Adversarial Data Augmentation Network for Speech Emotion Recognition

Lu YI and Man-Wai Mak
The Hong Kong Polytechnic University, Hong Kong SAR, China
E-mail: luicerain@gmail.com
E-mail: enmwmak@polyu.edu.hk

***Abstract*—Insufficient data is a common issue in training deep learning models. With the introduction of generative adversarial networks (GANs), data augmentation has become a promising solution to this problem. This paper investigates whether data augmentation can help improve speech emotion recognition. Unlike conventional GANs, we train a GAN with an autoencoder, where the input to the discriminator comes from the bottleneck layer of the autoencoder and the output of the generator. The synthetic samples can be obtained from the decoder, using the output of the generator as the decoder's input. The combined network, namely adversarial data augmentation network (ADAN), can generate samples that share common latent representation with the real data. Evaluations on EmoDB and IEMOCAP show that using OpenSmile features as input, the ADAN can produce augmented data that make an ordinary SVM classifier outperforms an RNN classifier with local attention and make a DNN competitive to some state-of-the art systems.**

I. Introduction

Emotion recognition plays an important role in natural human computer interaction. With the development of deep learning, impressive progress has been achieved in speech emotion recognition [1], [2], [3]. Instead of using hand-crafted spectral and prosodic features, deep belief networks can be used for feature learning and feature selection [4]. To classify the frame-based bottleneck features or to exploit the dynamic structure of frame-based features, long short-term memory recurrent neural networks (LSTM-RNN) have been used [5], [6]. Training a deep learning model requires a lot of data. Unfortunately, in many applications, acquiring labeled data is a big challenge. The data collection process is expensive and time-consuming. It is also difficult to define emotion in a precise way, because the emotion of some utterances may be ambiguous and could belong to more than one emotion type. In some cases, even professional annotators may not be unanimous in their decisions [7]. Therefore, it is important to address the data sparsity problem. In this paper, we propose an elegant solution based on the idea of Generative Adversarial Network (GAN) to enlarge the training set by generating fake samples.

Transfer learning [8] is a popular solution to the insufficient data problem. In particular, domain adaptation (a subset of transfer learning) is able to adapt a source-domain model to fit the target-domain data without requiring labeled data from the target domain. This technique has made significant progress in image classification [9]. Motivated by the achievements in image classification, transfer learning has been gradually applied to speech emotion recognition as well [10], [11].

The introduction of generative adversarial networks (GANs) [12] opens up new opportunities for addressing the insufficient training data problem. A typical GAN comprises two neural networks: a generator and a discriminator. These two networks act like two players aiming to win a zero-sum game. The generator is trained to map an arbitrary distribution to the data distribution, and the discriminator is trained to distinguish whether a sample comes from the data distribution (i.e., genuine) or from the generator (i.e., fake). Some researchers advocate that the GAN-based data augmentation technique can help to improve performance on recognition tasks [13], [14].

Another GAN-based approach is the adversarial autoencoder (AAE) [15]. In addition to minimizing the reconstruction errors, an AAE also needs to match the aggregated posterior distribution of the latent representation to an arbitrary prior distribution. Sahu *et al.* [16] exploited the AAE to synthesize speech emotion samples and applied the synthetic data for emotion classification. Their results demonstrate that adding synthetic data to the original training set helps to improve the performance of speech emotion classification. However, they also pointed out that the generated features do not follow the actual marginal distribution of the real samples.

In this work, we design an adversarial data augmentation network (ADAN) by combining an autoencoder with a GAN in a way that is different from the AAE. To address the problem of AAE highlighted above, we take feature learning into consideration. Specifically, instead of presenting the decoder with random vectors sampled from an arbitrary distribution, we present the decoder with emotion-aware latent vectors generated by a DNN. These emotion-aware latent vectors are made indistinguishable from the output vectors of the encoder through adversarial learning. As a result, the ADAN aims to generate samples that share common latent representation with the real data. Our experimental results demonstrate that this data augmentation approach can improve the speech emotion recognition on the EmoDB [17] and IEMOCAP [7] datasets.

This work was supported by the Hong Kong Research Grant Council, Project No. 152137/17E.

