An Attention Pooling based Representation Learning Method for Speech Emotion Recognition

Pengcheng Li[1], *Yan Song*[1], *Ian McLoughlin*[2], *Wu Guo*[1], *Lirong Dai*[1]

[1]National Engineering Laboratory of Speech and Language Information Processing
University of Science and Technology of China, Hefei, China
[2]School of Computing, University of Kent, Medway, UK

pclee@mail.ustc.edu.cn, {songy, guowu, lrdai}@ustc.edu.cn, ivm@kent.ac.uk

Abstract

This paper proposes an attention pooling based representation learning method for speech emotion recognition (SER). The emotional representation is learned in an end-to-end fashion by applying a deep convolutional neural network (CNN) directly to spectrograms extracted from speech utterances. Motivated by the success of GoogleNet, two groups of filters with different shapes are designed to capture both temporal and frequency domain context information from the input spectrogram. The learned features are concatenated and fed into the subsequent convolutional layers. To learn the final emotional representation, a novel attention pooling method is further proposed. Compared with the existing pooling methods, such as max-pooling and average-pooling, the proposed attention pooling can effectively incorporate class-agnostic bottom-up, and class-specific top-down, attention maps. We conduct extensive evaluations on benchmark IEMOCAP data to assess the effectiveness of the proposed representation. Results demonstrate a recognition performance of 71.8% weighted accuracy (WA) and 68% unweighted accuracy (UA) over four emotions, which outperforms the state-of-the-art method by about 3% absolute for WA and 4% for UA.

Index Terms: speech emotion recognition, high-level feature learning, convolutional neural network, second-order pooling.

1. Introduction

Speech emotion recognition (SER) is the task of automatically identifying human emotions from the analysis of utterances. With the rapid growth of speech-based human-computer interaction applications, including intelligent service robotics, automated call centers, and remote education, SER has attracted steadily increasing interest from researchers over the past two decades.

A key to the success of SER systems is to find an effective emotional representation for speech segments. This is challenging due to the complexity of emotional expressions and the lack of large datasets. Traditional SER methods generally consist of frontend frame-based feature extraction and backend utterance representation for classification or regression [1, 2, 3]. Slaney *et.al.* applied Gaussian Mixture Models (GMM) with Mel-Frequency Cepstral Coefficients (MFCC) for SER [1]. In [2, 3], the prosodic features were extracted to train Support Vector Machine (SVM) classifiers. However, these hand-crafted features may not be optimal for characterizing emotional information in speech, which would lead to unsatisfactory performance.

Motivated by the success of deep learning techniques in various application domains, such as large scale image and speech recognition [4, 5], several Deep Neural Network (DNN) or Convolutional Neural Network (CNN) based SER methods have recently been proposed [6, 7, 8, 9, 10, 11, 12]. In [6, 7], a multi-stage procedure was applied, in which the DNN and CNN network were trained for frontend feature extraction, followed by a backend emotion recognizer such as SVM and Extreme Learning Machine (ELM). More recent works have taken advantage of end-to-end training schemes [9, 11]. For example, Trigeorgis *et. al* [9] fed raw audio into a CNN for frontend feature extraction, followed by Long Short-Term Memory (LSTM) layers for emotional representation learning. The model parameters can be jointly optimized with back-propagation algorithms. In [8, 10] a max-pooling operation was applied over time to obtain an utterance representation from salient regions. Neumann *el. al* [12] further introduced an attention mechanism after the max-pooling operation. while Mirsamadi *et. al* [13] applied weighted pooling to an RNN output.

Despite recent improvements in deep learning based SER methods, several issues still exist. Firstly, speech emotional information may be embodied in both the temporal and frequency domains. However it is still unclear how to design a suitable neural network architecture to exploit temporal and frequency information in deriving an effective speech emotional representation. In [14, 15], 2D Time-Frequency (TF) LSTM and Grid-LSTM were proposed to model the variation over time and frequency for large scale automatic speech recognition (ASR). However, complex model architectures are prone to overfitting on a small scale dataset such as IEMOCAP [16]. Furthermore, simply average-pooling or max-pooling may be insufficient to derive effective representations for complex emotional expressions that require analysis of higher order statistics. Some recent works show the benefit of introducing an attention mechanism for representation learning [12, 13, 10]. However, they generally derive salient regions from the features in a bottom-up manner.

