SPEECH EMOTION RECOGNITION USING CAPSULE NETWORKS

Xixin Wu[1]*, Songxiang Liu*[1]*, Yuewen Cao*[1]*, Xu Li*[1]*, Jianwei Yu*[1]*, Dongyang Dai*[2]*, Xi Ma*[2]*,*
Shoukang Hu[1]*, Zhiyong Wu*[*1,2]*, Xunying Liu*[1]*, Helen Meng*[1,2]

[1]Department of Systems Engineering and Engineering Management,
The Chinese University of Hong Kong, China
[2]Tsinghua-CUHK Joint Research Center for Media Sciences, Technologies and Systems,
Graduate School at Shenzhen, Tsinghua University, Shenzhen, China
{wuxx, sxliu, ywcao, xuli, jwyu, skhu, zywu, xyliu, hmmeng}@se.cuhk.edu.hk,
{ddy17,max15}@mails.tsinghua.edu.cn

ABSTRACT

Speech emotion recognition (SER) is a fundamental step towards fluent human-machine interaction. One challenging problem in SER is obtaining utterance-level feature representation for classification. Recent works on SER have made significant progress by using spectrogram features and introducing neural network methods, e.g., convolutional neural networks (CNNs). However the fundamental problem of CNNs is that the spatial information in spectrograms is not captured, which are basically position and relationship information of low-level features like pitch and formant frequencies. This paper presents a novel architecture based on the capsule networks (CapsNets) for SER. The proposed system can take into account the spatial relationship of speech features in spectrograms, and provide an effective pooling method for obtaining utterance global features. We also introduce a recurrent connection to CapsNets to improve the model's time sensitivity. We compare the proposed model to previous published results based on combined CNN-long short-term memory (CNN-LSTM) models on the benchmark corpus IEMOCAP over four emotions, i.e., neutral, angry, happy and sad. Experimental results show that our model achieves better results than the baseline system on weighted accuracy (WA) (72.73% vs. 68.8%) and unweighted accuracy (UA) (59.71% vs. 59.4%), which demonstrates the effectiveness of CapsNets for SER.

***Index Terms*—** Speech Emotion Recognition, Capsule Networks, Spatial Relationship Information, Recurrent Connection, Utterance-level Features

1. INTRODUCTION

One important step towards intelligent human-machine interaction is speech emotion recognition (SER), since human emotion contains important information for machine response generation. Two of the most challenging problems in SER is the extraction of frame-based high-level feature representations and the construction of utterance-level features [1, 2]. Speech signals are considered to be approximately stationary in small frames. Some acoustic features extracted from short frames, e.g., pitch, are believed to be influenced by emotions and can provide detailed emotionally relevant local information. These frame-based features are often referred to as low-level features [3]. Based on the frame-based low-level features, neural networks are utilized to extract neural representations frame by frame, which are referred to as frame-based high-level feature representations [4]. However, the emotion recognition at utterance level requires a global feature representation, which contains both detailed local information and global characteristics related to emotion.

There have been many studies in this area. [4, 5] propose to apply neural network on low-level features, e.g., pitch, energy, to learn high-level features, i.e., the neural network outputs. Recently, SER has made great progress by introducing neural networks to extract high-level neural hidden representations directly from spectrogram features [6]. Directly applying neural networks to spectrograms overcomes the open question of how to choose effective low-level features as input [7], since the spectrograms contain rich information for emotion recognition and neural networks can learn the high-level feature representations oriented for the recognition task [6]. Convolutional neural networks (CNNs) are conventionally utilized to extract the high-level neural features [8, 10]. However, with the kernel-based convolution process, CNN is not sensitive to spatial relationship or orientation information of input features.

Based on the frame-based high-level features learned, various methods are used to construct the utterance-level features. [5] proposes to use extreme learning machine (ELM) upon utterance-level statistical features. The utterance-level features are statistics of segment-level deep neural networks (DNNs) output probabilities, where each segment is a stack of neighboring frames [4]. [6, 11] propose to introduce recurrent networks to increase model's ability to capture temporal information. However, only the final states of the recurrent layers are used for classification, which may lead to loss of detailed information for emotion classification, since all information is stored in the fixed-size final states. [10] explores pooling utterance-level features from high-level features output by CNNs with attention weights, because not all regions in the spectrogram contain information useful for emotion recognition. By pooling, it also avoids to squeeze all the information in one fixed-size vector [12]. However, pooling loses the spatial information, e.g., the position information of pitch in time and frequency axes, which is informative for emotion recognition.

