Journal of Theoretical and Applied Information Technology
15th November 2022. Vol.100. No 21

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

# DEEP LEARNING BASED SOUTH INDIAN SIGN LANGUAGE RECOGNITION BY STACKED AUTOENCODER MODEL AND ENSEMBLE CLASSIFIER ON STILL IMAGES AND VIDEOS

**RAMESH MANOHAR BADIGER[1] , DHARMANNA LAMANI[2]**

[1]Assistant Professor, Computer Science Department, Tontadarya College of Engineering, Gadag 582101, India

[2]Associate Professor, Department of ISE, SDMIT, Ujire 574240, India

E-mail: [1]rameshmbadiger@gmail.com, [2] dharmannasdmit@gmail.com

**ABSTRACT**

Recently, sign or gesture recognition has been challenged by concerns like high computational cost, occlusion of hands, and inaccurate tracking of hand signs and gestures. The existing models face difficulty in managing longer term sequential data, due to poor information learning and processing. To highlight the aforementioned concerns, a novel deep learning based ensemble model is proposed in this article. Firstly, the sign/gesture images are acquired from American Sign Language (ASL)-Modified National Institute of Standard and Technology (MNIST) and real time South Indian Sign Language (SISL) databases. In addition, K-means clustering with the Gaussian blur method is implemented for precisely segmenting the sign/gesture region. Next, the feature extraction is carried-out using Gray-level Co-occurrence Matrix (GLCM) features and AlexNet, and then the dimensionality of the extracted feature vectors are decreased using a deep learning model: stacked autoencoder. The dimensionally decreased feature vectors are fed to the ensemble classifier (Multi-Support Vector Machine (MSVM) and Naive Bayes) to classify 24 alphabets and 30 SISL classes on the ASL-MNIST and real time SISL databases. The extensive experiments demonstrated that the ensemble based stacked autoencoder model achieved 99.96% and 99.08% of accuracy on the ASL-MNIST and real time SISL databases, which are better related to the traditional machine learning classifiers.

**Keywords:** *Gesture, K-means Clustering, Multi Support Vector Machine, Naïve Bayes, Sign Language Recognition, Stacked Autoencoder*

## 1. INTRODUCTION

Sign language is a vision based inter-active language with complex and unique linguistics rules [1], and it is mainly used by people who are impaired in communicating and exchanging their thoughts, ideas and feelings using different body parts [2-3]. Sign language differs from one place to another based on its geographic location, but it has unique linguistic structures [4]. In recent decades, each nation has created its sign languages to communicate among the deaf and dumb communities [5-6]. Hence, manual sign/gesture recognition involves hand orientation, hand postures and hand movements [7]. The non-manual sign/gesture recognition involves lip movements, eye gaze, and facial expressions, where the recognition methods are generally categorized into two types vision based methods and data glove methods [8-9]. However, the existing models are very sensitive to lighting conditions and cannot be operated in the cluttered environment [10-11]. In addition to this, the existing models obtain minimum classification performance, due to over-lapping of the head, hand, skin color, and background color [12-13]. To overcome the above-mentioned problems and to achieve better gesture/sign recognition, a novel deep learning based ensemble model is implemented in this manuscript. The main contributions are listed below:

- Acquired raw images from ASL-MNIST and real time SISL databases and further, the Region of Interest (RoI) is segmented by using K-means clustering with Gaussian blur method.
- After segmenting RoI from ASL-MNIST and the real time SISL databases, the feature extraction is performed using GLCM feature and AlexNet model. The semantic space

between the extracted feature sub-sets is reduced by extracting local and deep learning feature vectors, where this process helps in achieving better classification results.

- The extracted multi-dimensional feature vectors are optimally decreased by proposing a stacked autoencoder, where it decreases computational complexity and running time of the proposed system.
- The developed ensemble classifier uses the optimum feature vectors for classifying 24 alphabets and 30 SISL classes on the ASL-MNIST and real time SISL databases, and the effectiveness of the ensemble based stacked autoencoder model is tested by means of sensitivity, accuracy, F1-measure, specificity and Matthews Correlation Coefficient (MCC). The experiments conducted on the ASL-MNIST and real time SISL databases showed that the proposed model achieved 99.96% and 99.08% of accuracy.

This article is prepared as follows: Some manuscripts related to sign language and gesture recognition are surveyed in Section 2. The brief theoretical description, simulation result and the conclusion of ensemble based stacked autoencoder model is represented in Sections 3, 4, and 5 respectively.

## 2. RELATED WORKS

Subramanian, et al. [14] introduced a new Media-Pipe-Optimized Gated Recurrent Unit (MOPGRU) algorithm for sign detection. As depicted in the resulting phase, the implemented MOPGRU algorithm obtained high learning efficiency, prediction accuracy, fast convergence and information processing capability related to existing sequential algorithms. However, the implemented model was computationally expensive, because it requires higher-end graphics processing units for achieving better classification results. Gangrade and Bharti [15] used Gaussian blur for decreasing the noise from the acquired gray sign images, and then, the hand segmentation was accomplished utilizing the background subtraction technique. Finally, the hand sign detection was performed by implementing the Convolutional Neural Network (CNN) model. The simulation investigation showed that the presented model efficiently detects ISL alphabet in the real time database with a high detection rate. The presented model works well with static ISL sign, but it does not manage continuous and dynamic signs. Kumar and Kumar [16] used Histogram of Oriented Gradients (HOG) and Extreme Learning Machine (ELM) for feature extraction and hand sign recognition. Still, the developed model needs to be extended for recognizing the dynamic ISL signs.

