Deep Learning Techniques for Speech Emotion Recognition : A Review

1st Sandeep Kumar Pandey, 2nd H.S.Shekhawat, 3rd S.R.M.Prasanna
Indian Institute of Technology Guwahati[1,2,3]
Guwahati, India
Indian Institute of Technology Dharwad[3]
Dharwad, India
sandeep.pandey@iitg.ac.in

***Abstract*—This paper presents an introduction to various deep learning techniques with the aim of capturing and classifying emotional state from speech utterances. Architectures such as Convolutional Neural Network(CNN) and Long Short-Term Memory(LSTM) have been used to test the emotion capturing capability from various standard speech represenations such as mel spectrogram, magnitude spectrogram and Mel-Frequency Cepstral Coefficients (MFCC's) on two popular datasets- EMO-DB and IEMOCAP. Experimental findings along with reasoning have been presented as to which architecture and feature combination is better suited for the purpose of speech emotion recognition. This work explores the widely used basic deep learning architectures used in literature.**

***Index Terms*—Deep Learning, speech emotion, recognition/identification**

I. Introduction

With the advent of technology, there is an ever arising need to make the Human-Computer Interaction (HCI) more natural. As speech is the most easiest and effective form of communication for human beings, its natural that we would want our machines to be able to understand us completely based on our speech commands. But apart from the message, speech signal carries secondary level information also such as gender, emotional states etc. As such, it becomes obvious that the HCI should also be able to capture those information and use it to make the interaction process smoother and contextual pertaining to the emotional state of the speaker. Therefore, speech emotion recognition (SER), which aims at identifying the emotional state from the speech utterances, have drawn particular interest among the researchers.

Over the past decade, several techniques have been employed to identify the emotional state of a speaker from his/her speech utterances.The traditional methods relied on extracting acoustic features from speech utterances such as MFCC, pitch and its harmonics, shimmer, jitter etc and subsequently classifying them using traditional classifiers like GMM, HMM, SVM. But the performance couldn't be improved beyond a certain limit as hand crafted features are not much capable in capturing a complex phenomena such as emotional state [1]. Over the past decade, the deep learning techniques have been employed in several fields of research such as image classification, speaker recognition, speech and handwriting recognition etc because of its ability to capture high level discriminative features which are otherwise not captured well in traditional hand crafted features.

In [2], emphasis have been given on extracting emotional 'context aware' features using a combination of Convolutional Neural Network (CNNs) and LSTM network in an end-to-end fashion, which outperforms the baseline Support Vector Regression and BLSTM-DRNNs over valence and arousal classification. In [3], different acoustic feature sets of varying complexity is evaluated using a Recurrent Neural Network (RNN) as well as CNN extracted features are evaluated, only to conclude that there is no clear winner. Several researchers have tried to extract emotion salient information directly from raw speech signals too, instead of levels of feature extraction and then feeding to deep learning architectures. In this regard, [4] utilised both the waveform and spectrogram in a parallel architecture with temporal convolutions for extracting complimentary information and finally used stacked BLSTM layers for the arousal and valence classification. This method showed slight improvement over the spectrogram only or waveform only models. Moreover in [5], raw speech waveform based front end layers to learn emotion specific ques within the network and end-to-end DNN setup for emotion identification has been proposed. Different architectures have been experimented such as - Time delay Neural Networks (TDNN) and unidirectional recurrent projected LSTM, also interleaving TDNN-LSTM setup with attention layer. With an objective to increase the efficiency of the architectures, attention mechanism became popular so as to focus on the emotional salient parts of the utterances. The work in [6], [7], [8] and [9] utilizes attention layer in addition to the conventional RNN or CNN layers to increase the emotion capturing capability of the models. Overall, deep learning is being widely used in emotion recognition studies as learning emotion ques from utterances is a tedious task and hand-crafted features are not much successful in capturing the emotional information. Many papers such as [10], [11], [12] and [13] have surveyed the emotion recognition work but their entire focus is on handcrafted features and traditional classifiers. A proper survey on the effectiveness of various features and deep learning architecture is still missing. This paper attempts to present a lucid and comprehensible study of simple deep learning architectures and how efficient they are

in capturing emotion salient information from various standard speech representations.

The rest of the paper is organized as follows. Section II describes the standard deep learning techniques in brief with the motivation behind using those architectures for the Speech Emotion Recognition (SER) task. Section III describes the datasets and the experiments conducted using various features and architectures. Section IV presents the results on the two datasets used and Section V concludes the paper with discussion on the architecture-feature combinations.