Recently, capsule networks (CapsNets) are proposed to overcome the disadvantage of CNNs in capturing spatial information [13]. A capsule contains a group of neurons maintaining activity vectors. The length of the vector represents the existence probability of the activity represented by the capsule, and the orientation captures the detailed instantiation parameters of the activity,

*Corresponding author

e.g., translation and rotation information. Based on the activity vectors output by the lower layer of capsules, a routing algorithm is used to couple the similar activity vectors to activate corresponding capsules in the upper layer. The CapsNet has been applied to several tasks and demonstrated its effectiveness [14, 15, 16]. However, the CapsNets used in previous work does not consider the temporal information, which is important in emotion recognition.

In this paper, we propose a sequential capsule structure, first slicing the input into windows and applying a CapsNet to the windows iteratively, then aggregating the CapsNet outputs and applying another CapsNet to the outputs to obtain utterance-level features. In this way, we adapt the CapsNet to incorporate temporal information. To further enhance this capability, we introduce recurrent connections between capsules at neighboring timesteps to the capsule structure. Experimental results demonstrate that the proposed capsule-based structure can significantly improve the overall accuracy of SER system.

This paper is organized as follows: the capsule-based structures are introduced in Section 2; the system architecture is illustrated in Section 3; Section 4 describes the setup and results of experiments; Section 5 draws the conclusion.

2. CAPSULE NETWORKS

2.1. Basic Capsule Structure

The idea of the capsule is to maintain a group of neurons, instead of a unique neuron, to capture both the existence probability and the spatial information. For the connection between capsule layers, the routing-by-agreement algorithm is applied to learn the hierarchical relationship between the learned spatial information in neighboring layers [13]. Compared to previous neural networks, e.g., CNNs, the CapsNets replace the scalar neuron with vector neuron, and the max-pooling method with dynamic routing method.

Referring to Fig. 1, assume that the capsules in layer l and $l+1$, e.g., window capsules layer and window emo-capsules layer, are $\boldsymbol{u}$ and $\boldsymbol{v}$ respectively. The j-th capsule $\boldsymbol{v}_j$ in layer $l+1$ is obtained by compressing the total input $\boldsymbol{s_j}$ as Eq. (1). This non-linear compressing function normalizes the input vector to have norm between 0 and 1, while keeping the orientation of the vector. The vector norm represents the existence probability of the activity represented by the orientation.

$$\boldsymbol{v}_j = \frac{||\boldsymbol{s}_j||^2}{1+||\boldsymbol{s}_j||^2}\frac{\boldsymbol{s}_j}{||\boldsymbol{s}_j||} \tag{1}$$

The input $\boldsymbol{s}_j$ is calculated by summing up all prediction vectors $\hat{\boldsymbol{u}}_{j|i}$ with weights c_{ij}, where the prediction vector $\hat{\boldsymbol{u}}_{j|i}$ is produced by multiplying $\boldsymbol{u}_i$, output of the i-th capsule in layer l, with a weight matrix $\boldsymbol{W}_{ij}$ as Eq. (3).

$$\boldsymbol{s}_j = \sum_i c_{ij}\hat{\boldsymbol{u}}_{j|i} \tag{2}$$

$$\hat{\boldsymbol{u}}_{j|i} = \boldsymbol{W}_{ij}\boldsymbol{u}_i + \boldsymbol{b}_{ij} \tag{3}$$

The weight c_{ij} is determined by a softmax upon the logits d_{ij}, which represents the log probability of whether $\boldsymbol{u}_i$ should be routed to $\boldsymbol{v}_j$.

$$c_{ij} = \frac{\exp(d_{ij})}{\sum_k \exp(d_{ik})} \tag{4}$$

When the outputs of lower capsule layer are different, the coupling coefficients are determined dynamically by the dynamic routing algorithm. Initially, $d_{ij} = 0$, which means the output in layer l is routed to all possible capsules in upper layer $l+1$ equally. In each iteration, c_{ij}, $\boldsymbol{s}_j$ and $\boldsymbol{v}_j$ are updated according to Eq. (1)-(3), and the logit values are updated according to $d_{ij} \leftarrow d_{ij} + \hat{\boldsymbol{u}}_{j|i} \cdot \boldsymbol{v}_j$, where $\cdot$ represents dot production. Finally, when the predefined iteration number is reached, the $\boldsymbol{v}_j$ at the final iteration is returned as output of the j-th capsule in layer $l+1$.

