2022 International Conference on Futuristic Technologies (INCOFT)
Karnataka, India. Nov 25-27, 2022

Analysis of Emotions from Speech using Hybrid Deep Learning Network Models

Chevella Anil Kumara
Assistant Professor
Electronics and Communication Engineering
VNR Vignana Jyothi Institute of Engineering and Technology,
Hyderabad-500090, India
anilkumar_chevella@vnrvjiet.in

Kancharla Anitha Sheelab
Professor, SMIEEE
Electronics and Communication Engineering
Jawaharlal Nehru Technological University
Hyderabad-85, India
kanithasheela@jntuh.ac.in

Naveen Kumar Vodnalac
Assistant Professor, SMIEEE
Electronics and Communication Engineering
VNR Vignana Jyothi Institute of Engineering and Technology,
Hyderabad-500090, India
naveenkumar_v@vnrvjiet.in

2022 International Conference on Futuristic Technologies (INCOFT) | 978-1-6654-5046-1/22/$31.00 ©2022 IEEE | DOI: 10.1109/INCOFT55651.2022.10094442

***Abstract*- In today's advanced digital world, Emotion plays a important role for communication to understand better each other in virtual environment and for Human to Machine Communication. Research has revealed the powerful role that emotion plays in shaping human social interaction. This has opened up a new research field called Automatic Emotion Recognition, having basic goal to understand and retrieve desired emotions. In the past several biometrics have been explored to recognize the emotional states such as facial expressions, speech, and physiological signals, etc. However, several inherent advantages make speech signals are good source for affective computing. The primary objective of this study is to assess various emotions from speech using Deep learning algorithms (CNN and LSTM). Traditional/Machine Learning approaches rely on manual feature extraction before classifying the emotional state, whereas Deep Learning networks widely used for emotional classification due to their advanced feature extraction mechanisms. In this implementation, we will use the most advanced hybrid deep learning models 1-D CNN+LSTM and 2-D CNN+LSTM, to extract features and classify seven distinct emotions based on their input data. Using the Emo-DB standard dataset, we trained and test the proposed network model and achieved a maximum accuracy of 91.51% and 95.52% for I-C CNN+LSTM and 2-D CNN+LSTM respectively.**

Key words: Deep learning; Speech Emotion Recognition; HCI; MTCNN; 1-D CNN+LSTM; 2-D CNN+LSTM.

I. INTRODUCTION

Emotion plays a significant role in daily interpersonal human interactions. This is essential for Human to Compter Interface (HCI) as well as intelligent decisions[1]. Emotional convey information describes about the mental state of an individual. This has opened up a new research field called Automatic Emotion Recognition, having basic goal to understand and retrieve desired emotions. In the past several biometrics have been explored to recognize the emotional states such as facial expressions, speech, and physiological signals, etc. However, several inherent advantages make speech signals are good source for affective computing. When working with data that contains spatial information such as images, a common difficulty encountered by Fully Connected Networks (FCN)[2] is the loss of spatial and local information as a result of dimensionality reduction. The Convolutional Neural Network (CNN) provides a solution to this issue. Convolutional layers conduct convolution operations on input information, and an FCN is then applied at the end of the process. There are several real-time application of detecting emotions from speech signal viz in the interface with robots, audio surveillance, web-based e-learning and commercial applications, etc.

As depicted in the figure 1, deep learning techniques are used to identify the emotional state of a speaker based on their speech. There are many possible network models/designs for feature extraction that can be modeled here.

- One Dimensional(1D) CNN
- Two Dimensional(2D) CNN
- Hybrid 1-D and 2-D CNN+LSTM

Fig. 1. Basic block diagram of Speech emotion Recognition

The convolution operation can only be carried out in a single direction when using 1D-CNN; in contrast, the convolution kernel used by 2D-CNN operates in two dimensions. The analysis of time series data utilising hybrid network models such as CNN+LSTM is another possible application of this. The Log Mel-Spectrogram is used as an input by the 2D-CNN model; this input is expressed as a 2D matrix.