Katoch, et al. [17] used background subtraction and skin color techniques to segment sign regions from the collected images. Further, Speeded up Robust Features (SURF) and bag of visual words for feature extraction and then the sign classification was accomplished using hybrid classifiers: CNN and SVM, but the presented hybrid model was computationally costly. Wadhawan and Kumar, [18] implemented deep learning based CNN model for robust ISL recognition. The effectiveness of the developed model was tested utilizing performance metrics like recall, F1-score and precision. The implemented deep learning based CNN model was computationally complex, where it needed an enormous number of sign images to attain superior classification performance. Additionally, Badhe and Kulkarni [19] integrated Otsu thresholding and background subtraction techniques for gesture segmentation. Then, the handcrafted features were utilized along with Artificial Neural Networks (ANNs) for gesture classification. As depicted in the future work, the factors like occlusions, lighting conditions, and background variations affect the presented model's effectiveness in classification.

In addition, Roy, et al. [20] integrated the hidden markov model and cam-shift tracker for effective gesture recognition. As a future extension, the classification performance can be further improved by incorporating the developed model with other deep learning classifiers. Additionally, Xiao, et al. [21] used Capsule Networks (CapsNet) for alphabetic letter and sign language digit recognition. Hence, the CapsNet model has achieved higher classification accuracy in ISL recognition, but it was computationally complex. Mannan et al. [22] used a hyper-tuned deep CNN model for sign recognition. The conducted experiments confirmed that the deep CNN model obtained higher accuracy compared to the existing state-of-the-art methods. Correspondingly, Fregoso, et al. [23] integrated Particle Swarm Optimizer (PSO) and CNN for feature optimization and sign language recognition. As stated earlier, the CNN model was computationally costly, while experimenting on the larger databases. To address the above-stated issues, a new deep learning based ensemble model is proposed in the current manuscript.

Journal of Theoretical and Applied Information Technology
15th November 2022. Vol.100. No 21

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

## 3. METHODOLOGY

In the hand sign/gesture recognition, the proposed system consists of five phases such as image acquisition: ASL-MNIST and real time SISL databases, Sign/gesture segmentation: K-means clustering with Gaussian blur method, feature extraction: GLCM feature with AlexNet model, feature optimization: stacked autoencoder model and sign/gesture detection: ensemble classifier (combination of MSVM and Naïve Bayes). The block diagram of the proposed system is determined in figure 1.

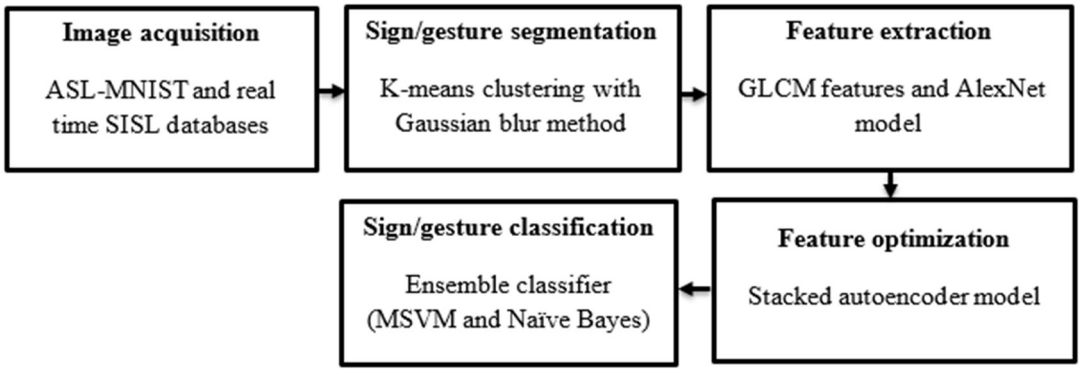


*Figure 1: Block-diagram of the proposed system*

### 3.1. Image acquisition

In this manuscript, the proposed ensemble model's effectiveness is tested on two databases. The ASL-MNIST database includes 34627 gray-scale sign images with a pixel size of 28 × 28. The ASL-MNIST database consists of 24 labeled classes in a range from zero to twenty-five. Classes nine and twenty-five (alphabets J and Z) are eliminated, due to improper gestural movements. The statistical description of the ASL-MNIST database is stated in table 1, and the sample images are represented in figure 2.

*Table 1: The statistical description of the ASL-MNIST database*

| Name | ASL-MNIST details |
|---|---|
| Database format | Comma Separated Values (CSV) file |
| Image size | 28 × 28 |
| Testing images | 6926 |
| Training images | 27701 |
| Total images | 34627 |

Database link: https://www.kaggle.com/datasets/datamunge/sign-language-mnist


*Figure 2: Sample images of ASL-MNIST database*

Journal of Theoretical and Applied Information Technology
15th November 2022. Vol.100. No 21

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

In the real-time SISL database, the images are captured utilizing smart-phone cameras. The parameter specifications followed during image acquisition are given as follows: mobile type: Samsung A51, light: normal day, image pixel size: $1080 \times 2400$ and focal length: f/2.0 aperture. Around 2000 sign images are captured in the real time SISL database which belongs to thirty SISL hand signs of Kannada, Telugu and Tamil languages. The sample images of the real time SISL database is represented in figure 3.

