Interspeech 2018
2-6 September 2018, Hyderabad

On Enhancing Speech Emotion Recognition using Generative Adversarial Networks

Saurabh Sahu[1], *Rahul Gupta*[2], *Carol Espy-Wilson*[1]

[1]Speech Communication Laboratory, University of Maryland, College Park, MD, USA
[2]Amazon.com, USA

{ssahu89,espy}@umd.edu, gupra@amazon.com

Abstract

Generative Adversarial Networks (GANs) have gained a lot of attention from machine learning community due to their ability to learn and mimic an input data distribution. GANs consist of a discriminator and a generator working in tandem playing a min-max game to learn a target underlying data distribution; when fed with data-points sampled from a simpler distribution (like uniform or Gaussian distribution). Once trained, they allow synthetic generation of examples sampled from the target distribution. We investigate the application of GANs to generate synthetic feature vectors used for speech emotion recognition. Specifically, we investigate two set ups: (i) a vanilla GAN that learns the distribution of a lower dimensional representation of the actual higher dimensional feature vector and, (ii) a conditional GAN that learns the distribution of the higher dimensional feature vectors conditioned on the labels or the emotional class to which it belongs. As a potential practical application of these synthetically generated samples, we measure any improvement in a classifier's performance when the synthetic data is used along with real data for training. We perform cross validation analyses followed by a cross-corpus study.

1. Introduction

Emotion recognition has wide applications in psychology, medicine and designing human-computer interaction systems [1]. In particular, using speech data for emotion recognition is popular because it's collection is easy, non-invasive and cheap. Given that datasets available for this task are typically limited in size, we explore synthetic feature generation and their utility for emotion recognition experiments. Generative adversarial networks (GANs) [2] are popular tools that computer vision researchers have used to generate real looking synthetic images [3] as well as for speech emotion recognition [4, 5]. We generate synthetic features to aid emotion classification using two schemes: (i) a vanilla GAN to generate a compressed version of the actual feature vectors and, (ii) a conditional GAN [6] to generate the actual higher dimensional feature vectors. The goal of our experiments is to assess the performance increase one can obtain with these synthetic features.

GANs have enhanced state of the art in several tasks such as image generation [7], image translation [8] and dialog generation [9]. More recently, they have also been applied to the task of emotion recognition [4, 5]. However, these works have focused on learning feature representations for emotion recognition. In this paper, we investigate the task of improving emotion classification accuracy using GANs. Initially, we train GAN models to imitate emotion utterance representations and generate synthetic samples. The synthetic datapoints are then used as features with/without real data and fed to a classifier. We observe increase in classification performances, indicating that even with only few hours of data, GANs can learn to generate synthetic samples learned on training data distribution. Finally, we do a cross validation study followed by cross-corpus experiments to obtain a more comprehensive assessment.

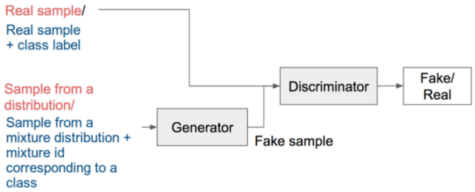


Figure 1: *Block representation of a GAN architecture. A vanilla GAN requires access to real samples from a dataset and samples from a probability density (depicted in red font). A conditional GAN also requires the class samples corresponding to real datasamples and a mixture probability density (depicted in blue font).*

2. Background: Generative Adversarial Networks

A vanilla GAN consists of two components: a generator, G and a discriminator, D. Given a random sample $\boldsymbol{z}$ from a probability distribution $p_{\boldsymbol{z}}$, the generator is responsible for generating a fake datapoint $G(\boldsymbol{z})$. The discriminator attempts to classify real samples $\boldsymbol{x}$ (drawn from a distribution p_{data}) against the one generated by the generator. The objective of training a GAN is to obtain a generator that can mimic real data such that the discriminator is incapable of differentiating between real and fake samples. GAN is trained using the following optimization on the GAN loss $V(D, G)$.

$$\begin{aligned}\min_G \max_D V(D,G) = {} & \mathbb{E}_{\boldsymbol{x}\sim p_{\text{data}}}[\log D(\boldsymbol{x})]+ \\ & \mathbb{E}_{\boldsymbol{z}\sim p_{\boldsymbol{z}}}[\log(1-D(G(\boldsymbol{z})))]\end{aligned} \tag{1}$$