II. Deep Learning Techniques for SER

Several deep learning architectures have been employed for the task of SER. Among the most popular ones are the feed-forward and the recurrent neural network architecture. A comparison among the three widely used architectures is discussed along with the merits and shortcomings.

A. Deep Neural Network

Feed-forward network architectures consists of multiple layers of affine transformation followed by non-linear functions which are executed in a sequential fashion. The layers in between the input and output layers are usually referred to as the hidden layers. The last layer in the network is the softmax layer, which gives the class probabilities pertaining to the different classes. The parameters for the different layers comprises of the weights and biases. The weights define the connection between the input neurons to the hidden units, and gets updated iteratively. The weight and bias update process is performed by minimizing the classification error, basically cross-entropy for the classification task, which is optimized using Stochastic Gradient Descent or optimizers such as Adam [14], AdagGrad [15] etc. The forward process of a Deep neural network can be described using the equations [16] -

$$h^{(l)} = y^{(l-1)}W^{(l)} + b^{(l)}, \tag{1}$$

$$y^{(l)} = \phi(h^{(l)}), \tag{2}$$

where, $h^{(l)} \in \mathcal{R}^{hidden}, l \in \{1, \cdots L\}$ represents the output of the l-th hidden layer before passing through the elementwise non-linear activation function ϕ. The input of the l-th hidden layer is $y^{(l-1)} \in \mathcal{R}^{input}$, which comes from the previous layer. $W^{(l)} \in \mathcal{R}^{input \times hidden}$ is a matrix of learnable weights for the layer l and $b^{(l)} \in \mathcal{R}^{hidden}$ is a vector of learnable bias for layer l. The activation function used in the L-the layer, which have nodes equal to the number of classes, is softmax which gives class probabilities as output. We use the feed-forward architecture as fully connected layers for CNN and LSTM output in our work.

B. Convolutional Neural Network

Convolutional neural Networks(CNN) is a popular variant in the feed-forward architecture. CNN utilizes the convolution operation to replace the affine transformation as in case of DNN. Among the many merits of CNN are the utilization of spatial information, which is of importance for image data, higher-level feature extraction from raw images directly and parameter sharing. CNN usually consists of a convolution layer which utilizes a set of convolutional filters to extract multiple local patterns at each local region in the input space and produce many feature maps. Mathematically, this layer can be written as [16]

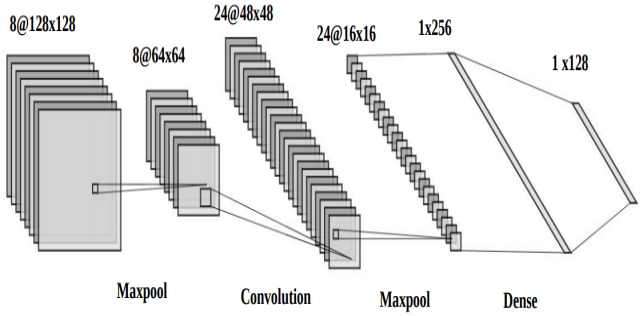

Fig. 1. A Convolutional Neural Network Architecture with pooling and Fully Connected layers.

$$(h_k)_{ij} = (W_k * q)_{ij} + b_k, \tag{3}$$

where, $(h_k)_{ij}$ denotes the (i, j) element of the k-th output feature map, q represents the input feature maps, W_k and b_k denotes the k-th filter and bias, respectively. The symbol $*$ represents a 2-D spatial convolution operation.

The convolutional layer is usually followed by a pooling layer which down samples the feature maps obtained through convolution operations, thereby keeping a single output from local regions of feature maps, thus reducing the number of parameters to be learned in the subsequent layers. Two pooling methods widely used are- max pooling, which takes the maximum value of the specified window for the pooling layer and average pooling, which takes the average of the values in the window specified for the pooling layer.

The convolutional layers in combination with pooling layers forms multiple layers of a deep architecture, before being finally followed by fully connected (FC) layers. The FC layers is responsible for integrating the output of the last layer for classification/regression tasks.The output of a fully-connected layer is calculated as described in section A.

C. Long Short Term Memory

Adding another feather to the cap of deep learning is the recurrent network architecture. It extends the notions of a feed forward architecture by adding self connections to units as well as hidden layers at previous time steps. Recurrent architectures are particularly relevent in scenarios where temporal sequence modelling is of concern such as speech processing, speech to text etc.

