Interspeech 2018
2-6 September 2018, Hyderabad

Stochastic Shake-Shake Regularization for Affective Learning from Speech

Che-Wei Huang, Shrikanth Narayanan

Signal Analysis and Interpretation Laboratory (SAIL), University of Southern California
cheweihu@usc.edu, shri@sipi.usc.edu

Abstract

We propose stochastic Shake-Shake regularization based on multi-branch residual architectures to mitigate over-fitting in affective learning from speech. Inspired by recent Shake-Shake [1] and ShakeDrop [2] regularization techniques, we introduce negative scaling into the Shake-Shake regularization algorithm while still maintain a consistent stochastic convex combination of branches to encourage diversity among branches whether they are scaled by positive or negative coefficients. In addition, we also employ the idea of stochastic depth to randomly relax the shaking mechanism during training as a method to control the strength of regularization. Through experiments on speech emotion recognition with various levels of regularization strength, we discover that the shaking mechanism alone contributes much more to constraining the optimization of network parameters than to boosting the generalization power. However, stochastically relaxing the shaking regularization serves to conveniently strike a balance between them. With a flexible configuration of hybrid layers, promising experimental results demonstrate a higher unweighted accuracy and a smaller gap between training and validation, i.e. reduced over-fitting, and shed light on the future direction for pattern recognition tasks with low resource.

Index Terms: Stochastic Shake-Shake Regularization, Stochastic Depth, Adversarial Training, Affective Computing, Speech Emotion Recognition

1. Introduction

Deep convolutional neural networks have been successfully applied to several pattern recognition tasks such as image recognition [3], machine translation [4] and speech recognition [5]. Currently, to successfully train a deep neural network, one needs either a sufficient number of training samples to implicitly regularize the learning process, or employ techniques like weight decay and dropout [6] and its variants to explicitly keep the model from over-fitting.

In the recent years, one of the most popular and successful architectures is the residual neural network (ResNet) [3]. The ResNet architecture was designed based on a key assumption that it is more efficient to optimize the residual term than the original task mapping. Since then, a great deal of effort in machine learning and computer vision has been dedicated to studying the multi-branch architecture.

Deep convolutional neural networks have also gained much attention in the community of affective computing [7] and human behavioral signal processing [8] mainly because of its outstanding ability to formulate discriminative features for the top-layer classifier [9, 10]. Usually the number of parameters in a model is far more than the number of training samples and thus it requires either effective data augmentation [10] or heavy regularization to train deep neural networks for affective computing. However, since the introduction of batch normalization [11], the gains obtained by using dropout for regularization have decreased [11, 12, 13]. Yet, architectures which consist of multiple branches, for example the ResNeXt architecture [14], or stochastic depth [13] have emerged as a promising alternative.

Regularization techniques based on multi-branch architectures such as Shake-Shake regularization [1] have delivered an impressive performance on standard image datasets including the CIFAR-10 [15]. In a clever way, Shake-Shake regularization utilizes multiple branches to learn different aspects of the relevant information and then a summation in the end follows to enforce information alignment among branches, in which it has been shown that specializing each branch to learn a certain amount of complementary information helps to alleviate over-fitting. Training a very deep neural network often suffers from a number of issues, including loss in the information flow in forward propagation and gradient vanishing. To address these issues, training deep neural networks with stochastic depth [13] reduces the network depth in expectation and thus facilitates efficient training while exploits expressiveness by the full depth at testing time. Recently, ShakeDrop regularization [2] applied stochastic scaling, even with negative coefficients, to forward and backward propagations in training a deep Pyramidal ResNet [16]. Stochastic depth with a linear decay rule was employed in ShakeDrop regularization to stabilize learning because of sensitivity to the survival probability.

In this work, we propose stochastic Shake-Shake regularized ResNeXt, complementary to the ShakeDrop on a deep Pyramidal ResNet [2]. The convex combination is key to the very success of Shake-Shake regularization for it forces branches to learn independent information [1]. To preserve this feature, we extend Shake-Shake regularization to incorporate negative scaling while still maintain a consistent stochastic convex combination of its branches. Second, similar to [13, 2], we employ stochastic relaxation as a straightforward method to control the strength of shaking when regularizing training in practice. Third, we demonstrate the applicability of stochastic Shake-Shake regularization on a much shallower network, compared to the deep Pyramidal ResNet of 272 layers in [2], for speech emotion recognition. The promising experimental results highlight the ability of the proposed model to nicely address the data sparsity issue in deep learning for affective computing.