To develop a system, ability to express one's deepest feelings, it is necessary to learn/extract the features. In order to acquire local and global emotion-related features from speech, we suggested hybrid deep learning network models such 1-Dimensional and 2-Dimesional CNN plus LSTM[3]–[5] and compared the aforesaid networks with 1-Dimensional CNN and 2-Dimesional CNN Networks without a LSTM network in terms of an accuracy of emotion recognition. Four local feature learning blocks (LFLB) and one LSTM layer are used to learn the local and global features of 1-dimensional and 2-dimensional speech signals in the hybrid Networks. LFLB, which is primarily composed of one convolutional layer and one max-pooling layer, was developed for the purpose of learning local correlations in addition to extracting hierarchical correlations.

The LSTM layer is used to learn long-term dependencies from the local characteristics that have been learned. Feature extraction in these proposed networks can take use of the network's strengths and overcome its weaknesses by combining LSTM with a convolutional neural network. As a time-varying signal, the speech signal requires particular processing in order to capture its temporal variations. As a result, the LSTM layer has been added to help identify long-term correlations between variables. There are two types of CNN+LSTM networks[6]: one for 1-dimensional raw audio clips and another for 2-dimensional Spectrogram of Speech signal.

II. METHODOLOGY

Extracting more distinguishing emotion features is one of the main tasks for researchers to recognize speech emotion. According to the difference of feature extraction methods, speech features can be classified as handcrafted features and learned features. Most of the extraction of handcrafted features are carefully designed using ingenious strategies and can be explained in more detail how it works and what it does. And the learnt features that are extracted by the various deep neural networks perform exceptionally well When it comes to prediction, As a result, deep feature learning is becoming increasingly popular for making predictions.

The following are our original contributions to the research work: 1) a local feature learning block (LFLB), which consists of one convolutional layer, one batch normalization (BN) layer, one exponential linear unit layer, and one max-pooling layer, is designed to extract local features; 2) to learn long-term dependencies from a sequence of local features, LSTM layer is introduced to build CNN+LSTM networks following the LFLB; 3) it is proved experimentally that 1D-CNN+LSTM and 2D-CNN+LSTM networks can learn lots of emotional features from raw audio utterances and spectrograms respectively. The entire structure of the proposed hybrid network is depicted in Figure 2.

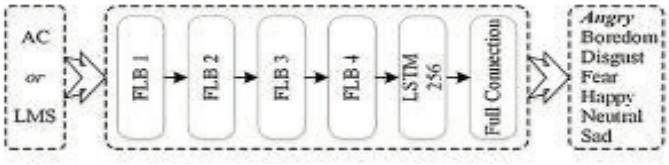

Fig. 2. Architecture of Proposed Hybrid network.

AC=Audio Clips

LMS= Log Mel-Spectrogram

A. Deep feature learning

Local and global features can be learned from raw audio clips by combining LFLB and LSTM. The convolution layer, Core layer of LFLB, is designed to processing a grid of data. It can learn a sequence feature where each member of the feature is a function of a small number of neighboring members of the input. Whereas LSTM is specialized for processing a sequence of values, each member of the learned feature is a function of the previous members of the output. Hence, We can learn high-level features containing both local information and long term contextual dependencies by combining the CNN with LSTM.

B. Local feature learning

A alternative for CNN that is designed to extract emotional traits is called a local feature learning block, or LFLB for short. As shown in Figure 3, each LFLB has the following layers: one convolutional layer, one batch normalisation (BN) layer[7], one ELU (exponential linear unit) layer, and one max-pooling layer.