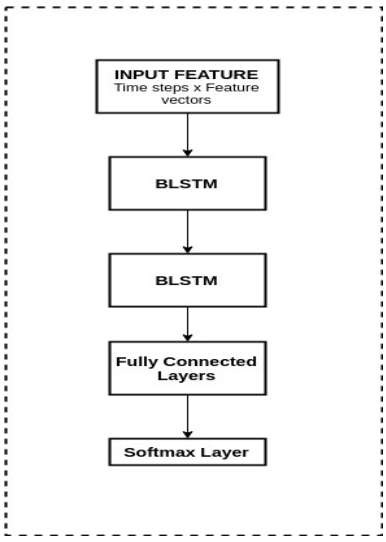

Fig. 2. A stacked Bi-directional LSTM with fully connected output layers.

The feed-forward step of the recurrent architecture can be mathematically represented as [16] [17]-

$$h_t^{(l)} = y_t^{(l-1)} W_y^{(l)} + s_{t-1}^{(l)} W_s^{(l)} + b^{(l)}, \tag{4}$$

where, h_t^l is the hidden layer output prior to activation , at time step t. $y_t^{(l-1)}$ is the output of the previous layer $(l-1)$ and input to layer l at time step t, $W_y^{(l)}$, $W_s^{(l)}$ are matrices of learnable weights for layer l and $s_{t-1}^{(l)}$ is the state of the layer at previous time step $(t-1)$.

RNN suffers from the problem of vanishing or exploding gradients because it tries to use the entire previous time frames. Long Short Term Memory (LSTM) is an advanced version of a simple RNN which tackles this problem by using gates to control information flow. The design mechanism of LSTMs are such that it learns the long-term dependencies in sequences, thus capturing global information over utterances. An LSTM can be mathematically represented by -

$$i_t^{(l)} = \sigma(W_{yi} y_t^{(l-1)} + W_{hi} h_{t-1}^l + W_{ci} c_{t-1}^l + b_i^{(l)}), \tag{5}$$

$$f_t^{(l)} = \sigma(W_{yf} y_t^{(l-1)} + W_{hf} h_{t-1}^l + W_{cf} c_{t-1}^l + b_f^{(l)}), \tag{6}$$

$$c_t^{(l)} = f_t c_{t-1} + i_t \tanh(W_{yc} y_t^{(l-1)} + W_{hc} h_{t-1}^l + + b_c^{(l)}), \tag{7}$$

$$o_t^{(l)} = \sigma(W_{yo} y_t^{(l-1)} + W_{ho} h_{t-1}^l + W_{co} c_t^l + b_o^{(l)}), \tag{8}$$

$$h_t^{(l)} = o_t^{(l)} \tanh(c_t^{(l)}), \tag{9}$$

where σ is non-linear activation -the sigmoid function. i, f, o are the input, forget and output gate respectively and c is the cell state or cell activation vector. y_t^l is the output vector at time instant t for layer l. W are the different weight matrices connecting the gates to the cell states and are trainable. For example, W_{ci} is the weight matrix from the cell to the gate vectors.

III. Experimental Evaluation

A. Datasets

The deep learning models were evaluated on two widely used Speech Emotion datasets with different languages - Emo-DB and IEMOCAP. The Berlin Emotional Database(Emo-DB) [18] is a German emotional speech corpus consisting of seven emotion categories i.e anger, disgust, fear, happiness, sadness, surprise and neutral recorded from ten German actors. The speech utterances were recorded using a sampling rate of 16 kHz with a 16-bit resolution and mono channel and the entire dataset comprises of 535 sentences. The average duration of the audio files is 3 seconds. We selected the four basic emotions- anger, happiness, neutral and sadness for our experiment purpose.

The Interactive Emotional Dyadic Motion Capture (IEMO-CAP) [19] is an English Emotional Speech database consisting of audiovisual data from 10 actors, of which only the audio part is used for our experimentation . It contains approximately 12 hours of audiovisual data which includes video, speech, motion capture of face and text transcriptions. It consists of dyadic sessions where actors perform improvisations or scripted scenarios which are specifically selected to elicit emotional expressions. Multiple annotators have annotated the utterances in both the categorical as well as dimensional labels. The categorical classes are- Anger, Happiness, Sadness, Neutral, Surprise, Disgust, Fear, Frustration and Excited. The Dimensional classes are- valence, activation and dominance.