In addition to the novelty aforementioned, the major contribution in this work lies in the identification of the shaking mechanism with its ability to constrain the learning process. We show via experiments on a shallow architecture that without relaxation, the shaking mechanism leads to significant over-fitting reduction but contributes little to the improvement of the generalization performance. On the other hand, we find the survival probability in stochastic relaxation convenient to trade off between over-fitting reduction and generalization power improvement. With a flexible configuration of hybrid layers, we present a model that is close to the sweet spot to strike a balance.

2. Related Work

2.1. Shake-Shake Regularization

Shake-Shake regularization [1] is a recently proposed technique to regularize training of deep convolutional neural networks for

10.21437/Interspeech.2018-1327

image recognition tasks. This regularization technique based on multi-branch architectures promotes stochastic mixtures of forward and backward propagations from network branches in order to create a flow of model-based adversarial learning samples/gradients during the training phase. Owing to it excellent ability to combat over-fitting even in the presence of batch normalization, the Shake-Shake regularized 3-branch residual neural network [1] has achieved one of current state-of-the-art performances on the CIFAR-10 image dataset.

An overview of a 3-branch shake-shake regularized ResNeXt is depicted in Fig. 1. In addition to the short-cut flow (in light gray), there are other two residual branches $\mathbf{B}(x)$, each of them consisting of a sequence of layers stacked in order: Batch Normalization, ReLU, Conv$H \times W$, Batch Normalization, ReLU, Conv$H \times W$, where Conv$H \times W$ represents a convolutional layer with filters of size $H \times W$ and ReLU is the rectified linear unit $\mathrm{ReLU(x)} = \max(0, \mathrm{x})$.

Shake-Shake regularization adds to the aggregate of the output of each branch an additional layer, called the Shaking layer, to randomly generate adversarial flows in the following way:

$$\text{ResBlock}^N(\mathbf{X}) = \mathbf{X} + \sum_{n=1}^{N} \text{Shaking}\left(\{\mathbf{B}_n(\mathbf{X})\}_{n=1}^{N}\right)$$

where in the forward propagation for $\mathbf{a} = [\alpha_1, \cdots, \alpha_N]$ sampled from the $(N-1)$-simplex (Fig. 1 (a))

$$\text{ResBlock}^N(\mathbf{X}) = \mathbf{X} + \sum_{n=1}^{N} \alpha_n \mathbf{B}_n(\mathbf{X}),$$

while in the backward propagation for $\mathbf{b} = [\beta_1, \cdots, \beta_N]$ sampled from the $(N-1)$-simplex and $\mathbf{g}$ the gradient from the top layer, the gradient entering into $\mathbf{B}_n(x)$ is $\beta_n \mathbf{g}$ (Fig. 1 (b)). At testing time, the expected model is then evaluated for inference by taking the expectation of the random sources in the architecture (Fig. 1 (c)).

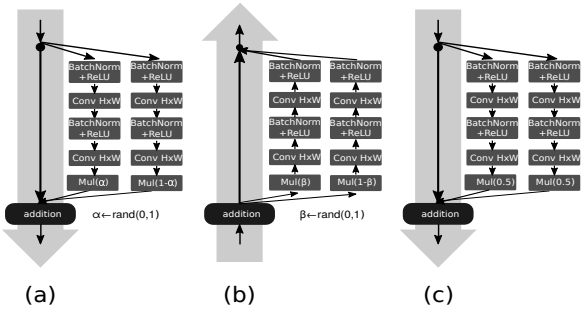

Figure 1: *An overview of a 3-branch shake-shake regularized residual block. (a) Forward propagation during the training phase (b) Backward propagation during the training phase (c) Testing phase. The coefficients α and β are sampled from the uniform distribution over $[0, 1]$ to scale down the forward and backward flows during the training phase.*