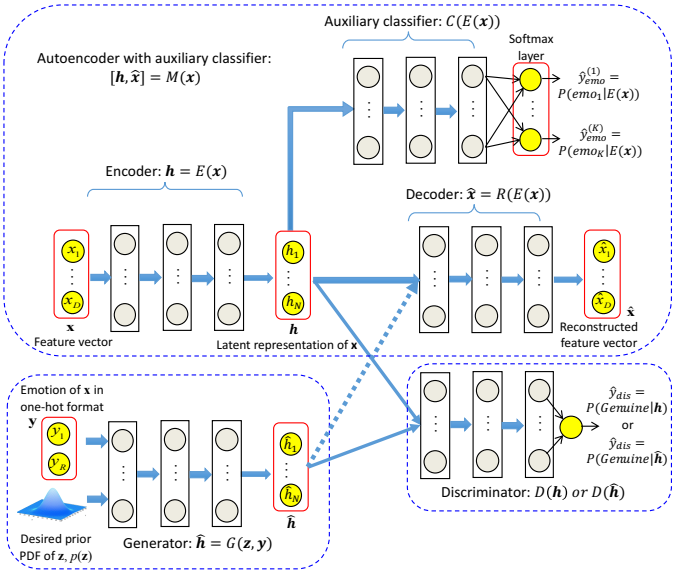


Fig. 1. The structure and data flow of an adversarial data augmentation networks (ADANs), the network comprises an autoencoder with an auxiliary classifier (top), a generator (lower-left) and a discriminator (lower-right). The dotted line is only used for data augmentation after training.

II. Adversarial Data Augmentation Network

A. Network Structure and Training Algorithm

Fig. 1 shows the structure of an adversarial data augmentation network (ADAN). It consists of an autoencoder $R(E(\mathbf{x}))$ with an auxiliary classifier $C(E(\mathbf{x}))$, a generator $G(\mathbf{z}, \mathbf{y})$ and a discriminator $D(\mathbf{h})$. The ADAN aims at: (i) learning a latent representation that retains emotion information, (ii) matching the posterior distribution $p(\hat{\mathbf{h}}|\mathbf{z}, \mathbf{y})$ to the posterior distribution $p(\mathbf{h}|\mathbf{x})$, and (iii) minimizing the reconstruction errors between $\mathbf{x}$ and $\hat{\mathbf{x}}$.

To achieve the three aims above, the three components in Fig. 1 are trained adversarially. Specifically, the encoder of the autoencoder is trained to learn a latent representation $\mathbf{h}$ in an N-dimensional space. The auxiliary classifier is to ensure that the latent representation is able to differentiate the emotion classes. The decoder is to reconstruct the emotion vectors in the original space. The generator takes samples drawn from an arbitrary distribution in an N-dimensional space and one-hot encoded emotion labels as input and generates samples in the latent space; its goal is to generate samples that are indistinguishable from the real samples in the latent space, i.e., $p(\mathbf{h}|\mathbf{x}) \approx p(\hat{\mathbf{h}}|\mathbf{z}, \mathbf{y})$. The discriminator aims at distinguishing whether a latent vector comes from the real data or from the generator. Compared with generating samples in the original space, the advantage of generating samples in the latent space is that the latter overcomes the difficulties of generating high-dimensional vectors.

To train the proposed network, we minimize the losses defined below:

$$\mathcal{L}_D^{(ADAN)} = \mathbb{E}_{p(\mathbf{x},\mathbf{y},\mathbf{z})}\Big\{ -\log D(E(\mathbf{x})) - \log(1 - D(G(\mathbf{z}, \mathbf{y})))\Big\} \tag{1}$$

$$\mathcal{L}_C^{(ADAN)} = \mathbb{E}_{p(\mathbf{x})}\Big\{ -\sum_{k=1}^{K} y_{emo}^{(k)} \log C(E(\mathbf{x}))_k \Big\} \tag{2}$$

$$\mathcal{L}_R^{(ADAN)} = \mathbb{E}_{p(\mathbf{x})}\Big\{ ||\mathbf{x} - R(E(\mathbf{x}))||^2 \Big\} \tag{3}$$

$$\mathcal{L}_E^{(ADAN)} = \mathbb{E}_{p(\mathbf{x})}\Big\{ ||\mathbf{x} - R(E(\mathbf{x}))||^2 - \sum_{k=1}^{K} y_{emo}^{(k)} \log C(E(\mathbf{x}))_k \Big\} \tag{4}$$

