

## Road and Sight Functions Detection and Recognition Using Stacked Sparse Auto Encoders (SSAE)

T. C. Kalaiselvi<sup>1</sup> and S. Swathi<sup>2</sup>

<sup>1</sup>\*Professor, <sup>2</sup>PG Scholar

<sup>1,2</sup>Department of ECE, Kongu Engineering College, Perundurai-638060, India

Corresponding Author: [kalaiselvi@kongu.ac.in](mailto:kalaiselvi@kongu.ac.in)

**Abstract:** Road accidents are one of the main sources of horribleness and death rate accounting one million passings for every year. In unfriendly rush hour conditions, the driver may not notify traffic signs which may cause mishaps. The main objective of the paper is to detect and recognize the traffic sign using deep learning model using Stacked Sparse AutoEncoders (SSAE) which uses the enriched dataset of German traffic signs to sustainably increase the accuracy and to give less computational time using SSAE than the previous approaches. The advancement of the framework incorporates following working stages, image preprocessing, detection and recognition. The proposed method uses RGB color segmentation and shape matching can be followed by Stacked Sparse AutoEncoders which gives the result of accuracy (100%) and processing time (5.5ms) per frame. SSAE introduces the sparse representation based on optimized kernel function to improve the image classification effect. It has an advanced level features from pixel intensities and the advanced level features produced with the help of autoencoders to give the better classification of traffic sign images. The created algorithm is executed using NodeMCU hardware platform.

**Index terms:** Stacked Sparse AutoEncoders (SSAE), Deep learning, Traffic sign detection and recognition, color segmentation, German traffic dataset, NodeMCU

### INTRODUCTION:

Driving car has become a significant movement of day by day life. Today, because of high speed of vehicles and physical and mental condition of the driver, road mishaps are more regular and lead to the demise of numerous lives. Psychological over-burden or carelessness of the driver prompts bogus acknowledgment of the surroundings and confusion are significant components in road mishaps. So the Traffic Sign Detection and Recognition (TSDR) system has been introduced to overcome this mishaps. However, the traffic signs can be classified using many different approaches. But the accuracy cannot be satisfied and the processing time also not fulfilled. Deep learning that utilizes different layers to continuously extract higher-level features from the raw input. It has a powerful learning ability which can effectively improve the image classification accuracy. The Stacked Sparse AutoEncoders (SSAE) has been introduced to give better accuracy and less computational time. Stacked Sparse AutoEncoders superior to any other approaches and analysis of general quality image of the traffic images reveals high variable information (Kavitha M. et al., 2020)[1].

The SegU-Net, has been used to segment traffic signs from individual frames of a video sequence and it is recognized based on a Convolutional Neural Network (CNN) architecture (U Kamal et al.,

2020)[2] and large scale structure of information in the traffic sign images are obtained by hierarchical significance detection method (Hee Seok Lee et al., 2018)[3]. In other methods, the proposed system utilized SURF feature of Indian traffic signs to train the Support Vector Machine (SVM) classifier. Here the system achieved 99% accuracy for cautionary class and 90% accuracy for informatory class (Altaf Alam et al., 2020)[4]. A novel TSR embedded system based on ZedBoard platform using segmentation technique on HSV space and ORB feature detector methods (W Farhat et al., 2018)[5] and to reduce detection time proposed to extract the traffic signs using SVM and CNN, to detect and classify the traffic signs (Yi Yang et al., 2015)[6] and demonstrated CNN architecture by implementing a real time traffic sign via camera attached to the vehicle and give voice notification to the driver. The proposed system achieved the accuracy 99.71% (Danyah A. Alghmgham et al., 2019)[7]. Then, proposed a combination of CNN and SVM performs a traffic sign detection and recognition. The YCbCr coloring is used which is input to the CNN for extracting and to categorize the color channels and SVM used for classification (Kavin Kumar K et al., 2018)[8]. Their proposed system achieved the accuracy 98.6% (Lai Y et al., 2018)[9]. Then proposed a deep convolutional neural network is deeper and wider than existing networks for hyperspectral image classification (H. Lee et al., 2017)[10]. Color and shape matching are used in an iterative optimization approach to segment the traffic signs. CNN perform better than traditional detectors and classifiers but it need to give further improved performance (Zhe Zhu et al., 2017)[11]. Random forest and Support Vector Machine (SVM) classifiers have been tried along with the new descriptor. The proposed system achieved the accuracy 94.21% on the GTSDDB dataset at a processing rate 8-10 frames/s (Ayoub Ellahyani et al., 2016)[12]. Circle detection algorithm has been newly developed to detect the traffic sign and an RGB-based color thresholding method is proposed. Histogram of Oriented Gradients (HOG) (Huang Z. et al., 2017)[13], Local Binary Patterns (LBP) and Gabor features are engaged within a support vector machine classification for traffic sign recognition and it achieved the accuracy of 97.04% (Selcan Kaplan Berkaya et al., 2016)[14].