*Figure 3: Sample sign images of the real time SISL database*

### 3.2. Sign/gesture segmentation

After acquiring images from ASL-MNIST and real time SISL databases, the sign/gesture segmentation is accomplished by using K-means clustering. Initially, the acquired images are partitioned into k-number of disjoint clusters or k-number of groups. Further, computes the k-centroids, and then identifies the clusters that have the nearest centroids using data points. In K-means clustering technique, Euclidean distance is used for determining the distance between nearest centroids, where each cluster is determined by its member objects and centroids. The steps involved in K-means clustering are listed as follows:

**1st step:** Initialize some clusters and centroids k [24].

**2nd step:** Compute Euclidean distance $d$ between image pixel and centroids $s_k$ using equation (1).

$$d = \|pixels\ (x,y) - s_k\| \quad (1)$$

**3rd step:** Based on Euclidean distance $d$, the pixel values are assigned to the nearest centroids.

**4th step:** After assigning all pixel values, the position of the centroids is recomputed using equation (2).

$$s_k = \frac{1}{k}\sum_{y\in s_k}\sum_{x\in s_k} pixels\ (x,y) \quad (2)$$

**5th step:** The following steps are repeated until tolerance or error value is satisfied. Then, the Gaussian blur method is used for decreasing the noise from the segmented grayscale images which help to obtain better classification results.

### 3.3. Feature extraction

After segmenting the sign/gesture regions, the AlexNet and GLCM features are applied for extracting feature vectors. Initially, AlexNet model extracts deep feature vectors from the segmented sign/gesture regions, where it consists of 8 pre-defined layers like 5 convolutional and 3 fully-connected layers. The following layers comprise two important functions such as max-pooling and leaky Rectified Linear Unit (ReLU) activation function. The AlexNet model extracts 392 deep feature vectors from the segmented sign/gesture regions [25-27].

Additionally, the GLCM features include 21 techniques for feature extraction such as difference variance, the sum of squares, inverse difference moment normalized, correlations, sum average, maximum probability, difference entropy, dissimilarity, inverse difference, contrast, cluster prominence, variance, homogeneity, information measure of correlation, sum entropy, inverse difference normalized, cluster shades, autocorrelations, entropy, energy, and the sum variance [28-29]. Around 1819 feature vectors are extracted by applying 21 GLCM techniques. Then, the feature level fusion technique is employed to combine 392 deep feature vectors, and 1819 GLCM feature vectors.

### 3.4. Feature optimization

After extracting the feature vectors, the dimensionality reduction is performed utilizing a stacked autoencoder model, which it performed superiorly in feature dimensionality reduction compared to the traditional models. The stacked autoencoder model is a feed forward neural network that consists of numerous hidden layers, an output layer, and an input layer, which are detailed in equations (3) and (4).

$$Z^{(l)} = y^{(l-1)} \times W^{(l)} + b^{(l)} \quad (3)$$

$$y^{(l)} = g(\,Z^{(l)}) \quad (4)$$

Where, $g(.)$ specifies a non-linear activation function, $W^{(l)} \in \mathbb{R}^{n_i \times n_0}$ states matrix of learnable biases $b$, $y^{(L)}$ denotes final layer output, $y^{(l-1)}$ indicates the output of previous layers $l-1$ and input of present layer $l$, $y^{(l)}$ indicates the model's input, $Z^{(l)}$ represents pre-activation layer of vector $l$, and $l \in [1, \dots L]$ indicates $l^{th}$ layer. In this model, the ReLU is applied as an activation function, which significantly improves the model's learning rate and computational effectiveness for better feature dimensionality reduction. In addition to this, the softmax non-linearity function is utilized for obtaining better probability interpretation in the output layer, and it is mathematically depicted in equation (5).

$$Softmax\left(Z^{(L)}\right) = \frac{expZ_k}{\sum_{k=1}^{K} expZ_k} \quad (5)$$

Where, $K$ represent output classes. In the stacked autoencoder model, the cross entropy loss function $Cr$ is employed for dealing with the optimization problems, which is mathematically mentioned in equation (6).

$$Cr = -\sum_{k=1}^{K} \hat{y}_k log(y_k^{(L)}) \quad (6)$$

Where, $\hat{y}_k \in \{0,1\}^k$ denotes encoded labels and $y^{(L)}$ states the model's output. In this article, the deep learning model: stacked autoencoder is used for learning higher dimensional feature vectors. Additionally, the mathematical formula of a stacked autoencoder model with hidden layers is determined in equation (7) [30-31].

$$h_e = a_1(W_e x) \; and \; \hat{x} = a_2(W_d \, h_e) \quad (7)$$

Where, $W_d$ and $W_e$ are matrices, which denote a linear combination of the inputs for decoding and encoding, $\hat{x}$ indicates reconstructed feature vectors, and $x$ specifies input feature vectors. In addition, $h_e$ indicates a bottle-neck layer that considers the low dimensional representation of the feature vectors, and $a_1$ and $a_2$ represents constant values. The hyper-parameter settings of the stacked autoencoder model are listed as follows: maximum iterations are 100, a number of hidden layers is 100, L2 weight regularization is 0.4, sparsity regularization is 4, and sparsity proportion is 0.150. The optimized 198 feature vectors are given to the ensemble classifier for sign/gesture recognition.