In the equation above, $D(\boldsymbol{x})$ and $D(G(\boldsymbol{z}))$ are the probabilities that $\boldsymbol{x}$ and $G(\boldsymbol{z})$ are inferred to be real sample by the discriminator. Note that in the optimization in equation 1, the generator attempts to fool the discriminator as it tries to minimize $V(D, G)$. During GAN training, we minimize the discriminator and generator losses as defined below and track them separately. Note that for discriminator loss, y is 1 if input is $\boldsymbol{x}$ and 0 if input is $G(\boldsymbol{z})$.

$$\begin{aligned}&\textbf{Disc. loss:} \; -y\log(D(\boldsymbol{x})) - (1-y)\log(1-D(G(\boldsymbol{z}))) \\ &\textbf{Gen. loss:} \; -\log(D(G(\boldsymbol{z}))), \text{where } \boldsymbol{x}\sim p_{\text{data}}, \boldsymbol{z}\sim p_{\boldsymbol{z}}\end{aligned} \tag{2}$$

Although there are many variants of GAN architectures, we also experiment with a conditional GAN [6]. Apart from the real data-points, conditional GAN also requires a class label for each

10.21437/Interspeech.2018-1883

data-point. The distribution $p_{\boldsymbol{z}}$ is chosen to be a mixture distribution (e.g. Gaussian Mixture Models), where each mixture component corresponds to a sample class. The objective of conditional GANs is to be able to generate fake samples for a class, when $\boldsymbol{z}$ is sampled from the corresponding component mixture in $p_{\boldsymbol{z}}$. Figure 1 provides a block diagram of vanilla/conditional GAN architectures.

3. Synthetic Sample Generation for Emotion Recognition

Training emotion recognition system often suffers from a lack of data availability. As GANs have been successful in generating images, we explore their applicability in generating data samples for training emotion recognition systems. Specifically, we focus on using vanilla and conditional GAN architectures to generate samples for each emotion class in our experiments and present our analysis. As convergence of the loss $V(D, G)$ is often problematic, we also list the tricks we use to achieve the same. We first describe the dataset we use for training the GAN models, followed by a description of sample generation strategy.

3.1. Database for GAN training

We use the Interactive Emotional Dyadic Motion Capture (IEMOCAP) dataset [10] for training GAN models. The dataset consists of five sessions of scripted and improvised interactions between two actors acting out real world situations. No two sessions have the same set of actors, enabling us to do a speaker independent leave-one-session-out five-fold cross validation. The database comes with the dyadic conversation segmented into utterances which are on an average about 5 seconds in duration. The utterances are then labeled by three annotators for emotion labels such as happy, sad, angry, excitement and, neutral (class labels are required for training conditional GANs). We only use utterances for which we obtain a majority vote regarding the ground truth label. Following [11], we combine the utterances in happy and excited class to get a "combined happy" class for our experiments. This was done to obtain a more balanced dataset, due to a small number of "happy" class instances. For our classification experiments we focused on a set of 5531 utterances shared amongst four emotional labels: neutral (1708), angry (1103), sad (1084), and happy (1636). Overall, this amounts to approximately 7 hours of data.

We use the 'emobase2010' feature set in openSMILE toolkit [12] hat gives us a 1582-dimensional fixed length representation for each of the utterances. It consists of several functionals computed from a set of acoustic low level descriptors [13]. Next, we discuss the GAN training using these 1582-dimensional representations for each utterance.

3.2. Sample generation using GAN

Below, we describe the experiments performed using vanilla and conditional GANs separately.

3.2.1. Sample generation using vanilla GAN

In this experiment, we train a simple GAN architecture without the label information. Our initial aim was to generate synthetic 1582-dimensional feature vectors from a simple distribution $p_{\boldsymbol{z}}$ which was set as a 2 dimensional Gaussian distribution with zero mean and unit variance. Consequently, the generator is a feed-forward neural network with two neurons in the input layer and 1582 neurons in the output layer. Our discriminator is also a feed-forward neural network with 1582 neurons in input layer followed by two hidden layers and an output node with sigmoid activation. However, we could not get the generator and discriminator losses (equation 2) to converge. Any attempt towards changing the architecture, learning rates, number of epochs didn't lead to a convergence of losses. We hypothesized that this issue stems from a high dimensionality of feature space and the resulting data sparsity. This prompted us to train a GAN to generate synthetic lower dimensional representations of the original higher dimensional representations.

