Emotion Identification from raw speech signals using DNNs

Mousmita Sarma[4], *Pegah Ghahremani*[1], *Daniel Povey*[1 2], *Nagendra Kumar Goel*[3],
Kandarpa Kumar Sarma[4], *Najim Dehak*[1]

[1]Center for Language and Speech Processing, Johns Hopkins University, Baltimore, MD, USA
[2]Human Language Technology Center of Excellence, Johns Hopkins University, Baltimore, MD, USA
[3]Go-Vivace Inc., McLean, VA, USA
[4] Department of Electronics and Communication Engineering, Gauhati University, Guwahati, Assam, India

(mousmita.s, kandarpaks)@gauhati.ac.in, (pghahre1, ndehak3)@jhu.edu, dpovey@gmail.com,
nagendra.goel@govivace.com

Abstract

We investigate a number of Deep Neural Network (DNN) architectures for emotion identification with the IEMOCAP database. First we compare different feature extraction frontends: we compare high-dimensional MFCC input (equivalent to filterbanks), versus frequency-domain and time-domain approaches to learning filters as part of the network. We obtain the best results with the time-domain filter-learning approach. Next we investigated different ways to aggregate information over the duration of an utterance. We tried approaches with a single label per utterance with time aggregation inside the network; and approaches where the label is repeated for each frame. Having a separate label per frame seemed to work best, and the best architecture that we tried interleaves TDNN-LSTM with time-restricted self-attention, achieving a weighted accuracy of 70.6%, versus 61.8% for the best previously published system which used 257-dimensional Fourier log-energies as input.

1. INTRODUCTION

Speech based emotion classification has been found to be gaining popularity for the development of emotionally sensitive Human Machine Interaction (HMI) systems. In the evolving setups of intelligent commercial dialogue systems and smart call centers, emotion information obtained from speech can be used as meta data to understand speaker's psychology and response. Human expresses emotional state related information through numerous subtle ways that may or may not be directly represented by common features such as mel filterbank or formant locations, pitch, voicing etc. Research has been primarily focused on deriving useful statistical feature sets from low level acoustic cues as well as on developing efficient machine learning based modeling strategy to learn the emotion dependent temporal and contextual variations of speech. Machine learning based models have been also used to derive high level features to represent the whole utterance from low level acoustic features. Recently, deep learning approaches are becoming popular for modeling emotion specific information from speech signals [1, 2, 3, 4, 5]. However, there is a recent trend in deep speech based system design which attempts to derive features of the input signal directly from raw unprocessed speech waveforms excluding the necessity of hard coded feature extraction outside the Deep Neural Network (DNN). Such approaches have shown appreciable reliability and observed state of art performance in speech recognition task [6, 7, 8]. In the domain of paralinguistic, Trigeorgis et al., 2016, used raw waveforms for speech emotion dimensional rating in a deep CNN framework [9]. Motivated by such success of raw waveforms, in this work we propose to use raw waveform front end layers to learn emotion specific cues within the network and design end to end DNN setup for categorical emotion identification task. The raw waveform front end [6] used in this work attempts to learn specific set of filters, which are jointly optimized with rest of the network and the filter bank is learned to optimize emotion identification objective.

A challenging issue in emotion identification task is effective modeling of the long temporal context. This is because emotion specific information lies on the long span of time to a great extent. We explore multiple DNN architecture to appropriately model such long term dependencies of emotion cues and provide a comparative analysis. We use temporal convolution in the form of time-delay neural network (TDNN) layers [10] and unidirectional recurrent projected Long Short Term Memory (LSTM) [11] layers individually with the raw waveform front end. We also experiment an interleaving TDNN with unidirectional LSTM (TDNN-LSTM) setup and time restricted attention mechanisms [12] which enables the DNN to be more attentive to emotionally sensitive portions of the speech. We use such time restricted attention layers with both LSTM and TDNN-LSTM setup and we observe that attention improves the accuracy significantly in both of these setups as well as helps reducing confusions among individual categories. Finally, we experiment statistics extraction layers which was previously used with the xvector setup of speaker and language identification [6, 13, 14]. We experiment all these temporal modeling setups individually with the frequency domain and time domain raw waveform front end and observe the best results with TDNN-LSTM-attention setup with time domain raw waveform front end.