In our work, we have used only the four basic emotions - Angry, Happy, Neutral and Sad. Since the number of utterances for Happy emotion is less compared to other emotion classes in IEMOCAP, we treated the excited and happy class as one to increase the number of utterances.

B. Feature Extraction

Three different feature types are investigated using the deep architectures - Magnitude Spectrograms , log-Mel Spectrograms and Mel-Frequency Cepstral Coefficients (MFCC) features. This three features are selected after reviewing several works on emotion recognition. For spectrograms as input feature, the time domain speech files are windowed using a hanning window of 20 ms window size with 10 ms overlap and magnitude spectrum which is a time-frequency representation of the speech is obtained. Zero padding is done to ensure that the spectrograms are of equal size, as required by a CNN input. In case of log-mel spectrograms , the computed magnitude spectrogram is mapped to a mel-scale and subsequently to a log operator to obtain the mel-spectrogram. The mel spectrogram are different from the magnitude spectrograms in the sense that the mel filter bank mimics the human ears perception and emphasizes the lower frequency region more than the higher frequencies. For LSTM architecture, MFCC features are calculated for the speech utterances over a sliding window of size 20 ms and shift 10 ms. The delta and double-delta features are calculated for each frame and concatenated with the MFCC's to form feature

vectors. Each frame represents an input to the LSTM at a time instant.

C. Architecture

The CNN architecture used for spectrogram and mel spectrogram [9] inputs consists of 3 convolutional layers with varied feature maps, dropout layers, and finally fully connected dense layers with softmax layer as the output. The input shape of the spectrogram image is 129×251. The size of the kernel for the first convolutional layer is 4×4 and 32 feature maps and for the subsequent convolutional layers 3×3 and 64 and 128 feature maps.The Convolutional layers capture higher level emotion salient 2D features from the spectrogram. Max-pooling of size 2×2 is used to reduce the size of the feature maps so as to reduce the number of trainable parameters. Fully connected Dense layers along with a softmax layer is used to generate class probabilities for each speech spectrograms.

The LSTM architecture used is bi-directional (BLSTM) to better capture the temporal dependencies in both the previous as well as the next time steps. LSTM of cell size 128 is used , which is followed by fully connected and softmax layers.The input for the BLSTM requires feature vectors with timesteps and as such the frames of features for the speech utterances is used as the time-steps and a sequence-to-one modelling is used to train the BLSTM.

Also, a combination of CNN and BLSTM is evaluated to perform end-to-end emotion recognition with the CNN being the feature extractor and LSTM modelling the temporal information with fully -connected layers for producing the class probabilities. This is motivated by the fact that hand-crafted features, directly fed to an LSTM network, is not sufficient enough to model the intricate variabilites of emotions in an utterance. As such, CNN is used as feature extractor to extract high -level features and subsequently LSTM is used as a global feature extractor which models the utterance level long-term dependencies.

Batch Normalization is used after the Convolution Layers to ensure that the data passing on to higher layers are normalized, which enhanced the stability and performance as well as fastens the training procedure of deep networks. [17] [20]. Dropout is also used to overcome over-fitting problem in Deep networks. [21]

D. Optimization and Hyper-parameter tuning

Since SER is a multi-class classification problem, cross-entropy loss is used. The models are optimized using Adam optimizer [14] [15]. The method computes individual adaptive learning rates for different parameters from estimates of first and second moments of the gradients. Adam is selected as it is computationally inexpensive and combines the benefit of both AdaGrad and RMSprop optimizers [15]. The values for the parameters of Adam were $\beta_1 = 0.9$ and $\beta_2 = 0.999$ with an initial learning rate of 0.001. Depending on the size of training data available, batch size of 10 for EMO-DB and 25 for IEMOCAP dataset is used and training is done for 20 epochs. A small batch size results in convergence to flat minimizers and help in generalization of the parameters learned to deal with unseen data. However, large batch-size tends to converge to sharp mimimizers which leads to poor generalization and degradation in the quality of the model [22]. The activation function used is RELU since it is computationally efficient and the likelihood of vanishing gradient is low and also shows better convergence than sigmoid activation [23]. Dropout with probability of 0.25 is used before the fully connected layers. Dropout helps in the generalization of the model by helping the model to learn more robust features. Different topologies were evaluated for each case by varying the number of Convolution layers and filters in CNN and the number of cells in LSTM and the best performing one is reported in this work.