To address these issues, we propose an attention pooling based representation learning method for SER, as shown in Fig. 1. A deep CNN is applied directly to the spectrogram extracted from speech, in which two groups of convolutional filters with different shapes are designed to capture both temporal and frequency domain context information. Results presented in Section 3.2 will show that convolution filter shapes may affect the effectiveness of emotion representation. Motivated by GoogLeNet [17], the learned features are further concatenated and fed into the following convolutional layers. For effective representation learning, a novel attention pooling method is proposed. Unlike existing attention-based SER methods, two attention maps, *i.e.* class-agnostic and class-specific, are combined for effective emotion representation. The first attention map is

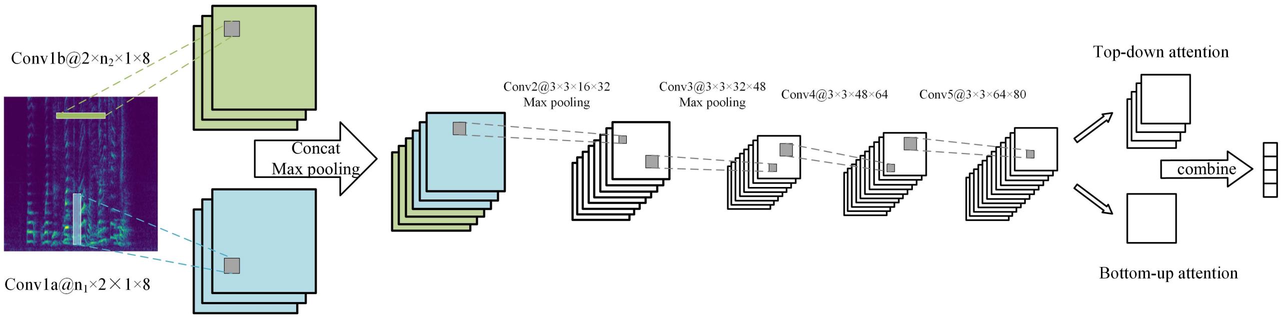

Figure 1: *The CNN architecture for attention pooling. The size of convolution filter is denoted by height* $\times$ *width* $\times$ $channel_{input}$ $\times$ $channel_{output}$. *The window size of max-pooling is* 2×2.

derived in a bottom-up manner, while the second is directly related to the emotion types. The effectiveness of the proposed representation is extensively evaluated using the IEMOCAP improvised dataset [16]. The results demonstrate an SER performance of 71.8% for WA and 68.0% for UA, both of which significantly outperform existing state-of-the-art methods.

2. Attention pooling based representation learning

In this section, we will first describe the feature extraction process, followed by a description of the CNN architecture for SER, as shown in Fig. 1. We will then introduce the two groups of convolution filters designed to capture time-specific and frequency-specific context information from the input spectrogram. Finally, the derivation of attention pooling is detailed.

2.1. Input spectrogram calculation

The CNN input is a spectrogram extracted from speech. Given an utterance, we segment it into 2 s segments for training, and use an overlap of 1 s to enable us to obtain more training data. Each segment is assigned to the same label as the corresponding utterance. For test utterances, the overlap is set to 1.6 s, and the final prediction is obtained by averaging the score for each segment.

A spectrogram is then calculated for each speech segment. A sequence of overlapping 40 ms Hamming windows is first applied to the speech signal, with a shift of 10 ms. Since it has been determined that a high frequency resolution can enhance system performance [11], a DFT of length 1600 (corresponding to 10 Hz grid resolution) is then calculated. We use a frequency range of 0 to 4 kHz for the DFT as input features. Following aggregation of the short-time spectra, the spectrogram is finally represented by a 400×200 matrix. The matrix is further processed by a normalization step in which we first linearly transform it to a range of [-1, 1], and then apply a μ-law expansion to each entry x as follows

$$F(x) = sgn(x)\frac{ln(1+\mu|x|)}{ln(1+\mu)}, -1 \leq x \leq 1 \qquad (1)$$

where $\mu = 255$. Compared with log-spectrogram post-processing, the μ-law expansion will decrease the difference between the maximum and minimum value, which in practice improves the stability of the training process.