2.2. ShakeDrop Regularization

Deep Pyramidal ResNet [16] was designed to uniformly distribute the burden concentrated at the down-sampling location in the ResNet [3] by gradually increasing the number of channels in each layer along the depth of the network. A deep Pyramidal ResNet that consists of 272 layers has demonstrated further improvement over the vanilla ResNet for image recognition. Other than the number of channels, the l-th residual block (ResBlock$_l$) in a Pyramidal ResNet is formulated in the same way as a vanilla ResNet:

$$\text{ResBlock}_l(\mathbf{X}_l) = \mathbf{X}_l + \mathbf{B}_l(\mathbf{X}_l). \tag{1}$$

In addition, training a very deep, up to 1200 layers, ResNet with stochastic depth [13] by randomly skipping a ResBlock also brings improvement on the CIFAR-10 dataset:

$$\text{ResBlock}_l(\mathbf{X}_l) = \mathbf{X}_l + b_l \mathbf{B}_l(\mathbf{X}_l), \tag{2}$$

where b_l is a Bernoulli random variable with $P(b_l = 1) = p_l$ and $P(b_l = 0) = 1 - p_l$. A linear decay rule for p_l is suggested over the uniform rule

$$p_l = 1 - \frac{l}{L}(1 - p_L), \tag{3}$$

where L is the number of layers and p_L is a fixed hyper-parameter for the L-th layer.

Recently, ShakeDrop regularization [2] proposed to apply scaling on the forward and backward propagations in training a deep Pyramidal ResNet:

$$\text{ResBlock}_l(\mathbf{X}_l) = \mathbf{X}_l + \alpha_l \mathbf{B}_l(\mathbf{X}_l), \tag{4}$$

where α_l is an uniform random variable for scaling the forward propagation. Unfortunately, Eq. (4) leads to instability during training. The authors used stochastic depth to overcome this instability and formulated it as

$$\begin{aligned} \text{ResBlock}_l(\mathbf{X}_l) &= \mathbf{X}_l + (b_l + \alpha_l - b_l\alpha_l)\mathbf{B}_l(\mathbf{X}_l) \quad (5) \\ &= \begin{cases} \mathbf{X}_l + \mathbf{B}_l(\mathbf{X}_l) & \text{if } b_l = 1. \\ \mathbf{X}_l + \alpha_l \mathbf{B}_l(\mathbf{X}_l) & \text{if } b_l = 0. \end{cases} \end{aligned}$$

Moreover, the ShakeDrop regularized Pyramidal ResNet achieved a significant improvement on image recognition when the forward scaling coefficients α_l was sampled from a uniform distribution over $[-1, 1]$ instead of $[0, 1]$.

However, since the Pyramidal ResNet model and stochastic depth are specifically related to very deep architectures, it remains unknown whether the gain by the combination of stochastic depth and negative scaling can translate to shallow networks for affective computing from speech.

3. Stochastic Shake-Shake Regularization

Since Shake-Shake regularization has demonstrated nice properties such as actively encouraging model-based diversity among branches, it would be worthwhile to keep the convex combination of branches when adding negative scaling to Shake-Shake regularization. In forward propagation, we formulate it by

$$\begin{aligned} \text{ResBlock}_l^N(\mathbf{X}_l) &= \mathbf{X}_l + s_l \sum_{n=1}^{N} \text{Shaking}\left(\{\mathbf{B}_{nl}(\mathbf{X}_l)\}_{n=1}^{N}\right) \\ &= \mathbf{X}_l + s_l \sum_{n=1}^{N} \alpha_{nl} \mathbf{B}_{nl}(\mathbf{X}_l) \end{aligned}$$

where s_l is a Bernoulli random variable with $P(s_l = 1) = P(s_l = -1) = 0.5$ and $\mathbf{a}_l = [\alpha_{1l}, \cdots, \alpha_{Nl}]$ is sampled from the $(N-1)$-simplex. Furthermore, we employ stochastic depth to bring more complexity into the architecture and it becomes

$$\text{ResBlock}_l^N(\mathbf{X}_l) = \mathbf{X}_l + s_l \sum_{n=1}^{N} (b_l + \alpha_{nl} - b_l\alpha_{nl})\mathbf{B}_{nl}(\mathbf{X}_l),$$

where b_l is a Bernoulli random variable with $P(b_l = 1) = p_l$.