TABLE I
RECOGNITION ACCURACIES WITH STANDARD DEVIATION FOR 5-FOLD CROSS VALIDATION OF VARIOUS ARCHITECTURES ON EMO-DB USING MAGNITUDE SPECTROGRAM,MEL-SPECTROGRAM AND MFCC FEATURES AS INPUT.

Feature used	Architecture		
	CNN	BLSTM	CNN +BLSTM
Spectrogram	53.11%(3.86%)	31.58%(5.63%)	65.71%
Mel-Spectrogram	**78.16 %(3.87%)**	51.90%(4.28%)	71.97%(3.42%)
MFCC	74.93%(2.09%)	**66.61 % (8.40 %)**	**82.35 %**

TABLE II
RECOGNITION ACCURACIES WITH STANDARD DEVIATION FOR 5-FOLD CROSS VALIDATION OF VARIOUS ARCHITECTURES ON IEMOCAP USING MAGNITUDE SPECTROGRAM,MEL-SPECTROGRAM AND MFCC FEATURES AS INPUT.

Feature used	Architecture		
	CNN	BLSTM	CNN +BLSTM
Spectrogram	42.02%(4.0%)	34.58%(2.68%)	**46.80%(2.23%)**
Mel-Spectrogram	47.04%(1.08%)	44.73%(1.55%)	**50.05%(3.0%)**
MFCC	45.34% (1.24%)	**46.21%(1.96%)**	45.61%(2.00 %)

IV. RESULTS

The SER performance is evaluated using the Weighted Accuracy (WA), which is the classification accuracy over all the speech utterances. Several architectures each for CNN , LSTM and CNN+LSTM have been experimented with to find the one giving the best accuracies and only architectures giving the best results have been presented. Confusion matrix for both the datasets are also reported.

1) Result for Emo-DB dataset: The number of Training and testing utterances for Emo-db are 271 and 68 respectively. Five-fold cross validation is performed to ensure that the results are not biased towards a particular train-test split. Table 1 presents the accuracies over various features and architecture combination. The best accuracy is 82.35%, achieved for CNN + BLSTM architecture with MFCC as input . The confusion matrix is reported for this architecture in Fig.3. As seen from the confusion matrix in Fig.4, happy speech is not being modelled well by the architecture, which can be attributed to the following two reasons - number of utterances for happy speech is less compared to other two classes and happy and

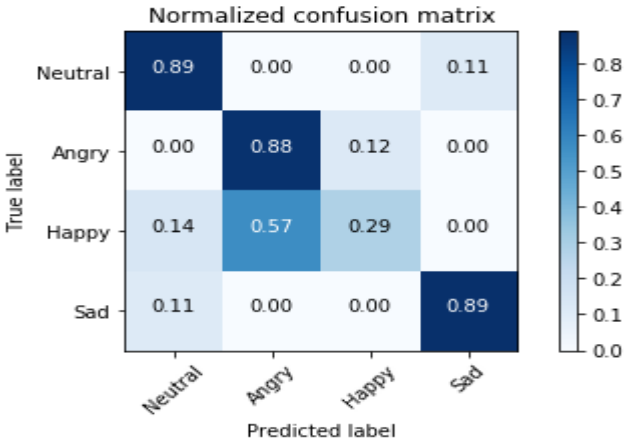

Fig. 4. Confusion matrix for CNN+LSTM architecture using MFCC as input for EMO-DB.

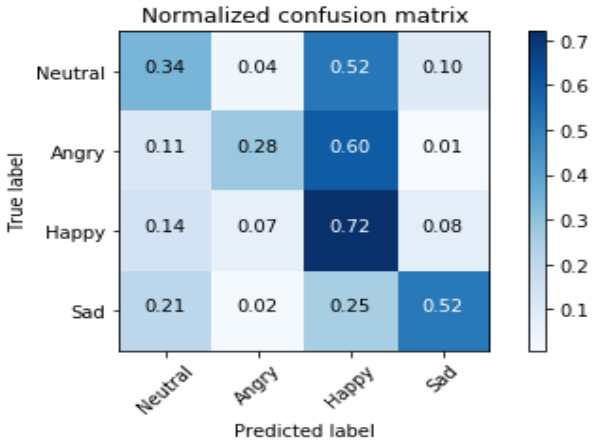

Fig. 6. Confusion matrix for CNN+LSTM architecture using Mel-Spectrogram as input for IEMOCAP.

angry being both belonging to the arousal category, the subtle difference is unable to being captured by the architecture.