2.2. CNN architecture

Assuming that the input feature maps have size (H, W, C_{in}), where H is the feature map height and W is the width, C_{in} is the number of channels. A standard CNN architecture will process it through multiple convolutional layers, in which a convolution layer computes the k_{th} output feature map C_{out_k} as follows:

$$C_{out_k} = b_i + \sum_{i=1}^{C_{in}} \omega(C_{out_k}, i) * input(C_{in_i}) \qquad (2)$$

where b is the bias, ω is the weight matrix, $*$ is the convolution operation. The CNN filters convolve the input feature maps, and output their learned representation. In a CNN architecture, a high-level representation can be learned through the application of multiple convolutional layers.

Once the features learned from a different time-frequency domain have been obtained, the next step for the CNN is extraction of a high-level representation for emotion recognition. Following the network design of image feature extraction [4] [18], we use 4 convolution layers with 3×3 sized filters, and use 2×2 max-pooling to down-sample the feature maps.

2.3. Time-specific and frequency-specific filters designed for input spectrogram

The role of the first convolution layer is to extract features from raw spectrograms. As convolution filters receive a rectangular part at each location, that means each output contains a specific range of time-frequency information. This rectangular time×frequency window is also called the receptive field. In SER systems, the receptive field may be important, but few works have investigated this in detail.

In order to explore the influence of receptive field and further obtain an appropriate time and frequency range for emotional representation learning, the process of Conv1 filter shape design is carried out in the following steps. Firstly the convolutional filter's receptive field on the frequency axis is set to a minimal value of 2, then the receptive field on the time axis is adjusted. This process aims to find the time span over which an emotional representation is predominantly confined, while at the same time minimizing the influence of frequency information. Then a similar process is applied to the frequency axis. Motivated by the inception module in [17], the channel concatenation is applied on the time-specific and frequency-specific feature maps, which aim to exploit both time and frequency information. In Section 3.2, experimental results from different filter shapes will be explored.

2.4. Attention pooling method

Generally, CNNs use several fully-connected (FC) layers to produce probability scores on target labels. However direct concatenating the convolutional features and feeding it into FC layers may result in over-parameterization, which makes training difficult, especially for a small-scale dataset. A pooling function that can downsample the feature maps and keep the representational ability of high-level features is important. In this work, two different pooling methods are investigated as follows;

2.4.1. *Global average pooling*

Global pooling uses a pooling window that covers the entire feature map. For feature map X with a size of $(HW) \times C$, where HW represents the reshaped 2-dimensional feature map and C is the channel size, the k_{th} prediction score can be computed using

$$score(k) = 1^T X W_k \tag{3}$$

where W_k is the $C \times 1$ class-specific fully-connected weight matrix, 1 is a vector of all ones and $1^T X$ sums the input feature. A frequently used method in CNN is Global Average Pooling (GAP), which averages the sum-pooled feature and outputs a $1 \times 1 \times C$ feature vector. GAP has been proven to be efficient in many computer vision tasks, such as in [4] [18].

2.4.2. *Attention pooling*

Simple global pooling methods can efficiently downsample the feature maps to 1-dimensional vectors. But these methods treat all input data equally, which implies that they don't consider regional saliency. Within spectrograms, many regions may have little relation to emotional information. Thus a better pooling method is needed to improve SER performance. Motivated by attention pooling proposed by Girdhar *et. al* [19], we introduce an attention pooling layer after Conv5.

Attention pooling is based on second-order pooling [20, 21], which has been proven useful for fine-grained classification tasks. This second-order pooling is implemented by first constructing the feature $X^T X$, which has a size of $C \times C$. Then using a $C \times C$ fully-connected weight matrix, and replacing the inner product by the trace. The k_{th} class prediction can be written as:

$$score(k) = Tr(X^T X W_k^T) \tag{4}$$

In practice, W_k is high-dimensional, leading to over-fitting problems when the dataset size is inadequate. The experiment reported in Section 3.3 will demonstrate this point. Therefore a low-rank estimation is needed, based upon which the attention pooling is proposed. Firstly the weight matrix W_k is decomposed into the product of two $C \times 1$ vectors:

$$score(k) = Tr(X^T X b_k a_k^T) \tag{5}$$

Using the cyclic property of the trace operation, where $Tr(ABC) = Tr(CAB)$, and noting that the trace of a scalar is itself, Equation (5) can be rewritten:

$$\begin{aligned} score(k) &= Tr(a_k^T X^T X b_k) \qquad (6)\\ &= a_k^T X^T X b_k \\ &= (X a_k)^T (X b_k) \end{aligned}$$