### 3.5. Sign/gesture recognition

The optimized feature values are fed as the input to the ensemble classifier for classifying hand signs and gestures. The ensemble classifier integrates MSVM and Naïve Bayes classifiers, and then, the best outcomes are voted out using weighted voting. The Naïve Bayes performs sign/gesture classification based on the maximum-posterior-decision rule. The Naïve Bayes has accomplished with an existing probability $pr$ function and a Gaussian function and it is stated in equations (8) and (9) [32-33].

$$Pr(f_1, f_2, \dots . f_n|C) = \prod_{i=1}^{n} pr(f_i|C) \quad (8)$$

$$Pr(f_i|C_i) = \frac{pr(C_i|f) \times pr(f)}{pr(C_i)} \quad (9)$$

Further, the association possibility is applied for classifying test data $C$, which is mathematically indicated in equation (10).

$$C_n = argmax \; pr(C_t) \prod_{i=1}^{n} pr(f_i|C_z), \text{ Where } t = 1,2 \dots \quad (10)$$

The MSVM comprises two techniques like One against All (1-a-a) and One against One (1-a-1) for sign/gesture classification. First, the 1-a-a technique creates a binary classification method for every class which effectively distinguishes the objects in the same classes, and the result of $i^{th}$ class in 1-a-a technique is compared with the 1-a-1 technique for achieving high output value. In addition, the MSVM classifier generates all possible two class classification methods from the training sets of $i^{th}$ class, but it trains only two out of $i^{th}$ class which results in $i \times (i-1)/2$ classifiers. The mathematical illustration of the MSVM is stated in equations (11-13) [34-35].

$$min\Phi(E, \xi) = 1/2 \sum_{m=1}^{o}(E_m) + C \sum_{i=1}^{o} \sum_{m \neq yi} \xi_i^m \quad (11)$$

$$\left(E_{yi} \times x_i\right) + U_{yi} \geq \left(E_{yi} \times x_i\right) + U_m + 2 - \xi_i^m \quad (12)$$

$$\xi_i^m \geq 0, i = 1,2,3 \dots o, m, yi \in \{1,2,3 \dots k\}, m \neq yi \quad (13)$$

The decision function in MSVM is mathematically determined in equation (14).

$$df(x) = arg\,max[(E_i \times x) + U_i], i = 1,2,3,..k \quad (14)$$

Where, $C$ indicates classes, $o$ specifies training data points, $\xi_i^m$ states slack variables, $yi$

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

states class of training data vectors $x_i$ and $k$ denotes user's positive constant. The hyper-parameters of the ensemble classifier are specified as follows: criterion is Gini, splitter is best, minimum samples spilt is two, maximum depth is none, minimum samples leaf is one, a degree in kernel function is three, tolerance of the termination criteria is 0.1, coast factor is five, and the kernel function is linear. The experimental result of the proposed model on the ASL-MNIST and real time SISL databases is specified in the next phase.

## 4. SIMULATION RESULTS

In this manuscript, the ensemble based stacked autoencoder model's efficacy is analyzed utilizing Matlab 2020 software environment and validated on a computer with configuration: Intel core i9 processor, 4TB hard disk, 8GB random access memory and windows 10 (64-bit) operating system. In this research, the ensemble-based stacked autoencoder model's efficacy is investigated using the evaluation measures like sensitivity, MCC, accuracy, F1-measure and specificity. The evaluation measures: sensitivity and specificity to identify the features of the sign/gesture and background regions. The accuracy is an important evaluation measure in sign/gesture recognition, because it finds how close the obtained results are to the true values. In addition, the parametric value of MCC lies between zero to one, where the ensemble based stacked autoencoder model is effective in the sign/gesture classification, while the parametric value is one. The F1-measure is a harmonic mean of precision and sensitivity values, where the mathematical depiction of the undertaken evaluation metrics: sensitivity, MCC, accuracy, F1-measure and specificity is specified in equations (15-19).

$$Sensitivity = \frac{TP}{TP+FN} \times 100 \quad (15)$$

$$MCC = \frac{TP \times TN - FP \times FN}{\sqrt{(TP+FP)(TP+FN)(TN+FP)(TN+FN)}} \times 100 \quad (16)$$

$$Accuracy = \frac{TP+TN}{TP+TN+FP+F} \times 100 \quad (17)$$

$$F1 - measure = \frac{2TP}{FP+2TP+FN} \times 100 \quad (18)$$

$$Specificity = \frac{TN}{TN+FP} \times 100 \quad (19)$$

Where, TP, TN, FP, and FN state true positive, true negative, false positive and false negative values.

### 4.1. Quantitative evaluation

In this section, the ensemble based stacked autoencoder model's efficacy is evaluated on the ASL-MNIST database in light of sensitivity, MCC, accuracy, F1-measure and specificity. As represented earlier, the ASL-MNIST database includes 34627 gray-scale sign images with a pixel size of $28 \times 28$, and it has 24 labeled classes. The experimental result of the ensemble based stacked autoencoder model on the ASL-MNIST database is represented in table 2. By inspecting table 2, the experimental analysis is performed with various classifiers: Ensemble, naïve Bayes, MSVM and SVM, and optimizers: firefly optimizer, reliefF, infinite and stacked autoencoder. Related to other combination results, the combination: Ensemble classifier with stacked autoencoder model has obtained high classification result with a sensitivity of 99.98%, MCC of 99.95%, the accuracy of 99.96%, F1-measure of 99.80%, and specificity of 99.82% on the ASL-MNIST database. The graphical presentation of the ensemble based stacked autoencoder model on the ASL-MNIST database is represented in figure 4. In this manuscript, the stacked autoencoder significantly optimizes the dimensions of the extracted feature vectors or selects the optimum relevant feature vectors. The incorporation of the stacked autoencoder model in the proposed system effectively reduces the running time and computational complexity.