All our results have been reported on the categorical emotion identification problem of Interactive Emotional Motion Capture (IEMOCAP) database [15]. We design a basline DNN setup with TDNN layers using a high resolution 23-dimensional mel frequency cepstral coefficients (MFCC). Experimental results prove the improvement obtained from the proposed raw waveform based DNN setups which learn features within the network over MFCC based DNN setup where hard coded features are used. We also experiment separately with time and frequency domain data driven filter learning approaches in the raw waveform setup. We also include a few intermediate experimental comparison regarding DNN training time and decode time dependencies on seeing more or less context. We find such dependencies plays a very critical role for emotion identification task. In our best DNN setup we observe 8.31% improvement in terms of weighted accuracy (WA) and 4.37% improvement in terms unweighted accuracy (UA) over 257-dimensional magnitude FFT vectors based DNN setup reported in [4].

The rest of the paper is organized as follows. Details of baseline MFCC based DNN and raw waveform based feature extraction front end is described in Section 2. Section 3 provides experimental details on temporal modeling using DNN, optimization and analysis of results. Conclusions are presented in Section 4.

2. Feature extraction in emotion identification

Here we describe the baseline MFCC based DNN setup and two raw waveform feature extraction front end setups for emotion identification task. Table 1 represents the results obtained from all three experiments. The neural network setup used in all these experiments are explained in details in Section 3.2.1.

Most speech systems use short term hand-crafted spectral and cepstral features based on fixed filters, such as MFCC or Mel filter-banks. In the first experiment, we use 23-dimensional MFCC features as input to DNN.

However using fixed filter may not be the most appropriate for final objective of minimizing emotion states classification error. In the next experiment in row 2 of Table 1, we use direct-from-signal setup described in [6] which attempts to learn filters within the DNN. We refer to this as time domain raw waveform front end in the rest of the description. The input frames are 40 ms long segments of raw waveform signal with 10 ms overlap. This raw waveform front end has a $1-d$ time convolution layer, which operates on 40 ms raw signal with step size 1.25 ms and the filter outputs are aggregated using two trainable Network-in-Network (NIN) nonlinearity layers introduced in [6]. We also used setup proposed in [16], where the signal first transformed into frequency domain and a trainable filter bank layer, which is modeled using linear transformation is jointly trained with rest of the network.

We observe that using direct-from-signal setup, we are able to improve the performance significantly compared to the baseline MFCC. We also observe that results of time domain raw waveform front end is better than learning feature from frequency domain. We need to do more experiments using complex domain filter learning to model phase information, which can be useful in learning emotion states. We use time domain feature extraction block in all results reported here after.

Table 1: *Effect of different feature extraction methods*

Feature extraction method	WA
MFCC	59.9
Time-domain	**65.5**
Frequency-domain	63.4

3. Experimental Details and Results

This section describes the experimental details and results. All our experiments are done using Kaldi toolkit [17]. All DNN based emotion identification setups described in this section have two common structures as shown in Fig 1 (a) and (b). The initial block contains raw waveform front end layers as described in Section 2. The temporal modeling layers are either TDNN or LSTM or a combination of the two along with attention layer. We use statistics pooling layer before softmax layer as in Fig 1(a) to get segment level emotion class output. DNN set ups where we use attention layer in the temporal modeling block, we use post processing in terms of averaging posteriors over frames outside the network to get the segment level emotion class output as in Fig 1(b).

