, C. (2012). Man vs. computer: Benchmarking machine learning algorithms for traffic sign recognition. *Neural Networks*, *32*, 323–332. doi:10.1016/j.neunet.2012.02.016 PMID:22394690

Sugiharto, A., & Harjoko, A. (2016). *Traffic sign detection based on HOG and PHOG using binary SVM and k-NN.* Paper presented at the 2016 3rd International Conference on Information Technology, Computer, and Electrical Engineering (ICITACEE). doi:10.1109/ICITACEE.2016.7892463

Tan, P.-N., Steinbach, M., & Kumar, V. (2016). *Introduction to data mining*. Pearson Education India.

Torresen, J., Bakke, J. W., & Sekanina, L. (2004, October). Efficient recognition of speed limit signs. In *Proceedings. The 7th International IEEE Conference on Intelligent Transportation Systems* (IEEE Cat. No. 04TH8749) (pp. 652-656). IEEE. doi:10.1109/ITSC.2004.1398978

Yang, Y., Luo, H., Xu, H., & Wu, F. (2015). Towards real-time traffic sign detection and classification. *IEEE Transactions on Intelligent Transportation Systems*, *17*(7), 2022–2031. doi:10.1109/TITS.2015.2482461

Youssef, A., Albani, D., Nardi, D., & Bloisi, D. D. (2016, October). Fast traffic sign recognition using color segmentation and deep convolutional networks. In *International conference on advanced concepts for intelligent vision systems* (pp. 205-216). Springer. doi:10.1007/978-3-319-48680-2_19

Yuan, Y., Xiong, Z., & Wang, Q. (2016). An incremental framework for video-based traffic sign detection, tracking, and recognition. *IEEE Transactions on Intelligent Transportation Systems*, *18*(7), 1918–1929. doi:10.1109/TITS.2016.2614548

Zhou, S., Liang, W., Li, J., & Kim, J.-U. (2018). Improved VGG model for road traffic sign recognition. *Computers. Materials & Continua*, *57*(1), 11–24. doi:10.32604/cmc.2018.02617

Zhou, W., Li, H., Lu, Y., & Tian, Q. (2012). Principal visual word discovery for automatic license plate detection. *IEEE Transactions on Image Processing*, *21*(9), 4269–4279. doi:10.1109/TIP.2012.2199506 PMID:22614654

Zhu, Y., Zhang, C., Zhou, D., Wang, X., Bai, X., & Liu, W. (2016). Traffic sign detection and recognition using fully convolutional network guided proposals. *Neurocomputing*, *214*, 758–766. doi:10.1016/j.neucom.2016.07.009

Zhu, Z., Liang, D., Zhang, S., Huang, X., Li, B., & Hu, S. (2016). Traffic-sign detection and classification in the wild. *Proceedings of the IEEE conference on computer vision and pattern recognition*. doi:10.1109/CVPR.2016.232

Utkarsh Dubey was an undergraduate B Tech student at the dept. of Electronics and Communication Engineering of Malaviya National Institute of Technology Jaipur during 2016-20. He has worked on deep learning and machine learning algorithms during his BTech Project. He is currently working as executive graduate trainee in Fidelity Investments, Bangalore, India.

Rahul Kumar Chaurasiya received the B.Tech. degree from the Maulana Azad National Institute of Technology, Bhopal, India, in 2009, and the M.E. degree in System Science and Automation from the Indian Institute of Science, Bangalore, in 2011. He received his Ph.D. degree in 2017 from National Institute of Technology, Raipur. He was a Senior Software Engineer with Brocade Communications Systems, Bangalore, in 2011-12. During 2013-19, he was Assistant Professor at the NIT, Raipur and MNIT Jaipur. Since 2020, he is with the Maulana Azad National Institute of Technology Bhopal as Assistant Professor. His research area includes Machine Learning, Pattern Recognition, Brain-Computer Interfacing, Optimization, Biomedical Signal Processing. He has authored several research articles in aforementioned areas.