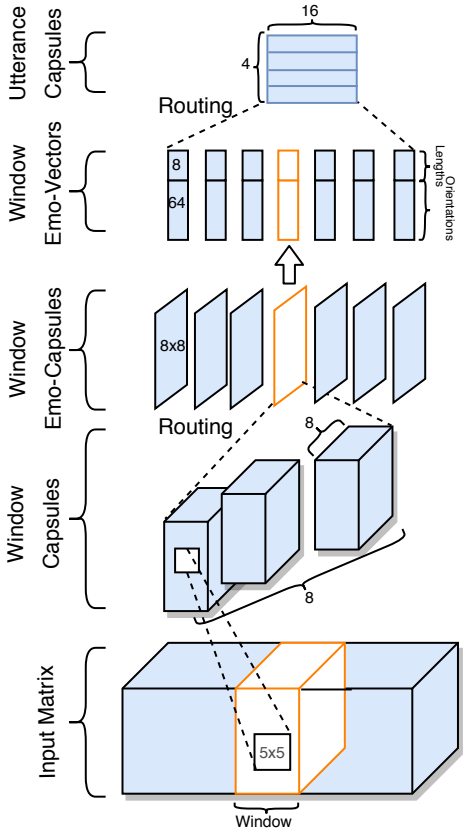

Fig. 1. Structure of SeqCaps: the input is first sliced into windows, and then CapsNets are applied to the windows. Finally the window outputs are aggregated and sent to another CapsNets to obtain utterance-level features.

2.2. Sequential Capsules (SeqCaps)

In the task of SER, the input data is a sequence of feature frames with variable length. It is impractical to feed the whole sequence to capsule layers. In order to optimize the model upon the whole sequence, we propose the structure of SeqCaps. As shown in Fig. 1, we first slice the input sequence into overlapped windows and apply capsule layers to each window, with the capsule layer weights shared across time steps. In each window, several separated convolutional layers are used on the input to obtain the capsules, denoted as window capsules. The output of the window capsules are routed to activate the window emo-capsules. The window emo-capsule outputs are converted to window emo-vectors. The window emo-vector $\boldsymbol{o}$ contains the orientations and lengths of all the N emo-capsules in one window:

$$\boldsymbol{o} = [\boldsymbol{v}_1^T, ..., \boldsymbol{v}_N^T, ||\boldsymbol{v}_1||, ..., ||\boldsymbol{v}_N||]. \tag{5}$$

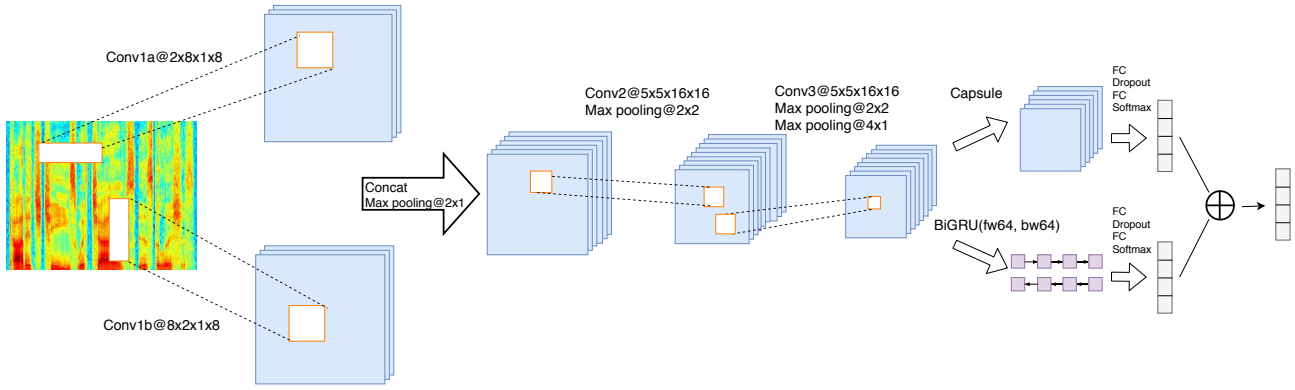

Fig. 2. Architecture of proposed system CNN_GRU-SeqCap.

All window emo-vectors are then routed to activate the utterance capsules.