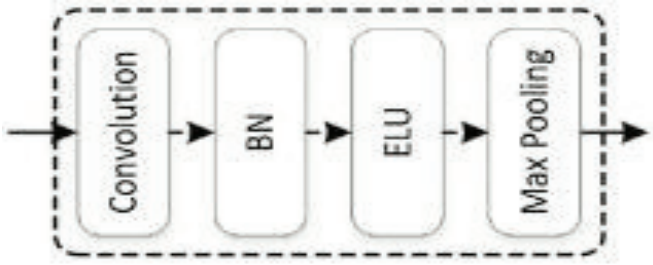

Fig. 3. LFLB (Local feature learning block)

The LFLB is built on top of several layers, the most important of which are the convolution and pooling layers. Connectivity and weight sharing are two of the most notable features of convolution. The following figure 4 represents the convolution and pooling layers of 1D and 2D Networks[2].

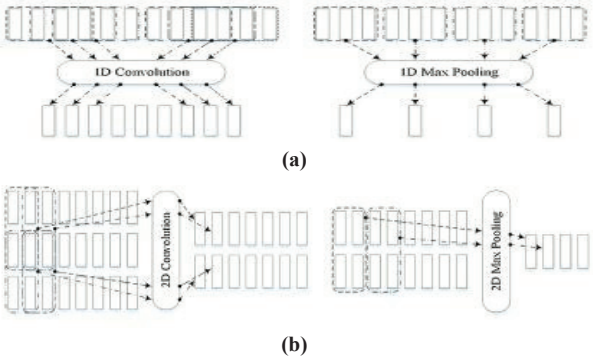

Fig. 4. (a) 1-D Covolution and Pooling (b) 2-D Covolution and Pooling

The BN layer helps to increase the performance of deep neural networks as well as their stability. It does this by normalising the activations of the convolutional layer at each batch. Because of the batch normalisation transformation, the mean activation stays relatively close to 0, while the activation standard deviation stays relatively close to 1. The output of the BN layer is specified by the ELU layer.

ELU is a special kind of activation function because, unlike most others, it can have negative values. Because of this, it can bring the mean of the activations closer to zero, which speeds up the learning process in constructed networks and leads to improved recognition accuracy. The pooling layer can strengthen the features so that they are less susceptible to noise and distortion. The max-pooling function is the non-linear function that is utilised the most frequently. It then outputs the greatest value of each of these sub-regions and separates the input into a collection of regions that do not overlap with one another.

It is possible to acquire the result h(n) of interest by applying a 1-Dimensional convolution layer to an input signal

of x(n), and then convolving the result of interest with the convolution kernel w(n) of size (n). Experiments in our lab use a one-dimensional convolution kernel called w(n).

$$h(n) = x(n) * w(n) = \sum_{m=-l}^{l} x(m).w(n-m) \quad (1)$$

However, in the case where x(i,j) is used as the input to the 2D convolution layer, the result z(i,j) can be obtained by convolving the signal x(i,j) with the convolution kernel w(i,j) that has a size range of axb. Within the context of our research, the 2D convolution kernel w(i,j) is also initialised in a random fashion.

$$z(i,j) = x(i,j) * w(i,j) = \sum_{s=-a}^{a} \sum_{t=-b}^{b} x(s,t).w(i-s,j-t) \quad (2)$$

It then normalises the previous layer's activations at each batch by inputting the convolved features into the BN layer. Mean and variance of the convolved features are maintained close to zero and one, respectively, by the BN layer. As a result of applying the ELU layer to the normalised features, the final features may be expressed as

$$h_i^l = \sigma(BN(b_i^l + \sum_j h_j^{l-1} * w_{ij}^l \quad (3)$$

Where h_i^l and h_j^{l-1} represent the ith output feature at the lth layer and the jth input feature at the (l-1)th layer; w_{ij}^l denotes convolution kernel between the ith and jth feature. Normalization of the features learned by the convolution layer is accomplished using the function BN(.). It is possible to express the ELU activation function of the network as

$$\sigma(x) = \begin{cases} x & x \geq 0 \\ \alpha(e^x - 1) & x < 0 \end{cases} \quad (4)$$