*Table 2: Experimental results of the ensemble based stacked autoencoder model on the ASL-MNIST database*

| Optimizers | Classifiers | Sensitivity (%) | MCC (%) | Accuracy (%) | F1-measure (%) | Specificity (%) |
|---|---|---|---|---|---|---|
| Firefly | Naïve Bayes | 90.76 | 90.55 | 90.30 | 90.88 | 90.82 |
| Infinite | | 92.78 | 92.20 | 91.76 | 91.27 | 92.08 |
| ReliefF | | 94.80 | 93.80 | 92.85 | 92.77 | 92.10 |
| Autoencoder | | 95.78 | 94.72 | 93.50 | 92.95 | 93.80 |
| Firefly | SVM | 92.90 | 94.22 | 93.68 | 93.13 | 94.44 |
| Infinite | | 94.85 | 95.91 | 94.46 | 94.32 | 94.60 |
| ReliefF | | 96.78 | 96.46 | 94.85 | 95.99 | 96.96 |

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

| | | | | | | |
|---|---|---|---|---|---|---|
| Autoencoder | | 96.90 | 96.85 | 95.60 | 96.78 | 97.94 |
| Firefly | MSVM | 93.98 | 94.28 | 94.25 | 95.07 | 96.38 |
| Infinite | | 95.34 | 95.46 | 95.90 | 96.46 | 97.38 |
| ReliefF | | 96.96 | 97.56 | 96.56 | 97.63 | 98.52 |
| Autoencoder | | 97.72 | 98.90 | 97.58 | 98.64 | 98.82 |
| Firefly | Ensemble | 97.60 | 97.85 | 98.10 | 96.40 | 97.90 |
| Infinite | | 98.90 | 98.80 | 98.28 | 97.94 | 98.18 |
| ReliefF | | 99.12 | 99.48 | 99.18 | 98.86 | 99.40 |
| Autoencoder | | 99.98 | 99.95 | 99.96 | 99.80 | 99.82 |

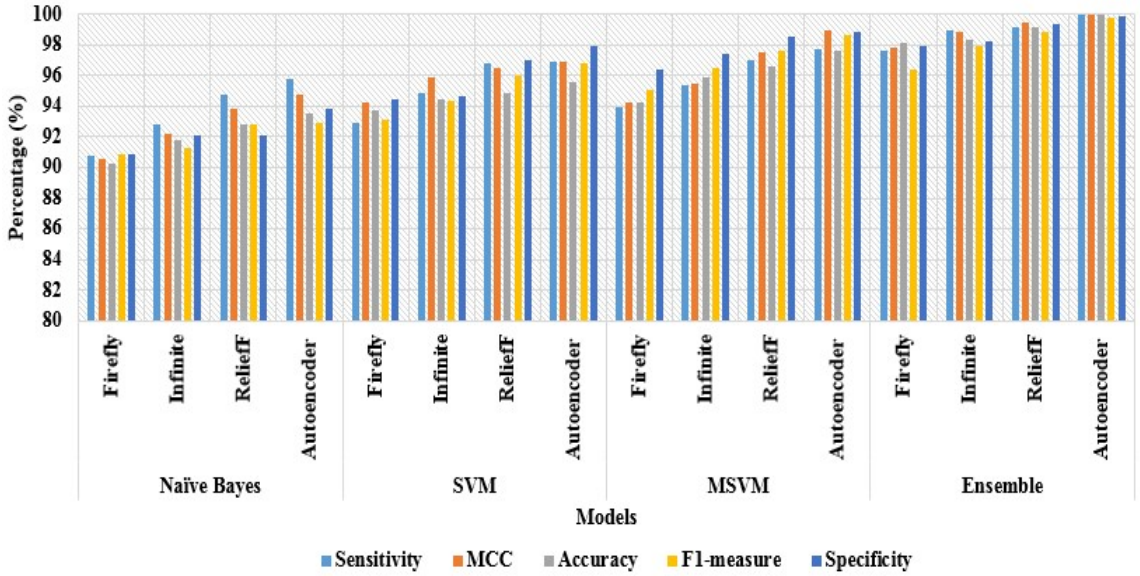


*Figure 4: Graphical validation of the ensemble based stacked autoencoder model on the ASL-MNIST database*

Similar to table 3, the experimental result of the ensemble based stacked autoencoder model on the real time SISL database is given in table 3. The real time SISL database has 2000 sign images with the pixel size of $1080 \times 2400$. As denoted in table 3, the combination: ensemble classifier with stacked autoencoder model has attained a maximum classification performance with 80:20% training and testing of data, and it is better related to other training percentages. By performing cross-validations, the computational time, variance, and bias of the ensemble based stacked autoencoder model is superiorly reduced. Further, the ensemble based stacked autoencoder model has obtained 98.72% of Sensitivity, 98.40% of MCC, 99.08% of Accuracy, 98.68% of F1-measure, and 98.24% of Specificity on the real time SISL database. The graphical validation of the ensemble based stacked autoencoder model on the real time SISL database is represented in figure 5. Related to the individual classifiers, the ensemble classifier makes superior predictions and achieves better classification performance. In addition, the ensemble classifier effectively decreases the dispersion of the model performance.

*Table 3: Experimental results of the ensemble based stacked autoencoder model on the real time SISL database*

| Optimizers | Classifiers | Sensitivity (%) | MCC (%) | Accuracy (%) | F1-measure (%) | Specificity (%) |
|---|---|---|---|---|---|---|
| Firefly | Naïve Bayes | 86.80 | 90.98 | 92.06 | 88.32 | 90.40 |
| Infinite | | 88.72 | 93.18 | 93.26 | 92.24 | 91.82 |
| ReliefF | | 90.98 | 94.36 | 94.54 | 92.88 | 91.92 |
| Autoencoder | | 94.84 | 95.72 | 95.87 | 94.38 | 92.84 |
| Firefly | SVM | 92.46 | 88.78 | 90.85 | 91.92 | 91.44 |
| Infinite | | 94.86 | 91.98 | 94.56 | 93.12 | 92.60 |
| ReliefF | | 95.14 | 92.84 | 94.90 | 94.44 | 93.14 |