2.3. Recurrent Capsules (RecCaps)

The temporal information of speech contains important cues for emotion recognition. In order to improve the model's ability of capturing temporal information, we introduce recurrent connections to the routing algorithm. We denote the output of j-th capsule in layer $l+1$ in window $t-1$ as $\boldsymbol{v}_{t-1,j}$ and output of i-th capsule in layer l in window t as $\boldsymbol{u}_{t,i}$. The prediction vector $\hat{\boldsymbol{u}}_{t,j|i}$ in window t is produced as Eq. (6):

$$\hat{\boldsymbol{u}}_{t,j|i} = \boldsymbol{W}^{u}_{ij}\boldsymbol{u}_{t,i} + \boldsymbol{W}^{o}_{ij}\boldsymbol{o}_{t-1} + \boldsymbol{b}_{ij}, \tag{6}$$

where $\boldsymbol{o}_{t-1}$ is the window-level emo-vector containing both the length and orientation information of all the N capsules in layer $l+1$ in window $t-1$:

$$\boldsymbol{o}_{t-1} = [\boldsymbol{v}^{T}_{t-1,1}, ..., \boldsymbol{v}^{T}_{t-1,N}, ||\boldsymbol{v}_{t-1,1}||, ..., ||\boldsymbol{v}_{t-1,N}||]. \tag{7}$$

Via this connection, the spatial information in the previous window can assist in determining the coupling coefficients and activating the activity which is salient in terms of time steps.

3. SYSTEM ARCHITECTURE

As demonstrated in previous work [4, 6], using spectrograms as input to CNNs can significantly improve the system accuracy. Hence, we also directly apply CNN layers to spectrograms. To capture the relationship information across frequencies and timesteps, we apply two separated CNN layers with kernel of 2×8 and 8×2. The outputs of these two separated CNN layers are concatenated together and passed through two convolutional layers and three max pooling layers as shown in Fig. 2, where the convolution filters are denoted by height$\times$width$\times$input channels$\times$output channels. These CNN layers are used to extract a representation with appropriate size which can be handled by the CapsNets.

We first explore using gated recurrent unit (GRU) architecture upon the CNN layers mentioned above in the system of CNN_GRU. The GRU layer is bidirectional with 64 cells per direction. The final state of forward GRU and the first state of backward GRU are concatenated and fed to a dense layer with 64 units activated by ReLU and dropped out with rate of 0.5. The dense layer output is then fed to a linear dense output layer with 4 units. A softmax function is applied to the final output to obtain the emotion probabilities. To compare with CNN_GRU, we build another system CNN_SeqCap, which adds SeqCaps upon the above mentioned CNN layers. The output of SeqCaps, containing four 16-dimension classes, is first reshaped into emo-vector, which contains both the class vector and the vector length, as in Eq. (7). The emo-vectors are then fed to two dense layers and softmax function with the same configuration as in CNN_GRU. The window used to slice the input matrix is set to size of 40 input steps with shift of 20 steps. The configuration of SeqCaps is shown in Fig. 1, where the connection weights between the input matrix and the window capsules are shared across input steps. The iteration number of the routing algorithm is set to 3.

We also explore the addition of RecCaps upon the above mentioned CNN layers, denoted as CNN_RecCap, to verify whether recurrent connection can help improve the performance. The architecture of CNN_RecCap is the same as CNN_SeqCap except that in CNN_RecCap the capsule components have recurrent connection in the routing from the window capsules to the window emo-capsules. To further improve the system's long-context view, we add another branch of GRU layer upon the CNN layers, parallel to the capsule branch. This system is denoted as CNN_GRU-SeqCap, as shown in Fig. 2. The outputs of the GRU layers are fed to a separated set of dense layers with the same configuration as in the capsule branch. The total loss of CNN_GRU-SeqCap is the unweighted sum of losses of the two branches. At the testing stage, the output probabilities of the capsule branch and the GRU branch are combined with the weights of λ and $1-\lambda$ respectively.

4. EXPERIMENTS AND ANALYSIS

4.1. Experimental Setup

We evaluate our proposed system on the corpus of interactive emotional dyadic motion capture (IEMOCAP) database [17], which is a common evaluation dataset. The corpus consists of five sessions, with two speakers in each session. We evaluate our systems on four emotions in the corpus, i.e., *Neutral*, *Angry*, *Happy* and *Sad*. Since the speech from scripted data may contain undesired relationship between linguistic information and the emotion labels, we only use the improvised data. Five-fold cross validation is adopted as [6]: 8 speakers from four sessions in the corpus are used as training data. One speaker from the remaining session is used as validation data, and the other one as test data.