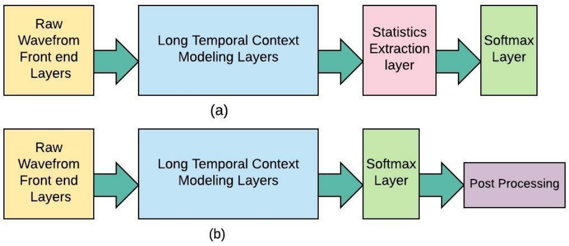
Raw Wavefrom Front end Layers
Long Temporal Context Modeling Layers
Statistics Extraction layer
Softmax Layer
(a)
Raw Wavefrom Front end Layers
Long Temporal Context Modeling Layers
Softmax Layer
Post Processing
(b)

Figure 1: *Lay out of the proposed end to end DNNs for emotion identification task*

We have used four emotional categories (neutral, angry, sad and happy) from the Interactive Emotional Dyadic Motion Capture (IEMOCAP) database [15]. The database consists of about 12 hours of audiovisual data (speech, video, facial motion capture) from five mixed gender pairs of male and female actors, at two recording scenarios: scripted and improvised speech. It is organized in five sessions, four of which are used for training and remaining one is used for testing. Each wave file has segment level emotion category label annotated by human annotators.

The performance of the emotion identification DNNs are reported using two parameters, weighted accuracy (WA) which is the overall classification accuracy and unweighted accuracy (UA) which is the average recall over the emotion categories.

3.1. Data perturbation

To increase the amount of data in the training set we perform data augmentation by means of amplitude and speed perturbation. For each speech signal 10 different amplitude modulated versions are created initially. Speed perturbation [18] is performed on the amplitude modulated signals with speed factors 0.9, 1.0 and 1.1. Effect of data perturbation on emotion identification task can be seen in Table 2.

Table 2: *Effect of data perturbation on emotion identification on our best setup without tuning decode time parameters*

Perturbation	WA	UA
No	60.27	48.84
Yes	66.07	57.215

3.2. Modeling long temporal context

Deep neural network is trained to classify different emotion states and training examples consists of variable length chunks of speech features with a single emotion state label. We use softmax layer at the end of network to give the network freedom to model any distribution over output and each emotion state is modeled as a separate output class.

One of the main issues in predicting emotion state is that the emotion cues can not be easily estimated over small span of time and we need to preserve the temporal context or use long examples to estimate emotional state of speakers. In this section, we compare different approaches to model temporal context. We use TDNN architecture which models long term temporal dependencies in models described in Section 3.2.1 through Section 3.2.3. One disadvantage with temporal modeling using TDNN is the linear increase in parameters and computation with increase in temporal context and non-uniform subsampling method helps to mitigate this issue. We also use only LSTM layers with and without attention for temporal modeling in recurrent way as described in Section 3.2.4.

Table 3: *Effect of long temporal modeling layers*

Temporal Modeling	WA	UA
TDNN-Statistics Pooling	65.5	55.3
TDNN-LSTM	59.5	56.4
TDNN-LSTM-Attention	66.3	60.3
LSTM	59.9	53.7
LSTM-Attention	63.4	56.2

3.2.1. Statistic pooling layer

We use TDNN layer as temporal convolution in this setup and the context used in TDNN layer are similar to setup in [14]. The statistic pooling layer [6, 14] used in this setup, which aggregates all available frame level inputs for intermediate layer in the network and outputs their mean and standard deviation. This layer operates on the entire segment and the mean and standard deviation are concatenated together and passed through feed forward layer and finally a softmax layer applied on them. Despite TDNN-LSTM setup, we use single emotion state label for whole example chunk in this setup and the result is reported in row 1 of Table 3. One disadvantage with this setup is that the speech and non-speech frame weights are similar in computing the mean and standard deviation in statistic pooling layer and the error is back-propagated uniformly from this layer across all time frames. This requires to use energy based SAD to filter out non-speech frames and not filtering non-speech frames results in large degradation in this setup. This issue is solved using TDNN-LSTM-attention setup (Section 3.2.3) and the non-speech frames are not removed in this setup. Having multiple labels per chunk in the set up of Section 3.2.3 results in back-propagating the error through frames for example chunk more frequently.

