Speech Emotion Recognition Using Deep Learning Methods

Dibbo Dey
Student
EEE,BUET
Dhaka
1906019@eee.buet.ac.bd

Mohammed Ismail Chowdhury
Student
EEE,BUET
Dhaka
1906058@eee.buet.ac.bd

Jarjis Mondal
Student
EEE,BUET
Dhaka
1806068@eee.buet.ac.bd

***Abstract*— Speech emotion recognition (SER) is a crucial task in human-computer interaction systems and affective computing. In this paper, we showed approaches for SER by utilizing 2D CNN LSTM and 1D CNN LSTM. We have used Bidiectional and Unidirectional LSTM in these approaches. We have showed that use of BILSTM increases accuracy, Our method capitalizes on the temporal dependencies present in speech signals by leveraging the sequential modeling capabilities of LSTM networks, while the 2D CNN layers effectively capture spatial features from the logmelspectrogram representations. The architecture is trained and evaluated on the Emotional Database (EMODB), a widely used dataset for emotion recognition in speech. Our validation accuracy was 80.68% for the method based on our mother research paper. Our own 2D CNN LSTM gives 62% accuracy and 2D BiLSTM gives 66.38% accuracy. 1D CNN LSTM gives 65.4% and 1D CNN-BiLSTM gives 66.15% accuracy. We adapted publicly available implementation from a GitHub repository, inspired by prior research in the field. Our experiments demonstrate that use of BiLSTM improves the accuracy.**

Keywords—LSTM, CNN, SER

I. Introduction

Speech is a fundamental mode of human communication, conveying not only linguistic information but also rich emotional cues essential for effective interpersonal interaction. Recognizing and understanding emotions conveyed through speech is therefore a critical task in various domains, including human-computer interaction, sentiment analysis, and mental health monitoring. Speech emotion recognition (SER) aims to automatically classify the emotional states of speakers based on their vocal expressions, enabling machines to perceive and respond to human emotions in a manner akin to human cognition. Traditional approaches to SER often rely on handcrafted features extracted from speech signals, such as pitch, intensity, and spectral characteristics. However, these methods often struggle to capture the complex and dynamic nature of emotional expressions, leading to limited performance in real-world scenarios. In recent years, deep learning techniques, particularly Convolutional Neural Networks (CNNs) and Recurrent Neural Networks (RNNs), have emerged as powerful tools for automatically learning discriminative features directly from raw data, revolutionizing various domains, including computer vision, natural language processing, and speech processing. In this paper, we implemented a SER using deep learning techniques, specifically a combination of 2D CNNs and Long Short-Term Memory (LSTM) networks, applied to log-mel spectrograms of speech signals following the research paper “Speech emotion recognition using deep 1D & 2D CNN LSTM networks” by Jianfeng Zhaoa,b, Xia Maoa, Lijiang Chena,∗. Log-mel spectrograms offer a compact and informative representation of the spectral content of speech signals, facilitating the extraction of discriminative features relevant to emotional content. The integration of 2D CNNs and LSTM networks enables our model to capture both spatial and temporal dependencies inherent in speech data, effectively capturing contextual information crucial for accurate emotion recognition. We have also utilized Bidirectional LSTM and showed the accuracy improvement after using BiLSTM. We conduct experiments on the Emotional Database (EMODB), a widely used dataset for SER research, to evaluate the performance of our proposed approach. By adopting a publicly available implementation from a GitHub repository, inspired by prior research in the field, we demonstrate promising results, achieving a validation accuracy of 80.68% on the EMODB dataset. The code that we implemented is based on the following github repository ::

https://github.com/Hbbbbbby/EmotionRecognition_2Dcnn-lstm/tree/main

Another code that we have used for 1D CNN-LSTM is based on the following public available github repository:

https://github.com/vandana-rajan/1D-Speech-Emotion-Recognition/blob/master/cnn1d.py

Our implementations will be uploaded in the following repository:

https://github.com/ExploringCodes/SER

II. Methodology

Data Collection :

For our project we have used Berlin Emodb database which is publicly available. It has 535 audio files. There are seven emotions. Ten professional speakers speak those emotional utterances. Seven emotions are angry, boredom, disgust, fear, happy, neutral and sadness. The audio are of varying length and all are sampled at 16 KHz. We have taken 3.5s of clip and perform the training based on that. Speech signal that is shorter than 3.5 s are zero padded to make all the dataset 3.5s in duration.