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

| | | | | | | |
|---|---|---|---|---|---|---|
| Autoencoder | | 95.38 | 93.62 | 95.42 | 95.36 | 94.98 |
| Firefly | MSVM | 88.48 | 94.48 | 93.88 | 92.80 | 94.04 |
| Infinite | | 93.27 | 95.06 | 94.18 | 94.86 | 95.44 |
| ReliefF | | 94.18 | 95.38 | 95.82 | 95.88 | 96.86 |
| Autoencoder | | 96.45 | 97.90 | 96.18 | 96.30 | 97.82 |
| Firefly | Ensemble | 95.94 | 94.92 | 95.26 | 92.92 | 96.90 |
| Infinite | | 96.34 | 95.68 | 96.68 | 95.94 | 97.06 |
| ReliefF | | 97.26 | 97.38 | 97.42 | 96.96 | 97.44 |
| Autoencoder | | 98.72 | 98.40 | 99.08 | 98.68 | 98.24 |

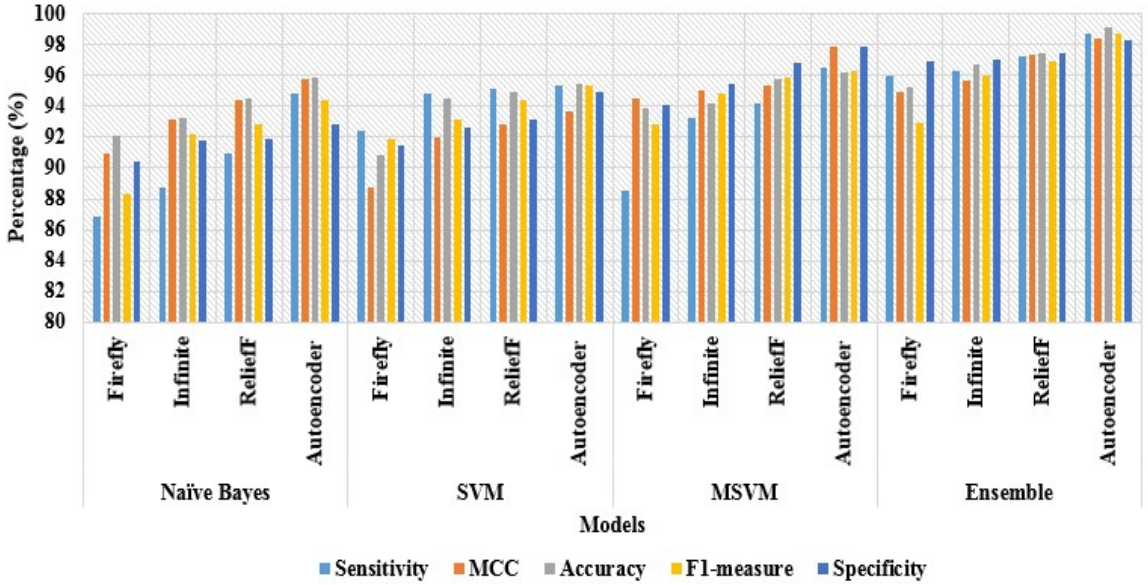


*Figure 5: Graphical validation of the ensemble based stacked autoencoder model on the real time SISL database*

### 4.2. Comparative evaluation

In this section, the comparative evaluation between the prior models and the proposed ensemble based stacked autoencoder model is specified in table 4 and figure 6. Mannan, et al. [22] implemented a hyper-tuned deep CNN model for sign language recognition. The experiments conducted on the ASL-MNIST database demonstrated that the implemented model achieved 99.67% of recognition accuracy. Fregoso et al. [23] integrated the PSO and CNN model for dimensionality reduction and sign language detection. As depicted in the resulting phase, the developed model has achieved 99.80% of recognition accuracy on the ASL-MNIST database. Related to the existing research manuscripts, the ensemble based stacked autoencoder model achieved significant classification performance with a recognition accuracy of 99.96% on the ASL-MNIST database.

As stated in the methodology section, feature optimization and sign/gesture classification are the two integral phases of this research. Where the extracted higher dimensional features are effectively optimized by a deep learning model: stacked autoencoder. The dimensionality reduction diminishes the computational complexity of the proposed system to linear based on order of magnitude and input size. Further, the running time of the ensemble based stacked autoencoder model is 30 and 54.1 seconds on the ASL-MNIST and real time SISL databases, which are minimum compared to the state-of-the-art methods. As represented in the literature section, the major problems: computational cost and complexity are effectively decreased by implementing the ensemble based stacked autoencoder model.

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

*Table 4: Comparative evaluation between the prior models and the proposed ensemble based stacked autoencoder model*

| Models | Recognition accuracy (%) |
|---|---|
| Hyper-tuned deep CNN [22] | 99.67 |
| PSO-CNN II [23] | 99.80 |
| Ensemble based stacked autoencoder | 99.96 |