## **MATERIALS AND METHODS:**

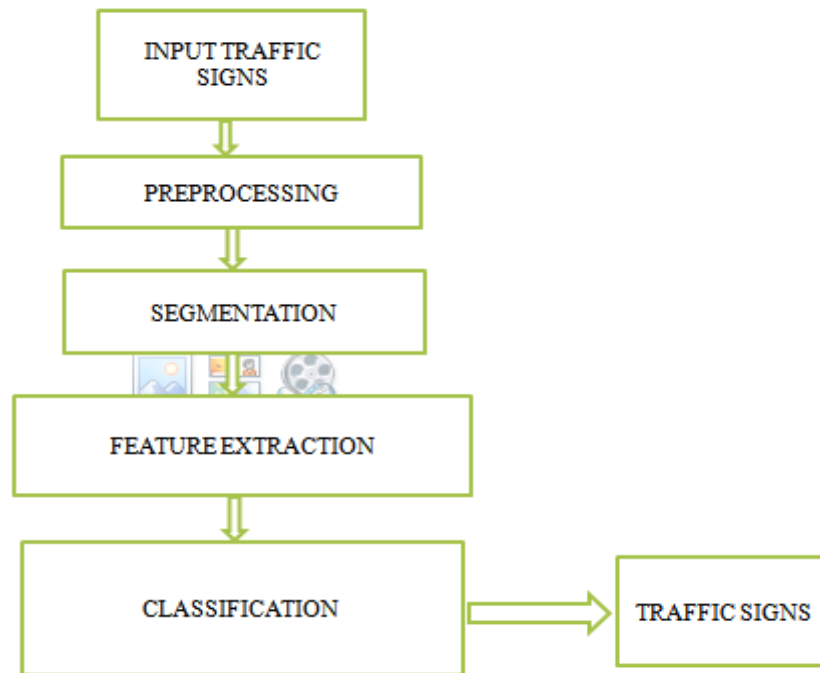
### **Proposed system**

In proposed method, the traffic sign is detected and recognized using Stacked Sparse Autoencoders. It has three main stages preprocessing, detection and recognition. The detection stage is used to find the region of interest which contains a traffic sign. The recognition stage using SSAE network finds the traffic sign and recognize them. The gathered dataset which can be given as an input to the SSAE architecture for training, validation and testing purpose. If the SSAE is trained, it is ready for classification and to give the finest accuracy and less computational time compared to previous approaches. Figure.1 shows the flowchart of proposed system and this is implemented in a NodeMCU hardware platform.

### **Dataset**

A traffic sign database is a basic prerequisite in building up any traffic sign detection and recognition system. German traffic data set has been used in this proposed method. The traffic data set is in video format with the dimension of 1280\*720. Frame rate of the data set is about 30 frames

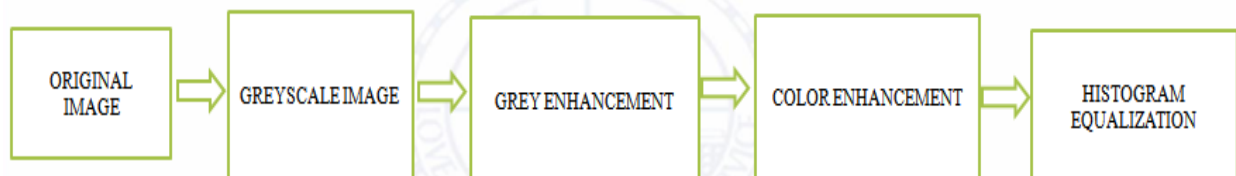
per sec.



**Figure 1. Flowchart of proposed system**

### Preprocessing

The first step is to train the dataset for that the video data set is trimmed to 1 minute. Then for further processing the trimmed video is converted into frames. After conversion, 2096 frames are obtained from trimmed video data set per minute. Preprocessing must be done to the dataset to improve the performance. It is utilized to eliminate exceptions and normalize the information with the goal that they take a structure to effectively used to make a model. Preprocessing is a fundamental prior before training the models.



**Figure.2. Block diagram for preprocessing**

Different dimensions of RGB images which can be led to preprocessing the images before given them to the specified network[9]. The images were transformed to greyscale images and the method used for preprocessing is histogram equalization method. Grey scale images are made out of shades of grey, the contrast ranges from black to white that is from weakest intensity to strongest intensity. The pixel value of a grayscale image ranges from 0-255. The color image into a grayscale image is converting the RGB values (24 bit) into grayscale value (8 bit). The greyscale enhancement is used to increase the contrast of greyscale image and to remove the noise and smoothing the image using gaussian filter. The gaussian filter is a non uniform low pass filter in

order to obtain the smoothening by reducing the noise. Gaussian filtering is simply to replace each pixel value in an image with the usage of gaussian filter mask. When sigma  $\sigma = 2$ , it produces an enhanced grey image compare to other values given to sigma. When sigma  $\sigma = 6$ , it produces a lower enhanced grey image.

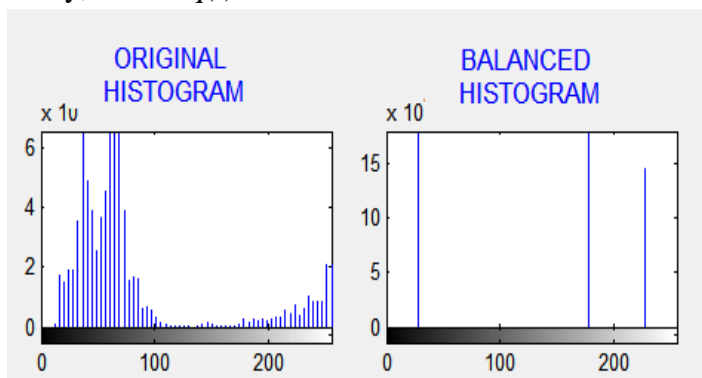


**Figure.3. Improved grey enhancement when sigma=2 for speed limit sign**



**Figure.4. Lower grey enhancement when sigma=6 for speed limit sign**

Intensity values can be adjusted with the help of histogram equalization . Normally original image contains low contrast, most of the pixel values esteems in the middle of the intensity range. The output image provided by “histeq” has an uniformly appropriated pixel values all through the range. It can be denoted by,  $J = \text{histeq}(I)$