Deep Feature Learning:

In traditional machine learning algorithm features are handcrafted and various feature extraction and feature transformation methods are used. However in case of Deep

Learning features are learned automatically. This is a great advantage of Deep Neural Network.

The features that we consider for the SER is Logmelspectrogram which is achieving good accuracy in many database in a deep learning framework.

LogMelSpectogram(LMS) basically perform STFT on the speech signal so that speech signal of 1D transforms to 2D of frequency and time. However the frequency that we get in case of LMS is not perceived by human just by their magnitude, rather there exists a scale. Hence, to take account of human perception we convert those frequencies into Mel Frequencies. Then we take logarithm of the magnitude to obtain LMS.

In case of 2D CNN-LSTM based on the research paper the architecture is demonstrated in the following table.

Table 2
The layer parameters of the 2D CNN LSTM network. The output dimension is represented as height × width × number. M × N is the size of the low-level features. The kernel size K of 2 F is the number of the emotions. 2C1 and 2P1 are the convolutional layer and the max-pooling layer of 2 LFLB1, and so on.

Name		Output Dim	Kernel Size	Stride
2 LFLB1	2C1	M × N×64	3 × 3	1 × 1
	2P1	M/2 × N/2 × 64	2 × 2	2 × 2
2 LFLB2	2C2	M/2 × N/2 × 64	3 × 3	1 × 1
	2P2	M/8 × N/8 × 64	4 × 4	4 × 4
2 LFLB3	2C3	M/8 × N/8 × 128	3 × 3	1 × 1
	2P3	M/32 × N/32 × 128	4 × 4	4 × 4
2 LFLB4	2C4	M/32 × N/32 × 128	3 × 3	1 × 1
	2P4	M/128 × N/128 × 128	4 × 4	4 × 4
2 L		–	256	–
2 F		–	K	–

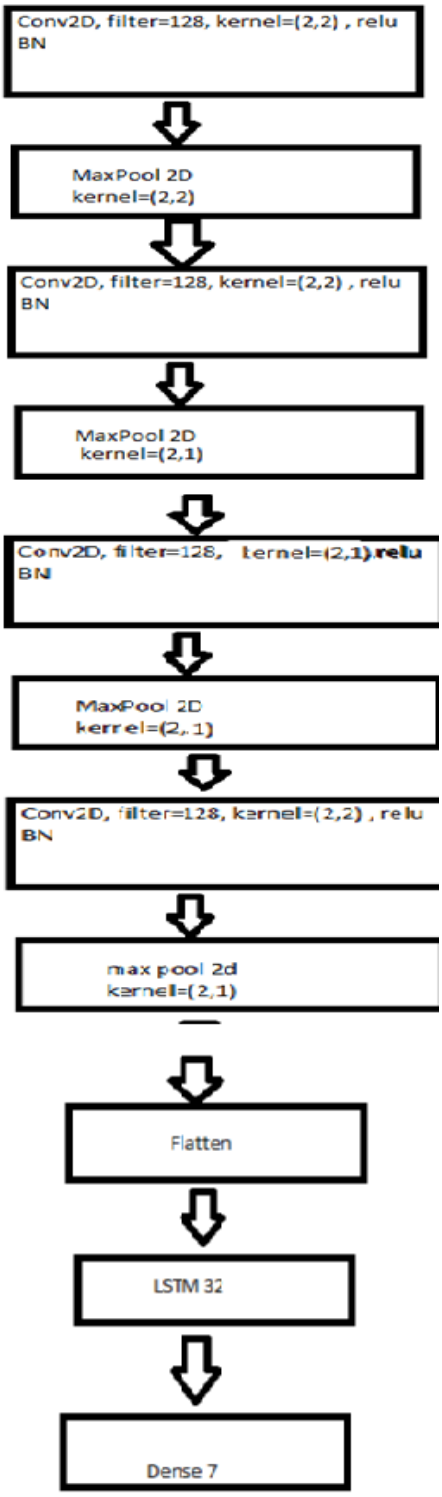
The methodology employs a deep learning architecture designed to extract high-level emotional features from log-mel spectrograms of speech signals. The architecture consists of several key components: four Local Feature Learning Blocks (LFLBs), one Long Short-Term Memory (LSTM) layer, and one fully connected layer (FCL), as illustrated in Table 2