$$\mathcal{L}_G^{(ADAN)} = \mathbb{E}_{p(\mathbf{z},\mathbf{y})}\Big\{ \log(1 - D(G(\mathbf{z}, \mathbf{y})) - \sum_{k=1}^{K} y_{emo}^{(k)} \log C(G(\mathbf{z}, \mathbf{y}))_k \Big\} \tag{5}$$

where $()_k$ denotes the k-th element of a vector, G stands for the generator, R for the decoder, E for the encoder, D for the discriminator and C for the auxiliary classifier. If the generator is trained with inverted labels [12], the generator loss becomes

$$\mathcal{L}_G^{(ADAN)} = \mathbb{E}_{p(\mathbf{z},\mathbf{y})}\Big\{ -\log(D(G(\mathbf{z}, \mathbf{y})) - \sum_{k=1}^{K} y_{emo}^{(k)} \log C(G(\mathbf{z}, \mathbf{y}))_k \Big\}. \tag{6}$$

After training, the auxiliary classifier C and the discriminator D are removed. Then the remaining parts are used for data augmentation by connecting the generator G to the decoder R (the dotted arrow in Fig. 1). The augmentation procedure will be described in Section III.

B. Advantages of ADAN

If the autoencoder and the auxiliary classifier of an ADAN are well trained and fixed, then the remaining parts are similar to a vanilla GAN. This motivates us to train an autoencoder and a vanilla GAN separately. However, this training approach leads to poorly performed augmented data in our experiments. This may be due to the fundamental problems of GAN; that is to say, GANs are hard to train because of the gradient vanishing problem and gradient instability problem in the generator [18], especially when the amount of training received by the generator and the discriminator is not carefully balanced. Imagine a situation where the distributions of the fake and real samples are fairly different at the beginning of training. Then, the fake and real samples could be easily classified by the discriminator. If the discriminator becomes so good that the error gradient received by the generator become zero, then the generator will be prohibited to learn anything.

The gradient vanishing problem in GAN can be overcome by the ADAN. The reason is explained as follows. When training begins, the weights and bias terms of the generator

and the encoder are initialized randomly with zero mean and variable variances. Because both the feature vectors and weights in E have zero mean and `tanh` is used as the activation function, the expectation of $\mathbf{h}$ is almost zero.[1] Similar situation occurs in the generator. This means that at the early stage of training, the distributions of $\mathbf{h}$ and $\hat{\mathbf{h}}$ are largely overlapped, which causes difficulty for the discriminator to differentiate these two groups of latent vectors. With such a high cross-entropy loss from D, the generator will receive non-zero error gradient and the gradient vanishing problem will not occur. During the course of training, the auxiliary classifier and the decoder will encourage E to form class-dependent clusters in the latent space, while the adversarial training of G (through D, the first term of (5)) will make $\hat{\mathbf{h}}$'s similar to $\mathbf{h}$'s. Another strategy used by ADAN to avoid gradient vanishing is to inject cross-entropy loss of C to G through the second term of (5). This means that even if the gradient of the first term in (5) is zero, we still have the gradient of the second term to update G.

While the AAE and the proposed ADAN rely on an autoencoder to create a latent space, in the ADAN, the autoencoder is trained and used in a very different and potentially much better way. In particular, instead of using a mixture of Gaussians to generate class-dependent random samples [15], ADAN uses a DNN (the lower-left network in Fig. 1) to generate class-dependent latent vectors, where the class information comes from the one-hot encoded labels. Because the DNN can be trained to maximize class information in the generated vectors, the ADAN avoids a potential problem in AAEs in which the synthetic samples follow an arbitrary distribution rather than the actual data distribution [16]. The auxiliary classifier will also encourage the encoder to form class-dependent clusters in the latent space, which facilitates the generator to learn and mimic the distributions of genuine latent vectors. In short, the proposed network can improve the quality of the synthetic samples compared to the vanilla GAN and AAE.

III. Experiments

A. Datasets

The Berlin Database of Emotional Speech (EmoDB) [17] and the Interactive Emotional Dyadic Motion Capture (IEMOCAP) [7] were used in the experiments.

EmoDB contains seven categories of emotional speech spoken by ten speakers. All speakers spoke the same set of verbal content in an anechoic chamber. It is a very small dataset which comprises 535 utterances.