Since e is Euler's number, the additional alpha constant should be positive (> 0). It is at this point that the features are sent to the max-pooling layer. The non-linear down-sampling function is performed by the pooling layer, which diminishes the feature's resolution. The max-pooling layer's features can be described as

$$h_k^l = \max_{\forall_P \in \Omega_k} h_p^l \quad (5)$$

Where k=pooling region with index k,

h_p^l =Input feature of the lth max-pooling layer

h_k^l =output feature of lth pooling layer

C. Global feature learning

Long-term relationships between sequences are learned from the LSTM architecture. So it is stacked upon the LFLB to learn contextual dependencies from the learned local feature sequences[9]. Input, output, forget, and a cell state are all used by the LSTM to add or delete information from the block state. Following are some equations that describe the process of updating an LSTM unit at each and every time step[6]. It is possible to express the relationship between an LSTM unit's inputs and outputs as follows:

$$i_t = \sigma(w_i[h_{t-1}, x_t] + b_i) \quad (6)$$

$$\hat{c}_t = \tanh(w_c[h_{t-1}, x_t] + b_c) \quad (7)$$

$$c_t = f_t * c_{t-1} + i_t * \hat{c} \quad (8)$$

$$f_t = \sigma(w_f.[h_{t-1}, x_t] + b_f) \quad (9)$$

$$o_t = \sigma(w_o.[h_{t-1}, x_t] + b_o) \quad (10)$$

$$h_t = o_t * \tanh(c_t) \quad (11)$$

Where ct represents the LSTM unit state; w, b are parameter matrices and vector; and

ft, it and ot=gate vectors;

σ =sigmoid function,

C, h=hyperbolic tangents;

* =Hadamard product.

III. EXPERIMENTAL STUDIES

Before, going to develop the proposed hybrid network models we trained and tested the emotional speech database on 1D-CNN[10], 2D-CNN[2] and compared with the recognition accuracy of proposed Hybrid(one and two dimensional CNN plus LSTM) Network models. When an audio clip, represented by a one-dimensional vector, is fed into one dimensional proposed hybrid network, and a spectrogram, represented by a two-dimensional vector, is fed into the two-dimensional network, the LFLBs learn local features for all four network models.

Following the reshaping of Hybrid network models, the LFLB features are fed into the LSTM layer for further processing. After that, the contextual dependencies are learned from the local hierarchical features that have been inputted. Figure 5 depicts the 1D-CNN+LSTM learning of local features and contextual dependencies, while Figure 6 depicts the 2D-CNN+LSTM learning of the same information. This means that the output features from the LSTM layer include both local and long-term context information. A fully connected layer is then created, which is directly linked to an LSTM layer. The layer that is entirely interconnected can be expressed as follows:

$$z^l = b^t + z^{l-1}.w^t \quad (12)$$

Using the entered features, Softmax classifies the data and predicts what will happen. To solve the challenge of multi-class classification, Softmax can be used as a generalisation of logistic regression.

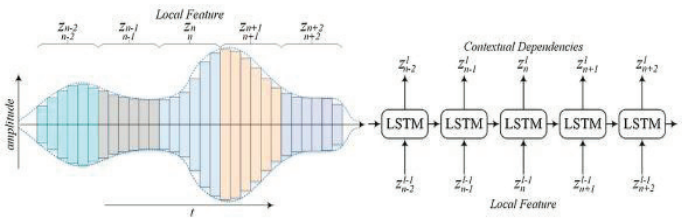

Fig. 5. Architecture of 1-Dimensional CNN+LSTM

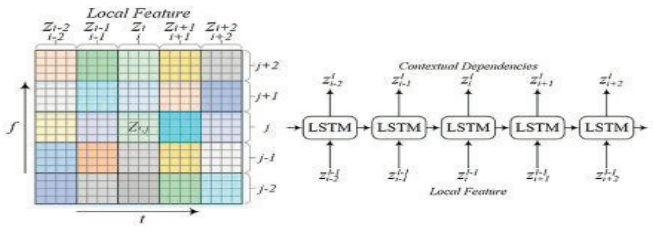

Fig. 6. Architecture of 2-Dimensional CNN+LSTM

There are multiple possible interpretations of the class label y. It is possible to define the Softmax function as

$$z_i = \sum w_{ji} * h_j \tag{14}$$

$$\text{Softmax}(z_i) = p_i = \frac{e^{z_i}}{\sum_{j=0}^{k} e^{z_j}}, i = 0,1 \dots k \tag{15}$$