**Figure.5. Before histogram and after histogram of speed limit sign**

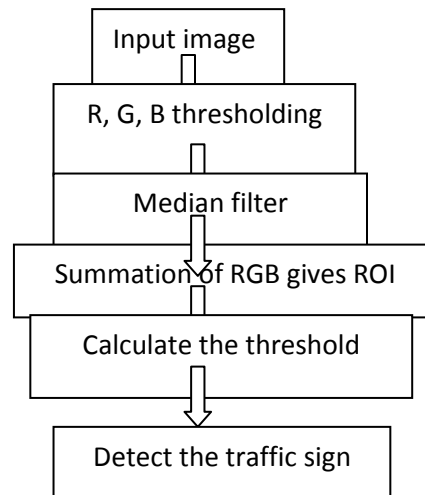
Different signs	Original image	Greyscale conversion	After histogram
Speed limit sign			
Pedestrian crossing sign			
Stop sign			
No entry sign			

**Figure.6. Table for preprocessing of different traffic signs**

### Traffic sign detection:

Traffic sign segmentation is the most important in the area of traffic sign detection. It is mainly used to reduce the noise in order to reduce the complexity. So the efficiency and accuracy can be improved automatically. Figure.7 shows the detection of traffic sign, first the input image can be passed to R, G, B space where the thresholding is carried out. The threshold value mainly depends on the intensity or brightness of the time. The threshold value can be allotted to R, G, B separately. After thresholding, the median filter can be applied. Filtering of ROI leads to removal

of background. The sum of the three colors is the Region of Interest (ROI). ROI is used to give binary image, so the pixel in the ROI set to one and outside the ROI set to zero. After detecting the traffic sign, the next step is to classify the traffic sign using SSAE network.



**Figure.7. Flowchart for detection of traffic sign**

The threshold depends upon the performance of training images. The redThresh = 0.14, which denotes the threshold for red component and figure.9 shows the image after thresholding when the traffic sign is in red. If the threshold value for red is 0.18, it does not detect the color of red. The greenThresh = 0.05, which denotes the threshold for green component and the blueThresh = 0.15, which denotes the threshold value for green component. The red, green, blue layer can be denotes

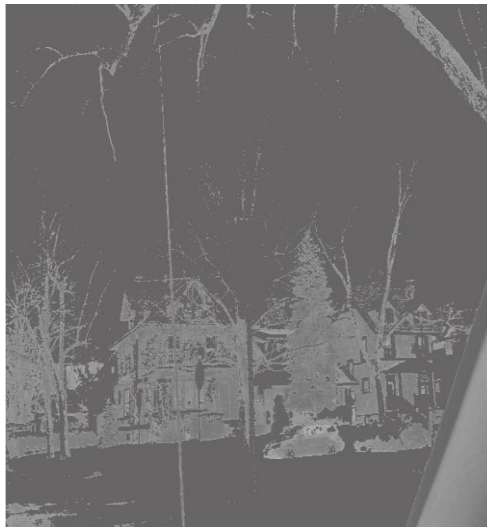
as, for red,  $image\_R(k\_c, k\_r) < R\_thres$

for green,  $image\_G(k\_c, k\_r) < G\_thres$

for green,  $image\_B(k\_c, k\_r) < B\_thres$



**Figure.8. After Thresholding (Red)**



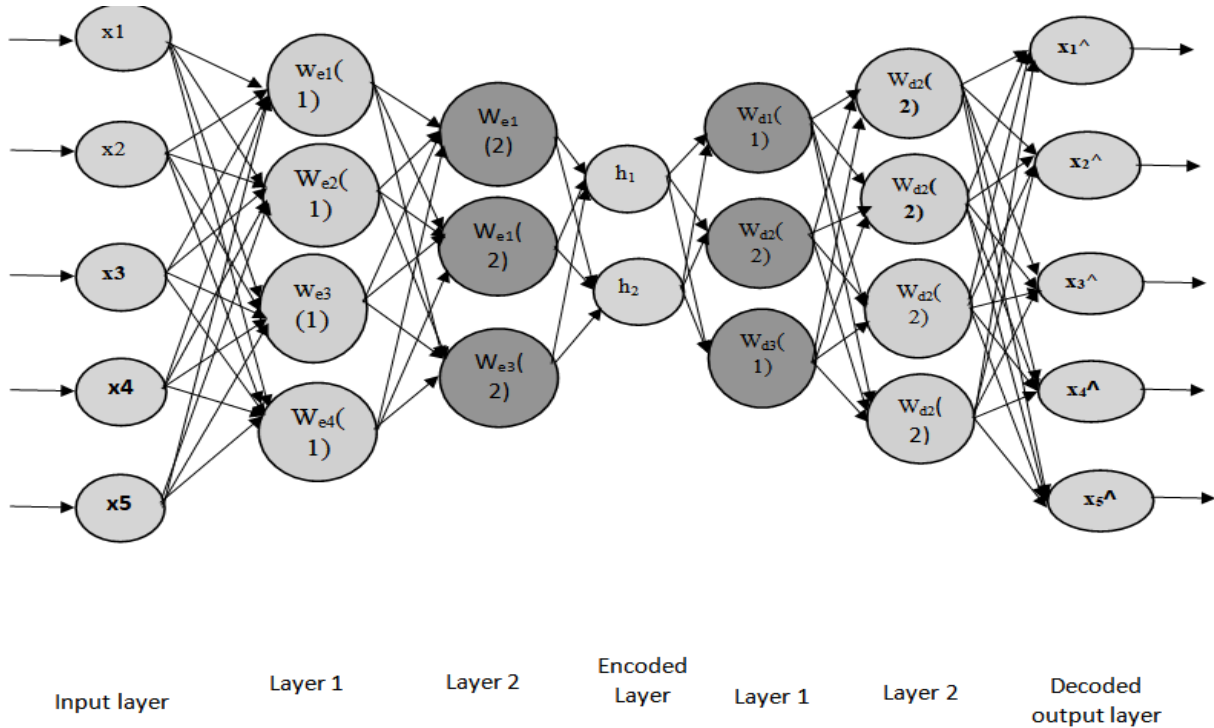
**Figure.9. After Thresholding (Green and Blue)**