For a 3-branch stochastic Shake-Shake regularized ResNeXt, it could be further simplified as

$$\text{ResBlock}_l^2(\mathbf{X}_l) = \mathbf{X}_l + c_{1l}\mathbf{B}_{1l}(\mathbf{X}_l) + c_{2l}\mathbf{B}_{2l}(\mathbf{X}_l), \quad (6)$$

where

$$\begin{aligned} c_{1l} &= b_l + \alpha_{1l} - b_l\alpha_{1l}, \\ c_{2l} &= b_l + \text{sgn}(\alpha_{1l}) - \alpha_{1l} - b_l(\text{sgn}(\alpha_{1l}) - \alpha_{1l}), \end{aligned} \quad (7)$$

and $\text{sgn}(x)$ returns the sign of x.

Here we deliberately maintain a consistent stochastic convex combination of residual branches by applying a random sign to the sampled vector $\mathbf{a}$ and keep the nice property inherited from Shake-Shake regularization. What we have done is to decompose the α_l in ShakeDrop regularization into its sign and absolute value and represent them respectively by new random variables s_l and $\{\alpha_{nl}\}$ for stochastic Shake-Shake regularization. The scaling coefficients β_{nl} for backward propagations can also be extended in a similar approach to incorporate negative value.

To demonstrate the effectiveness, we propose to benchmark the following models.

1. **Baseline**: A 3-branch ResNeXt that consists of 5 convolutional layers with

   $$p_l = 1.0$$

   for every Shaking layer.

2. **Shake-Shake regularized 3-branch ResNeXt** with

   $$p_l = 0.0, \quad s_l = 1^1$$

   for every Shaking layer.

3. **Stochastic Shake-Shake regularized 3-branch ResNeXt** with

   $$p_l = 1 - \frac{l}{L}(1 - p_L)$$

   for every Shaking layer, where $p_L = 0.50$.

4. **Hybrid stochastic Shake-Shake regularized 3-branch ResNeXt**

   (a) with $p_1 = 0.50, \quad p_2 = 0.0, \quad s_2 = 1$

   (b) with $p_1 = 0.75, \quad p_2 = 0.0, \quad s_2 = 1$

For all of the proposed models, we constrain β_{nl} to be sampled uniformly from $[0, 1]$ for simplicity.

The first three models have been covered in the previous two sections; however, during experiments we found that hybrid models with a mixture of different shaking mechanisms provide more flexibility and hence we present the last two models to showcase the advantage and compare them with the first three models. In particular, Model 4 has stochastic Shake-Shake regularization at the output of the first ResBlock, where Model $4(a)$ has $p_1 = 0.5$ and Model $4(b)$ has $p_1 = 0.75$, and regular Shake-Shake regularization, in particular with $s_2 = 1$, at the output of the second ResBlock.

[1] Although we write $s_l = 1$, what we actually control during experiments is the range for α_{nl}.

4. Experiments

4.1. Datasets

We use six publicly available emotion corpora to demonstrate the effectiveness of the proposed models, including the eNTERFACE'05 Audio-Visual Emotion Database [17], the Ryerson Audio-Visual Database of Emotional Speech and Song (RAVDESS) [18], the Interactive Emotional Dyadic Motion Capture (IEMOCAP) database [19], the Berlin Database of Emotional Speech (Emo-DB) [20], the EMOVO Corpus [21] and the Surrey Audio-Visual Expressed Emotion (SAVEE) [22]. Some of these corpora are multi-modal in which speech, facial expression and text all convey a certain degree of affective information. However, in this paper we solely focus on the acoustic modality for experiments.

Corpus	No. Actors	No. Utterances			
		joy	anger	sadness	fear
eNTERFACE	42	207	210	210	210
RAVDESS	24	376	376	376	376
IEMOCAP	10	720	1355	1478	0
Emo-DB	10	71	127	66	69
EMOVO	6	84	84	84	84
SAVEE	4	60	60	60	60
Total	96	1518	2212	2274	799

Table 1: *An overview of these selected corpora, including the number of actors and the distribution of utterances in the emotional classes.*

We formulate the experimental task into a sequence classification of 4 classes, including joy, anger, sadness and fear. We perform sub-utterance sampling [23] by dividing long utterances into several short segments of 6.4-second long with the same label to limit the length of the longest utterance; for example, a 10-second long angry utterance is replaced by two angry segments corresponding to the first 6.4 and the last 6.4 seconds of the original utterance with some overlapping part. In this way, we also slightly benefit from data augmentation. As a result, we obtain 6803 emotional utterances from the aggregated corpora. Table 1 summarizes the information about these six corpora. For the evaluation, we adopt a 4-fold cross validation strategy by splitting the actor set and maintain the distribution as gender and corpus uniform as possible, similar to [24].