Thus the prediction score on the k_{th} class can be seen as the combination of two attention map Xa_k and Xb_k. Instead of making them both class-specific, Xb_k can be set to class-agnostic, so that $b_k = b$. Then the final attention model can be obtained, by combining the class-specific top-down attention Xa_k and class-agnostic bottom-up attention Xb:

$$score(k) = (Xa_k)^T (Xb) \tag{7}$$

The combination of top-down and bottom-up maps are motivated by biological vision [22], which indicates it can also be simply implemented by a element-wise multiplication and global pooling:

$$score(k) = 1^T (Xa_k \circ Xb) \tag{8}$$

3. Experiments and Analysis

3.1. Experiment setting

We use the Interactive Emotional Dyadic Motion Capture database (IEMOCAP) [16] for all experiments. IEMOCAP contains approximately 12 hours of audiovisual data performed by 10 skilled actors. The entire database is divided into 5 sections, each containing one male and one female actor. According to the recording scenarios, the data can be further divided into an improvised speech section, and a scripted speech section. Each utterance in the dataset is annotated by multiple annotators into 8 emotion labels. Following previous works [10, 11], we choose 4 emotion types for our experiments (namely neutral, happy, angry and sad) from the improvised speech for study, since scripted data may contain undesired contextual information.

Adopting the methodology of previous works [10, 11, 23, 24], we perform a 10-fold cross-validation using a leave-one-out strategy. In each training process, 9 speakers are used as training data and the remained one is used for testing data. For CNN training we make use of the PyTorch [25] deep learning framework. The optimization method is standard Stochastic Gradient Descent (SGD) with a mini-batch size of 64. We use a Nesterov momentum of 0.9 and a weight decay of 0.0001. The CNNs are trained over 50 epochs. The initial learning rate is 0.05, reducing by a factor of 10 at the 21, 31 and 41 epochs. We adopt a batch normalization [26] layer after each convolution layer and the activation function used is the Rectified Linear Unit (ReLU). The objective function for optimization uses the cross-entropy (CE) criterion. The SER performance is evaluated using the following metrics:
Weighted Accuracy (WA), which is the classification accuracy of all utterances.
Unweighted Accuracy (UA), which averages the accuracy of each individual emotion class.

3.2. Experimental with time-specific and frequency-specific filters

In order to find optimal filter shapes while excluding other factors, the first convolution layer is tested independently. We directly applied GAP to its feature map and used the fully-connected layer (16:4) to produce prediction scores. Dozens of configurations were investigated, including heights in the range 2. . . 100 (corresponding to 20. . . 1000 Hz) and widths in the range 2. . . 80 (corresponding to 20. . . 800 ms). These results are shown in Fig. 2 and reveal the following: (1) When increasing height (frequency), WA and the accuracy of the neutral class also increases. However the trend flattens at a height of around 10 (100 Hz) beyond which further increases in receptive field frequency range offer no significant improvement. (2) When increasing the width (time), WA firstly increases to a peak at a

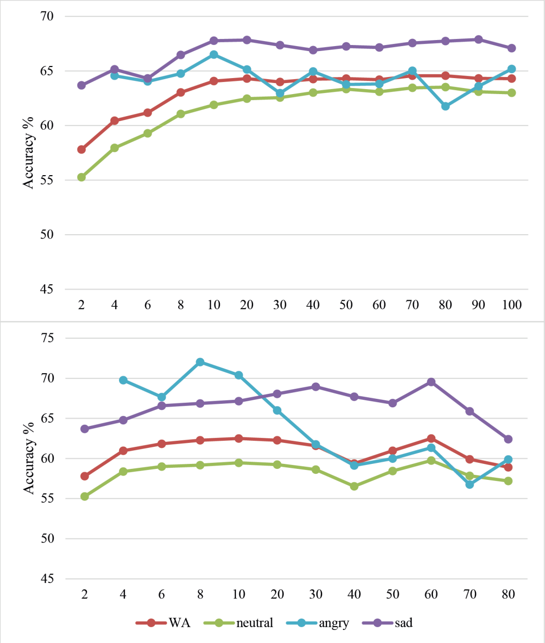

Figure 2: *WA and accuracies of neutral, happy and sad on different filter shapes. Top: fix filter width at 2 (20 ms) and adjust height between 2 and 100 (20 and 1000 Hz), bottom: fix height at 2 (20 Hz) and adjust width between 2 and 80 (20 and 800 ms)*

width of around 8 (80 ms), then gradually decreases. Interestingly, the accuracy of the angry class decreases rapidly when the receptive field time span becomes large, which indicates that angry emotions are expressed through short-time representations. (3) In these experiments, the happy class achieved the worst accuracy and doesn't relate to filter shape, hence is excluded from the figure for clarity. This may indicate that the happy class is better learned from a higher-level representation.