**Figure.10.After applying median filter**

### **Traffic sign recognition using Stacked Sparse Autoencoders (SSAE)**

After detecting the traffic sign, the ROI has been passed to the SSAE network for recognition. SSAE is a neural network which is an unsupervised learning algorithm which uses back propagation to produce the output value almost equal to the input value. It tries to make the input into two components, one is encoder with fully connected layers which compress the input into a smaller manner that has less dimensions called bottleneck and the bottleneck tries to reconstruct the compressed input using decoder to give the specified output.



**Figure.11. Structure of SSAE**

SSAE solves the complex function easily in the deep learning model. The more number of sparse autoencoders forms a deep network called SSAE. It has a sparse constraint that is generally sigmoid function. If neuron is in active mode, the output is one and the other neuron is subdued, the output is zero. Here the  $x_1, x_2, x_3, \dots, x_m$  is the image to be trained. It has three layers input layer L1, hidden layer L2, output layer L3 and its objective is to make output signal  $\hat{x}$  same as input signal  $x$ . It has parameters  $(W, b)$ . The output  $\hat{x}$  that is similar to  $x$ . If the inputs  $x$  are the pixel intensity values from a  $10 \times 10$  image (100 pixels) so  $n = 100$ , and there are  $s_2 = 50$  hidden units in layer L2. Note that we also have  $y \in R^{100}$ . There are only 50 hidden units, the network is compelled to learn input in a compressed representation. ie., hidden unit activations vectors are only given  $a^{(2)} \in R^{50}$ , it must try to reconstruct the 100-pixel input  $x$ . Equation (1) denotes the activation value of  $j$  is,

$$\hat{\rho}_j = \frac{1}{m} \sum_{i=1}^m [a_j^{(2)}(x^i)] \dots \dots \dots (1)$$

The samples can be correspond to the activation value of node  $j$  are averaged and the equation (2) shows the constraints are,

$$\hat{\rho}_j = \rho \dots \dots \dots (2)$$

Here,  $\rho$  is the hidden layer of the sparse parameter,  $\rho$  is zero then the value is  $\rho = 0.08$  that is only 8% can be activated. Equation (3) denotes, to achieve this, add an extra penalty term to our optimization objective that penalizes  $\hat{\rho}_j$  deviating significantly from will choose the following:

$$\sum_{j=1}^{s_2} \rho \log \frac{\rho}{\hat{\rho}_j} + (1 - \rho) \log \frac{1-\rho}{1-\hat{\rho}_j} \dots \dots \dots (3)$$

Here,  $s_2$  denotes hidden layer and the index  $j$  is summing over the hidden units in our network.

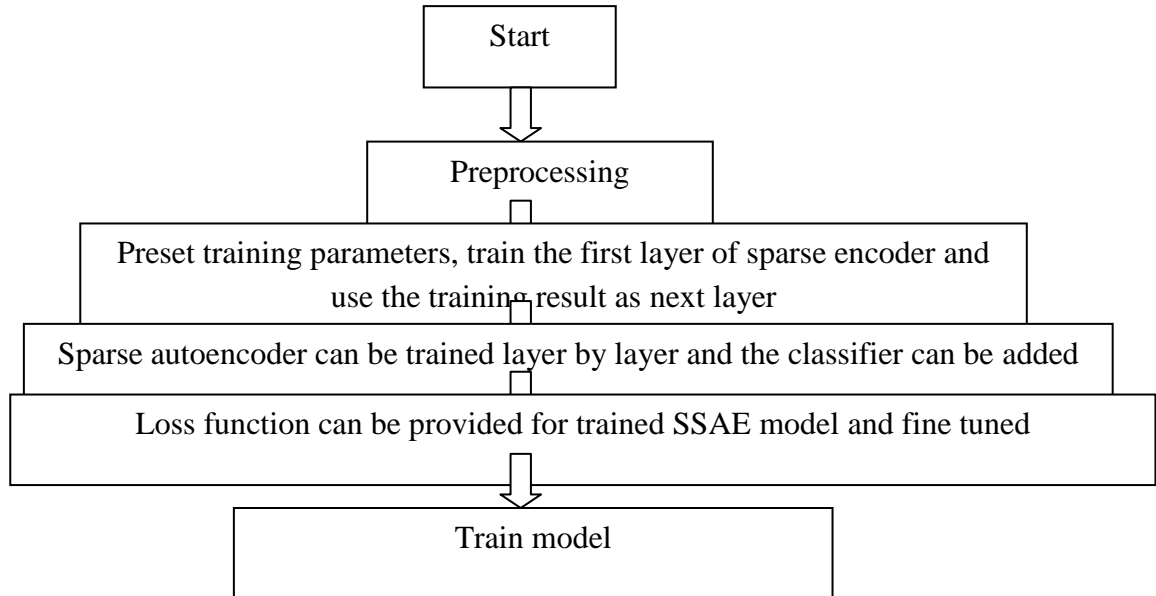


The equation (4) denotes the concept of KL divergence based on the penalty term and can also be written as:

$$\sum_{j=1}^{S_2} KL(\rho || \hat{\rho}_j) \dots\dots\dots(4)$$

where  $KL(\rho || \hat{\rho}_j) = \rho \log \frac{\rho}{\hat{\rho}_j} + (1 - \rho) \log \frac{1-\rho}{1-\hat{\rho}_j}$  is the Kullback-Leibler (KL) divergence between mean  $\rho$  and mean  $\hat{\rho}_j$ . The equation (5) shows the overall cost function is,

$$J_{sparse}(W, b) = J(W, b) + \beta \sum_{j=1}^{S_2} KL(\rho || \hat{\rho}_j) \dots\dots\dots (5)$$



**Figure.12. Flowchart for SSAE training model**

In SSAE, the dataset is preprocessed before the training process, with a size 256\*256 and the mean value also calculated and processed for each pixel in the dataset. In the training process, the image contains sparse features are extracted and the data characteristics can be learned layer by layer. So the efficiency and accuracy can be increased.

Method	Accuracy	Processing time	System environment
LBP + SVM	80%	450s	i3 processor, 500GB capacity, 3GB RAM, Hard disk-1TB
HOG + SVM	87%	135s	i3 processor @2.50 GHz, 4GB RAM
CNN	98%	25s	NVIDIA Geforce GTX 1050 Ti GPU, Intel i5 CPU
SSAE	100%	5.5ms	Intel i5 processor @3.5GHz, 4GB RAM

**Figure.13. Table for previous and proposed classification accuracy and processing time.**

## Performance of SSAE

The performance of the proposed system based on the german traffic dataset . Compare to other algorithms, the average processing time for the proposed is short. ROI method used for detection and sparse representation used for recognition achieves good performance. Both gaussian filter and histogram equalization method are used for successful image preprocessing, utilizing principal component analysis for dimensionality decrease and a better classification accuracy achieved. Despite the fact that the running environment of the algorithm included GPU and CPU, the average processing time was still moderately long. Deep learning techniques can still be additionally improved because of the complex structure of the training model, the large amount of calculation, the long training time and the poor real-time performance. All the traffic signs are correctly classified as true negative. Therefore the proposed system achieves 100% accuracy and 100% specificity. The result is divided into four section, ie True Positive (TP), True Negative (TN), False Positive (FP), False Negative (FN). FP does not detect the sign correctly. FN detect the sign as nonsign region. TP detect the sign correctly. TN detect the sign correctly as a nonsign region. Figure14 shows the specificity and accuracy obtained for the proposed system contains 2096 frames per minute.

$$\text{Specificity} = \text{TN} / (\text{TN} + \text{FP}) = 100\%$$

$$\text{Accuracy} = (\text{TN} + \text{TP}) / (\text{FP} + \text{FN} + \text{TP} + \text{TN}) = \left[ \begin{array}{l} \text{TP} = 0 \\ \text{FN} = 0 \end{array} \quad \begin{array}{l} \text{FP} = 0 \\ \text{TN} = 6 \end{array} \right] = 100\%$$

The overall processing time of the traffic sign per frame for detection and recognition system is 5.5ms and figure 15 shows the performance of SSAE for each epochs and the Mean Square Error (MSE) also calculated for each epochs, for SSAE layer = 1, MSE = 2.91667

