Pertanika J. Sci. & Technol. 29 (3): 1937 - 1961 (2021)

PERTANIKA JOURNALS

SCIENCE & TECHNOLOGY

Journal homepage: http://www.pertanika.upm.edu.my/

An Analysis of Emotional Speech Recognition for Tamil Language Using Deep Learning Gate Recurrent Unit

Bennilo Fernandes* and Kasiprasad Mannepalli

Department of Electronics & Communication Engineering, Koneru Lakshmaiah Education Foundation, Guntur, Andhra Pradesh, India

ABSTRACT

Designing the interaction among human language and a registered emotional database enables us to explore how the system performs and has multiple approaches for emotion detection in patient services. As of now, clustering techniques were primarily used in many prominent areas and in emotional speech recognition, even though it shows best results a new approach to the design is focused on Long Short-Term Memory (LSTM), Bi-Directional LSTM and Gated Recurrent Unit (GRU) as an estimation method for emotional Tamil datasets is available in this paper. A new approach of Deep Hierarchal LSTM/BiLSTM/GRU layer is designed to obtain the best result for long term learning voice dataset. Different combinations of deep learning hierarchal architecture like LSTM & GRU (DHLG), BiLSTM & GRU (DHBG), GRU & LSTM (DHGL), GRU & BiLSTM (DHGB) and dual GRU (DHGG) layer is designed with introduction of dropout layer to overcome the learning problem and gradient vanishing issues in emotional speech recognition. Moreover, to increase the design outcome within each emotional speech signal, various feature extraction combinations are utilized. From the analysis an average classification validity of the proposed DHGB model gives 82.86%, which is slightly higher than other models like DHGL (82.58), DHBG (82%), DHLG (81.14%) and DHGG (80%). Thus, by comparing all the models DHGB gives prominent outcome of 5% more than other four models with minimum training time and low dataset.

Keywords: Bi-Directional Long Short-Term Memory, emotional recognition, Gated Recurrent Unit, Long Short-Term Memory

ARTICLE INFO

Article history:
Received: 04 February 2021
Accepted: 01 April 2021
Published: 31 July 2021

DOI: https://doi.org/10.47836/pjst.29.3.37

E-mail addresses:
bennij05@gmail.com (Bennilo Fernandes)
mkasiprasad@gmail.com (Kasiprasad Mannepalli)
* Corresponding author

ISSN: 0128-7680
e-ISSN: 2231-8526

INTRODUCTION

Deep learning is applied often for identify a large course of various layers of cells generated by neural networks. For many years neural network is utilized in emotional recognition, in brief. Deep architecture mechanism is practiced in huge area may even though there are some other tech innovations in such development and rapid, the significant rise in processing method using GPUs has done it much easier to analyzes bigger sets of data (Kumar et al., 2017; Mannepalli et al., 2016a; Li et al., 2014). Deep learning models have several applications in various fields, but in the domain of speech processing in specific, numerous accomplishments were seen. For instance, a machine learning design called Convolutional Neural Networks (CNNs) is designed to replicate the actions of the cerebral system (Rao et al., 2018; Schwarz et al., 2015; Ravanelli et al., 2016). The issue of speaker identification contains data from time-series. Feedforward neural networks are unilateral in which the outcomes of one layer are directed to the next layer. Such feedforward networks are unable to preserve past data. Furthermore, so if Deep Neural Network (DNN) is optimized to evaluate speech recognition predefined challenges are caused, such as separate talking rates and spatial dependencies (Srivastava et al., 2014; Mannepalli et al., 2016b; Ioffe & Szegedy, 2015). DNNs can hardly design review the existing acoustic screen windows that they are unable to describe various talking rates (Abdel-Hamid et al., 2014; Sastry et al., 2016; Zhang et al., 2016; Rao & Kishore, 2016). A whole other system course that includes sequences in the input units to anticipate the value of a particular period venture by maintaining the relevant data at the previous iteration is the Recurrent Neural Network (RNN). The whole process facilitates RNNs to control different talking rates.

In this article, the characteristics of LSTM, BiLSTM, and GRU were analyzed for emotional voice recognition implemented to Tamil emotional information set and used a suitable clustering technique. To classify data sets, distinct user defined classifiers are based end-to-end utilizing Connectionist Temporal Classification (CTC) (Liu et al., 2014). The integrated research in the community of emotional voice recognition and machine learning. The methodology gives the brief introduction about RNN and its layers. Followed by the feature extraction variables that are implemented. Then the dataset collection and its details were described in brief perspective. With the collected data research performance and the outcome are reported and followed by the conclusion and discussions are provided by comparing the other models.

MATERIALS AND METHODS

Historically, conceptual models are based for computer vision. Usually, iterative methods consist of Maximum-A-Posteriori (MAP) assessment, Gaussian Mixture Models (GMMs), and Hidden Markov Models (HMM) (Kishore et al., 2016; Ravanelli et al., 2017; Zhou et al., 2016). Such existing methods require significant knowledge (i.e., understanding of a specific word) and fully Automated Speech Recognition (ASR) document preprocessing. For several problems, dialect simulation is crucial, including such speaker recognition, machine translation or channels offered. To plot patterns to feature vectors, both RNNs and LSTMs have also been used. As outlined, LSTM communication techniques are seen to be greater than regular modeling RNNs for Context Free Language (CFL) and Context Sensitive Language (CSL) (Hochreiter & Schmidhuber, 1997; Graves et al., 2013; Krizhevsky et al., 2012). Additionally, Graves et al. (2013) also introduced deep LSTM RNNs and assessed the voice recognition structure. On a led to a large language processing problem, the quality of LSTM, RNN and DNN configurations has been analyzed and compared (Sak et al., 2014; Chen et al., 2015; Weninger et al., 2015). A few tests are done also on TIMIT voice information source utilizing BiLSTM, deep BiLSTMs, RNNs, and modified configurations. BiLSTM systems were used during the phonology training set. Outcomes have also established that symmetric LSTM connection information unlike nonlinear LSTM and conventional RNNs on clip-wise phonetic categorization.

The classification data demonstrate that bi - directional LSTM is an appropriate design for speaker recognition under which data sets is essential. Recurrent neural networks of Gated Recurrent Units (GRU) have been created (Erdogan et al., 2015; Eyben et al., 2013; Pascanu et al., 2013). GRUs has some similarity to LSTMs, both of which have been modelled to deal with long-term dependence. GRUs do, even then, get a fewer component unlike LSTMs. All other designs were used for harmonic soundtrack simulation and for voice recognition projects. The findings demonstrate which GRUs are highly useful as LSTMs. The method is linked, which uses a symmetric RNN to accelerate up the quality of voice recognition. To assess and analyze the outcomes of normal appearing channels, notably bi - directional RNN, bidirectional LSTM, and bidirectional GRU, shows better outcome for the collected Tamil emotional database.

Recurrent Neural Network (RNN)

The training algorithm composed of cells is RNN or Recurring Neural Network. As everyone and every nerve cell is using its inner memory for storing records of various instruction, it is indeed primarily important when analyzing data sets. In RNN, with exception of preceding neural networks in which past simulations output must be identified in ability to forecast the next results, the sector relies on supplied data. It is possible to think of RNN also as system which conveys all arithmetic or remembers every sequence that was occurred so far What this implies exactly would be that input signals are considered by RNN one would be the recent information, and the prior computational complexity behave as the next input (Jozefowicz et al., 2015; Chung et al., 2014; Laurent et al., 2016). It includes an input data, neurons in the hidden layer, like many other neural networks.

LSTM

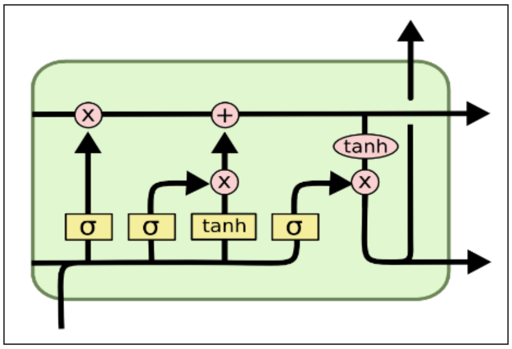


Figure 1. Sequential LSTM layer internal architecture

The intense active learning for storing significant data using Neural Network have not ever attains the optimum level of the system. This is majorly when the regressive procedure known decaying slope, the relative error disappears. In 1997, Hochreiter and Schmidhuber analyzed the problem of back propagation and made a new algorithm named Long-Short Term Memory (LSTM) for Neural Networks. In LSTM layer, there have been 4 new hidden layers introduced to the neural network and titled gates (Figure 1) (Weninger et al., 2015).

BiLSTM

Its efficiency at period "t" in BiLSTM itself is not like past because next sections including its pattern in a singular segment. Such as two parallel RNNs, symmetric RNNs, it only means going forward, or it ends up going downward and computes the mutual production between both RNNs based on everyones previous hidden. In this study, humans then use multilayered idea of the LSTM model, using two-layer system with both primitive and the forward passage in our procedure. The total principle of its recommended layered BiLSTM.

The interior design can be seen in the given Figure 2, which symbolizes the bidirectional RNN preprocessing step and combines this same internal layer including forward and regressive passage in the output units (Graves et al., 2013). The system is verified on 50% information, which again is divided again from learning process and utilize pass-entropy to calculate the failure rate also in test dataset. Adam optimization with 0.001 training data can be used for production efficiency. Either forward or regressive passage throughout the deep BiLSTM system consists of materials that deepen everyones system to measure the

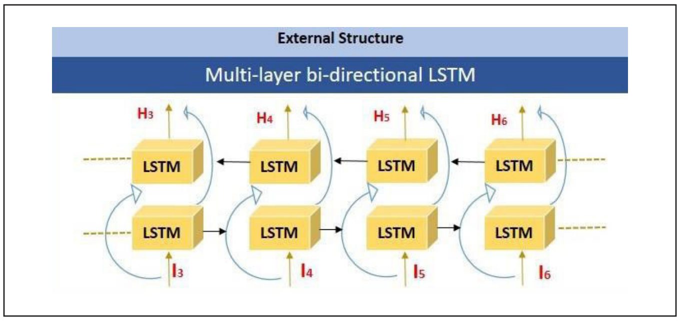


Figure 2. BiLSTM layer internal architecture

outcome in moment again from previous then next pattern as its system was performing in both paths.