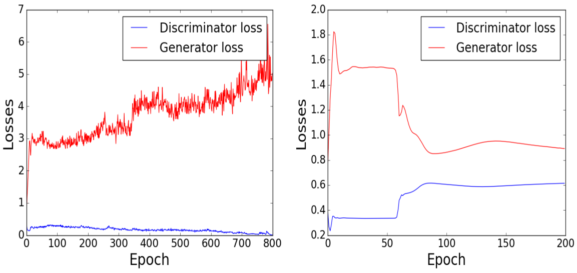


Figure 2: *Adversarial losses for GANs trying to estimate the actual high dimensional distribution (left) and their reduced 2-dimensional representation (right). Note how the errors can't converge while trying to estimate the higher dimensional distribution using a vanilla GAN.*

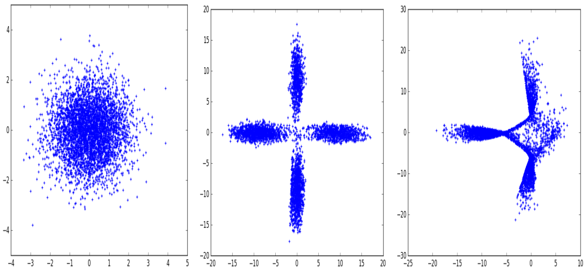


Figure 3: *GAN was trained to transform a simple 2-D Gaussian distribution (left) to a lower-dimensional representation of the 1582-D feature vectors resembling mixture of four Gaussian components (center). After training, the generator's output distribution is shown on the right.*

We use an adversarial auto-encoder framework [4] to get the lower dimensional representations which has been shown to map the higher dimensional features onto a 2D space while preserving the cluster structure/relationship between feature vectors efficiently. The compressed feature representations resemble a GMM with four components, each GMM component corresponding to an emotion class [4]. The output layer of generator now had two neurons and so does the input layer of discriminator. Input to generator are samples from a zero mean unit variance Gaussian distribution. Figure 2 shows the convergence of adversarial errors with the older and newer set up. While we had difficulty achieving convergence in synthetically generating 1582 dimensional feature vectors, vanilla GAN convergences when the target distribution was a 2 dimensional representation of original feature vectors. We generate synthetic data-points using the trained GAN and plot them in Figure 3. We observe that we roughly obtain the four component GMM distribution.

In our later experiments, we use the generated samples for classification. Each synthetic data-point is assigned to an emotion class based on the GMM component yielding the highest membership for the generated data point.

3.2.2. *Sample generation using conditional GAN*

We now focus on architectures that can generate synthetic higher dimensional feature vectors. We hypothesize that for a GAN to converge while trying to learn the distribution of higher dimensional representations, we need to provide it with more information. Conditional GAN is one such example where the synthetic data generation is conditioned on labels. Given a set of data-points $\mathbf{x} \sim \mathbf{p}_{\text{data}}$ and their corresponding labels $\mathbf{y}$, a vanilla GAN models the distribution $\mathbf{p(x)}$ while a conditional GAN learns the conditional distribution $\mathbf{p(x|y)}$. In our experiments, we chose $p_{\boldsymbol{z}}$ to be a mixture of four component GMM, with the target as modeling p_{data} in the 1582 dimensional feature space. As is done typically for a conditional GAN, each mixture component in the GMM corresponds to a particular emotion. The class information for the real data-points as well as GMM components information during optimization are provided as one-hot encoded vectors. We use several tricks to train a conditional GAN as described in detail below.

First, we split the data-points in the IEMOCAP dataset into a training (4 sessions) and validation set (1 session). For the baseline conditional GAN, we randomly initialize the generator and discriminator parameters. The learning rates of the generator and discriminator and the number of epochs for which they were trained are kept the same. Figure 4(a) shows the plot of adversarial errors for the training and development splits, indicating a lack of convergence. Next we investigated the effect of initializing the network parameters based on a pre-trained network. We initialize the generator with decoder weights of a pre-trained adversarial auto-encoder (Figure 1 in [4]). Figure 4(b) shows the plot of adversarial errors for the training and development splits. We observe that while the losses converge both on the training and development sets, the discriminator error is very low. This indicates that the discriminator is still able to distinguish between real data-points and the fake data produced by the generator. To improve the generator, we further incorporate two changes in our training scheme: (i) keeping the generator's learning rate higher than the discriminator (0.001 vs 0.0001 respectively) and, (ii) training the generator for five iterations for every iteration of discriminator training. Figure 4(c) shows that this leads to a higher discriminator loss, indicating the generator is able to produce data-points that can fool the discriminator. Training is not only stable but error convergence plots show that this training procedure also generalizes to the validation split. We refer to this model as improved conditional GAN.