IEMOCAP contains the utterances of ten actors participating in dyadic interactions. In this study, we considered four emotions: *angry*, *happy*, *neutral*, and *sad*. This amounts to 4490 utterances. The data can be divided into improvised sessions and scripted sessions. Because the actors need to change their emotional states within the scripted sessions, the emotions in the scripted sessions are more ambiguous than those in the improvised sessions [7]. We used all of the utterances belonging to these four emotion classes for training emotion classifiers and for performance evaluation. But we only used the improvised sessions for training the ADAN and for data augmentation.

B. Emotion Features

We used OpenSmile [19] to extract emotion features specified in Interspeech 2011 Speaker State Challenge [20] for EmoDB and Interspeech 2010 Paralinguistic Challenge for IEMOCAP [21]. For both datasets, we removed the features with zero variances. The features were then normalized independently by z-norm.

C. Evaluations

For IEMOCAP, we applied leave-one-session-out cross validation (LOSO-CV) to ensure that no testing data were involved in either data augmentation or training of emotion classifiers. IEMOCAP consists of five sessions, each with a male and a female speaker. For each fold in the LOSO-CV, we used four sessions for training and the remaining one for testing. For EmoDB, we applied leave-one-speaker-out cross validation. Thus, we performed a 10-fold cross-validation on EmoDB and 5-fold cross-validation on IEMOCAP. We used both weighted accuracy (WA) and unweighted average recall (UAR) for performance comparison.

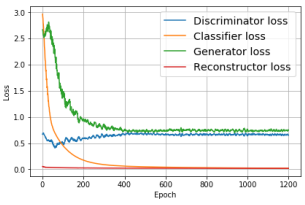
D. Experimental Setup

For each dataset, we carried out three steps: train ADANs, create augmented data, and train emotion classifiers.

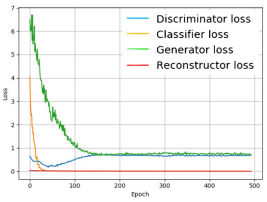
We set the dimension of the latent vectors to 100. For each training vector, we randomly drew a sample from a 100-dimensional Gaussian distribution $\mathcal{N}(\mathbf{0}, \mathbf{I})$ as the input to the generator. For each epoch, we trained the discriminator M times to minimize the cross-entropy of classifying real and synthetic data (1), followed by freezing its weights. Then the autoencoder and auxiliary classifier were trained ((2) – (4)), followed by freezing the weights of the auxiliary classifier. After that, we trained the generator to maximize the cross-entropy of the discriminator (1^{st} term of (6)) and to minimize the cross-entropy of the auxiliary classifier (2^{nd} term of (6)). When the network converged, we presented the one-hot emotion labels and Gaussian random vectors $\mathbf{z}$ to the generator, the synthetic latent vectors were then passed to the decoder (the dashed arrow in Fig. 1) to produce augmented data in the original space. The emotion labels determined the number of samples for each class that we wanted to obtain. We created 10 augmented sets, each with the same size as the original set, and augmented the synthetic data to the initial training set to train emotion classifiers.

The components in the ADAN are fully-connected neural networks with two hidden layers. The number of hidden neurons is 800 for the encoder and the decoder, while it is 100 for the remain parts. For each epoch, one pass of stochastic gradient descent was applied to the discriminator, i.e., $M = 1$. The learning rate for all subnetworks in the

[1] If random variable X and Y are independent, $\mathbb{E}\{XY\} = \mathbb{E}_X\{X\}\mathbb{E}_Y\{Y\}$.

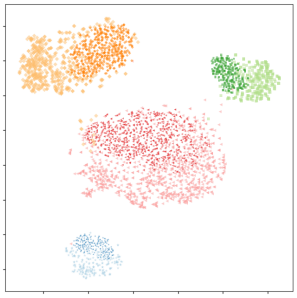


(a) EmoDB

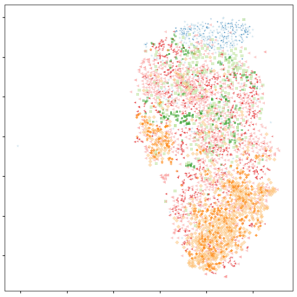


(b) IEMOCAP

Fig. 2. Cross entropy and mean squared error losses during the course of training of the ADAN using (a) EmoDB and (b) IEMOCAP as training data.