4.2. Experimental Setup

We extract the spectrograms of each utterance with a 25ms window for every 10ms using the Kaldi [25] library. Cepstral mean and variance normalization is then applied on the spectrogram frames per utterance. To equip each frame with a certain context, we splice it with 10 frames on the left and 5 frames on the right. Therefore, a resulting spliced frame has a resolution of 16×257. Since emotion involves a longer-term mental state transition, we further down-sample the frame rate by a factor of 8 to simplify and expedite the training process.

We establish a baseline of 3-branch ResNeXt that consists of only 5 convolutional layers and list the details in Table 2. For each utterance, a simple mean pooling is taken at the output of the residual block to form an utterance representation before it is transformed by the fully connected layers. We avoid explicit temporal modeling layers such as a long short-term memory recurrent network because our focus is to investigate the effectiveness by shaking the ResNe(X)t. Note that a Shaking layer has no parameter to learn and hence the model size does not change during this work. We implement the Shaking layer as well as the entire network architecture using the PyTorch [26] library. Only the Shake-Shake combination [1] is used and shaking is

Models	Layers	No. Params
3-Branch ResNeXt [w/ shake reg.]	Conv(4,2,16) + (Shortcut, Branch$(64) \times 2$) + [Shaking+] (Shortcut, Branch$(128) \times 2$) + [Shaking+] Mean-Pooling + Dropout(0.5) + ReLU + Linear(4)	34.3 K
Branch(F)	BatchNorm + ReLU + Conv(4,2,F) + BatchNorm + ReLU + Conv(4,2,F)	

Table 2: *Network architecture, layers and the number of parameters in the baseline and proposed models. BatchNorm stands for batch normalization, ReLU for the rectified linear unit, Conv(N, H, W) for a $2D$ convolutional layer with N filters of size $H \times W$, Dropout(p) for dropout with a dropping probability of p and Linear(N) for a fully connected layer with N nodes. The Mean-Pooling layer represents the temporal pooling for generating an utterance representation.*

applied independently per frame. As a preliminary study, we leave other combinations for future work. Due to class imbalance in the aggregated corpora, the objective function for training is the weighted cross-entropy, where the class weight is inversely proportional to the class size. The models are learned using the Adam optimizer [27] with an initial learning rate of 0.001 and the training is carried out on an NVIDIA Tesla K80 GPU. We use a mini-batch of 64 utterances across all model training and let each experiment run for 300 epochs.

4.3. Experimental Results

Table 3 summarizes the statistics from experiments to benchmark the proposed models, including the un-weighted accuracy, UA (%), the performance gap between training and validation UAs, and the number of epochs required to achieve the optimal UA. All of these numbers are averaged over four-fold cross validation for an estimate.

Model	**UA**	**Gap**	**Epoch**
1	59.27	8.52	58.75
2	59.74	1.69	220
3	57.26	-2.76	166.5
4(a)	59.13	-1.03	159.25
4(b)	**60.87**	**3.72**	225.5

Table 3: *Summary of the evaluations on the proposed models.*

First of all, we observe that two models, the ones based on Shake-Shake regularization (Model 2) and a hybrid shaking mechanism (Model $4(b)$), outperform the baseline by a respective margin of 0.47% and 1.6%. In addition, Model $4(a)$ also gives a performance comparable to the baseline. Model 3, on the other hand, degrades the performance by a margin of 2.01%.