3.2.2. TDNN-LSTM

In this setup, we use temporal convolution in the form of TDNN layers along with LSTM layers. We use interleaving of temporal convolution with unidirectional LSTM, which reported to outperform bidirectional LSTM [19]. We use per-frame objective, where all frames has same emotion state label for each utterance. We use higher frame rate at lower layers of LSTM and TDNN layer in the network and we decrease layer frame rate with layer depth. The layer wise context of temporal modeling block is similar to $config$ 1 of Table 4 except that this set up does not have an attention layer. The results of this set up is shown in the row 2 of Table 3.

3.2.3. TDNN-LSTM with time-restricted attention

We exploit time-restricted self-attention mechanism, where the input and output sequence lengths are the same and it attends at particular frame only sees input from a limited number of frames to the left and right. Time-restricted attention layer [12] is used as the last layer along with TDNN and unidirectional LSTM layer. The architecture of TDNN-LSTM-attention set up contains interleaving TDNNs and LSTMs with an attention layer after the last LSTM layer. The layer wise context of temporal modeling block of this set up is shown in $config$ 1 of Table 4. The dimensions of projection and the recurrence are one quarter the cell dimension. We found 128 cell dimension optimal for the current emotion identification task and with the recurrence of dimension 32 and the output of LSTM of dimension 64. The LSTMs operates with a recurrence that spans 3 time steps. The attention layer used has 12 heads, a context of $[-5, 2]$, a key-dimension of 40 and value dimension of 60. In this setup we use per frame dropout using the dropout schedule method described in [20] where entire vector is forced to be zero or one. The dropout schedule is expressed as a piece wise linear function on the interval $[0, 1]$, where $f(0)$ gives the dropout proportion at the start of training and $f(1)$ gives the dropout proportion after seeing all the data. A dropout schedule of the form $0, 0@0.20, p@0.5, 0@0.75, 0$ is used in this setup, where p is 0.3 in the results reported here. Thus, the dropout probability is 0 at $f(0)$, 0 at $f(0.2)$, 0.3 at $f(0.5)$, 0 at $f(0.75)$ and 0 at $f(1)$. In this set up we average frame posteriors outside the network to get an segment level aggregate from the frame level posteriors. The performance of this set up is shown in row 3 of Table 3. We give some extra left context at the time of decoding which provides flexibility to the network regarding number of frames it sees in addition to what was provided during training and we evaluate the model several times to tune this length of decode time context. We also observe improvement by using a longer training chunk using fixed length examples during training. Details of fixed length versus variable length training example experiments are reported in Section 3.3.

Table 4: *Layer wise context of temporal modeling block for TDNN-LSTMP-attention set up*

	Config1		Config2	
Layer	Context	Layer-type	Context	Layer-type
1	[-1, 0, 1]	TDNN	[-1,0,1,2]	TDNN
2	[0]	LSTM^1	[-3,0,3,6]	TDNN
3	[-3, 0, 3]	TDNN	[0]	LSTM^1
4	[-3, 0, 3]	TDNN	[-6,0,6,12]	TDNN
5	[0]	LSTM^1	[0]	LSTM^2
6	[-3, 0, 3]	TDNN	[-12,0,12, 24]	TDNN
7	[-3, 0, 3]	TDNN	[-5,2]	Attention
8	[0]	LSTM^1	[-12,0,12]	TDNN
9	[-5, 2]	Attention		

LSTM^1: delay time=-3
LSTM^2: delay time=-6

3.2.4. LSTM with and without time restricted attention

We use three unidirectional LSTM layers in this set up with a cell dimension 128 and recurrent and non recurrent projection dimension of 32. In a prior work on language identification task [21] it was suggested we perform pooling over time recurrent layer to reduce redundancy. We experiment using a max pooling layer after the last LSTM layer and observe improvement in the accuracy. A comparison of results of LSTM with and without time pooling is shown in Table 5. We also add time restricted attention on this LSTM only set up. We observe significant improvement using time restricted attention layer as a final layer in the LSTM only setup as shown in the row 4 and 5 of Table 3. We use similar dropout schedule in this set up as described in Section 3.2.3.