GRU

RNNs constitute another very role to fulfil capable of learning short and long-term voice constraints. After all, RNNs could even possibly obtain spatial features in a vibrant way, enabling this same system to openly calculate the type of relevant data to be used for every other sequence of the moment. And these so-called gated RNNs, where its main premise is to incorporate a stacking method to help support this same data exchange thru the varying time stages, are a popular pattern.

Shortest path problems were also remedied inside this design relatives by developing better "alternate routes" where the patterns could even disable appropriate information stages shown in Figure 3. The latest book system called GRU, which is premised only on four multiplier doors, will have spurred to a notable effort to clarify LSTMs. The conventional GRU building is characterized, through specific, by given Equations 1-4:

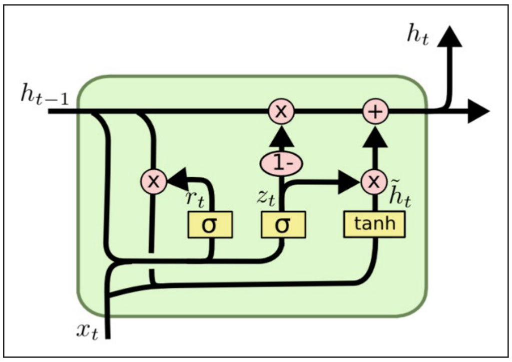


Figure 3. GRU layer internal architecture

$$z_t = \sigma(W_z x_t + U_z h_{t-1} + b_z) \qquad [1]$$

$$r_t = \sigma(W_r x_t + U_r h_{t-1} + b_r) \qquad [2]$$

$$\tilde{h}_t = tanh(W_h x_t + U_h(h_{t-1} \Theta\, r_t) + b_h) \qquad [3]$$

$$h_t = z_t \,\Theta\, h_{t-1} + (1 - z_t) \,\Theta\, \tilde{h}_t) \qquad [4]$$

In which, in both, z_t and r_t were also formulas relating to the official release and refresh gates, even as h_t is really the scale parameter for both the existing time t. There are element-wise algebraic expressions indicated with Θ. Logistic sigmoid $\sigma\,(.)$ operates seem to be the detections between both gates, which restrict z_t and r_t to ideals scale between 0 and 1. The existing original input x_t (e.g., a variable of function generators) feeds the system, whereas the method variables are now the matrix W_z, W_r, W_h (the feed-forward connexions) and U_z, U_r, U_h (the repeated strength training). Eventually, the design requires b_z, b_r, b_h adaptable partiality matrices, which are introduced prior to applying the variations.

Just like illustrated in Equation 3, intermolecular interaction between both the gene transcription h_{t-1} and the candidate running level h_t is 1d, the present state node $\widetilde{h_t}$. This same measurement variables are set by the z_t formative assessments, which chooses what else their transactions would be updated by components. The critical feature besides learning these dependencies was that sequential imputation. In reality, unless z_t is near to just one, this same opening area is held constant and therefore can remain constant after an unreasonable sequence of iterations in moment. From other side, if z_t is near to 0, this same system strongly favors h_t, which varies depending extra strongly mostly on existing hidden and output states closer to it. The assert of the applicant $\widetilde{h_t}$.

Feature Extraction

Mel Frequency Cepstral Coefficients (MFCC). MFCC is determined by the characteristics of listening in the human ear, which simulates the human auditory system using a nonlinear frequency unit. The Fast Fourier Transform (FFT) technique is optimally used to transform, as explained in Equation 5, each sample frame from the time domain into the frequency domain.

$$S[k] = \sum_{n=0}^{N-1} s[n].e^{\frac{-j2\pi nk}{N}}, 0 \leq k \leq N-1 \qquad [5]$$

The mel filter bank is composed of overlapping triangular filters with the cutoff frequencies determined by the two adjacent filters' centre frequencies. The filtration has centre frequencies linearly distributed, and fixed mel scale bandwidth. The logarithm seems to have the impact, mentioned in Equation 6, of shifting multiplier into addition.

$$F[m] = \log \sum_{n=0}^{N-1} |x[k]|^2 H_m[k], 0 \leq m \leq M \qquad [6]$$

Ultimately, to find the MFCC, the Discrete Cosine Transform (DCT) of the log wavelet packet energy is computed as Equation 7.

$$c[n] = \sum_{m=1}^{M} s[n].e^{\frac{-j2\pi nk}{N}}, 0 \leq k \leq N-1 \qquad [7]$$

MFCC Delta

MFCC Delta, also known as variance and maximum speed coefficients. The features of MFCC vector explains only the power spectral functions of single frames, but in speech data the information will be obtained in dynamic values and more variation in features, what the trajectories of the MFCC features extracted are done with over time. It gives an out turns that calculating and appending the MFCC trajectories to the vectors of real and original features increases the performance of ASR by quite a bit (if we have 12 MFCC coefficients, we would also get 12 delta coefficients, which would combine to give a length 24 feature vector). The following Equation 8 is employed to calculate the delta coefficients.

$$d_t = \frac{\sum_{n=1}^{N} n(c_{t+n} - c_{t-n})}{2\sum_{n=1}^{N} n^2} \qquad [8]$$

Where d_t is a delta coefficient, t frames are computed in terms of the static coefficients c_{t+n} to c_{t-n}. A typical value for N is 2.

The Bark Scale

It is based on the key throughput idea, is predictable underneath 500 Hz. e Bark scale outcomes from portraying an entire band of wavelengths with sequences of critical bands and not allowing to merge them. The Bark 1 to 24 numbers are the 24th critical band in the proceedings. Equivalent Rectangular Bandwidth ERB respective logarithmic and sequential; that every dimension is like Bark scale as it also offers an approximation of bandwidths of high noise filters, and therefore utilizes rectangular (unachievable recognition) band-pass filters to efficiently optimize filter modelling. The case hardening conversion would be between ERB and Hertz.

Spectral Kurtosis

The component associated with the execution of extracting features is spectral kurtosis, but it symbolizes the statistical relationship from both voice samples. Throughout the transmissions, the spectral kurtosis could still be described as the value of kurtosis of the variables of voice, and therefore is described as Equation 9,

$$F_5 = \frac{a_4\{S^*(m), S^*(m), S^*(m), S^*(m)\}}{a_2\{S^*(m), S^*(m)\}^2} \qquad [9]$$

During which $S^*(m) \in \{S(m), S^c(m)\}$ the complicated conjugate of the process parameters $S^c(m)$ is demonstrated by $S(m)$ as well as the accumulated fourth and second order are stated by a_4 and a_2.

Spectral Skewness

The spectral skewness demonstrates the irregularities in the spectrum ‘s distribution of the voice signal on it is average rating. The spectral skewness further assumes the energy level of it is spectrum via transfer. Unless the energy size is low upon this distribution left side, it will be very strong if the spectral variable of skewness contains its speech signal.

Dataset

The emotional voice signals are recorded through mobile apps for training and research. All inputs are captured in 44KHz frequency mono signal. The samples collected were utilized for the simulation purpose. Speech information is obtained from 10 individual male and female speakers individually. Every speaker has been asked to utter 10 times each sentence in different emotions like anger, happy, sad, fear, disgust, neutral and boredom. Both male and female speakers report a total of 1400 emotional speech data samples. These samples were taken into consideration for this design flow analysis. A sentence-based samples were recorded by students of arts. For testing purpose, the samples were collected with co-working faculty to identify their emotions during their counselling period. Totally 50 samples were collected with same 44KHz through same mobile apps. Thus these 50 samples were tested to identify the emotions of working faculty. Since the training data base were collected by the professional actors, taking that Tamil emotional data as base the testing emotion database can be identified with more accuracy and perfection.

RESULTS AND DISCUSSION

With sequential data input, the emotional speech database was analyzed in this design layer. The speech signal was converted to LSTM / BiLSTM / GRU Layers as sequential vectors and then passed to them. The MFCC, MFCC delta, Bark spectrum, Spectral kurtosis & Spectral Skewness are the extraction characteristics selected for this design analysis (Figure 4). For testing and training, all the characteristics were examined and concatenated for each speech data to identifies its mean and standard deviation. The vector feature per sequence is assigned to 20 and total number of feature overlapping is 10. With these characteristics the evaluation for different design layer structures that have been fixed. Adam is the optimizing algorithm used here. The Adam optimization algorithm is applied to back-propagation, which has recently seen wider adoption for deep learning applications in computer vision and artificial intelligence processing.

Integration, some of the common features of Adam, is directly forwarded for experimentation. Effectiveness in computation. Tiny specifications for recollection. Wavelet transform for adjusting patterns diagonal direction. Well suited for problems with information- and/or parameter-size. Good for goals that are non-stationary. For very noisy/ or scattered gradients, an effective algorithm. Hyper-parameter interpretation is user-friendly and usually includes minor changes. Optionally, during each single era, the data must optimize the training weights numerous times. The volume of material which is included in almost every transformation in sub-epoch weight is known as the size of the batch. For example, with a 50-voice test set, an entire batch size would be 1000, a 500 or 200 or 100 mini batch size, and batch size will define the deep function of training and testing of data, thus mini batch size is set to 250 and for the evaluation, the number