(a) Latent (IEMOCAP)



(b) Original (IEMOCAP)

Fig. 3. (a) Latent representation and (b) data in the original space after augmentation for four emotions in IEMOCAP improvised sessions. Markers of the same color belong to the same emotion. Lighter colors represent the real samples, while darker colors represent the synthetic samples.

ADAN was set to 0.0001. Support vector machines (SVMs) and simple deep neural networks (DNNs) were trained for emotion classification. Hyperbolic tangent was used in the ADAN and the DNN classifier for IEMOCAP. ReLU was used in the DNN classifier for EmoDB. Radial basis function kernels were used in the SVM classifier for EmoDB. The Adam optimizer [22] and Xavier weight initialization [23] were used in all DNNs. The scikit-learn toolbox was used for implementing the SVMs and Tensorflow was used for implementing the DNNs.

IV. Results

The curves in Fig. 2 clearly show that except for the discriminator loss, all losses decrease until convergence. The increase in the discriminator loss suggests that the synthetic samples causes adversity in the discriminator.

For visualization purpose, the t-SNE tool [24] was used to project the latent vectors and the feature vectors onto a 2-dimension space, as shown in Fig. 3. In the figure, emotional states are represented by different colors, with darker and lighter colors corresponding to synthetic and genuine vectors, respectively. Since we only used the improvised sessions of IEMOCAP for training the ADAN, the figure only reflects the data in that part. Clear emotion clusters can be observed in the latent space, and synthetic vectors are very close to the genuine vectors. For the original data space, we observe that the synthetic samples do not only close to the genuine ones, but also follow the actual data distribution. More importantly, mode collapse does not occur, which is a common problem in vanilla GANs.

TABLE I
Comparison of weighted accuracy (WA) on EmoDB dataset.

Methods	**WA** (%)
Perception by human (Burkhardt *et al.* [17])	87.50
GMM/SVM (Luengo *et al.* [25])	78.30
SVM with IS11_Speaker_State (Mak [26])	80.56
DNN with IS11_Speaker_State (Mak [26])	80.19
Augmentation based on data duplication	82.06
Augmentation based on adding noise	82.06
Augmentation based on SMOTE	82.43
ADAN + SVM (proposed)	80.93
ADAN + DNN (proposed)	**83.74**

TABLE II
Performance of SVM classifiers on IEMOCAP with and without data augmentation.

	Improvised only		**Improvised & Scripted**	
Training data	**WA**(%)	**UAR**(%)	**WA**(%)	**UAR**(%)
real only	67.94	60.06	64.74	58.25
synthetic only	67.76	58.10	59.00	52.52
real + synthetic	67.89	**61.32**	65.01	**59.01**

As EmoDB is a small dataset, it is expected that it can benefit a lot from data augmentation. Table I shows the weighted accuracy achieved by different methods. For this small dataset, SVMs can easily beat the DNNs. However, after data augmentation using our proposed ADAN, we can train a DNN that outperforms the SVM by more than 3% absolute. Better DNNs were also trained based on other data augmentation techniques, such as duplicating the training data, adding noise to feature vectors and applying the synthetic minority over-sampling technique (SMOTE) [27]. Table I shows that the classifier based on the proposed ADAN achieves the best performance. The performance of SVMs is also improved after data augmentation, which indicates that the synthetic data have positive effect on finding better decision boundaries in the SVMs. This promising result suggests that data augmentation is beneficial for speech emotion recognition and our proposed ADAN can generate real-like samples.

Since the emotions in EmoDB are highly distinguishable, it may be easy to synthesize the emotion data. In order to obtain more convincing results, we conducted our experiments on the IEMOCAP dataset in which different emotions are more confusing. Thus, to mimic its data distribution is more challenging.

Table II shows the performance of the SVM classifiers on IEMOCAP using real, synthetic, and both real and synthetic data. Evidently, with the synthetic data, the performance can be slightly improved. We have also used confusion matrices for further analysis. We find that the classification accuracy of "Happy" increases after data augmentation. But augmenting more "Happy" data could not further improve performance. This may be due to the feature vectors themselves, of which the emotion "Happy" is confusable with other emotions. Ma *et al.* [28] pointed out that the neutral speech segments from sentences labeled with "Happy" are very similar to other emotions. The augmented data may help the classifier to better

TABLE III
CLASSIFICATION PERFORMANCE OF THE PROPOSED METHOD AND EXISTING METHODS ON IEMOCAP IMPROVISED AND SCRIPTED SESSIONS.