And last, the predicted class label $\tilde{y}$ was computed by using the formula:

$$\tilde{y} = arg \max_i p_i \tag{16}$$

The experimental data was randomly divided into two sets, with the training set receiving eighty percent of the data and the testing set receiving the remaining twenty percent of the data. Our research aims to recognise speech emotion with excellent generalisation performance and high accuracy. Because of this, only the most accurate and well-fitting models are recorded throughout tests. If a model's validation accuracy stops improving throughout training, it will have a better predictive performance than the other models. As seen in the figure 7, the training accuracy does not reach its maximum at the same time as the validation accuracy does.

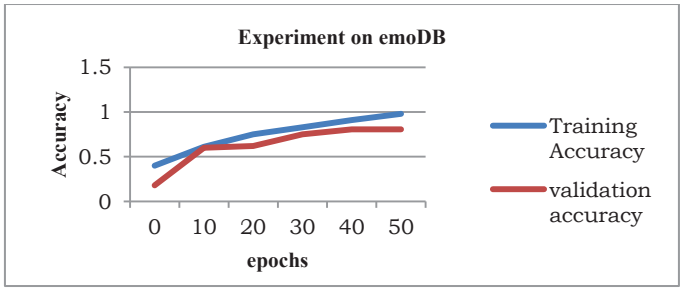

Fig. 7. Predictive Performance of Hybrid 1-D CNN+LSTM

IV. EXPERIMENTAL SETUP AND RESULTS

The following Deep Learning Techniques were modeled, trained and tested for emotion recognition from speech signal.

1D-CNN

Following are the parameters that should be used for the one-dimensional CNN model that we suggested here for recognising emotions from speech.

- Six convolution layers, with a kernel size of 1 x 5, One fully connected layer.
- It utilised zero padding and ReLu activation, both of which served to retain information along the edges.
- The maximum pooling procedure was performed on a size 1 x 8 and includes a dropout of 0.1 as well as batch normalisation.
- With this network model we trained a network with 500 epochs and achieved an accuracy of 85.47%

The Table 1 represents the confusion matrix for better accuracy.

TABLE I. CONFUSION MATRIX FOR 1-D CNN

	Ang	Sur	Dis	Fea	Hap	Neu	Sad
Ang	86.78	0	0.67	6.00	1.99	0	4.56
Sur	0	87.65	0	1.23	1.23	9.88	0
Dis	4.35	4.35	82.61	4.35	2.17	2.17	0
Fea	4.75	0	3.45	88.35	2.9	3.45	0
Hap	6.45	2.23	4.23	0	86.09	1.00	0
Neu	0	0	0	0	0	95.50	4.50
Sad	0	1.61	2.57	2.19	0	3.62	90.01

Where, Su=Surprise, Di=Disgust, An=Angry, Ha=Happy, Ne=Neutral, Sa=Sad, Fe=Fear,

2D-CNN

The convolution operation can only be carried out in one direction when using 1-dimensional CNN; in contrast, the convolution kernel used by 2-dimensional-CNN can carry out operations in two dimensions. The study of time series data is another possible application for this. The factors that made up our experiment were:

- ✓ Five layers of convolution, each with a filter size of 3x3.
- ✓ It utilised zero padding to protect the integrity of the information on the edges and ReLu activation.
- ✓ A dropout of 0.1 and batch normalisation were incorporated in the max-pooling procedure, which was performed on a size 2 x 2 matrix.
- ✓ With this network model we trained a network with 500 epochs and achieved an accuracy of 88.36%

The table 2 represents the confusion matrix for better accuracy of 2D-CNN.