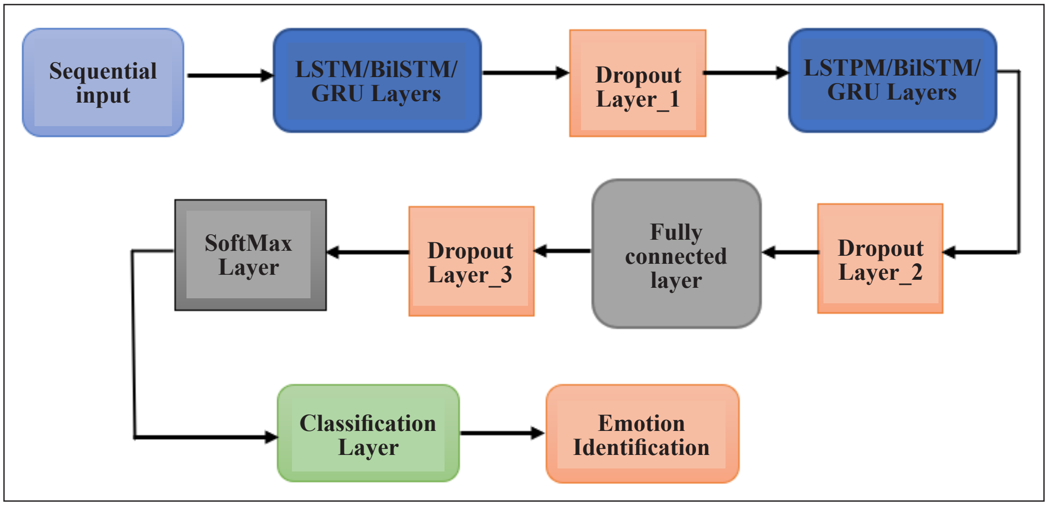


Figure 4. Proposed design flow architecture

of hidden layers is 500 and the initial learning rate is 0.005 and the max epoch is 10. Well after the epoch increases, the iteration can increase the efficiency by continuously training data, but the accuracy and loss during iteration remain the same. The accuracy level of the training dataset after 10 epochs has not been modified. The timeline for the learning rate is piecemeal. Dropout is a method that addresses both problems. This prevents overfitting and provides a way to effectively combine numerous different neural networks exponentially. The word dropout refers to the dropping out of units in a neural network (hidden and visible). In the simplest case, each unit is maintained with a fixed probability p, independent of other units, where p can be selected using a validation set or simply set to 0.5, which seems to be almost optimal for several networks and operations. The optimal retention probability, however, for the input units is generally closer to 1 than to 0.5. For design layer analysis three dropout layers were accomplished after each LSTM /BiLSTM /GRU. The probability values are 0.5 each. The LSTM /BiLSTM /GRU design layer was analyzed by fixing all these parameters.

Deep Hierarchal LSTM & GRU (DHLG) Model

As mentioned before the input speech signal is converted into the sequential data and processed to the dropout layer. The performance of the models is analyzed to reach a conclusion that DHLG model generates confusion matrix with 10-fold cross validation. Since cross folding is random each evaluation output shows different accuracy level a mean of 5 evaluation was considered for DHLG accuracy rate.

Among the average 10 folds cross valuation fold 2 and 4 shows 88.6% of accuracy and fold 3 and 7 shows 85.72% of accuracy, where other folds also show better performance of accuracy around 65 -80% (Figure 5). In the testing phase 50 samples of emotions were given as input for analysis of emotional recognition. From the 5 evaluation the best and higher accuracy level obtained in DHLG is 81.1%.

Now by analyzing the time factor the five evaluation the time taken to training and evaluation of classification timings were considered and shown in Figure 6. While taking the mean value it is clear that for training of DHLG takes around 7.11 mins and to evaluate the classification it takes around 1.06 mins.

From Figure 7 the accuracy level in each simulation is established. As the cross-validation folds are random the accuracy level changes randomly. But it lies in the range of 76 to 82. In each simulation the training time and the evaluation time also varies, but only seconds of variation can be identified. Among the 5 simulation results, in fifth iteration higher range of results is identified i.e., 81.14%.

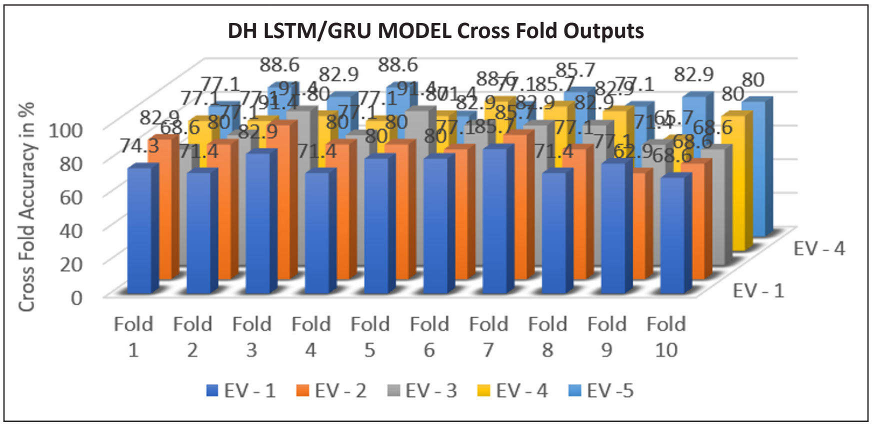


Figure 5. DHLG Cross fold output for multiple evaluation

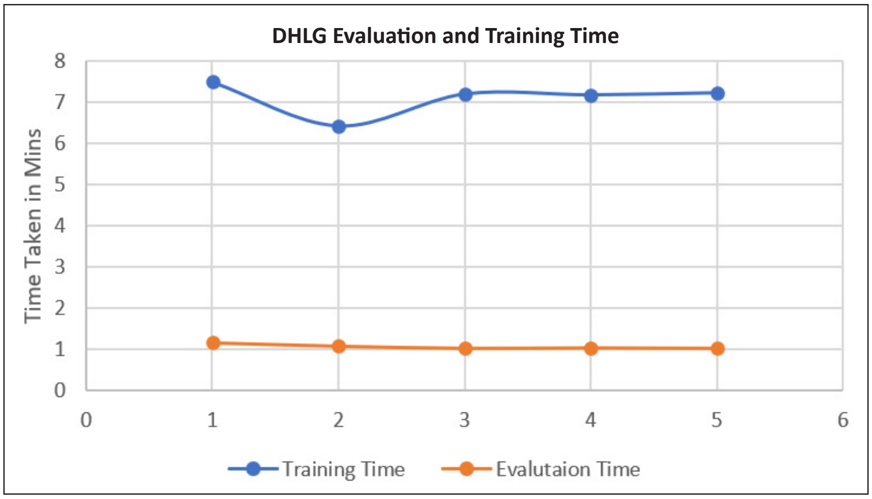


Figure 6. DHLG performance of evaluation time and training time

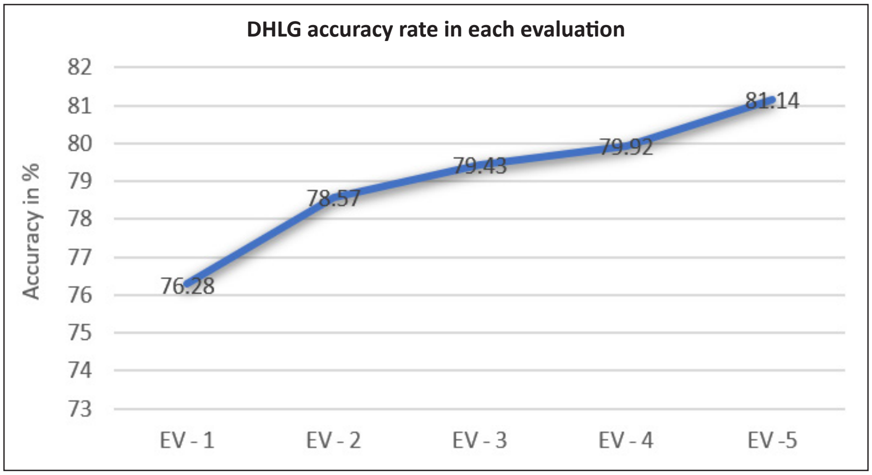


Figure 7. DHLG accuracy rate for 5 evaluations

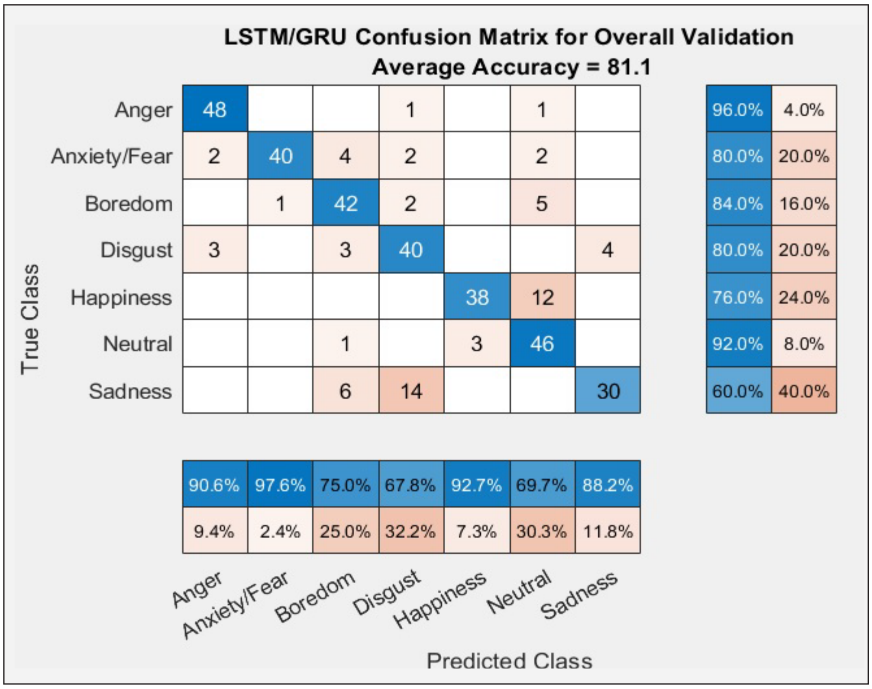


Figure 8. Cross fold confusion Matrix for DHLG

Finally, by considering the higher accuracy level iteration, Figure 8 clearly shows that for the given input Tamil database DHLG model gives 81.1% of efficiency. In the confusion matrix, emotions like anger and neutral gives higher rate of 96% and 92%. As this model also lags in other emotional states. Happy and Neutral emotions lags in DHLG model. Only 76% and 60% of accuracy is obtained in both states and shows lowest of all emotion recognition. Fear, Boredom and Disgust shows 80% and 84% of accuracy.