Local Feature Learning Blocks (LFLBs): The architecture incorporates four LFLBs, denoted as 2LFLB1, 2LFLB2, 2LFLB3, and 2LFLB4. Each LFLB utilizes two-dimensional convolutional and pooling kernels to learn local features with local correlations from the input log-mel spectrograms. The convolution kernels have a fixed size of 3 × 3, a stride of 1 × 1, and employ SAME padding to maintain spatial dimensions. The number of convolution kernels varies across LFLBs, with 64 kernels in 2LFLB1 and 2LFLB2, and 128 kernels in 2LFLB3 and 2LFLB4. Additionally, the max-pooling operation is applied with a kernel size of 2 × 2 in the first LFLB and 4 × 4 in subsequent blocks.

- Long Short-Term Memory (LSTM) Layer: Following the four LFLBs, the features extracted are reshaped into a temporal sequence and inputted into the LSTM layer (2L). The LSTM layer is responsible for capturing contextual dependencies among the learned local features, thereby incorporating global contextual information into the feature representation.
- Fully Connected Layer (FCL) and Softmax Classifier: Subsequently, the features outputted from the LSTM layer are fed into a fully connected layer (2F), which generalizes these features into the output space. Finally, a softmax classifier is applied at the top layer to predict the emotional states based on the learned features, which encode both local correlations and global contextual dependencies.

2D CNN-LSTM Time Distributed:

We have made a modification of the above model by not performing 2D CNN on entire LMS but rather we have divided the LMS into a group of sequences each having a number of samples. Then we perform CNN on each of these portions independently. Finally we perform flattening and put the sequence in the LSTM block. LSTM learns the temporal dependencies and finally the fully connected layer perform classification The architecture is given in figure-1.



.

Figure-1

In the architecture we have used LFLB similar to our mother research paper. However the shape of the LSTM here is (128,219) and shape of the input to CNN layer is (535,18,128,5,1). Hence we have divided the time frame of LMS of 219 length into 18 sequences each having length of 5 samples. We have performed CNN on each of these 128*5*1 block independently. Finally features for 18 sequences obtained after flattening the CNN outputs which goes to LSTM layer. In this case LSTM cell number will be 18.

2D CNN-BILSTM Time Distributed:

The architecture of 2D CNN-BILSTM is same as 2D CNN LSTM except that we are using BiLSTM instead of 2D CNN LSTM. The model summary is

Layer (type)	Output Shape	Param
time_distributed_26 (TimeDistributed)	(None, 18, 127, 4, 128)	640
time_distributed_27 (TimeDistributed)	(None,18,127,4,128)	512
time_distributed_28 (TimeDistributed)	(None,18,63,2,128)	0
time_distributed_29 (TimeDistributed)	(None,18,62,1,128)	65664
time_distributed_30 (TimeDistributed)	(None,18,62,1,128)	512
time_distributed_31 (TimeDistributed)	(None,18,31,1,128)	0
time_distributed_32 (TimeDistributed)	(None,18,30,1,128)	32896
time_distributed_33 (TimeDistributed)	(None,18,30,1,128)	512
time_distributed_34 (TimeDistributed)	(None,18,15,1,128)	0
time_distributed_35 (TimeDistributed)	(None,18,14,1,128)	32896
time_distributed_36 (TimeDistributed)	(None,18,14,1,128)	512
time_distributed_37 (TimeDistributed)	(None,18,7,1,128)	0
time_distributed_38 (TimeDistributed)	(None,18,496)	0
Bidirectional (Bidirectionnal)	(None,64)	237824
Dense_2(Dense)	(None,7)	455

Total params: 372423 (1.42 MB)

Trainable params: 371399 (1.42 MB)

Non-trainable params: 1024 (4.00 KB)

1D CNN-LSTM:

1 D CNN-LSTM architecture is same as 2D CNN LSTM except that here 1D CNN is being used. The model for 1D CNN-LSTM is based on the following table:

Table 1
The layer parameters of the 1D CNN LSTM network. The output dimension is given by length × number. L is the length of the audio clip. The kernel size K of 1F is the number of the emotions. 1C1 and 1P1 are the first convolutional layer and the max-pooling layer of 1LFLB1, and so on.