TABLE II. CONFUSION MATRIX FOR 2-D CNN

	Ang	Sur	Dis	Fea	Hap	Neu	Sad
Ang	89.24	0	1.58	4.53	2.05	0	2.60
Sur	0	87.65	0	1.23	1.23	9.88	0
Dis	4.35	4.35	82.61	4.35	2.17	2.17	0
Fea	4.75	0	3.45	88.35	2.9	3.45	0
Hap	2.45	2.63	3.83	0	91.09	0	0
Neu	0	0	0	0	0	98	2.00
Sad	0	1.53	2.12	4.17	0	5.62	86.56

Lastly, the Hybrid Deep Learning network models were made with four local LFLBs and one LSTM layer, as well as six LFLBs and one LSTM layer. Each LFLB had a filter size of 3x3, a stride of 1, and no padding. The number of convolution kernels in the first and second LFLBs was 64, while the number in the third and fourth LFLBs was 128, and the number in the fifth and sixth LFLBs was 256 as shown tables 3 and 4. And the proposed hybrid networks were trained and tested, and their categorical accuracy was assessed using the Berlin Emo-DB [9], [11]database with different number of epochs.

1D-CNN+LSTM

For 1D-CNN+LSTM Network models, we achieved the categorical accuracy is **77.35% and 79.96% for 4LFLB+LSTM and 6LFLB+LSTM** respectively with fully connected neurons is 64, batch size=32 and epochs=50, learning rate 0.0001 and momentum is 0.9 with stochastic gradient Descente optimizer. For 100 epochs, categorical accuracy is **78.65% and 81.53% for 4LFLB+LSTM and 6LFLB+LSTM** respectively with same hyper parameters.

Similarly for 300 epochs **90.94% and 91.51%** and also for 500 epochs, categorical accuracy is **71.80% and 80.25% for 4LFLB+LSTM and 6LFLB+LSTM** respectively with same hyper parameters. The tables 5 and 6 summarize the network accuracies and confusion matrix for better accuracy network model.

2D-CNN+LSTM

TABLE III. STRUCTURE OF 4 LFLB+LSTM

Block Name		Output Dim	Kernal/Filter Size	Stride
LFFB1	1C1	L x 64	3	1
	1P1	L/4 x 64	4	4
LFFB2	1C2	L/4 x 64	3	1
	1P2	L/16 x 64	4	4
LFFB3	1C3	L/16 x 128	3	1
	1P3	L/64 x 128	4	4
LFFB4	1C4	L/64 x 128	3	1
	1P4	L/256 x 128	4	4
L		-	256	-
F		-	k	-

Similarly like 1D-CNN+LSTM Network model, for 2D-CNN+LSTM Network model also, we achieved the categorical accuracy is **79.50% and 80.85% for 4LFLB+LSTM and 6LFLB+LSTM** respectively with fully connected neurons is 64, batch size=32 and epochs=50, learning rate 0.0001 and momentum is 0.9 with stochastic gradient Descente optimizer. For 100 epochs, categorical accuracy is **82.45% and 84.76% for 4LFLB+LSTM and 6LFLB+LSTM** respectively with same hyper parameters.

TABLE IV. STRUCTURE OF 6 LFLB+LSTM

Block		Output Dim	Kernel Size	Stride
LFFB1	1C1	L x 64	3	1
	1P1	L/4 x 64	4	4
LFFB2	1C2	L/4 x 64	3	1
	1P2	L/16 x 64	4	4
LFFB3	1C3	L/16 x 128	3	1
	1P3	L/64 x 128	4	4
LFFB4	1C4	L/64 x 128	3	1
	1P4	L/256 x 128	4	4
LFFB5	1C5	L/256 x 128	3	1
	1P5	L/256 x 128	4	4
LFFB6	1C6	L/512 x 256	3	1
	1P6	L/512 x 256	4	4
L		-	512	-
F		-	k	-