$$\text{layer} = 2, \text{MSE} = 0.833333$$

$$\text{layer} = 3, \text{MSE} = 0.133187$$

$$\text{layer} = 4, \text{MSE} = 0.0239962$$

$$\text{layer} = 5, \text{MSE} = 0$$

$$\text{layer} = 6, \text{MSE} = 1.0551\text{e-}29$$

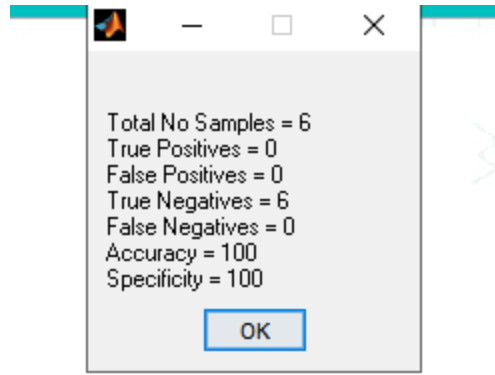
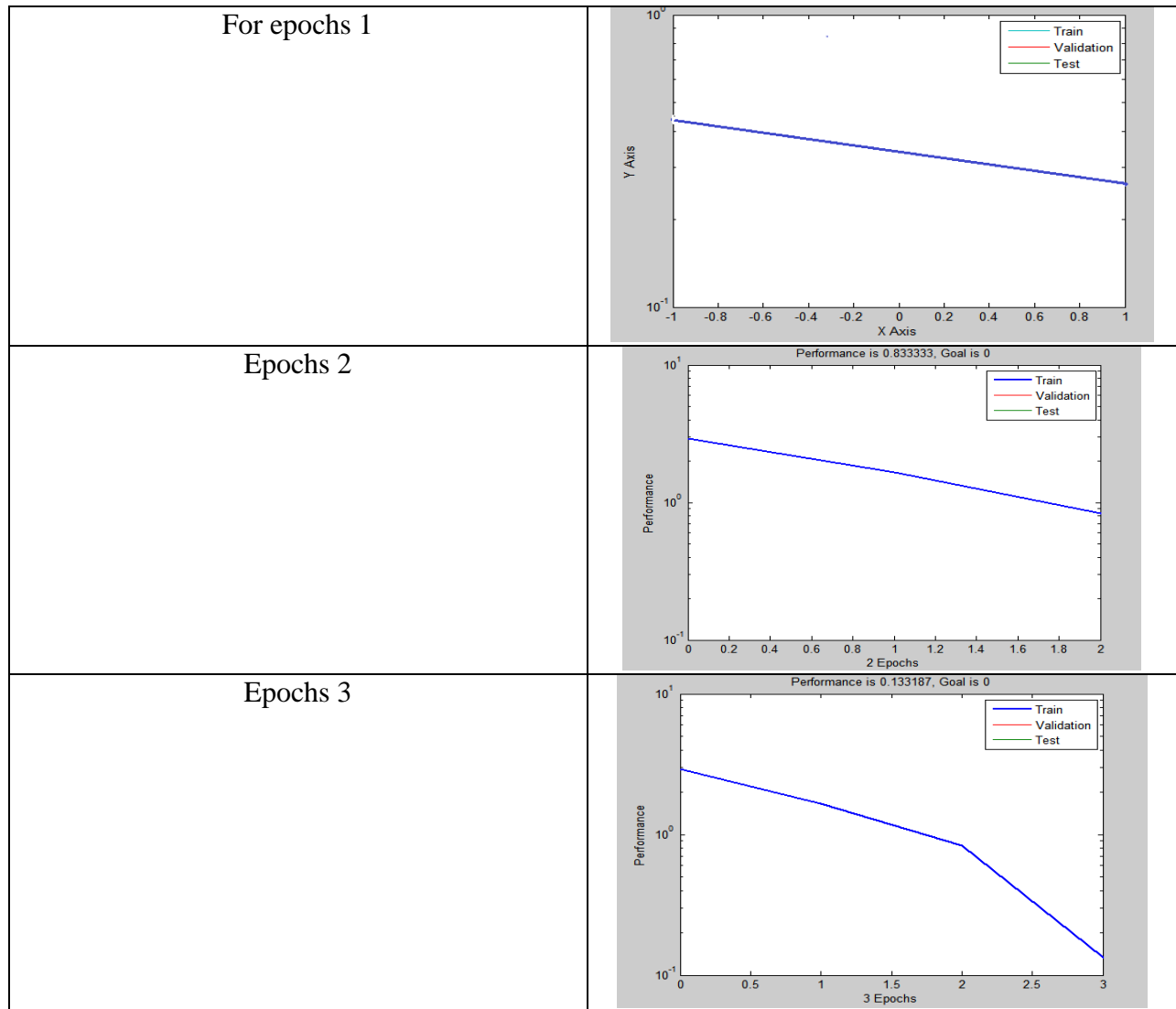
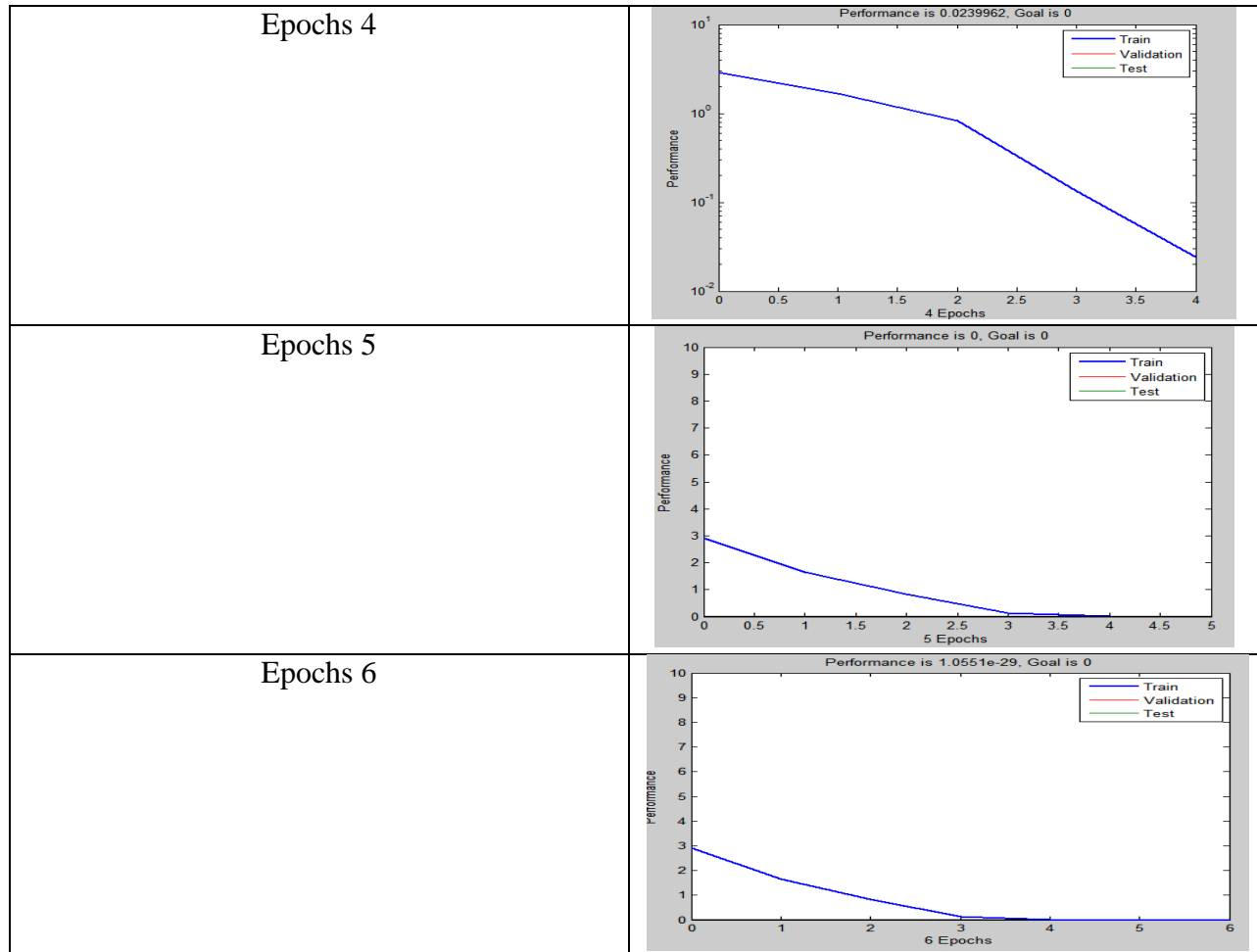