Name		Output Dim	Kernel Size	Stride
1 LFLB1	1C1	L×64	3	1
	1P1	L/4 × 64	4	4
1 LFLB2	1C2	L/4 × 64	3	1
	1P2	L/16 × 64	4	4
1 LFLB3	1C3	L/16 × 128	3	1
	1P3	L/64 × 128	4	4
1 LFLB4	1C4	L/64 × 128	3	1
	1P4	L/256 × 128	4	4
1 L		–	256	–
1 F		–	K	–

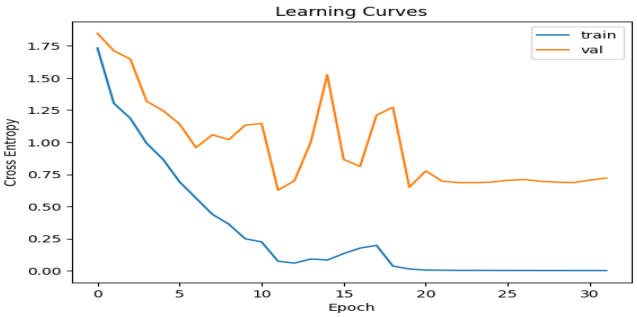
1D CNN-BiLSTM:

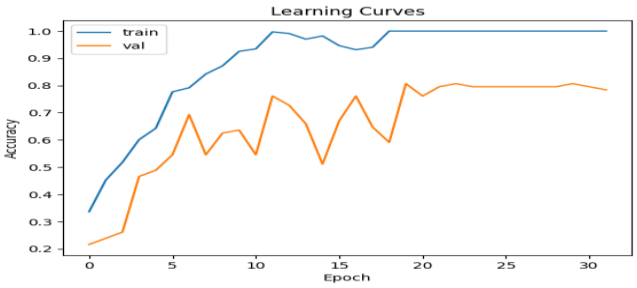
1D CNN-BiLSTM architecture is same as 1D CNN-LSTM. The only difference is in the LSTM cells where we are using BiLSTM instead of LSTM.

III. RESULT

The result of our research indicates that the best validation accuracy obtained by following the architecture of the research paper that we have followed. The architecture that we have implemented show less accuracy but we have showed that use of BiLSTM improves accuracy in our architecture.

Learning curve for the 2D CNN LSTM:

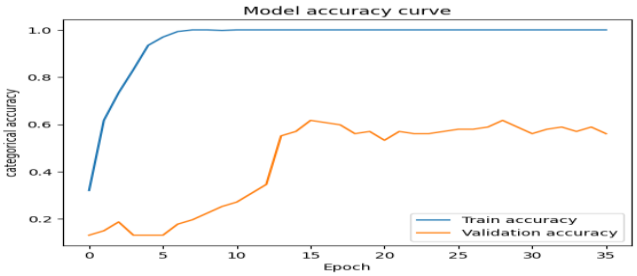




The best validation accuracy is 80.68%

2D CNN-LSTM Time Distributed:

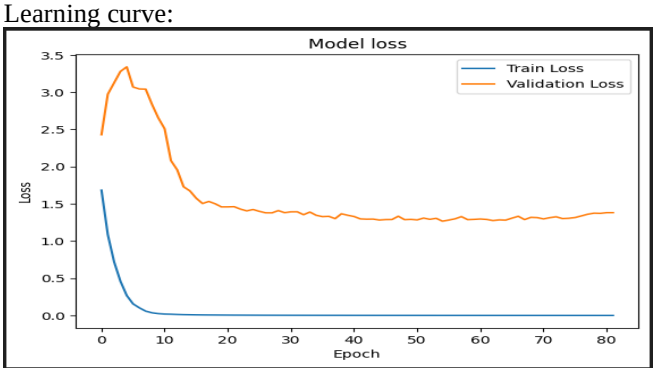
Learning curve:

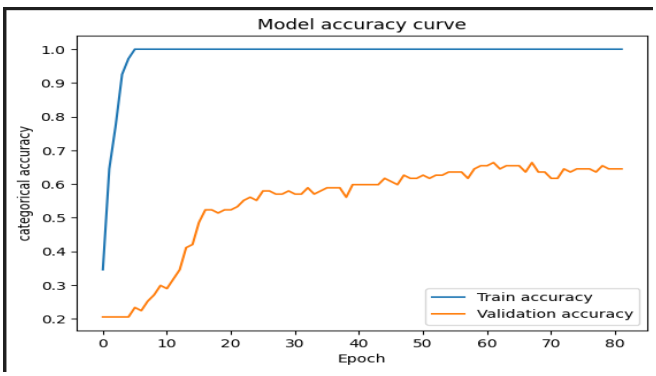


Best validation accuracy is 62%

2D CNN-BILSTM Time Distributed:

Learning curve:



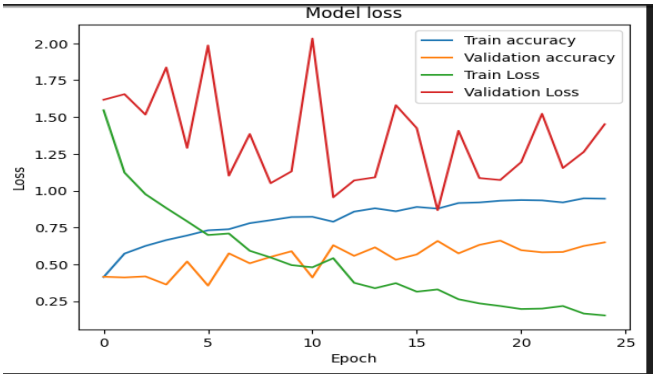


Best Validation accuracy is 66.38%

1D CNN LSTM :
Learning curve:

Best validation accuracy is 65.4 %

1D CNN-BiLSTM:
Moder learning curve:



The best accuracy is 66.15%

Comparison:

Model	Accuracy
2D CNN LSTM (from research paper)	80.68%
2D CNN LSTM Time Distributed	62%
2D CNN-BiLSTM Time Distributed	66.38%
1D CNN-LSTM (research paper)	65.4%
1D CNN-LSTM time distributed	66.15%

IV Discussion:

In this experiment we have performed CNN-LSTM on LMS. Use of CNN-LSTM learn global and local feature from the LMS. We have also used our own designed architecture which even though performed less accuracy still gives an indication of that use of BILSTM improves accuracy.
From the result we can see that BiLSTM improves accuracy by learning sequence from the previous and forward cells as well.
We still have some more investigative issues here. In the research paper the CNN was performed on the whole LMS and the output of CNN has 128 features. The paper considers that these 128 features can be considered as sequence. How is this the case requires further investigation. Also our dataset is quite small with only 535 data. Hence use of data augmentation techniques could have improved our accuracy. In future research we will explore data augmentation for improving accuracy.

Overfitting is one of the main problem of deep learning model training. To reduce the overfitting we have used the Batch Normalisaton layer and early stopping.

V Conclusion

We have performed SER on EmoDB database using five deep learning model. The first one is from the research paper and the remaining four models are designed. The highest accuracy is obtained from the architecture that is based on research paper. Our models only show that improved accuracy upon using BiLSTM. The only place where we have obtained more accuracy than the research paper is where we have used BiLSTM layer.

Since EmoDB database small, the data augmentation techniques are crucial for improving accuracy. In Future research we will utilize data augmentation and transfer learning for improving accuracy.

VI REFERENCES

[1] J. Zhao, X. Mao, and L. Chen, "Speech emotion recognition using deep 1D & 2D CNN LSTM networks," School of Electronics and Information Engineering, Beihang University, Beijing, China and School of Information Engineering, Inner Mongolia University of Science & Technology, Baotou, China.J. Clerk Maxwell, A Treatise on Electricity and Magnetism, 3rd ed., vol. 2. Oxford: Clarendon, 1892, pp.68–73.

[2] https://github.com/Hbbbbbby/EmotionRecognition_2Dcnn-lstm/tree/main

[3] https://github.com/vandana-rajan/1D-Speech-Emotion-Recognition/blob/master/cnn1d.py