Deep Hierarchal BiLSTM & GRU (DHBG) Model

The DHBG models is analyzed to reach a conclusion that DHBG technique generates confusion matrix with 10-fold cross validation as final classification output. As like DHLG max 5 times the simulation is evaluated to find the consistent in accuracy level. Among 5 simulation the average 10 folds cross valuation fold 3 shows 90.02% of accuracy and fold 3 and 9 shows 82% of accuracy, where other folds also show better performance of accuracy around 72 -80% (Figure 9). In the testing phase same 50 samples of data used in DHLG is utilized for analysis. Each iteration there is a small variation in identification of emotional recognition. The average accuracy level for 5 evaluation is around 82.0%.

Now by analyzing the time factor the five evaluation the time taken to training and evaluation of classification timings were considered from Figure 10. While taking the mean value it is clear that for training of DHBG takes around 16.33 mins and to evaluate the classification it takes around 1.184 mins.

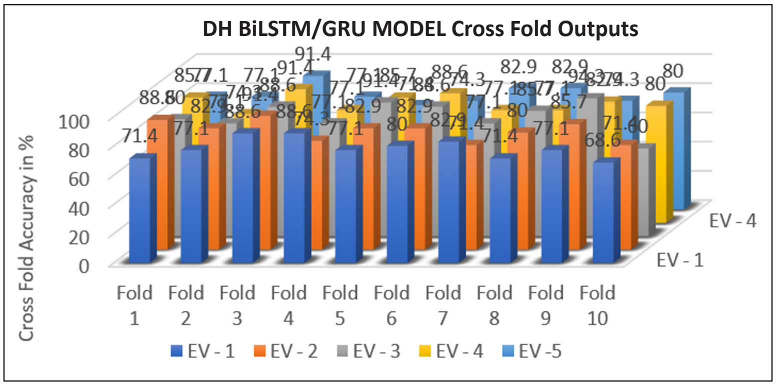


Figure 9. DHBG Cross fold output for multiple evaluation

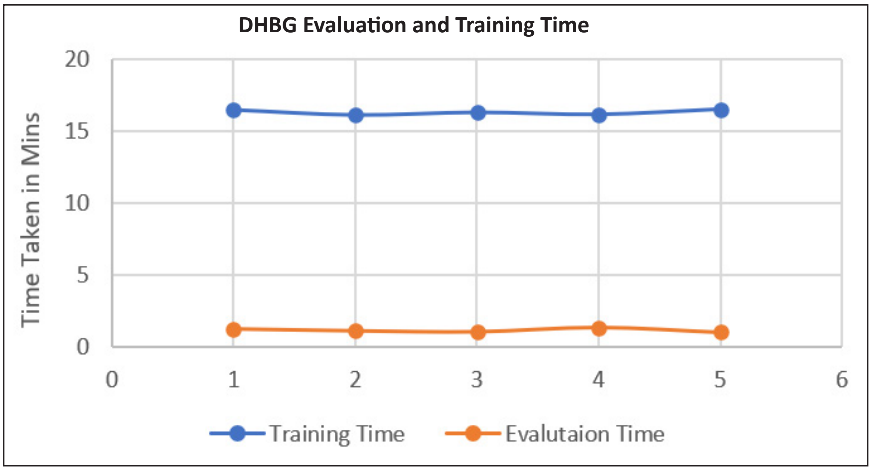


Figure 10. DHBG Performance of Evaluation time and Training Time

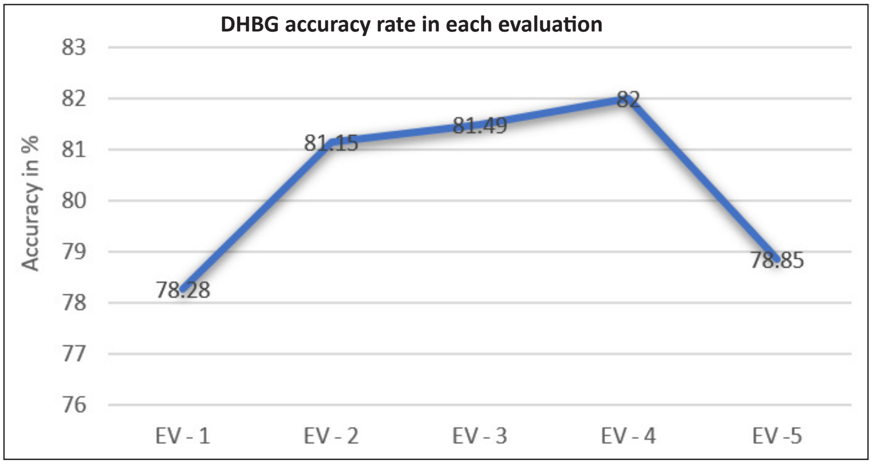
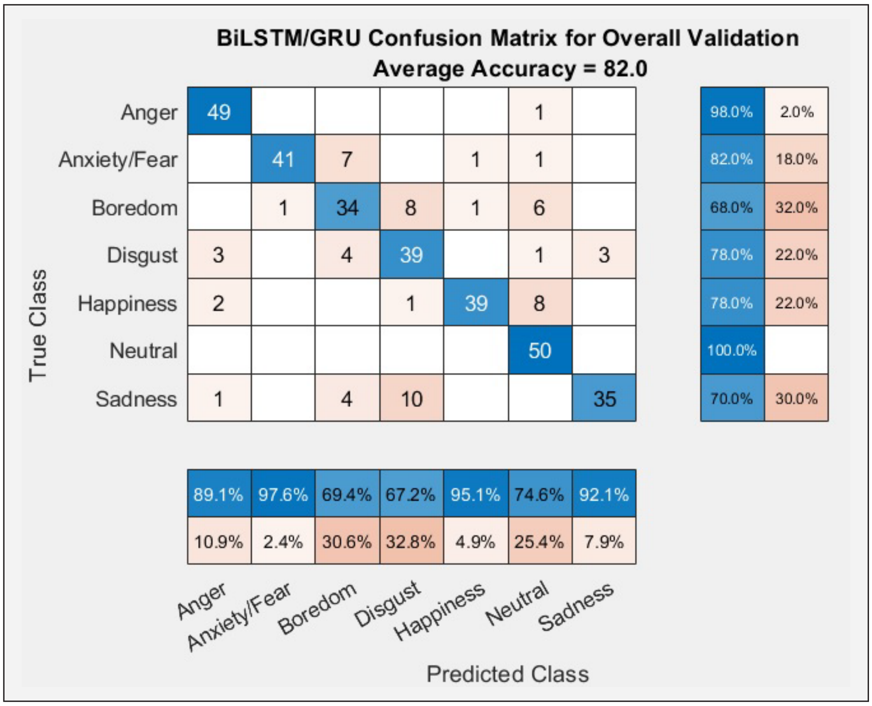


Figure 11. DHBG accuracy rate for 5 evaluations



	Anger	Anxiety/Fear	Boredom	Disgust	Happiness	Neutral	Sadness		
Anger	49					1		98.0%	2.0%
Anxiety/Fear		41	7		1	1		82.0%	18.0%
Boredom		1	34	8	1	6		68.0%	32.0%
Disgust	3		4	39		1	3	78.0%	22.0%
Happiness	2			1	39	8		78.0%	22.0%
Neutral						50		100.0%	
Sadness	1		4	10			35	70.0%	30.0%
	89.1%	97.6%	69.4%	67.2%	95.1%	74.6%	92.1%		
	10.9%	2.4%	30.6%	32.8%	4.9%	25.4%	7.9%		

Figure 12. Cross fold confusion Matrix for DHBG

From Figure 11 the accuracy level in each simulation is established. As the cross-validation folds are random the accuracy level changes randomly. But it lies in the range of 78 to 82. In each simulation the training time and the evaluation time also varies, but only few seconds of variation can be identified. Among the 5 simulation results, in fourth iteration shows higher range of results is identified i.e., 82%.

Finally, by considering the higher accuracy level iteration, Figure 12 clearly shows that for the given input Tamil database DHBG model gives 82% of efficiency. In the confusion matrix, emotions like anger and neutral gives higher rate of 98% and 100%. As like DHLG this model also lags in other emotional states. Boredom and Sadness emotions lags in DHBL model. Only 68% and 70% of accuracy is obtained in both boredom and sadness states and shows lowest of all emotion recognition. Fear, Happiness and Disgust shows 78% and 82% of accuracy.

Deep Hierarchal GRU & LSTM (DHGL) Model

The DHGL models is analyzed to reach a conclusion that technique generates confusion matrix with 10-fold cross validation as final classification output. As like DHLG max 5 times the simulation is evaluated to find the consistent in accuracy level. Among 5 simulation the average 10 folds cross valuation fold 6 shows 89.52% of accuracy and fold 2 and 3 shows 82% of accuracy, where other folds also show better performance of accuracy around 68 -78% (Figure 13).

In the testing phase same 50 samples of data used in DHLG is utilized for analysis. Each iteration there is a small variation in identification of emotional recognition. The average accuracy level for 5 evaluation is around 80.74%.

Now by analyzing the time factor the five evaluation the time taken to training and evaluation of classification timings were considered from Figure 14. While taking the mean value it is clear that for training of DHGL takes around 5.48 mins and to evaluate the classification it takes around 1.122 mins.