Based on the WA results, we separately choose the best filter shapes of 10×2 and 2×8, and concatenate them as separate channels in Conv1. There are two aspects to this channel concatenation: Firstly, the following layers are able to receive a large region in both time and temporary domain, offering more information for high-level representation learning. Secondly, compared with using large filters in both dimensions, this significantly reduces parameters and hence reduces the chance of over-fitting (which is especially problematic in small-scale tasks such as this).

3.3. Experimental with different pooling methods

To evaluate the attention pooling method, we report results from three different pooling methods, including

CNN_TF_GAP, the baseline CNN architecture. Following the observation from Section 3.2, we use 8-channel 2×8 filters and 8-channel 10×2 filters. These are then concatenated by channel in Conv1.

CNN_TF_Bilinear replaces the GAP with second-order pooling, using the original implementation in [20]. The feature map produced by Conv5 is multiplied by its transposition to generate 6400-dimensional features, followed by $l2$-normalization and signed square-root. Finally, a fully-connected layer (6400:4) is employed to obtain the final prediction score.

CNN_TF_Att.pooling where the GAP is replaced by attention pooling. For implementation, we use a 1×1 convolution layer after Conv5 to generate a top-down attention map (with size $H \times W \times 4$, corresponding to 4 emotion labels), and use another 1×1 convolution layer to generate bottom-up attention maps (with size $H \times W \times 1$). A softmax operation is then applied to the bottom-up attention map. Finally, the two types of attention map are element-wise multiplied and spatially averaged to obtain prediction scores for all 4 emotion classes.

Table 1: *UA and WA of proposed networks and previous works (in %), obtained by 10-fold cross validation*

Model	WA	UA
DNN-ELM [6][24]	57.91	52.13
RNN-ELM [24]	62.85	63.89
CNN-LSTM [11]	68.80	59.40
CNN-LSTM (two-step predictor)	67.30	62.00
CNN_TF_GAP	71.35	67.54
CNN_TF_Bilinear	69.17	64.16
CNN_TF_Att.pooling	**71.75**	**68.06**

These architectures are run through the strategy mentioned in Section 3.1. Then UA and WA are computed and summarized in Table 1. The performances of previous works [6, 11, 24] which have the same configurations, are also listed for comparison. The experimental results show that our baseline network has already outperformed state-of-the-art results and achieved 71.35% (WA) and 67.54% (UA), which proves the effectiveness of our network design. *CNN_TF_Bilinear* performs slightly worse than the *CNN_TF_GAP* baseline. This is due to over-fitting, caused by the large number of parameters in the fully-connected layer. The performance of attentional pooling, named as *CNN_TF_Att.pooling*, is shown in the last row of Table 1. It has the highest WA and UA among all configurations. In our work filters were selected by their WA score. We believe that more careful selection, for example considering performance on specific emotions, can further improve the accuracy.

4. Conclusions

In this paper, an attention pooling based CNN has been proposed for the SER task. Specifically, we firstly demonstrated that filter shape plays an important role in emotional representation learning through a series of experiments. From this, different shape filters are defined for the first layer to capture time and frequency representations. These representations are concatenated and fed into high-level representation learning. An attention pooling layer is further introduced on the learned representation to model both time and frequency saliency. The experiments demonstrate that the proposed CNN has excellent WA and UA, which separately achieve 71.75% and 68.06%, an absolute increase of about 3% and 4% respectively over previous works. These results demonstrate the strong emotional representation ability of the proposed CNN architecture.

5. Acknowledgements

This work is supported by National Natural Science Foundation of China grant No.U1613211.

6. References

[1] M. Slaney and G. McRoberts, “Baby ears: a recognition system for affective vocalizations,” in *Acoustics, Speech and Signal Processing, 1998. Proceedings of the 1998 IEEE International Conference on*, vol. 2, May 1998, pp. 985–988 vol.2.

[2] F. Yu, E. Chang, Y.-Q. Xu, and H.-Y. Shum, “Emotion detection from speech to enrich multimedia content,” *Advances in multimedia information processing*, pp. 550–557, 2001.