TABLE V. 1D-CNN+LSTM MODEL ACCURACIES VS NUMBER OF EPOCHS

Network	No. of Epochs	Accuracy
4LFLB+LSTM	50	77.35%
	100	78.65%
	300	90.94%
	500	71.80%
6LFLB+LSTM	50	79.96%
	100	81.53%
	300	91.51%
	500	80.25%

Similarly for 300 epochs **93.64% and 95.11%** and also for 500 epochs, categorical accuracy is **85.72% and 87.95% for 4LFLB+LSTM and 6LFLB+LSTM** respectively with same hyper parameters. The below tables 7 and 8 summarize the network accuracies and confusion matrix for better accuracy network model. Along with the network structure and Hyper Parameters, the accuracy of Deep Learning Algorithms mostly Depends upon the size of the Database and also size of the Train and Test Dataset.

TABLE VI. CONFUSION MATRIX OF 1D-CNN+LSTM FOR 300 EPOCHS WITH 6LFLB+LSTM

	Ang	Sur	Dis	Fea	Hap	Neu	Sad
Ang	95.28	0	0	0.29	3.15	0	0.79
Sur	0	87.65	0	1.23	1.23	9.88	0
Dis	4.35	4.35	82.61	4.35	2.17	2.17	0
Fea	1.45	0	1.45	92.75	2.9	1.45	0
Hap	8.45	0	4.23	0	87.32	0	0
Neu	0	0	0	0	0	100	0
Sad	0	1.61	1.61	0	0	1.61	95.01

TABLE VII. 2D-CNN+LSTM MODEL ACCURACIES VS NUMBER OF EPOCHS

Network	No. of Epochs	Accuracy
4LFLB+LSTM	50	79.50%
	100	82.45%
	300	93.64%
	500	85.72%
6LFLB+LSTM	50	80.85%
	100	84.76%
	300	95.11%
	500	87.95%

TABLE VIII. CONFUSION MATRIX OF 2D-CNN+LSTM FOR 300 EPOCHS WITH 6LFLB+LSTM

	Ang	**Sur**	**Dis**	**Fea**	**Hap**	**Neu**	**Sad**
Ang	99.21	0	0	0	0.79	0	0
Sur	0	94.05	0	0	0	3.7	2.25
Dis	1.73	2.15	96.12	0	0	0	0
Fea	1.45	0	1.45	92.75	2.9	1.45	0
Hap	6.63	0	3.72	0	89.65	0	0
Neu	0	2.50	0	0	0	97.00	0.50
Sad	0	1.38	1.61	0	0	0	97.01

V. CONCLUSION

Deep learning networks for speech emotion recognition are discussed in this chapter. Raw audio files are being studied to see if there is a better way to understand local correlations and global context from them. In order to learn local features, LFLB uses a one convolutional, Batch Normalization, Exponential linear unit, and max-pooling layer. An LSTM layer is used to process altered local characteristics learned by LFLBs. Learned contextual dependencies can be extracted from inputted local characteristics in LSTM's layer. Hybrid networks are able to learn characteristics with both local and long-term context in mind.

Berlin Emo-DB, a database of emotional speech, was used to test the hybrid CNN+LSTM networks that were developed. There are 535 statements that can be interpreted in all applied emotions from regular conversation. The audio clips that were used have a sampling rate of 16 kilohertz. The length of the raw audio clips that were utilised is eight seconds in length. The duration of the audio clip will be cut down to eight seconds if it is greater than that. Otherwise, it is padded to a length of eight seconds. At a sample rate of 16 khz, a 128000-bit vector can be used to describe the audio clip.

In our tests, we used 128000-bit vectors as the input to a one dimensional hybrid (CNN+LSTM) network, The FFT window length is set to 2048, and the hop length is set to 512, while the log-mel spectrogram is being computed as a process. As a result, a log-mel spectrogram is constructed with 251 frames and 128 mel frequency bins. Either a grid or a sequence can be visualised while looking at a log-mel spectrogram. During our trials, the 128 x 251 matrices were used as the input to the 2-Dimensional Hybrid network. Therefore, a high-level feature can be learned by the 2D-CNN+LSTM network from the 2D image-like patches.