We calculate spectrograms from the speech signal in IEMOCAP

Table 1. WA and UA of proposed and baseline systems

Model	WA(%)	UA(%)
CNN_LSTM[6]	68.80	59.40
CNN_GRU	67.02	51.84
CNN_SeqCap	69.86	56.71
CNN_RecCap	70.62	58.17
CNN_GRU-SeqCap	**72.73**	**59.71**

Table 2. Confusion matrix of CNN_SeqCap

Actual \ Predict	Neutral	Angry	Happy	Sad
Neutral	**84.22%**	5.26%	0.58%	9.94%
Angry	29.03%	**68.39%**	0%	2.58%
Happy	73.75%	18.64%	**1.69%**	5.92%
Sad	26.90%	0.29%	0.29%	**72.52%**

Table 3. Confusion matrix of CNN_RecCap

Actual \ Predict	Neutral	Angry	Happy	Sad
Neutral	**83.85%**	2.92%	3.11%	10.12%
Angry	34.06%	**59.42%**	2.17%	4.35%
Happy	76.19%	5.56%	**11.90%**	6.35%
Sad	19.29%	1.93%	1.29%	**77.49%**

and split those spectrograms into 2-s segments. Segments split from one sentence share the same emotion label. The training is conducted based on the 2-s segments. It is only during the testing stage that the whole original spectrogram is used for evaluation. The spectrograms are extracted with 40-ms Hanning window, 10-ms shift and DFT of length 1600 (for 10Hz grid resolution). The frequency range of 0-5.12KHz is used, ignoring the rest. Following aggregation of the short-time spectra, the spectrogram is finally represented by a $N \times M$ matrix, where $N \leq 200$ corresponds to the segment length and $M = 512$ according to the selected frequency grid resolution. We normalize the whole dataset to have zero mean and unit variance.

To evaluate the systems' performance, we use two common evaluation metrics:

- **Weighted Accuracy (WA)** – the accuracy of all samples in the test data.
- **Unweighted Accuracy (UA)** – the average of class accuracies in the test set.

Weight initialization is important to the convergence of CapsNets [15]. We use Xavier initializer for both CNN and capsule layers. We use a batch size of 16. Cross-entropy criterion is used as the training objective function. The Adam algorithm is adopted for optimization with parameters of $\beta_1 = 0.9$, $\beta_2 = 0.999$ and $\epsilon = 1e\text{-}8$ [18]. The learning rate in the first 3 epochs is set to 0.001. After that, the learning rate is decayed dynamically determined by the average of training losses of the last 100 training steps. The learning rate is reduced to 0.0005, 0.0002 and 0.0001 gradually, when the average training loss is reduced by a factor of 10. The models are all trained for 20 epochs and then optimized on the validation set with respect to the weighted accuracy.

4.2. Experimental Results

As shown in Table 1, both the CNN_SeqCap and the CNN_RecCap systems outperform the baseline system CNN_GRU. The CNN_GRU-SeqCap system achieves the best performance. In our experiments, we find that the CapsNets have better performance in emotions of *Neutral*, *Angry* and *Sad*, while have worse results in the category of *Happy*, as shown in Table 2. It is already reported in previous works that the category of *Happy* is difficult to be recognized because of limited training data and its special emotion characteristics, i.e., the *Happy* emotion relies on context contrastive information more than the other categories. In order to improve the CapsNets in capturing context information, in the system CNN_RecCap, we introduce recurrent connection to the routing process upon the window capsules. From the results shown in Table 3, it can been found that the *Happy* category is improved significantly from 1.69% to 11.9%. The system performance improves from 69.86% to 70.62%, and 56.70% to 58.17% on WA and UA respectively, which demonstrates the effectiveness of the recurrent connection.

CapsNets are good at capturing detailed spatial information for distinguishing emotions. However, the coupling process in the routing algorithm will tend to cluster to the emotion categories that are distributed evenly across time- and frequency-axis. In the sentences with *Happy* emotion, there may be many frames without obvious happy characteristics, thus CapsNets can not recognize the category of *Happy* as well as the other three categories. As suggested in [6, 11], RNN has strong ability to capture long context characteristics. In the system CNN_GRU-SeqCap, we combine the strengths of CapsNets and RNN by setting $\lambda = 0.6$. CNN_GRU-SeqCap achieves better performance than CNN_SeqCap and CNN_RecCap and the baseline system CNN_LSTM, as shown in Table 1. This demonstrates that CapsNet is effective, and the two kind of network, RNN and CapsNet, can be merged complementarily.