We use the generated samples by the conditional GAN to improve emotion classification. The synthetic samples generated are assigned a class based on the corresponding GMM component in $p_{\boldsymbol{z}}$. In the next section, we describe our classification setup using the synthetically generated samples

4. Classification using synthetically generated samples

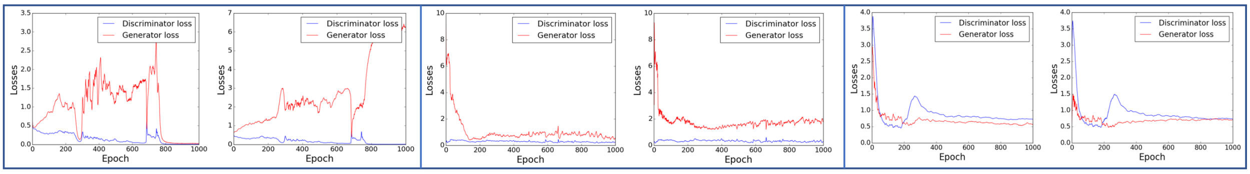
While the convergence of loss functions are a helpful tool to judge the capability of a trained GAN, we also investigate if the generated samples could aid emotion classification. To this end, we perform three sets of evaluations: (i) in-domain evaluation using synthetic samples as training set with and without real data (ii) in-domain evaluation on synthetic samples as test set and, (iii) a cross-corpus evaluation using a combination of real and synthetic data. For the simple GAN, we generate two dimensional representations of utterances to mimic the two dimensional representation learned by adversarial auto-encoders on real data. Additionally, we use the conditional GAN to generate 1582-dimension feature vectors to mimic the real data distribution. The corresponding emotion classes for the data-points generated using vanilla and conditional are identified as specified in section 3.2.1 and 3.2.2, respectively. We train models to classify an utterance amongst the four emotion classes and use Unweighted Average Recall on test sets as our evaluation metric. We briefly describe each of these experiments below, followed by results using vanilla and conditional GAN.

4.1. Synthetic samples in training set

In this experiment, we perform a leave one session out cross-validation experiment on the IEMOCAP dataset. Given that each session contains a unique pair of participants, this evaluation is also speaker independent. We train the vanilla/conditional GAN on four IEMOCAP sessions and generate synthetic samples. We train a Support Vector Machine (SVM) model with radial basis function as kernel, SVM$_{\text{van}}$, on the 2-dimensional projection space learned by the adversarial auto-encoder and train it under three conditions: (i) using only the synthetic samples generated by the vanilla GAN, (ii) using only the real samples in the four training sessions and, (iii) using a combination of both synthetic and real samples. The trained models are evaluated on the 2-dimensions representations of the test set, as yielded by the adversarial auto-encoder. Similarly, after obtaining samples from the conditional GAN, we train another SVM model, SVM$_{\text{con}}$, on the 1582-dimensional openSMILE feature space. We again perform the three sets of experiments as mentioned above. SVM$_{\text{con}}$ is evaluated on the test partition in the 1582-dimensional openSMILE feature space. The results for this experiment is listed in Table 1. It is clearly evident that by using only the synthetically generated samples for training the SVM we beat the chance accuracy by a big margin. It is worth noting that in case of simple GAN, the SVMs performance trained with only synthetic data is comparable to that of a SVM trained with actual 2D code vectors. This could probably be because the 2D code vectors follow a specific distribution enforced by the adversarial auto-encoder framework and not just any random distribution. The specific distribution being the mixture of four Gaussian components is not as complex as real world distributions and hence the GAN model could easily learn that distribution. Furthermore, from Table 1 it can be seen that while appending the real feature vectors with synthetic feature vectors from baseline conditional GAN can hurt the performance slightly thats not the case when appending them with synthetic data points generated from improved conditional GAN. An improvement in UAR in both cases shows the potential of using synthetically generated data along with real data for training and classification purposes.

4.2. Synthetic samples in test set

In this experiment, in addition to using the real dataset to train the GANs, we also use them to train a SVM for emotion recognition. The synthetic samples were used in the test set. The objective of this experiment is to assess the similarity between real and synthetic data by using a model trained on real data to classify synthetic data. In case of vanilla GAN, the generated 2D representations were used as test set while the compressed 2D representations were used for training the SVM. For conditional GAN the higher dimensional feature vectors were used for training the SVM which was evaluated on the synthetically generated test set. Results are shown in Table 2. As expected for the higher dimensional features the results shown are similar to what was obtained when the synthetic samples were used for training. For 2D samples the high accuracy suggests its much easier to estimate simpler dimensional distribution than a higher



(a) (b) (c)