[3] I. Luengo, E. Navas, I. Hernáez, and J. Sánchez, “Automatic emotion recognition using prosodic parameters,” in *Proc. of INTERSPEECH*. Citeseer, 2005.

[4] A. Krizhevsky, I. Sutskever, and G. E. Hinton, “Imagenet classification with deep convolutional neural networks,” in *International Conference on Neural Information Processing Systems*, 2012, pp. 1097–1105.

[5] L. Deng and D. Yu, “Deep learning: Methods and applications,” *Foundations & Trends in Signal Processing*, vol. 7, no. 3, pp. 197–387, 2014.

[6] K. Han, D. Yu, and I. Tashev, “Speech Emotion Recognition using Deep Neural Network and Extreme Learning Machine,” in *Proceedings of Interspeech*, no. September, 2014, pp. 223–227.

[7] Q. Mao, M. Dong, Z. Huang, and Y. Zhan, “Learning salient features for speech emotion recognition using convolutional neural networks,” *IEEE Transactions on Multimedia*, vol. 16, no. 8, pp. 2203–2213, 2014.

[8] D. Bertero and P. Fung, “A First Look into A Convolutional Neural Network for Speech Emotion Detection,” in *IEEE International Conference on Acoustics, Speech and Signal Processing(ICASSP)*, 2017, pp. 5115–5119.

[9] 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 *2016 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2016, pp. 5200–5204.

[10] Z. Aldeneh and E. M. Provost, “Using Regional Saliency for Speech Emotion Recognition,” in *ICASSP, IEEE International Conference on Acoustics, Speech and Signal Processing - Proceedings*, no. 3, 2017, pp. 2741–2745.

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

[12] M. Neumann and N. T. Vu, “Attentive convolutional neural network based speech emotion recognition: A study on the impact of input features, signal length, and acted speech,” in *Proc. of INTERSPEECH*, 2017, pp. 1263–1267.

[13] 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.

[14] J. Li, A. Mohamed, G. Zweig, and Y. Gong, “Exploring multidimensional lstms for large vocabulary ASR,” in *IEEE International Conference on Acoustics, Speech and Signal Processing*, 2016, pp. 4940–4944.

[15] B. Li, T. N. Sainath, A. Narayanan, J. Caroselli, M. Bacchiani, A. Misra, I. Shafran, H. Sak, G. Pundak, and K. Chin, “Acoustic modeling for Google home,” in *INTERSPEECH*, 2017, pp. 399–403.

[16] 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.

[17] C. Szegedy, W. Liu, Y. Jia, P. Sermanet, S. Reed, D. Anguelov, D. Erhan, V. Vanhoucke, and A. Rabinovich, “Going deeper with convolutions,” in *2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2015, pp. 1–9.

[18] K. He, X. Zhang, S. Ren, and J. Sun, “Deep residual learning for image recognition,” in *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2016, pp. 770–778.

[19] R. Girdhar and D. Ramanan, “Attentional pooling for action recognition,” *ArXiv e-prints*, Nov. 2017. [Online]. Available: http://arxiv.org/abs/1711.01467

[20] T.-Y. Lin, A. RoyChowdhury, and S. Maji, “Bilinear CNN models for fine-grained visual recognition,” in *International Conference on Computer Vision (ICCV)*, 2015.

[21] ——, “Bilinear CNNs for fine-grained visual recognition,” in *Transactions of Pattern Analysis and Machine Intelligence (PAMI)*, 2017.

[22] V. Navalpakkam and L. Itti, “An integrated model of top-down and bottom-up attention for optimizing detection speed,” in *2006 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR’06)*, vol. 2, 2006, pp. 2049–2056.

[23] R. Xia and Y. Liu, “DBN-ivector Framework for Acoustic Emotion Recognition,” in *Proceedings of Interspeech*, 2016, pp. 480–484.

[24] J. Lee and I. Tashev, “High-level Feature Representation using Recurrent Neural Network for Speech Emotion Recognition,” in *Interspeech*, 2015, pp. 1537–1540.

[25] A. Paszke, S. Gross, S. Chintala, G. Chanan, E. Yang, Z. DeVito, Z. Lin, A. Desmaison, L. Antiga, and A. Lerer, “Automatic differentiation in pytorch,” in *NIPS Workshop*, 2017.

[26] S. Ioffe and C. Szegedy, “Batch normalization: Accelerating deep network training by reducing internal covariate shift,” pp. 448–456, 2015.