From Figure 15 the accuracy level in each simulation is established. As the cross-validation folds are random the accuracy level changes randomly. But it lies in the range of 76 to 82. In each simulation the training time and the evaluation time also varies, but

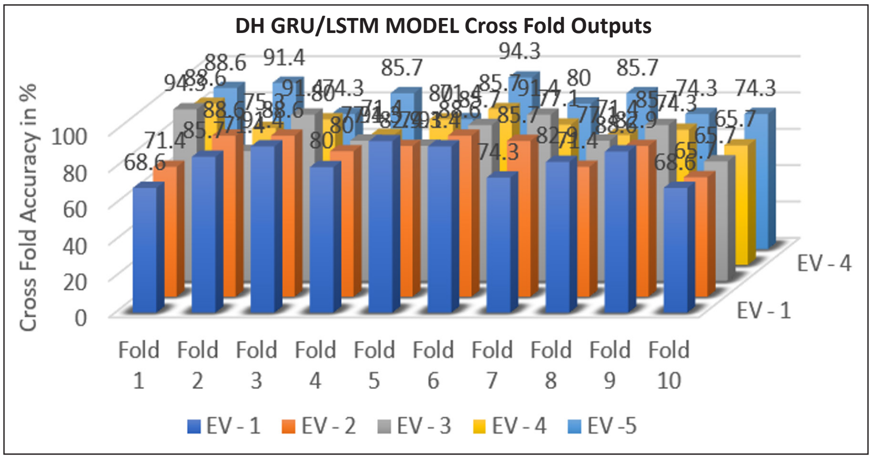


Figure 13. DHGL Cross fold output for multiple evaluation

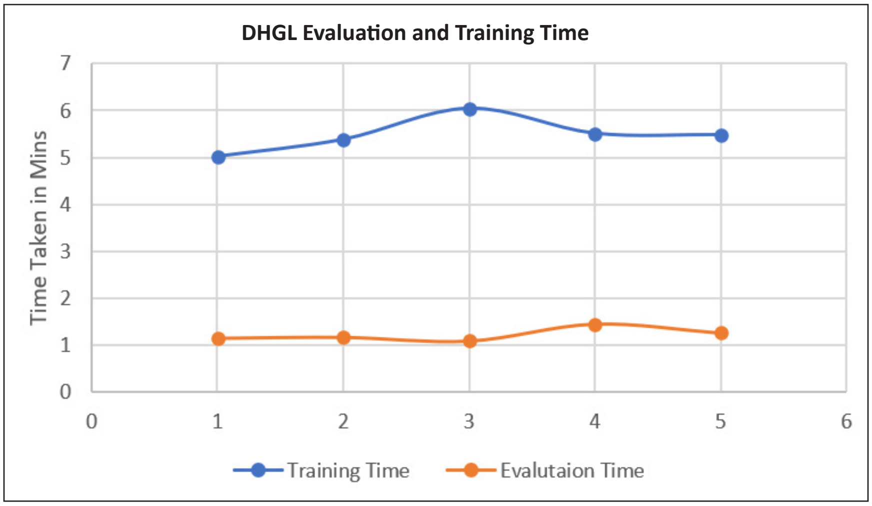


Figure 14. DHGL performance of evaluation time and training time

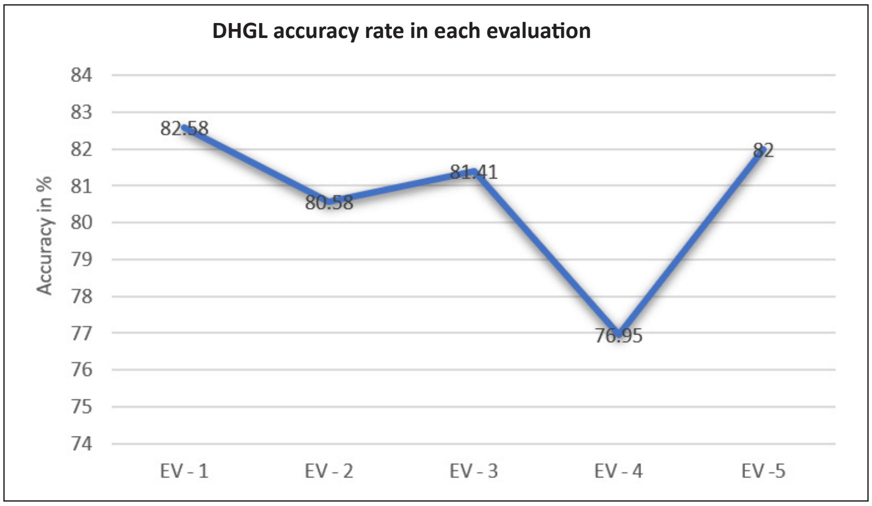
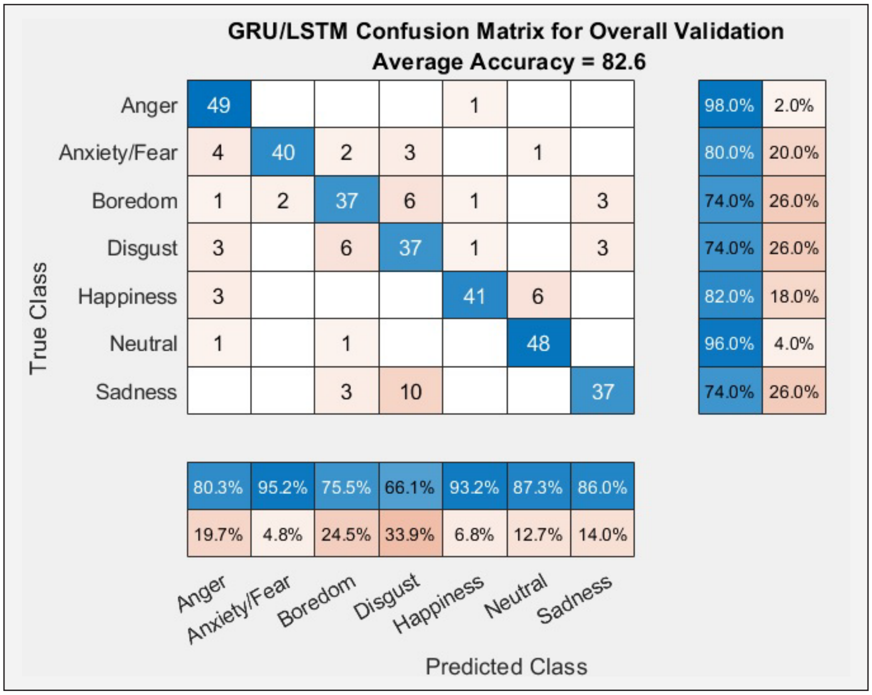


Figure 15. DHGL accuracy rate for 5 evaluations

only few seconds of variation can be identified. Among the 5 simulation results, in first iteration shows higher range of results is identified i.e., 82.58%.

Finally, by considering the higher accuracy level iteration, Figure 16 clearly shows that for the given input Tamil database DHGL model gives 82.58% of efficiency. In the confusion matrix, emotions like anger and neutral gives higher rate of 98% and 96%. As like DHBG this model also lags in other emotional states. Boredom, Didgust and Sadness emotions lags in DHGL model. Only 74% of accuracy is obtained in all boredom. disgust and sadness state and shows lowest of all emotion recognition. Fear and Happiness shows 82% and 74% of accuracy.



True Class \ Predicted Class	Anger	Anxiety/Fear	Boredom	Disgust	Happiness	Neutral	Sadness		
Anger	49				1			98.0%	2.0%
Anxiety/Fear	4	40	2	3		1		80.0%	20.0%
Boredom	1	2	37	6	1		3	74.0%	26.0%
Disgust	3		6	37	1		3	74.0%	26.0%
Happiness	3				41	6		82.0%	18.0%
Neutral	1		1			48		96.0%	4.0%
Sadness			3	10			37	74.0%	26.0%
	80.3%	95.2%	75.5%	66.1%	93.2%	87.3%	86.0%		
	19.7%	4.8%	24.5%	33.9%	6.8%	12.7%	14.0%		

Figure 16. Cross fold confusion Matrix for DHGL

Deep Hierarchal GRU & BiLSTM (DHGB) Model

The DHGB models is analyzed to reach a conclusion that technique generates confusion matrix with 10-fold cross validation as final classification output. As like DHLG max 5 times the simulation is evaluated to find the consistent in accuracy level. Among 5 simulation the average 10 folds cross valuation fold 7,6 and 4 shows 84.02% of accuracy and fold 1, 5 and 9 shows 81% of accuracy, where other folds also show better performance of accuracy around 74 -80% (Figure 17).

In the testing phase same 50 samples of data used in DHLG is utilized for analysis. Each iteration there is a small variation in identification of emotional recognition. The average accuracy level for 5 evaluation is around 81.03%.

Now by analyzing the time factor the five evaluation the time taken to training and evaluation of classification timings were considered from Figure 18. While taking the mean value it is clear that for training of DHBG takes around 6.23 mins and to evaluate the classification it takes around 1.104 mins.

From Figure 19, the accuracy level in each simulation is established. As the cross-validation folds are random the accuracy level changes randomly. But it lies in the range of 78 to 82. In each simulation the training time and the evaluation time also varies, but only few seconds of variation can be identified. Among the 5 simulation results, in first iteration shows higher range of results is identified i.e., 82.86%.