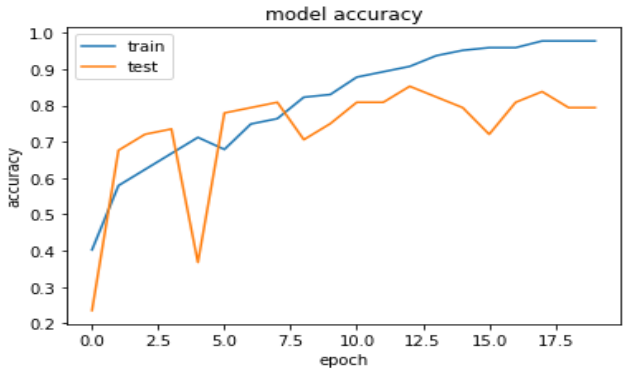

Fig. 3. The CNN+LSTM networks training and testing accuracy over epochs on Emo-DB.

2) Result for Iemocap Dataset: : Iemocap Dataset is divided into train and validation sets containing around 4424 and 1107 utterances respectively comprising of both the scripted and improvised scenario of the dataset. Cross validation of 5 fold is applied and the architecture-feature combinations are tested. Table 2 presents the recognition accuracies with the best accuracy being obtained for CNN+BLSTM architecture with Mel-Spectrogram as feature, which is in line with the result stated in [17]. Confusion matrix reported for this case in Fig.5 shows that the architecture is successful in modelling the four basic emotions even in near-natural emotional utterances. The drop in accuracy for IEMOCAP as compared to Emo-DB can be attributed to the more naturalness of elicited emotions in IEMOCAP.

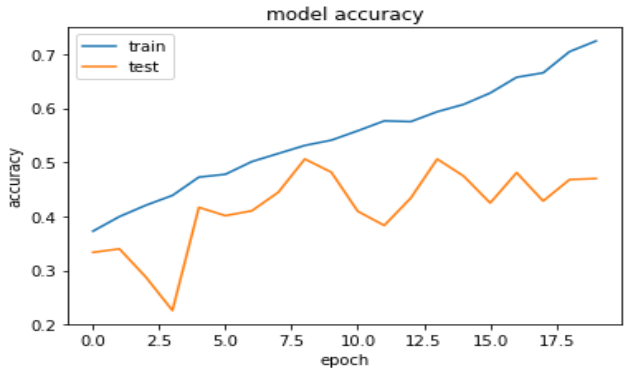

Fig. 5. The CNN+LSTM networks training and testing accuracy over epochs on IEMOCAP using Mel-Spectrogram as input.

V. Conclusion

We presented a comprehensive review of the popular deep learning algorithms for speech emotion recognition. Various features are tested such as Magnitude spectrogram, Log-Mel Spectrogram and MFCCs in combination with the architectures to reveal the best feature-architecture combination. Experiments conducted on the two widely used datasets - Emo-DB and IEMOCAP shows that the architecture performs good with Log-Mel Spectrograms in combination with CNN+LSTM architecture. The models suffers from the drawback of overfitting when the data is less as in Emo-DB case, which is taken care of using regularization techniques such as Dropout and Batch Normalization.

References

[1] S. Zhang, S. Zhang, T. Huang, and W. Gao, "Speech emotion recognition using deep convolutional neural network and discriminant temporal pyramid matching," *IEEE Transactions on Multimedia*, vol. 20, no. 6, pp. 1576–1590, 2018.

[2] G. Trigeorgis, F. Ringeval, R. Brueckner, E. Marchi, M. A. Nicolaou, B. Schuller, and S. Zafeiriou, "Adieu features? end-to-end speech emotion recognition using a deep convolutional recurrent network," in *Acoustics, Speech and Signal Processing (ICASSP), 2016 IEEE International Conference on*. IEEE, 2016, pp. 5200–5204.