Methods	**WA** (%)	**UAR**(%)
RNN (Mirsamadi *et al.* [29])	63.50	58.80
AAE & SVM (Sahu *et al.* [16])	-	58.38
Augmentation based on data duplication	64.37	58.34
Augmentation based on adding noise	64.30	58.40
Augmentation based on SMOTE	62.47	58.52
ADNN + SVM (proposed)	65.01	**59.01**

TABLE IV
CLASSIFICATION PERFORMANCE OF THE PROPOSED METHOD AND END-TO-END SYSTEMS ON IEMOCAP IMPROVISED SESSION ONLY.

Methods	**WA** (%)	**UAR**(%)
2-D ACRNN ([30])	-	62.40
3-D ACRNN ([30])	-	64.74
Variable-Length DNN ([28])	71.45	64.22
ADAN + SVM (proposed)	67.89	61.32
ADAN + DNN (proposed)	66.80	62.83

recognize the emotion of happy but the help is still limited.

The accuracies obtained by using the synthetic data only for training are comparable with those obtained by using real data for training. Compared with AAE [16], which achieves only UAR of 33.75% by using synthetic data only, the synthetic samples generated by our proposed network are more like real data, which achieves an UAR of 52.52%.

Table III and Table IV compare the performance of our proposed network with some end-to-end systems [28], [30] that do not use OpenSMILE features as input. An advantage of end-to-end systems is that they can capture the emotion information from waveforms or time-frequency representations through supervised learning. Typically, they can outperform systems that are based on handcrafted features extracted by OpenSMILE. That explains why even with data augmentation, our ADAN + DNN in Table IV could not beat these end-to-end systems. Nevertheless, ADAN is a very general data augmentation method, which can be readily applied to create training data for these end-to-end systems. This will be an interesting future work to pursuit.

V. CONCLUSIONS

Insufficient data in speech emotion datasets is a problem in training deep learning models for speech emotion recognition. A lack of training data would lead to over-fitting in complex models. In this paper, an adversarial data augmentation network is proposed to find a latent space, generate samples in the latent space and produce synthetic samples in the original space for data augmentation. We demonstrated that this data augmentation network can produce emotion-rich augmented samples that are beneficial for training emotion classifiers. We have only considered the simple case in which the inputs to the data augmentation network are OpenSmile emotion vectors. Nevertheless, the augmentation network is general enough for generating other types of vectors.

REFERENCES

[1] K. Han, D. Yu, and I. Tashev, "Speech emotion recognition using deep neural network and extreme learning machine," in *Proc. Interspeech*, 2014, pp. 223–227.

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

[3] A. Stuhlsatz, C. Meyer, F. Eyben, T. Zielke, G. Meier, and B. Schuller, "Deep neural networks for acoustic emotion recognition: Raising the benchmarks," in *Proceedings of the IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2011, pp. 5688–5691.

[4] Y. Kim, H. Lee, and E. M. Provost, "Deep learning for robust feature generation in audiovisual emotion recognition," in *Proceedings of the IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2013, pp. 3687–3691.

[5] S. Ghosh, E. Laksana, L. P. Morency, and S. Scherer, "Representation learning for speech emotion recognition." in *Proc. Interspeech*, 2016, pp. 3603–3607.

[6] J. Lee and I. Tashev, "High-level feature representation using recurrent neural network for speech emotion recognition," in *Proc. Interspeech*, 2015, pp. 1537–1540.

[7] 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.

[8] S. J. Pan and Q. Yang, "A survey on transfer learning," *IEEE Transactions on Knowledge and Data Engineering*, vol. 22, no. 10, pp. 1345–1359, 2010.

[9] M. Wang and W. Deng, "Deep visual domain adaptation: A survey," *Neurocomputing*, vol. 312, pp. 135–153, 2018.

[10] J. Deng, Z. Zhang, E. Marchi, and B. Schuller, "Sparse autoencoder-based feature transfer learning for speech emotion recognition," in *Proceedings of the Humaine Association Conference on Affective Computing and Intelligent Interaction*, 2013, pp. 511–516.