Second, compared to the performance gap between training and validation by the baseline, all of the shaking regularized models have achieved a much smaller gap. It appears that shaking regularization contributes more on constraining the optimization of model parameters than boosting the generalization power. In fact, the authors in [2] also reported that with shaking regularization the training loss is much larger than zero in the end of 300-epoch or 1800-epoch training, whereas a deep Pyramidal ResNet can reach almost zero in the end of training. By benchmarking Model 2 and Model 3, one can easily infer that negative scaling has introduced a much stronger strength of regularization which instead hurts the optimization. In Model 4, negative scaling only applies to the first shaking layer. By controlling the survival probability to relax shaking, Model $4(a)$ and Model $4(b)$ both achieve a competitive performance and maintain a smaller gap.

Metric\Model	**2**	**3**	**4(a)**	**4(b)**
UA	0.291	0.916	0.594	0.061
Gap	0.001	0.003	0.004	0.040

Table 4: *P-values for statistical hypothesis test of improvement with respect to the baseline (**Model 1**).*

To gain more insight into the the experimental results in Table 3, we perform one-sided paired t-test (df=3) between the baseline and each of the shaking regularized models. In Table 4, it is clear that the shaking mechanism helps to significantly reduce **GAP**, i.e. reduced over-fitting. Based on the statistical analysis of **UA** by Model 2, one may observe that applying shaking alone only leads to insignificant improvement of generalization performance. However, with a flexible configuration of hybrid layers, Model 4 delivers a significant improvement of **UA** and **Gap** with respective 90% and 95% confidence. The correlation between residual branches are presented in Table 5. Except for Model 3 and $4(a)$, residual branches are able to capture independent information as observed in [1]. These highly correlated branches in Model 3 and $4(a)$ may suitably explains their under-performing behaviors as well.

Block\Model	**1**	**2**	**3**	**4(a)**	**4(b)**
1	-0.025	0.045	0.079	0.719	-0.066
2	0.046	-0.036	0.612	0.064	0.033

Table 5: *Correlation between branches.*

Finally, we find that all of the shaking regularized models converges at a slower speed. In particular, the baseline took an average of 58.75 epochs to search for the optimal network parameters, while all other models generally spent several multiples of the number of epochs that the baseline took. We did not use early stopping in this work. Nevertheless, these results suggest when using early stopping in training, one should pay more patience in optimizing the model parameters of a shaking regularized model.

5. Conclusions

We proposed stochastic Shake-Shake regularization based on multi-branch architectures to mitigate over-fitting in affective learning from speech. Stochastic Shake-Shake regularization allows negative scaling while still maintains a consistent stochastic convex combination of branches in order to encourage model-based diversity among branches. However, multiplying flow of forward propagation with a negative scalar incurs a much stronger strength of regularization that effectively hinders the optimization of network parameters. We found that randomly relaxing the shaking mechanism conveniently serves to control the regularization strength. Arguably, it would be more usable in practice than adjusting the range of random variables α_{nl} or β_{nl} as investigated in [1]. Finally, we have demonstrated that a stochastic Shake-Shake regularized 3-branch ResNeXt outperforms the ResNeXt for speech emotion recognition by a clear margin of 1.6% with reduced over-fitting, from which one can infer stochastic depth and negative scaling are also applicable to shallow networks and speech related tasks.

6. References

[1] X. Gastaldi, "Shake-Shake Regularization," in *International Conference on Learning Representations Workshop*, 2017.

[2] Y. Yamada, M. Iwamura, and K. Kise, "ShakeDrop Regularization," *arXiv:1802.02375*, 2018.

[3] K. He, X. Zhang, S. Ren, and J. Sun, "Deep Residual Learning for Image Recognition," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016.

[4] J. Gehring, M. Auli, D. Grangier, D. Yarats, and Y. N. Dauphin, "Convolutional Sequence to Sequence Learning," 2017, arXiv:1705.03122.

[5] O. Abdel-Hamid, A.-R. Mohamed, H. Jiang, L. Deng, G. Penn, and D. Yu, "Convolutional Neural Networks for Speech Recognition," *IEEE/ACM Trans. Audio, Speech and Lang. Proc.*, vol. 22, no. 10, pp. 1533–1545, Oct. 2014.

[6] N. Srivastava, G. Hinton, A. Krizhevsky, I. Sutskever, and R. Salakhutdinov, "Dropout: A Simple Way to Prevent Neural Networks from Overfitting," *J. Mach. Learn. Res.*, vol. 15, no. 1, Jan. 2014.