Figure.14. Accuracy and specificity achieved for proposed system

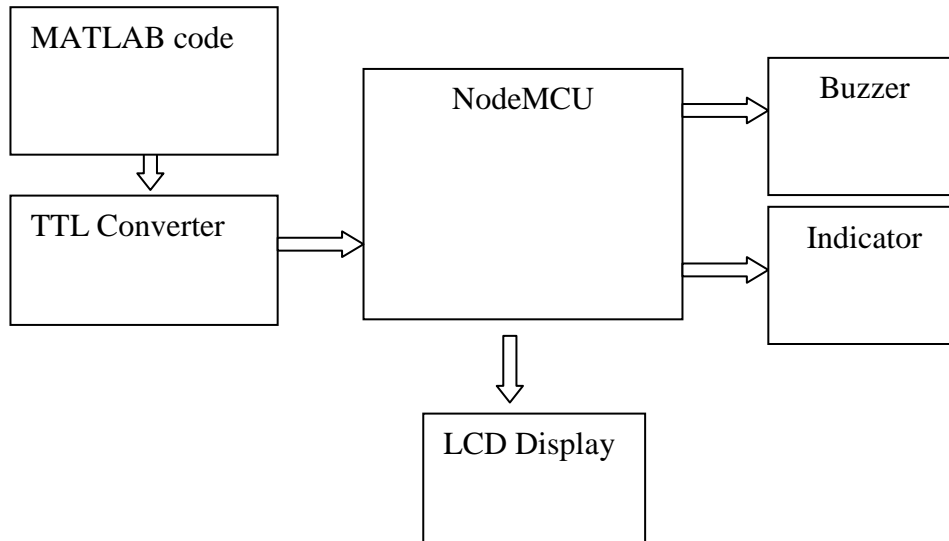




**Figure.15. Performance characteristics of SSAE for each epochs**

### Hardware implementation

The hardware implementation done using NodeMCU. NodeMCU is an ease open source IoT stage. It initially included firmware which runs on the ESP8266 Wi-Fi SoC from Espressif systems and hardware based on the ESP-12 module. It has RAM of 128 KB and 4MB of flash memory to store information and programs. In-built Wi-Fi / Bluetooth and Deep Sleep Operating highlights has an high processing power and make it ideal for IoT projects. The microprocessor underpins RTOS and the adjustable clock frequency from 80MHz to 160MHz. It is controlled utilizing micro USB jack and VIN pin. It supports UART, SPI, and I2C interface. Figure.15 shows the block diagram for the hardware implementation, the detected and recognized traffic signs in MATLAB has been sent to Transistor – Transistor Logic (TTL) converter. It is a logic family built from bipolar junction transistor. It performs both logic function and the amplifying function as opposed to Resistor Transistor Logic (RTL). It acts as an interface between hardware and software. The MATLAB code sent to NodeMCU through TTL convertor and it is displayed in LCD display.



**Figure.16. Block diagram for hardware implementation**

## RESULT AND DISCUSSION

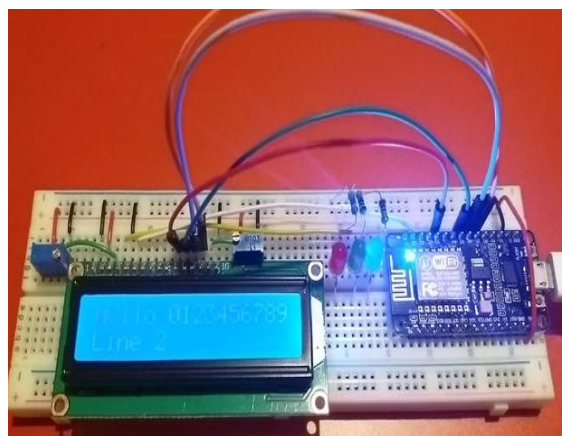
The detection and recognition of traffic sign based on german traffic sign dataset. The recognition process is done by using Stacked Sparse AutoEncoders. SSAE represents the pixel intensity in high level representation in an unsupervised manner. It solves the complex functions and develops a deep learning model with versatile approximation capacity. The algorithm uses sparse representation to rectify the optimization problem of classifier. The accuracy and processing time of SSAE gives better result. The software implementation is performed using MATLAB r2013a. The accuracy achieves 100% in the proposed method and the processing time is 5.5ms per frame. With the help of simulation result, the method successfully detect and recognize the traffic signs precisely with quick reaction. Figure 17 and figure 18 shows the classification of stop sign and no entry sign. The road sign detection implemented using NodeMCU (ESP8266) and the result displayed in the LCD display. It recognizes the road signs like pedestrian, speed limit, stop etc ..