According to the findings, the newly developed Hybrid networks are capable of learning and modelling high-level abstractions of emotional data while also learning specific properties. It is clear from Table 9 that 2-Dimensional CNN plus LSTM network outperforms the 1-dimensional CNN plus LSTM network in terms of overall performance. As far as average accuracy goes, the two dimensional hybrid network is superior to other well-established feature representations and algorithms. as well as detecting spatial and regional/temporal data properties, proposed hybrid network has proven to be an effective technique.

TABLE IX. DIFFERENT NETWORK MODEL ACCURACIES FOR SER

Network Model	**Accuracy %**
1-Dimensional CNN	85.47
2-Dimensional CNN	88.36
1-Dimensional CNN plus LSTM	91.51
2-Dimensional CNN plus LSTM	95.52

Even though the deep neural networks that were shown in this study have shown enhanced performance in speech emotion recognition, there are still a great many areas that need to be improved.

REFERENCES

[1] M. Turk, "Perceptive media: Machine perception and human computer interaction," *Jisuanji Xuebao/Chinese J. Comput.*, vol. 23, no. 12, pp. 1235–1244, 2000.

[2] Y. Eom and J. Bang, "Speech Emotion Recognition Using 2D-CNN with Mel-Frequency Cepstrum Coefficients," *J. Inf. Commun. Converg. Eng.*, vol. 19, no. 3, pp. 148–154, 2021, doi: 10.6109/jicce.2021.19.3.148.

[3] "CNN,RNN,LSTM," [Online]. Available: https://cs231n.github.io/rnn/#lstm.

[4] I. Sutskever, O. Vinyals, and Q. V. Le, "Sequence to sequence learning with neural networks," *Adv. Neural Inf. Process. Syst.*, vol. 4, no. January, pp. 3104–3112, 2014.

[5] A. Sherstinsky, "Fundamentals of Recurrent Neural Network (RNN) and Long Short-Term Memory (LSTM) network," *Phys. D Nonlinear Phenom.*, vol. 404, no. March, pp. 1–43, 2020, doi: 10.1016/j.physd.2019.132306.

[6] J. Zhao, X. Mao, and L. Chen, "Speech emotion recognition using deep 1D & 2D CNN LSTM networks," *Biomed. Signal Process. Control*, vol. 47, pp. 312–323, 2019, doi: 10.1016/j.bspc.2018.08.035.

[7] S. Ioffe and C. Szegedy, "Batch normalization: Accelerating deep network training by reducing internal covariate shift," *32nd Int. Conf. Mach. Learn. ICML 2015*, vol. 1, pp. 448–456, 2015.

[8] M. S. Hossain and G. Muhammad, "An Audio-Visual Emotion Recognition System Using Deep Learning Fusion for a Cognitive Wireless Framework," *IEEE Wirel. Commun.*, vol. 26, no. 3, pp. 62–68, 2019, doi: 10.1109/MWC.2019.1800419.

[9] K. S. Rao, S. G. Koolagudi, and R. R. Vempada, "Emotion recognition from speech using global and local prosodic features," *Int. J. Speech Technol.*, vol. 16, no. 2, pp. 143–160, 2013, doi: 10.1007/s10772-012-9172-2.

[10] Y. Li, C. Baidoo, T. Cai, and G. A. Kusi, "Speech Emotion Recognition Using 1D CNN with No Attention," *ICSEC 2019 - 23rd Int. Comput. Sci. Eng. Conf.*, pp. 351–356, 2019, doi: 10.1109/ICSEC47112.2019.8974716.

[11] "Emo-DB." https://www.kaggle.com/datasets/piyushagni5/berlin-database-of-emotional-speech-emodb.