Table 5: *Effect of time pooling in the LSTM setup*

Temporal Modeling	WA	UA
LSTM	54.5	48.9
LSTM with max pooling	59.9	53.7

3.3. Variable-length vs. fixed-length training

The training and test utterances used in our emotion identification setup have variable lengths in range of 0.6 to 10 seconds and we need high accuracy on short segments during test time. It is challenging to get utterance-level representation, which is normalized over different length. It is important to minimize network sensitivity to speech duration. One solution is to train the network on chunks of different durations. In this section, we investigate the effect of training using variable length chunks, where the output is generated from the entire utterance if it is shorter than 6 second and compare the results with dividing the utterance into fixed length chunks and randomize chunks and use them for training. Table 6 presents the results using models trained on fixed and variable-length examples. The model configuration used in all experiments are described in Table 4.

In the experiment in row 1, the length of example chunks are varied from 1 to 6 seconds and the entire utterance is used as example if it is shorter than 6 seconds. Size of mini-batches are a function of example length (e.g. mini-batch sizes used for examples with length 100 and 200 are 128 and 64.) and total number of frames are almost equal in different mini-batches. The network configuration, $Config2$, is used in this experiment that is described in Table 4. The use of future context information in unidirectional LSTM is accomplished using delayed prediction of the output label. We use delay time 3 and 6 for LSTM layers in our experiments. We use effective temporal context and decay time for LSTM layer and it helps to get generalized to unseen sequence length and this is equivalent to maximum number of frames that can be remembered via LSTM layer. We use decay time 100 frames in this experiments and the error is back-propagated through 100 effective frames. We use larger decay time in this experiment to remember longer frames for long example chunks. The network learns emotion states on longer chunks more easily and the frame level cross entropy objectives improved faster using variable length utterances and the model converges within 30 epochs.

In the experiments in row 2, we used fixed length chunk with 50 frames and the network is trained for same number of epochs. As expected, it is harder to learn emotion states over 0.5 seconds and the network converges slower and it needs to train for longer time. The interesting point about this setup is higher randomization. Since long chunks in variable length setup is segmented into subsegments with smaller size (e.g. 600 frames are segmented into 6 subsegments with 100 frames.) and the network use these subsegments randomly in different mini-batches during training, which results in more randomness during training and it can help for better convergence in SGD and it can be the reason for improvement in accuracy for fixed chunk length setup.

The results in row 3 and 4 are trained on fixed chunks with length 50 and 100 frames respectively and the training epochs are increased to 100. The result shows that, the network needs longer training time to learn emotion states using fixed length chunks. The results reported within parenthesis in row 3 and 4 are the weighed accuracy obtained by providing an effectively unlimited left context during decode time. It means the network is allowed to reuse hidden state activations from previously computed chunk. As can be seen from the table unlimited left context helps only with smaller training chunk. We also re-evaluate the best set up without using dropout and the result is reported within parenthesis (as $*$) of row 5 of the Table 6.

Table 6: *Effect of training example chunk length. The numbers inside parenthesis are results using looped decoding.(*training without dropout)*

chunk length	epoch	WA	UA
$100 - 600^1$	30	65	53.0
50^2	30	60.78	53.93
50^2	100	66.4 (67.2)	60.3 (58.7)
$100 - 600^2$	100	69.3 (64.43)	58.4 (53.2)
100^2	100	**70.1** (66.8) (*56.49)	**60.7**(58)
200^2	100	68.22 (65.32)	57.77 (54.9)

1: Config2 is used in this setup.
2: Config1 is used in this setup.