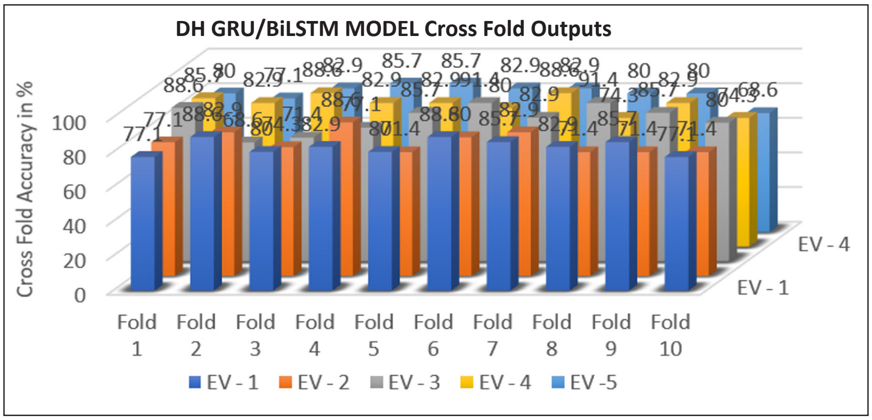


Figure 17. DHGB Cross fold output for multiple evaluation

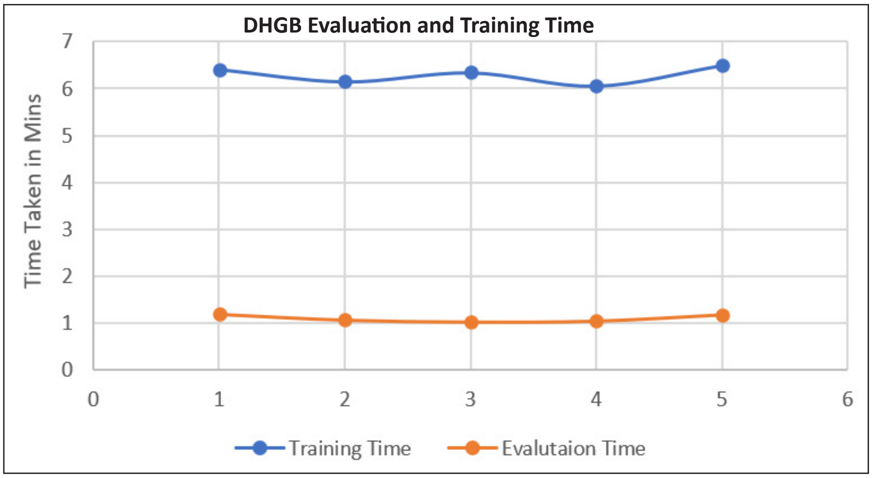


Figure 18. DHGB performance of evaluation time and training time

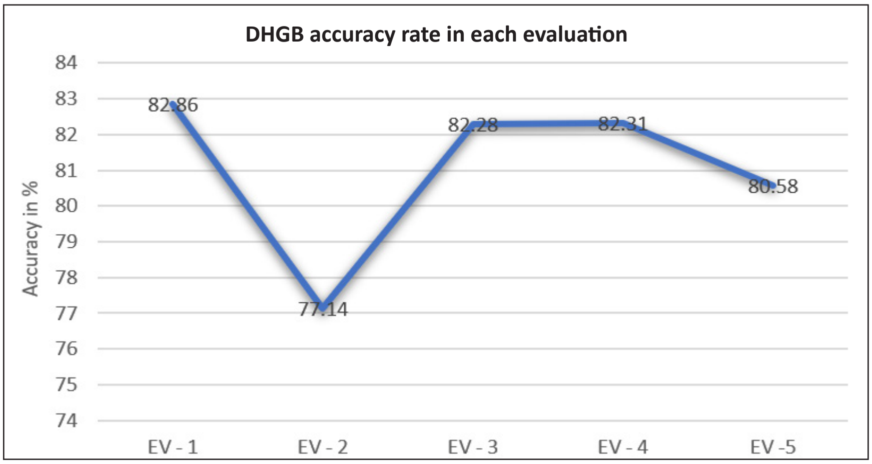
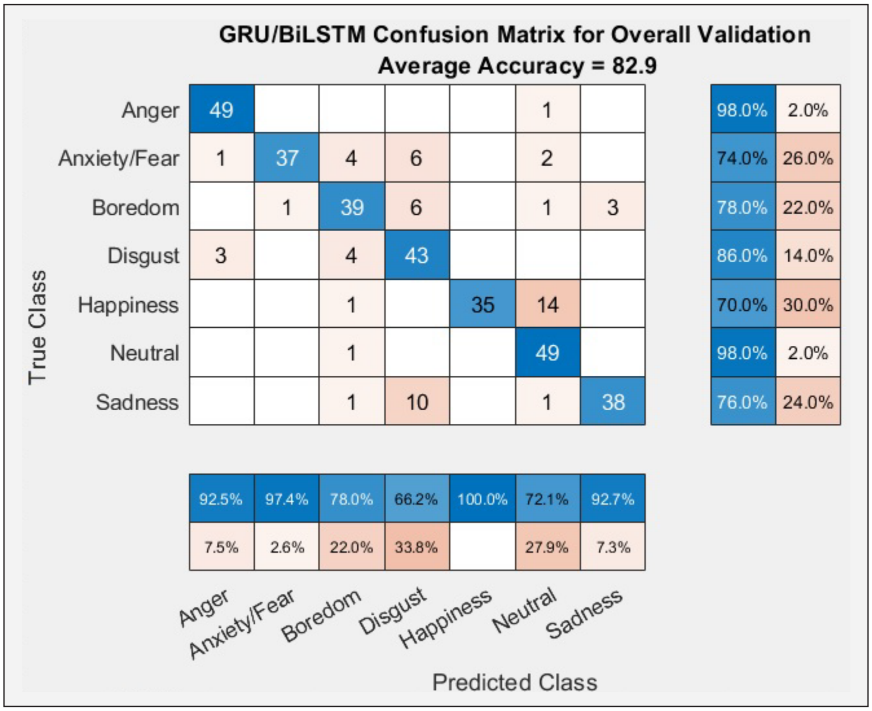


Figure 19. DHGB performance of evaluation time and training time



	Anger	Anxiety/Fear	Boredom	Disgust	Happiness	Neutral	Sadness		
Anger	49					1		98.0%	2.0%
Anxiety/Fear	1	37	4	6		2		74.0%	26.0%
Boredom		1	39	6		1	3	78.0%	22.0%
Disgust	3		4	43				86.0%	14.0%
Happiness			1		35	14		70.0%	30.0%
Neutral			1			49		98.0%	2.0%
Sadness			1	10		1	38	76.0%	24.0%
	92.5%	97.4%	78.0%	66.2%	100.0%	72.1%	92.7%		
	7.5%	2.6%	22.0%	33.8%		27.9%	7.3%		

Figure 20. Cross fold confusion Matrix for DHGB

Finally, by considering the higher accuracy level iteration Figure 20, clearly shows that for the given input Tamil database DHGB model gives 82% of efficiency. In the confusion matrix, emotions like anger and neutral gives higher rate of 98%. As like DHLG this model also lags in other emotional states. Happiness and fear emotions lags in DHGB model. Only 70% and 74% of accuracy is obtained in both boredom and sadness states and shows lowest of all emotion recognition. Boredom, and sadness shows 76% and 78% of accuracy.

Deep Hierarchal GRU & GRU (DHGG) Model:

The DHBG models is analyzed to reach a conclusion that DHGG technique generates confusion matrix with 10-fold cross validation as final classification output. As like DHBG max 5 times the simulation is evaluated to find the consistent in accuracy level. Among 5 simulation the average 10 folds cross valuation fold 3 & 9 shows 83.98% of accuracy and fold 4, 5 and 7 shows 80% and 83% of accuracy, where other folds also show better performance of accuracy around 73 -78% (Figure 21). In the testing phase same 50 samples of data used in DHLG is utilized for analysis.

Each iteration there is a small variation in identification of emotional recognition. The average accuracy level for 5 evaluation is around 80%. Now by analyzing the time factor the five evaluation the time taken to training and evaluation of classification timings were