[11] J. Gideon, S. Khorram, Z. Aldeneh, D. Dimitriadis, and E. M. Provost, "Progressive neural networks for transfer learning in emotion recognition," in *Proc. Interspeech*, 2017, pp. 1098–1102.

[12] I. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. Courville, and Y. Bengio, "Generative adversarial nets," in *Advances in neural information processing systems*, 2014, pp. 2672–2680.

[13] A. Antoniou, A. Storkey, and H. Edwards, "Data augmentation generative adversarial networks," *arXiv preprint arXiv:1711.04340*, 2017.

[14] Z. Zhang, J. Han, K. Qian, C. Janott, Y. Guo, and B. Schuller, "Snore-GANs: Improving automatic snore sound classification with synthesized data," *IEEE Journal of Biomedical and Health Informatics*, vol. PP, no. 99, pp. 1–1, 2019.

[15] A. Makhzani, J. Shlens, N. Jaitly, I. Goodfellow, and B. Frey, "Adversarial autoencoders," in *Proceedings of the 3rd International Conference on Learning Representations (ICLR)*, 2016.

[16] S. Sahu, R. Gupta, G. Sivaraman, W. AbdAlmageed, and C. Espy-Wilson, "Adversarial auto-encoders for speech based emotion recognition," in *Proc. Interspeech*, 2017, pp. 1243–1247.

[17] F. Burkhardt, A. Paeschke, M. Rolfes, W. F. Sendlmeier, and B. Weiss, "A database of German emotional speech," in *Proceedings of the 9th European Conference on Speech Communication and Technology*, 2005.

[18] M. Arjovsky, S. Chintala, and L. Bottou, "Wasserstein generative adversarial networks," in *International Conference on Machine Learning*, 2017, pp. 214–223.

[19] F. Eyben, F. Weninger, F. Gross, and B. Schuller, "Recent developments in Opensmile, the munich open-source multimedia feature extractor," in *Proceedings of the 21st ACM International Conference on Multimedia*, 2013, pp. 835–838.

[20] B. Schuller, S. Steidl, A. Batliner, F. Schiel, and J. Krajewski, "The INTERSPEECH 2011 speaker state challenge," in *Proc. Interspeech*, 2011, pp. 3201–3204.

[21] B. Schuller, S. Steidl, A. Batliner, F. Burkhardt, L. Devillers, C. Müller, and S. S. Narayanan, "The INTERSPEECH 2010 paralinguistic challenge," in *Proc. Interspeech*, 2010, pp. 2794–2797.

[22] D. P. Kingma and J. L. Ba, "Adam: A method for stochastic optimization," in *Proceedings of the 3rd International Conference on Learning Representations (ICLR)*, 2015.
[23] X. Glorot and Y. Bengio, "Understanding the difficulty of training deep feedforward neural networks," in *Proceedings of the 13th International Conference on Artificial Intelligence and Statistics*, 2010, pp. 249–256.
[24] L. V. d. Maaten and G. Hinton, "Visualizing data using t-SNE," *Journal of machine learning research*, vol. 9, no. Nov, pp. 2579–2605, 2008.
[25] I. Luengo, E. Navas, and I. Hernáez, "Feature analysis and evaluation for automatic emotion identification in speech," *IEEE Transactions on Multimedia*, vol. 12, no. 6, pp. 490–501, 2010.
[26] M. W. Mak, "Feature selection and nuisance attribute projection for speech emotion recognition," Department of Electronic and Information Engineering, The Hong Kong Polytechnic University, Tech. Rep., Dec 2016.
[27] N. V. Chawla, K. W. Bowyer, L. O. Hall, and W. P. Kegelmeyer, "SMOTE: synthetic minority over-sampling technique," *Journal of artificial intelligence research*, vol. 16, pp. 321–357, 2002.
[28] 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.
[29] S. Mirsamadi, E. Barsoum, and C. Zhang, "Automatic speech emotion recognition using recurrent neural networks with local attention," in *Proceedings of the IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP)*. IEEE, 2017, pp. 2227–2231.
[30] M. Chen, X. He, J. Yang, and H. Zhang, "3-D convolutional recurrent neural networks with attention model for speech emotion recognition," *IEEE Signal Processing Letters*, vol. 25, no. 10, pp. 1440–1444, 2018.