3.4. Summary of findings

In this work we propose to use raw speech waveform based end to end DNN for categorical emotion identification. We describe experimental results in two parts: effect of feature extraction front end and long temporal context modeling. In the raw waveform based DNN set up, we use 1-d time convolution layer and two NIN layers and we observe 4-5% improvement compared to MFCC as shown in Table 1. The second part of the experimental work is centered around long temporal context modeling. We compare five different set ups for temporal modeling namely TDNN with statistics pooling layer, TDNN-LSTM, TDNN-LSTM-attention, LSTM and LSTM-attention set ups. Here, we also investigated different ways to aggregate information over the duration of an utterance. We tried approaches with a single label per utterance on time aggregation inside the network and approaches with a label per frame. We observe TDNN-LSTM-attention which is our deepest set up with 12 layers (including both feature extraction and temporal modeling) gives best WA and UA of 66.4% and 60.3 % respectively. In this set up we use separate label per frame. We also experiment with variable and fixed length examples with this TDNN-LSTM-attention set up as described Section 3.3. The WA and UA improved further to 70.1% and 60.7% respectively when we use large chunk in fixed length example trainings. Our results outperforms previously reported results [4, 5] on the same emotion identification problem.

4. Conclusion

We describe experimental results obtained while designing end to end DNN for categorical emotion identification task. We design raw waveform front end layers which attempts to learn emotion specific features within the network to optimize emotion identification objective. We also experiment several DNN architecture for long temporal context modeling to capture emotion cues lies in the long span of speech. We observe that TDNN-LSTM-attention set up while trained with fixed length examples with longer chunk of 1 second duration outperforms all other set ups. In our future work we plan to investigate emotion identification in multi- dimensional space.

5. References

[1] J. Niu, Y. Qian and K. Yu, "Acoustic Emotion Recognition using Deep Neural Network," in *19^{th} International Symposium on Chinese Spoken Language Processing, Proceedings*, 2014.

[2] M. Neumann and N. Thang Vu, "Speech Emotion Recognition Using Deep Neural Network and Extreme Learning Machine," in *Interspeech 2014 – 15^{th} Annual Conference of the International Speech Communication Association, September 14-18, Singapore, Proceedings*, 2014.

[3] J. Lee and I. Tashev, "High-level Feature Representation using Recurrent Neural Network for Speech Emotion Recognition," in *Interspeech 2015 – 16^{th} Annual Conference of the International Speech Communication Association, September 6-10, Dresden, Germany, Proceedings*, 2015.

[4] S. Mirsamadi, E. Barsoum and C. Zhang, "Automatic Speech Emotion Recognition using Recurrent Neural Networks with Local Attention," in *ICASSP 2017 – IEEE International Conference on Acoustics, Speech and Signal Processing, April 15-20, Calgary, Alberta, Canada, Proceedings*, 2017.

[5] M. Neumann and N. Thang Vu, "Attentive Convolutional Neural Network based Speech Emotion Recognition: A Study on the Impact of Input Features, Signal Length, and Acted Speech," in *Interspeech 2017 – 18^{th} Annual Conference of the International Speech Communication Association, August 20-24, Stockholm, Sweden, Proceedings*, 2017.

[6] P. Ghahremani, V. Manohar, D. Povey and S. Khudanpur, "Acoustic modelling from the signal domain using CNNs," in *Interspeech 2016 – 17^{th} Annual Conference of the International Speech Communication Association, September 8-12, San Francisco, CA, USA, Proceedings*, 2016.

[7] D. Palaz, M. Magimai-Doss , R. Collobert , "Analysis of cnn-based speech recognition system using raw speech as input," in *Interspeech 2015 – 16^{th} Annual Conference of the International Speech Communication Association, September 6-10, Dresden, Germany, Proceedings*, 2015.

[8] N. Jaitly and G. Hinton, "Learning a better representation of speech sound waves using restricted Boltzmann machines," in *ICASSP 2011 – IEEE International Conference on Acoustics, Speech and Signal Processing, May 22-27, Prague, Czech Republic, Proceedings*, 2011.

[9] G. Trigeorgis , F. Ringeval , R. Brueckner , E. Marchi , M. A. Nicolaou , B. Schuller, S. Zafeiriou, "Adieu Features? End-to- End Speech Emotion Recognition using a Deep Convolutional Recurrent Network," in *ICASSP 2016 – IEEE International Conference on Acoustics, Speech and Signal Processing, May 20-25, Shanghai, China, Proceedings*, 2016.