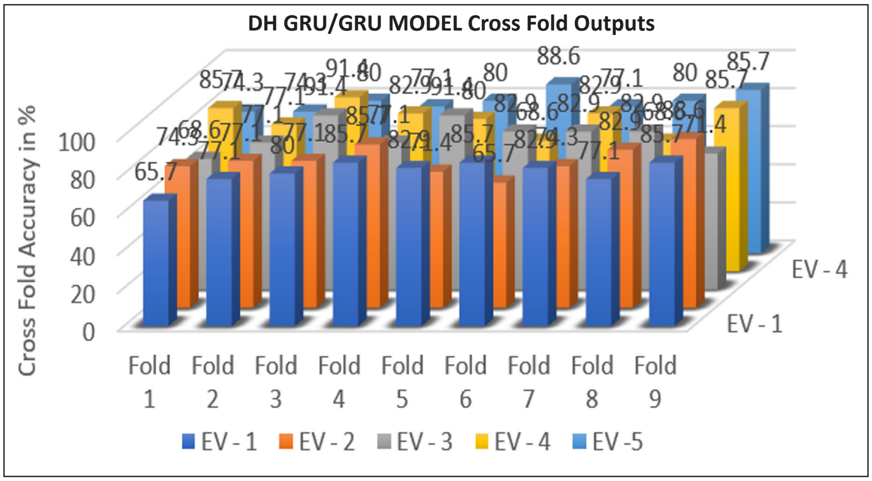


Figure 21. DHGG Cross fold output for multiple evaluation

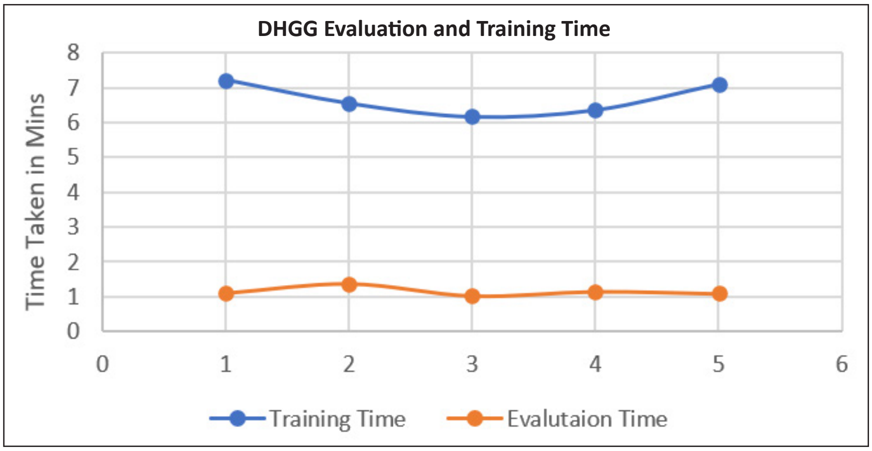


Figure 22. DHGG Performance of Evaluation time and Training Time

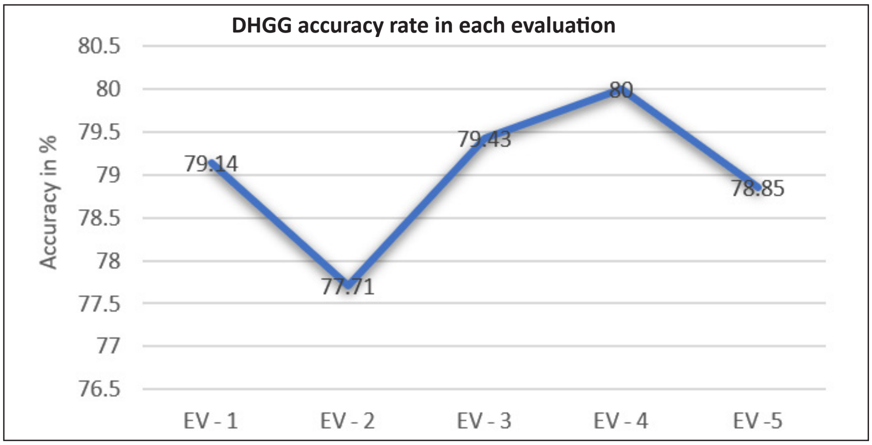
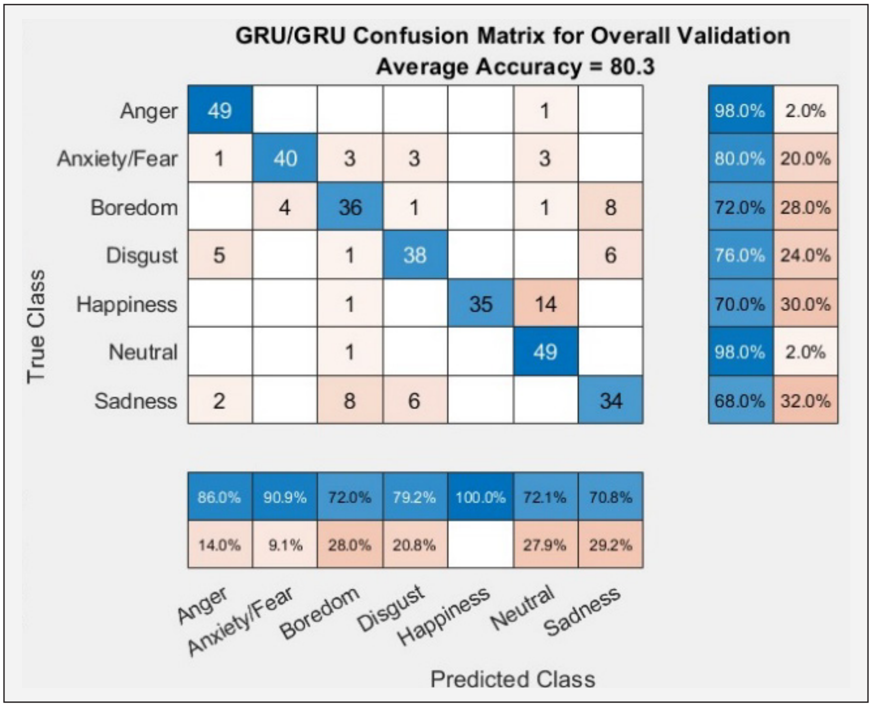


Figure 23. DHGG accuracy rate for 5 evaluations



	Anger	Anxiety/Fear	Boredom	Disgust	Happiness	Neutral	Sadness		
Anger	49					1		98.0%	2.0%
Anxiety/Fear	1	40	3	3		3		80.0%	20.0%
Boredom		4	36	1		1	8	72.0%	28.0%
Disgust	5		1	38			6	76.0%	24.0%
Happiness			1		35	14		70.0%	30.0%
Neutral			1			49		98.0%	2.0%
Sadness	2		8	6			34	68.0%	32.0%
	86.0%	90.9%	72.0%	79.2%	100.0%	72.1%	70.8%		
	14.0%	9.1%	28.0%	20.8%		27.9%	29.2%		

Figure 24. Cross fold confusion Matrix for DHGG

considered from Figure 22. While taking the mean value it is clear that for training of DHBG takes around 6.62 mins and to evaluate the classification it takes around 1.14 mins.

From Figure 23, the accuracy level in each simulation is established. As the cross-validation folds are random the accuracy level changes randomly. But it lies in the range of 77 to 80. In each simulation the training time and the evaluation time also varies, but only few seconds of variation can be identified. Among the 5 simulation results, in fourth iteration shows higher range of results is identified i.e., 80.03%.

Finally, by considering the higher accuracy level iteration Figure 24, clearly shows that for the given input Tamil database DHGG model gives 80.03% of efficiency. In the confusion matrix, emotions like anger and neutral gives higher rate of 98%. As like DHLG this model also lags in other emotional states. Happiness and Sadness emotions lags in DHGG model. Only 70% and 68% of accuracy is obtained in both happiness and sadness states and shows lowest of all emotion recognition. Fear, Boredom and Disgust shows 80% and 72% of accuracy.

Tables 1 and 2 shows the overall performance of the entire designs. Comparing with all the models DHBG shows better performance than the other models. Also, DHGG also achieves equal performance to DHLG. Both the models give average accuracy of 82% for the collected Tamil emotional database shown in Figure 25. Now comparing the training time for all models DHBG lags when compared with other models.

Table 1
Cross fold accuracy of DH LG/BG/GL/GB/GG layers

Fold Accuracy/ Methodology	DHLG	DHBG	DHGL	DHGB	DHGG
Fold 1	77.1	85.7	68.6	77.1	**85.7**
Fold 2	88.6	74.3	85.7	88.6	**77.1**
Fold 3	82.9	90.4	91.4	80	**91.4**
Fold 4	88.6	77.1	80	82.9	**82.9**
Fold 5	71.4	85.7	92.3	94.4	**80**
Fold 6	77.1	88.6	91.4	88.6	**68.6**
Fold 7	85.7	77.1	74.3	85.7	**82.9**
Fold 8	77.1	77.1	82.9	82.9	**68.6**
Fold 9	82.9	82.9	88.6	85.7	**85.7**
Fold 10	**80**	**80**	**68.6**	**77.1**	**77.1**

Table 2
Overall performance of DH LG/BG/GL/GB/GG models

Overall Performance (5 Iteration)	DHLG	DHBG	DHGL	DHGB	DHGG
Best Accuracy	81.14	82	82.58	82.86	**80**
Average accuracy	78.968	80.452	80.704	81.034	**79.026**
Best Training Time	7.19	16.15	5.02	6.06	**6.17**
Average Training Time	7.114	16.334	5.486	6.29	**6.676**
Best Evaluation Time	1.03	1.05	1.09	1.03	**1.03**
Average Evaluation Time	**1.068**	**1.184**	**1.224**	**1.104**	**1.15**

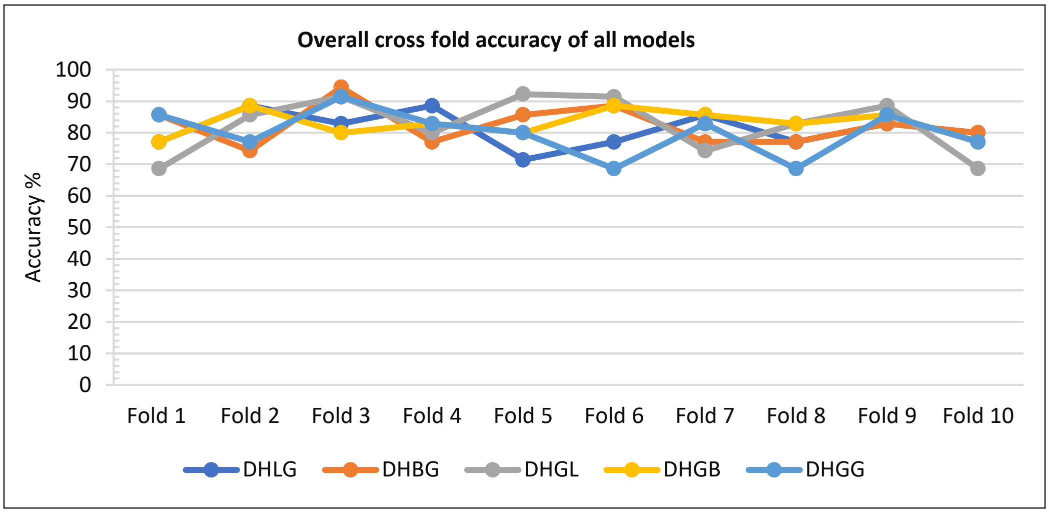


Figure 25. Overall graphical representation of all models cross fold accuracy

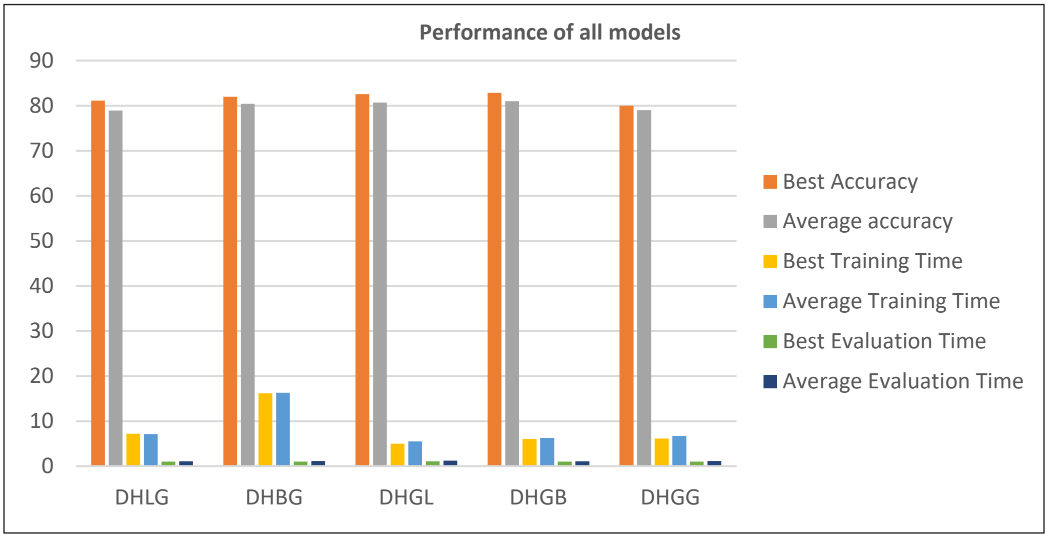


Figure 26. Overall performance of all models

Even though DHBG shows equal performance towards DHLG and DHGG, it takes more time for training the database. More than half of the time is reduced in DHLG. Around 17 mins were taken to train the database in DHBG model, whereas DHLG takes around 7.24 mins to complete the training and DHGG takes 6.36 for training the dataset.

After training the testing is done to identify the different emotional classification for the input 50 samples. In testing also DHBG shows lower evaluation time than other models, it takes only 1.36 mins to complete the evaluation. But other models have better evaluation and training time and its lagging in accuracy level shown in Figure 26.