**Figure.17. Stop sign detected**



**Figure.18. No entry sign detected**



**Figure.19. .Hardware implementation**

## CONCLUSION

The traffic sign detection and recognition system using german dataset achieved successfully using SSAE and implemented by NodeMCU with better performance. SSAE plays important role in achieving better results. The dataset contains 2096 frames per minute. It must be preprocessed before given to the network. The SSAE architecture by actualizing a reasonable system that can take real time traffic sign by means of camera appended to vehicle, groups them to the comparing sign, and afterward give voice notice to driver. Future work incorporates by expanding the size of the dataset and distributing it with the goal that it very well may be utilized by different analysts for benchmarking purposed. Work will likewise proceed on growing more strong and computationally ease acknowledgment system.

## REFERENCE

- 1) Kavitha M., Rajdakshan S. B., Tamilselvan S. and Mohamed Fardhin M. (2020), Framework for Cancer Detection using Deep Wavelet Autoencoder and Neural Network in Brain Images, Biosci. Biotech. Res. Comm, Special Issue, Vol.13, No.3, Pp.172-175.
- 2) Kamal U., Tonmoy T. I., Das S. and Hasan M. K. (2020), Automatic Traffic Sign

- Detection and Recognition Using SegU-Net and a Modified Tversky Loss Function With L1-Constraint, IEEE Transactions on Intelligent Transportation Systems, Vol.21, Issue 4.
- 3) Hee Seok Lee and Kang Kim (2018), Simultaneous Traffic Sign Detection and Boundary Estimation Using Convolutional Neural Network, IEEE Transactions on Intelligent Transportation Systems, Vol. 19, No. 5, Pp. 1652-1663.
  - 4) Altaf Alam, Zainul Abidin Jaffery (2016), A Video Based Indian Traffic Sign Classification, IEEE Transactions on Intelligent Transportation Systems, Vol.8, No.6.
  - 5) Farhat W., Sghaier S., Faiedh H. (2018), Design of Efficient Embedded System for Road Sign Recognition, Journal of Ambient Intelligence and Humanized Computing, Springer, Vol.10, Pp.491-507.
  - 6) Yang Y., Luo H., Xu H., Wu F.(2017), Towards Real Time Traffic Sign Detection and Classification, IEEE Transactions on Intelligent Transportation Systems, Vol.17, Issue 7, Pp. 2022-2031.
  - 7) Danyah A. Alghnham, Ghazanfar Latif, Jaafar Alghazo, Loay Alzubaidi (2019), Autonomous Traffic Sign Detection and Recognition Using Deep CNN, Journal of Traffic and Transportation Engineering, Elsevier, Vol.6, Issue 2, Pp.109-131.
  - 8) Kavin Kumar K, Meera Devi T, Maheswaran S. (2018), An Efficient Method for Brain Tumor Detection Using Texture Features and SVM Classifier in MR Images, Asian Pacific Journal of Cancer Prevention, Vol.19,No.10, Pp.2789-2794.
  - 9) Lai Y., Wang N., Yang Y., and Lin L. (2018), Traffic signs recognition and classification based on deep feature learning, 7th International Conference on Pattern Recognition Applications and Methods (ICPRAM), Pp. 622-629.
  - 10) Lee H. and Kwon H. (2017), Going deeper with contextual CNN for hyperspectral image classification, IEEE Transactions on Image Processing, Vol.26, No.10, Pp.4843-4855.
  - 11) Zhu Z., Lu J., Martin R. R. and Hu S. (2017), An Optimization Approach for Localization Refinement of Candidate Traffic Signs, IEEE Transactions on Intelligent Transportation Systems, Vol.18, No.6.
  - 12) Ayoub Ellahyani, Mohamed El Ansari, Ilyas El Jaafari (2016), Traffic sign detection and recognition based on random forests, Applied Soft Computing, Elsevier, Vol. 46, Pp. 805-815.
  - 13) Huang Z., Yu Y., Gu J. and Liu H. (2017), An Efficient Method for Traffic Sign Recognition Based on Extreme Learning Machine, in IEEE Transactions on Cybernetics, Vol. 47, No. 4, Pp. 920-933.
  - 14) Selcan Kaplan Bargaya, Huseyin Gunduz, Ozgur Ozsen, Cuneyt Akinlar, Serkan Gunal (2016), On Circular Traffic Sign Detection and Recognition, Journal of Traffic and Transportation Engineering, Elsevier, Vol.48, Issue 2, Pp. 67-75.
  - 15) Maheswaran S, Vivek B, Sivaranjani P, Sathesh S, Pon Vignesh K (2020), "Development of Machine Learning Based Grain Classification and Sorting with Machine Vision Approach for Eco-Friendly Environment", Journal of Green Engineering, Vol. 10, No. 3, Pp.526-543.