5. CONCLUSION

In this paper, we propose a novel architecture for speech emotion recognition (SER) based on capsule networks (CapsNets). The CapsNets can take into account the spatial relationship of activities in speech features, e.g., pitch and formant frequencies, which are important for emotion recognition, and provide an effective pooling method for constructing utterance-level feature representation for emotion recognition. To enable the CapsNets to consider temporal information, we propose to introduce recurrent connection to the routing algorithm between capsule layers. Experimental results demonstrate that the recurrent connection can significantly improve the CapsNets' performance. Our proposed system CNN_GRU-SeqCap outperforms the baseline system on both weighted accuracy and unweighted accuracy, which also demonstrates the effectiveness of CapsNets in SER. In the future, we plan to investigate extension of the CapsNet's ability to capture the emotions of which the characteristics are variable across time.

6. ACKNOWLEDGEMENT

This work is partially supported by National Natural Science Foundation of China-Research Grants Council of Hong Kong (NSFC-RGC) joint fund (61531166002, N_CUHK404/15).

7. REFERENCES

[1] B. Schuller, A. Batliner, S. Steidl, and D. Seppi, "Recognising realistic emotions and affect in speech: State of the art and lessons learnt from the first challenge," *Speech Communication*, vol. 53, no. 9-10, pp. 1062–1087, 2011.

[2] M. El Ayadi, M. Kamel, and F. Karray, "Survey on speech emotion recognition: Features, classification schemes, and databases," *Pattern Recognition*, vol. 44, no. 3, pp. 572–587, 2011.

[3] S. Mirsamadi, E. Barsoum, and C. Zhang, "Automatic speech emotion recognition using recurrent neural networks with local attention," in *Proc. ICASSP*. IEEE, 2017, pp. 2227–2231.

[4] J. Lee and I. Tashev, "High-level feature representation using recurrent neural network for speech emotion recognition," in *Proc. INTERSPEECH*, 2015.

[5] K. Han, D. Yu, and I. Tashev, "Speech emotion recognition using deep neural network and extreme learning machine," in *Fifteenth annual conference of the international speech communication association*, 2014.

[6] A. Satt, S. Rozenberg, and R. Hoory, "Efficient emotion recognition from speech using deep learning on spectrograms," in *Proc. INTERSPEECH*, 2017, pp. 1089–1093.

[7] M. Tahon and L. Devillers, "Towards a small set of robust acoustic features for emotion recognition: challenges," *IEEE/ACM Transactions on Audio, Speech and Language Processing (TASLP)*, vol. 24, no. 1, pp. 16–28, 2016.

[8] W. Zheng, J. Yu, and Y. Zou, "An experimental study of speech emotion recognition based on deep convolutional neural networks," in *Affective Computing and Intelligent Interaction (ACII), 2015 International Conference on*. IEEE, 2015, pp. 827–831.

[9] Y. Zhang, J. Du, Z. Wang, and J. Zhang, "Attention based fully convolutional network for speech emotion recognition," *arXiv preprint arXiv:1806.01506*, 2018.

[10] P. Li, Y. Song, I. McLoughlin, W. Guo, and L. Dai, "An attention pooling based representation learning method for speech emotion recognition," in *Proc. INTERSPEECH*, 2018, pp. 3087–3091.

[11] X. Ma, Z. Wu, J. Jia, M. Xu, H. Meng, and L. Cai, "Emotion recognition from variable-length speech segments using deep learning on spectrograms," in *Proc. INTERSPEECH*, 2018, pp. 3683–3687.

[12] J. Cho, R. Pappagari, P. Kulkarni, J. Villalba, Y. Carmiel, and N. Dehak, "Deep neural networks for emotion recognition combining audio and transcripts," in *Proc. INTERSPEECH*, 2018, pp. 247–251.

[13] S. Sabour, N. Frosst, and G. Hinton, "Dynamic routing between capsules," in *Advances in Neural Information Processing Systems*, 2017, pp. 3856–3866.

[14] W. Zhao, J. Ye, M. Yang, Z. Lei, S. Zhang, and Z. Zhao, "Investigating capsule networks with dynamic routing for text classification," in *Proc. EMNLP*, 2018.

[15] J. Bae and D. Kim, "End-to-end speech command recognition with capsule network," in *Proc. INTERSPEECH*, 2018, pp. 776–780.

[16] M. Turan and E. Erzin, "Monitoring infant's emotional cry in domestic environments using the capsule network architecture," in *Proc. INTERSPEECH*, 2018, pp. 132–136.

[17] 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, pp. 335, 2008.

[18] D. P. Kingma and J. L. Ba, "Adam: a method for stochastic optimization," in *Proc. ICLR*, 2015.