In most efficient way GRU and followed by BiLSTM gives better performance in RNN followed by GRU and LSTM is slightly less. Comparing the cross fold from above table in DHGB fold 5 gives more accuracy rate of 94.4% and in DHGL also fold 5 yields more accuracy of 92.3%. The DHBG model takes more time for training and evaluation, where other techniques take slightly less time of 5.02 mins and 1.03 mins. The results obtained from different models are generated and presented effectively in this paper. Thus, the further research in design layers can enhance this model and optimize it with lots of computation and data.

CONCLUSION

The purpose of this design work is to introduce a deep learning technique and to design the concepts in different emotional datasets among speech stimuli. Very precisely, the objective was to improve a system (classifier) which might decide whether a particular system recognize the emotions for the collected Tamil language database. Towards this end, the design implemented in this architecture focused on Deep Hierarchal LSTM / BiLSTM and GRU that to a large extent recognizes various emotional speech envelopes. The purpose of using the RNN in the voice envelope direction is to bring the network to attain more ability, through its precision, error, training, and evaluation time. Thus, it analyzes the emotional speech signal and its efficiency. Thus, comparing the classification analysis of proposed system based on LSTM / BiLSTM / GRU and the findings proved that the DHGB model performs better than the other four models for the given dataset. In the seven basic emotions anger and neutral has higher rate of identification level about 98% and emotions like fear and happiness has average accuracy rate of about 86% and only 74% of accuracy is achieved for other disgust, sadness, and boredom emotions. The classification accuracy of 82.86% is achieved with minimal losses in DHGB and the time required for preparation and assessment is also lesser than other models and further the models can be improved by different optimization methods for training the dataset, hence accuracy level can be increased with higher rate.

ACKNOWLEDGEMENT

Authors would like to acknowledge the KL University, AP, India for supporting this work and the faculty members who supported them for making the data available for this research.

REFERENCES

Abdel-Hamid, O., Mohamed, A. R., Jiang, H., Deng, L., Penn, G., & Yu, D. (2014). Convolutional neural networks for speech recognition. *IEEE ACM Transactions on Audio, Speech, and Language Processing*, 22(10), 1533-1545. https://doi.org/10.1109/TASLP.2014.2339736.

Chen, Z., Watanabe, S., Erdogan, H., & Hershey, J. R. (2015, September 6-10). Speech enhancement and recognition using multi-task learning of long short-term memory recurrent neural networks. In *Sixteenth Annual Conference of the International Speech Communication Association* (pp. 3274-3278). Dresden, Germany. https://doi.org/10.1109/SLT.2016.7846281

Chung, J., Cho, K., & Bengio, Y. (2014, December 8-13). Empirical evaluation of gated recurrent neural networks on sequence modeling. In *NIPS 2014 Deep Learning and Representation Learning Workshop* (pp. 2342-2350). Montreal, Canada. https://doi.org/10.5555/3045118.3045367.

Erdogan, H., Hershey, J. R., Watanabe, S., & Le Roux, J. (2015). Phase-sensitive and recognition-boosted speech separation using deep recurrent neural networks. In *2015 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)* (pp. 708-712). IEEE Conference Publication. https://doi.org/10.1109/ICASSP.2015.7178061.

Eyben, F., Weninger, F., Squartini, S., & Schuller, B. (2013). Real-life voice activity detection with lstm recurrent neural networks and an application to hollywood movies. In *2013 IEEE International Conference on Acoustics, Speech and Signal Processing* (pp. 483-487). IEEE Conference Publication. https://doi.org/10.1109/ICASSP.2015.7178061.

Graves, A., Jaitly, N., & Mohamed, A. R. (2013). Hybrid speech recognition with deep bidirectional LSTM. In *2013 IEEE workshop on automatic speech recognition and understanding* (pp. 273-278). IEEE Conference Publication. https://doi.org/10.1109/ASRU.2013.6707742.

Hochreiter, S., & Schmidhuber, J. (1997). Long short-term memory. *Neural Computation, 9*(8), 1735-1780. https://doi.org/10.1162/neco.1997.9.8.1735.

Ioffe, S., & Szegedy, C. (2015, July 7-9). Batch normalization: Accelerating deep network training by reducing internal covariate shift. In *International conference on machine learning* (pp. 448-456). Lille, France. https://doi.org/10.5555/3045118.3045167.

Jozefowicz, R., Zaremba, W., & Sutskever, I. (2015, July 7-9). An empirical exploration of recurrent network architectures. In *International conference on machine learning* (pp. 2342-2350). Lille, France. https://doi.org/10.5555/3045118.3045367.

Kishore, P. V. V., & Prasad, M. V. D. (2016). Optical flow hand tracking and active contour hand shape features for continuous sign language recognition with artificial neural networ. *International Journal of Software Engineering and its Applications, 10*(2), 149-170. https://doi.org/10.1109/IACC.2016.71

Krizhevsky, A., Sutskever, I., & Hinton, G. E. (2012). Imagenet classification with deep convolutional neural networks. *Advances in Neural Information Processing Systems, 25*, 1097-1105. https://doi.org/10.1145/3065386.

Kumar, K. V. V., Kishore, P. V. V., & Kumar, D. A. (2017). Indian classical dance classification with adaboost multiclass classifier on multi feature fusion. *Mathematical Problems in Engineering, 20*(5), 126-139. https://doi.org/10.1155/2017/6204742.

Laurent, C., Pereyra, G., Brakel, P., Zhang, Y., & Bengio, Y. (2016). Batch normalized recurrent neural networks. In *2016 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)* (pp. 2657-2661). IEEE Conference Publication. https://doi.org/10.1109/ICASSP.2016.7472159.

Li, J., Deng, L., Gong, Y., & Haeb-Umbach, R. (2014). An overview of noise-robust automatic speech recognition. *IEEE/ACM Transactions on Audio, Speech and Language Processing, 22*(4), 745-777. https://doi.org/10.1109/TASLP.2014.2304637

Liu, Y., Zhang, P., & Hain, T. (2014). Using neural network front-ends on far field multiple microphones based speech recognition. In *2014 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)* (pp. 5542-5546). IEEE Conference Publication. https://doi.org/10.1109/ICASSP.2014.6854663.

Mannepalli, K., Sastry, P. N., & Suman, M. (2016a). FDBN: Design and development of fractional deep belief networks for speaker emotion recognition. *International Journal of Speech Technology, 19*(4), 779-790. https://doi.org/10.1007/s10772-016-9368-y

Mannepalli, K., Sastry, P. N., & Suman, M. (2016b). MFCC-GMM based accent recognition system for Telugu speech signals. *International Journal of Speech Technology, 19*(1), 87-93. https://doi.org/10.1007/s10772-015-9328-y

Pascanu, R., Mikolov, T., & Bengio, Y. (2013). On the difficulty of training recurrent neural networks. *Proceedings of Machine Learning Research, 28*(3), 1310-1318. https://doi.org/10.5555/3042817.3043083.

Rao, G. A., & Kishore, P. V. V. (2016). Sign language recognition system simulated for video captured with smart phone front camera. *International Journal of Electrical and Computer Engineering, 6*(5), 2176-2187. https://doi.org/10.11591/ijece.v6i5.11384

Rao, G. A., Syamala, K., Kishore, P. V. V., & Sastry, A. S. C. S. (2018). Deep convolutional neural networks for sign language recognition. *International Journal of Engineering and Technology (UAE), 7*(Special Issue 5), 62-70. https://doi.org/10.1109/SPACES.2018.8316344

Ravanelli, M., Brakel, P., Omologo, M., & Bengio, Y. (2016). Batch-normalized joint training for DNN-based distant speech recognition. In *2016 IEEE Spoken Language Technology Workshop (SLT)* (pp. 28-34). IEEE Conference Publication. https://doi.org/10.1109/SLT.2016.7846241.

Ravanelli, M., Brakel, P., Omologo, M., & Bengio, Y. (2017). A network of deep neural networks for distant speech recognition. In *2017 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)* (pp. 4880-4884). IEEE Conference Publication. https://doi.org/10.1109/ICASSP.2017.7953084.

Sak, H., Senior, A. W., & Beaufays, F. (2014). Long short-term memory recurrent neural network architectures for large scale acoustic modeling. *Proceeding of Interspeech, 22*(1), 338-342. https://doi.org/10.1007/s10772-018-09573-7

Sastry, A. S. C. S., Kishore, P. V. V., Prasad, C. R., & Prasad, M. V. D. (2016). Denoising ultrasound medical images: A block based hard and soft thresholding in wavelet domain. *International Journal of Measurement Technologies and Instrumentation Engineering (IJMTIE), 5*(1), 1-14. https://doi.org/10.4018/IJMTIE.2015010101

Schwarz, A., Huemmer, C., Maas, R., & Kellermann, W. (2015). Spatial diffuseness features for DNN-based speech recognition in noisy and reverberant environments. In *2015 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)* (pp. 4380-4384). IEEE Conference Publication. https://doi.org/10.1109/ICASSP.2015.7178798.

Srivastava, N., Hinton, G., Krizhevsky, A., Sutskever, I., & Salakhutdinov, R. (2014). Dropout: A simple way to prevent neural networks from overfitting. *Journal of Machine Learning Research, 15*, 1929-1958. https://doi.org/10.5555/2627435.2670313.

Weninger, F., Erdogan, H., Watanabe, S., Vincent, E., Le Roux, J., Hershey, J. R., & Schuller, B. (2015). Speech enhancement with LSTM recurrent neural networks and its application to noise-robust ASR. In *International conference on latent variable analysis and signal separation* (pp. 91-99). Springer. https://doi.org/10.1007/978-3-319-22482-4_11

Zhang, Y., Chen, G., Yu, D., Yaco, K., Khudanpur, S., & Glass, J. (2016). Highway long short-term memory rnns for distant speech recognition. In *2016 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)* (pp. 5755-5759). IEEE Conference Publication. https://doi.org/10.1109/ICASSP.2016.7472780

Zhou, G. B., Wu, J., Zhang, C. L., & Zhou, Z. H. (2016). Minimal gated unit for recurrent neural networks. *International Journal of Automation and Computing, 13*(3), 226-234. https://doi.org/10.1007/s11633-016-1006-2.