[3] J. Wagner, D. Schiller, A. Seiderer, and E. André, "Deep learning in paralinguistic recognition tasks: Are hand-crafted features still relevant?" *Proc. Interspeech 2018*, pp. 147–151, 2018.
[4] Z. Yang and J. Hirschberg, "Predicting arousal and valence from waveforms and spectrograms using deep neural networks," *Proc. Interspeech 2018*, pp. 3092–3096, 2018.
[5] M. Sarma, P. Ghahremani, D. Povey, N. K. Goel, K. K. Sarma, and N. Dehak, "Emotion identification from raw speech signals using dnns," *Proc. Interspeech 2018*, pp. 3097–3101, 2018.
[6] S. Mirsamadi, E. Barsoum, and C. Zhang, "Automatic speech emotion recognition using recurrent neural networks with local attention," in *Acoustics, Speech and Signal Processing (ICASSP), 2017 IEEE International Conference on*. IEEE, 2017, pp. 2227–2231.
[7] Z. Zhao, Y. Zheng, Z. Zhang, H. Wang, Y. Zhao, and C. Li, "Exploring spatio-temporal representations by integrating attention-based bidirectional-lstm-rnns and fcns for speech emotion recognition," *Proc. Interspeech 2018*, pp. 272–276, 2018.
[8] P. Li, Y. Song, I. McLoughlin, W. Guo, and L. Dai, "An attention pooling based representation learning method for speech emotion recognition," *Proc. Interspeech 2018*, pp. 3087–3091, 2018.
[9] M. Chen, X. He, J. Yang, and H. Zhang, "3-d convolutional recurrent neural networks with attention model for speech emotion recognition," *IEEE Signal Processing Letters*, vol. 25, no. 10, pp. 1440–1444, 2018.
[10] M. El Ayadi, M. S. Kamel, and F. Karray, "Survey on speech emotion recognition: Features, classification schemes, and databases," *Pattern Recognition*, vol. 44, no. 3, pp. 572–587, 2011.
[11] C.-N. Anagnostopoulos, T. Iliou, and I. Giannoukos, "Features and classifiers for emotion recognition from speech: a survey from 2000 to 2011," *Artificial Intelligence Review*, vol. 43, no. 2, pp. 155–177, 2015.
[12] Z. Zeng, M. Pantic, G. I. Roisman, and T. S. Huang, "A survey of affect recognition methods: Audio, visual, and spontaneous expressions," *IEEE transactions on pattern analysis and machine intelligence*, vol. 31, no. 1, pp. 39–58, 2009.
[13] S. G. Koolagudi and K. S. Rao, "Emotion recognition from speech: a review," *International journal of speech technology*, vol. 15, no. 2, pp. 99–117, 2012.
[14] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," *CoRR*, vol. abs/1412.6980, 2014. [Online]. Available: http://arxiv.org/abs/1412.6980
[15] S. Ruder, "An overview of gradient descent optimization algorithms," *CoRR*, vol. abs/1609.04747, 2016. [Online]. Available: http://arxiv.org/abs/1609.04747
[16] H. M. Fayek, M. Lech, and L. Cavedon, "Evaluating deep learning architectures for speech emotion recognition," *Neural Networks*, vol. 92, pp. 60–68, 2017.
[17] J. Zhao, X. Mao, and L. Chen, "Speech emotion recognition using deep 1d & 2d cnn lstm networks," *Biomedical Signal Processing and Control*, vol. 47, pp. 312–323, 2019.
[18] F. Burkhardt, A. Paeschke, M. Rolfes, W. F. Sendlmeier, and B. Weiss, "A database of german emotional speech," in *Ninth European Conference on Speech Communication and Technology*, 2005.
[19] C. Busso, M. Bulut, C.-C. Lee, A. Kazemzadeh, E. Mower, S. Kim, J. N. Chang, S. Lee, and S. S. Narayanan, "Iemocap: Interactive emotional dyadic motion capture database," *Language resources and evaluation*, vol. 42, no. 4, p. 335, 2008.
[20] S. Ioffe and C. Szegedy, "Batch normalization: Accelerating deep network training by reducing internal covariate shift," *arXiv preprint arXiv:1502.03167*, 2015.
[21] L. Deng, D. Yu *et al.*, "Deep learning: methods and applications," *Foundations and Trends® in Signal Processing*, vol. 7, no. 3–4, pp. 197–387, 2014.
[22] N. S. Keskar, D. Mudigere, J. Nocedal, M. Smelyanskiy, and P. T. P. Tang, "On large-batch training for deep learning: Generalization gap and sharp minima," *CoRR*, vol. abs/1609.04836, 2016. [Online]. Available: http://arxiv.org/abs/1609.04836
[23] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "Imagenet classification with deep convolutional neural networks," in *Advances in neural information processing systems*, 2012, pp. 1097–1105.