[7] P. Ekman, E. R. Sorenson, W. V. Friesen *et al.*, "Pan-cultural Elements in Facial Displays of Emotion," *Science*, vol. 164, no. 3875, pp. 86–88, 1969.

[8] S. S. Narayanan and P. Georgiou, "Behavioral Signal Processing: Deriving Human Behavioral Informatics from Speech and Language," *Proceedings of IEEE*, vol. 101, no. 5, pp. 1203–1233, May 2013.

[9] Q. Mao, M. Dong, Z. Huang, and Y. Zhan, "Learning Salient Features for Speech Emotion Recognition Using Convolutional Neural Networks," *IEEE Trans. Multimedia*, vol. 16, no. 8, pp. 2203–2213, 2014.

[10] C. Huang and S. S. Narayanan, "Deep Convolutional Recurrent Neural Network with Attention Mechanism for Robust Speech Emotion Recognition," in *IEEE International Conference on Multimedia and Expo (ICME)*, 2017.

[11] S. Ioffe and C. Szegedy, "Batch Normalization: Accelerating Deep Network Training by Reducing Internal Covariate Shift," in *Proceedings of the 32nd International Conference on Machine Learning (ICML)*, 2015.

[12] S. Zagoruyko and N. Komodakis, "Wide Residual Networks," in *BMVC*, 2016.

[13] G. Huang, Y. Sun, Z. Liu, D. Sedra, and K. Weinberger, "Deep Networks with Stochastic Depth," in *ECCV*, 2016.

[14] S. Xie, R. Girshick, P. Dollr, Z. Tu, and K. He, "Aggregated Residual Transformations for Deep Neural Networks," *arXiv:1611.05431*, 2016.

[15] A. Krizhevsky, "Learning Multiple Layers of Features from Tiny Images," *Technical Report*, 2009.

[16] D. Han, J. Kim, and J. Kim, "Deep Pyramidal Residual Networks," *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 6307–6315, 2017.

[17] O. Martin, I. Kotsia, B. M. Macq, and I. Pitas, "The eNTERFACE'05 Audio-Visual Emotion Database," in *Proceedings of the International Conference on Data Engineering Workshops*, 2006.

[18] S. R. Livingstone, K. Peck, and F. A. Russo, "RAVDESS: The Ryerson Audio-Visual Database of Emotional Speech and Song," in *Proceedings of the 22nd Annual Meeting of the Canadian Society for Brain, Behaviour and Cognitive Science*, 2012.

[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, Nov 2008.

[20] F. Burkhardt, A. Paeschke, M. Rolfes, W. F. Sendlmeier, and B. Weiss, "A Database of German Emotional Speech," in *Proceedings of Interspeech*, 2005.

[21] G. Costantini, I. Iaderola, andrea Paoloni, and M. Todisco, "EMOVO Corpus: an Italian Emotional Speech Database," in *Proceedings of the 9th International Conference on Language Resources and Evaluation*, 2014.

[22] S.-U. Haq and P. J. Jackson, *Machine Audition: Principles, Algorithms and Systems*. IGI Global, 2010, ch. Multimodal Emotion Recognition.

[23] G. Keren, J. Deng, J. Pohjalainen, and B. Schuller, "Convolutional neural networks with data augmentation for classifying speakers native language," in *Proceedings of Interspeech*, 2016.

[24] C. Huang and S. S. Narayanan, "Shaking Acoustic Spectral Subbands Can Better Regularize Learning in Affective Computing," in *Proceedings of the IEEE International Conference on Audio, Speech and Signal Processing*, 2018.

[25] D. Povey, A. Ghoshal, G. Boulianne, N. Goel, M. Hannemann, Y. Qian, P. Schwarz, and G. Stemmer, "The KALDI speech recognition toolkit," in *Proceedings of the IEEE Workshop on Automatic Speech Recognition and Understanding*, 2011.

[26] "Pytorch: Tensors and Dynamic Neural Networks in Python with Strong GPU Acceleration," 2017, https://github.com/pytorch/pytorch.

[27] D. P. Kingma and J. Ba, "Adam: A Method for Stochastic Optimization," *arXiv:1412.6980*, 2014.