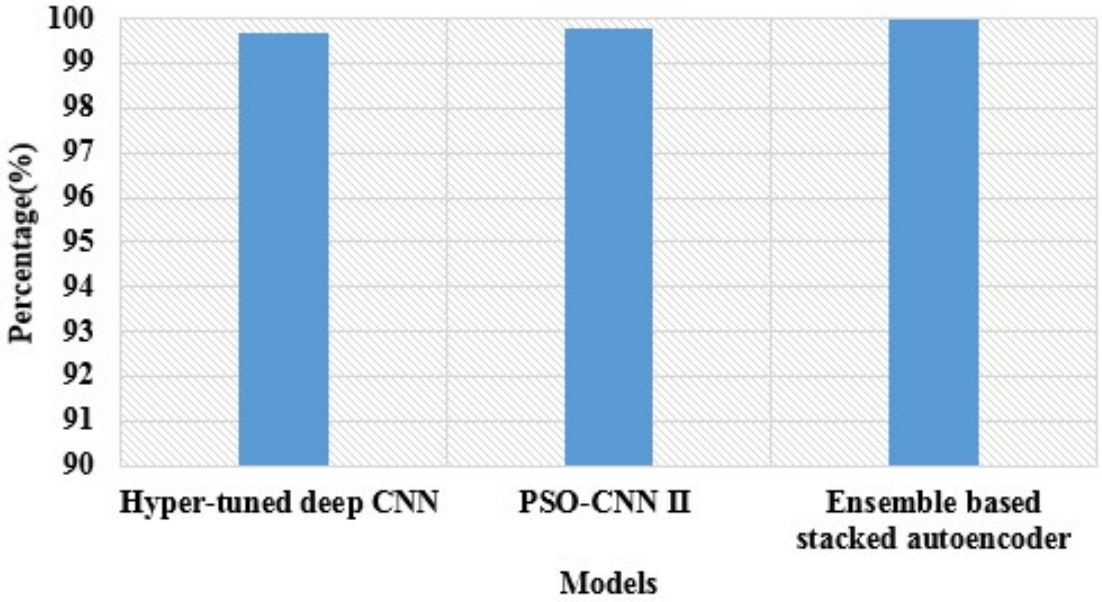


*Figure 6: Graphical comparison between the prior models and the proposed ensemble based stacked autoencoder model*

## 5. CONCLUSION

The ensemble based stacked autoencoder model is implemented in this manuscript for effective sign/gesture recognition. The developed ensemble based stacked autoencoder model includes four important steps, firstly, the sign/gesture regions are segmented from the ASL-MNIST and real time SISL databases using k-means clustering with Gaussian blur technique. Then, the discriminative feature vectors are extracted by implementing GLCM features and AlexNet, which are further dimensionally reduced using a stacked autoencoder model. This action helps in the reduction of computational complexity and running time, and further, the selected features are classified by proposing an ensemble classifier (naïve Bayes and MSVM) and it classifies 24 alphabetical and 30 Indian sign classes on the ASL-MNIST and real time SISL databases. In the quantitative evaluation phase, the undertaken evaluation measures like sensitivity, MCC, accuracy, F1-measure and specificity demonstrated the effectiveness of the ensemble based stacked autoencoder model. The developed model has achieved 99.96% and 99.08% of classification accuracy on the ASL-MNIST and real time SISL databases. Further, the ensemble based stacked autoencoder model has shown superior performance using computational complexity and running time. In a real time sign/gesture recognition, the proposed model fail to meet the requirements, especially in the grammatical aspects of continuous signs. Therefore, as the future extension, a novel deep learning classifier can be incorporated with the ensemble based stacked autoencoder model to further improve sign/gesture recognition on the large unstructured databases.

## REFERENCES:

[1] J. Joy, K. Balakrishnan, and M. Sreeraj, "SignQuiz: a quiz based tool for learning fingerspelled signs in Indian sign language using ASLR", *IEEE Access*, Vol. 7, 2019, pp. 28363-28371.

[2] T. Raghuveera, R. Deepthi, R. Mangalashri, and R. Akshaya, "A depth-based Indian sign language recognition using microsoft Kinect", *Sādhanā*, Vol. 45, No. 1, 2020, pp. 1-13.

[3] J. Gangrade, J. Bharti, and A. Mulye, "Recognition of Indian sign language using ORB with bag of visual words by Kinect sensor", *IETE Journal of Research*, 2020, pp.1-15.

[4] R. Gupta, and A. Kumar, "Indian sign language recognition using wearable sensors and multi-label classification", *Computers & Electrical Engineering*, Vol. 90, 2021, pp. 106898.

[5] R. Gupta, and S. Rajan, "Comparative analysis of convolution neural network models for continuous Indian sign language classification", *Procedia Computer Science*, Vol. 171, 2020, pp. 1542-1550.

[6] S.G. Praveena, and C. Jayasri, "Recognition and Translation of Indian Sign Language for Deaf and Dumb People", *International Journal of Information and Computing Science*, Vol. 6, 2019.

[7] G.A. Rao, and P.V.V. Kishore, "Selfie video based continuous Indian sign language recognition system", *Ain Shams Engineering Journal*, Vol. 9, No. 4, 2018, pp. 1929-1939.

[8] N. Aloysius, and M. Geetha, "Understanding vision-based continuous sign language recognition", *Multimedia Tools and Applications*, Vol. 79, No. 31, 2020, pp. 22177-22209.

[9] D.A. Kumar, A.S.C.S. Sastry, P.V.V. Kishore, and E.K. Kumar, "3D sign language recognition using spatio temporal graph kernels", *Journal of King Saud University-Computer and Information Sciences*, 2018.

[10] G.A. Rao, and P.V.V. Kishore, "Selfie sign language recognition with multiple features on adaboost multilabel multiclass classifier", *Journal of Engineering Science and Technology*, Vol. 13, No. 8, 2018, pp. 2352-2368.

[11] M. Jebali, A. Dakhli, and M. Jemni, "Vision-based continuous sign language recognition using multimodal sensor fusion", *Evolving Systems*, Vol. 12, No.4, 2021, pp.1031-1044.