[10] V. Peddinti, D. Povey and S. Khudanpur, "A time delay neural network architecture for efficient modeling of long temporal contexts," in *Interspeech 2015 – 16^{th} Annual Conference of the International Speech Communication Association, September 6-10, Dresden, Germany, Proceedings*, 2015.

[11] H. Sak, A. Senior and F. Beaufays, "Long Short-Term Memory Recurrent Neural Network Architectures for Large Scale Acoustic Modeling," in *Interspeech 2014 – 15^{th}Annual Conference of the International Speech Communication Association, September 14-18, Singapore, Proceedings*, 2014.

[12] D. Povey, H. Hadian, P. Ghahremani, Ke. Li and S. Khudanpur, "A Time- Restricted Self-attention Layer for ASR," in *ICASSP 2018 – IEEE International Conference on Acoustics, Speech and Signal Processing, April 15-20, Calgary, Alberta, Canada, Proceedings*, 2018.

[13] D. Snyder, D. Garcia-Romero, D. Povey and S. Khudanpur, "Deep Neural Network Embeddings for Text-Independent Speaker Verification," in *Interspeech 2017 – 18^{th} Annual Conference of the International Speech Communication Association, August 20-24, Stockholm, Sweden, Proceedings*, 2017.

[14] D. Snyder, D. Garcia-Romero, G. Sell, D. Povey and S. Khudanpur, "X-VECTORS: Robust DNN Embeddings for Speaker Recognition," in *ICASSP 2018 – IEEE International Conference on Acoustics, Speech and Signal Processing, April 15-20, Calgary, Alberta, Canada, Proceedings*, 2018.

[15] C. Busso, M. Bulut, C. Lee, A. Kazemzadeh, E. Mower, S. Kim, J. N. Chang, S. Lee, and S. S. Narayanan, "IEMOCAP: Interactive emotional dyadic motion capture database," *Journal of Language Resources and Evaluation*, vol. 42, no. 4, pp. 335–359, 2008.

[16] P. Ghahremani, H. Hadian. L. Hang, D. Povey, S. Khudanpur "Acoustic Modeling from Frequency-Domain Representations of Speech," *Interspeech 2018 – 19^{th} Annual Conference of the International Speech Communication Association, September 2-8, Hyderabad, India, Proceedings*, 2018.

[17] D. Povey, A. Ghoshal, G. Boulianne, L. Burget, O. Glembek , N. Goel , M. Hannemann, P. Motlcek , Y. Qian , P. Schwarz, J. Silovsky, G. Stemmer, K. Vesely, "The Kaldi Speech Recognition Toolkit," in *IEEE 2011 Workshop on Automatic Speech Recognition and Understanding, Hilton Waikoloa Village, Big Island, Hawaii, US, Proceedings*, 2011.

[18] T. Ko, V. Peddinti, D. Povey and S. Khudanpur, "Audio Augmentation for Speech Recognition," in *Interspeech 2015 – 16^{th} Annual Conference of the International Speech Communication Association, September 6-10, Dresden, Germany, Proceedings*, 2015.

[19] V. Peddinti, Y. Wang, D. Povey, S. Khudanpur, "Low latency acoustic modeling using temporal convolution and LSTMs," *IEEE Signal Processing Letters*, vol. 25, issue. 3, pp. 373–377, 2018.

[20] G. Cheng, V. Peddinti, D. Povey, V. Manohar, S. Khudanpur,Y. Yan, "An exploration of dropout with LSTMs," in *Interspeech 2017 – 18^{th} Annual Conference of the International Speech Communication Association, August 20-24, Stockholm, Sweden, Proceedings*, 2017.

[21] T. N. Trong, V. Hautamaki, K. A. Lee, "Deep Language: a comprehensive deep learning approach to end-to-end language recognition," in *Odyssey 2016 The Speaker and Language Recognition Workshop, June 21-24, Bilbao, Spain, Proceedings*, 2011.