[12] N. Krishnaraj, M.G. Kavitha, T. Jayasankar, and K.V. Kumar, "A Glove based approach to recognize Indian Sign Languages", *International Journal of Recent Technology and Engineering (IJRTE)*, Vol. 7, 2019, pp. 1419-1425.

[13] J. Imran, and B. Raman, "Deep motion templates and extreme learning machine for sign language recognition", *The Visual Computer*, Vol. 36, No. 6, 2020, pp. 1233-1246.

[14] B. Subramanian, B. Olimov, S.M. Naik, S. Kim, K.H. Park, and J. Kim, "An integrated mediapipe-optimized GRU model for Indian sign language recognition", Scientific Reports, Vol. 12, No. 1, 2022, pp. 1-16.

[15] J. Gangrade, and J. Bharti, "Vision-based hand gesture recognition for Indian sign language using convolution neural network", *IETE Journal of Research*, pp. 1-10, 2020.

[16] A. Kumar, and R. Kumar, "A novel approach for ISL alphabet recognition using Extreme Learning Machine", *International Journal of Information Technology*, Vol. 13, No. 1, 2021, pp. 349-357.

[17] S. Katoch, V. Singh, and U.S. Tiwary, "Indian Sign Language recognition system using SURF with SVM and CNN", *Array*, Vol. 14, 2022, pp. 100141.

[18] A. Wadhawan, and P. Kumar, "Deep learning-based sign language recognition system for static signs", *Neural computing and applications*, Vol. 32, No. 12, 2020, pp. 7957-7968.

[19] P.C. Badhe, and V. Kulkarni, "Artificial neural network based Indian sign language recognition using hand crafted features", *11th International Conference on Computing, Communication and Networking Technologies (ICCCNT), IEEE*, 2020, pp. 1-6.

[20] P.P. Roy, P. Kumar, and B.G. Kim, "An efficient sign language recognition (SLR) system using Camshift tracker and hidden Markov model (hmm)", *SN Computer Science*, Vol. 2, No. 2, 2021, pp. 1-15.

[21] H. Xiao, Y. Yang, K. Yu, J. Tian, X. Cai, U. Muhammad, and J. Chen, "Sign language digits and alphabets recognition by capsule networks", *Journal of Ambient Intelligence and Humanized Computing*, Vol. 13, No. 4, 2022, pp. 2131-2141.

[22] A. Mannan, A. Abbasi, A.R. Javed, A. Ahsan, T.R. Gadekallu, and Q. Xin, "Hypertuned deep convolutional neural network for sign language recognition", *Computational Intelligence and Neuroscience*, 2022.

[23] J. Fregoso, C.I. Gonzalez, and G.E. Martinez, "Optimization of convolutional neural networks architectures using pso for sign language recognition", *Axioms*, Vol. 10, No. 3, 2021, pp. 139.

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

[24] R.C. Hrosik, E. Tuba, E. Dolicanin, R. Jovanovic, and M. Tuba, "Brain image segmentation based on firefly algorithm combined with k-means clustering", *Stud. Inform. Control*, Vol. 28, No. 2, 2019, pp. 167-176.

[25] M. Lv, G. Zhou, M. He, A. Chen, W. Zhang, and Y. Hu, "Maize leaf disease identification based on feature enhancement and DMS-robust alexnet", *IEEE Access*, Vol. 8, 2020, pp. 57952-57966.

[26] H. Alaskar, N. Alzhrani, A. Hussain, and F. Almarshed, "The implementation of pretrained AlexNet on PCG classification", *In International conference on intelligent computing,* Springer, Cham, pp. 784-794, 2019.

[27] S. Sun, T. Zhang, Q. Li, J. Wang, W. Zhang, Z. Wen, and Y. Tang, "Fault diagnosis of conventional circuit breaker contact system based on time-frequency analysis and improved AlexNet", *IEEE Transactions on Instrumentation and Measurement*, Vol. 70, pp.1-12, 2020.

[28] A. Tassi, and M. Vizzari, "Object-oriented lulc classification in google earth engine combining snic, glcm, and machine learning algorithms", *Remote Sensing*, Vol. 12, No. 22, pp. 3776, 2020.

[29] P.K. Mall, P.K. Singh, and D. Yadav, "December. Glcm based feature extraction and medical x-ray image classification using machine learning techniques", *In 2019 IEEE Conference on Information and Communication Technology*, pp. 1-6, 2019.

[30] M. Yu, T. Quan, Q. Peng, X. Yu, and L. Liu, "A model-based collaborate filtering algorithm based on stacked AutoEncoder", *Neural Computing and Applications*, Vol. 34, No. 4, pp. 2503-2511, 2022.

[31] A. Sagheer, and M. Kotb, "Unsupervised pre-training of a deep LSTM-based stacked autoencoder for multivariate time series forecasting problems", *Scientific reports*, Vol. 9, No. 1, pp. 1-16, 2019.

[32] S. Chen, G.I. Webb, L. Liu, and X. Ma, "A novel selective naïve Bayes algorithm", *Knowledge-Based Systems*, Vol. 192, 2020, pp. 105361.

[33] H. Zhang, L. Jiang, and L. Yu, "Attribute and instance weighted naive Bayes", *Pattern Recognition*, Vol. 111, pp. 107674, 2021.

[34] A. Krishnaswamy Rangarajan and R. Purushothaman, "Disease classification in eggplant using pre-trained VGG16 and MSVM", *Scientific reports*, Vol. 10, No. 1, 2020, pp. 1-11.

[35] Y. Guo, Z. Zhang, and F. Tang, "Feature selection with kernelized multi-class support vector machine", *Pattern Recognition*, Vol. 117, pp. 107988, 2021.