Figure 4: *Convergence of generator and discriminator errors for training set (left) and validation sets (right) when (a) the generator of conditional GAN was randomly initialized (baseline-conditional), (b) the generator of conditional GAN was initialized using decoder's weights of a previously trained adversarial auto-encoder and, (c) along with weight initialization other schemes were used for better convergence (improved-conditional).*

Table 1: *Classification results of a SVM trained using different set-ups involving the synthetically generated 2-D representation of the real 1582-D openSMILE and the synthetically generated higher dimensions samples with and without real data. Feature vectors of same dimensionality were used in the test set*

Dataset	UAR (%)
Chance accuracy	25.00
Only synthetic 2D code vectors	55.38
Only 2D code vectors	56.72
2D code vector + Synthetic	**57.58**
Only improved-conditional	34.09
Only real openSMILE	59.42
Real openSMILE + baseline-conditional	59.20
Real openSMILE + improved-conditional	**60.29**

Table 2: *Classification results of using synthetically generated vectors in the test set*

Dataset	UAR (%)
2D code-vectors	97.09
Improved-conditional	35.23

dimensional complex distribution.

4.3. Cross corpus experiments

Having studied the convergence of GAN architectures and evaluating the quality of synthetically generated samples produced by them in a single corpora setting, we now move to performing cross-corpus evaluations. The objective of this experiment is to investigate if synthetically generated samples can be used during classification on an external corpus (as opposed to being applicable for only in-domain tasks). We use IEMOCAP for training and MSP-IMPROV [14] as our testing set. MSP-IMPROV, like IEMOCAP, also has actors participating in dyadic conversations which has then been segmented into utterances and annotated by evaluators. There are 7798 utterances in total spanned across the same four emotion classes. However, the distribution across classes was highly unbalanced with the number of utterances belonging to happy/neutral class more than three times the number of angry/sad samples. This prompted us to use it as a test set rather than training set. The loss curves for a conditional GAN with the same set-up used in cross-validations experiments is shown in Figure 5. We observe that the adversarial errors converge even if the test set is a different corpus than the training set. Results in Table 3 show a similar trend as cross-validation results.

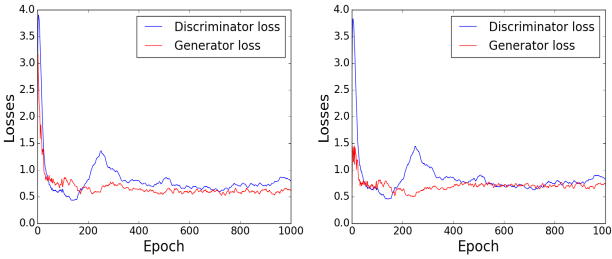


Figure 5: *Loss curves showing the error convergence for a conditional GAN on training set (left) and test set (right) for cross-corpus experiments.*

Table 3: *Classification results of a SVM trained on synthetic samples along with real training samples for different scenarios for cross-corpus experiments*

Dataset	UAR (%)
Only synthetic 2D	40.17
Only 2D code vectors	41.27
2D code vector + Synthetic	41.54
Only real openSMILE	45.14
Only improved-conditional	33.96
Real openSMILE + baseline-conditional	43.79
Real openSMILE + improved-conditional	**45.40**

5. Conclusions

It is encouraging to observe that even with smaller datasets, the adversarial errors of a GAN can be made to converge. With more data it is expected that GANs will be able to learn a more generalized distribution/manifold where the openSMILE feature vectors lie. The experiments on conditional GAN show that a generator's job to estimate a more complex PDF from a simpler PDF is more complex than a discriminator's job which is to distinguish between fake and real samples. Hence, we had to incorporate tricks like updating the generator more times for a single update of discriminator or keeping the learning rate of generator more than that of a discriminator. We also experimented with reducing the number of trainable parameters in a discriminator but it didn't help in this case by a larger amount. While we see an improvement in performance of SVM when real data is appended with synthetic data, however the improvement isn't much. This is probably because the synthetic vectors after all are sampled from a distribution that mimics the real data distribution, something which the SVM classifier is already using for training. Also the smaller size of dataset might be hampering the capabilities of our GAN models. Cross corpus results showing similar trend as cross-validation indicate that the models are indeed generalizable across datasets with different priors.

In the future, we aim to further analyze other GAN architectures for the task of emotion classification [15]. A similar application of GANs could also be extended to other tasks within the study of emotion classification [16], as well as to tasks such as psychotherapy [17] and medicine [18].

6. References

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

[2] 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.

[3] A. Radford, L. Metz, and S. Chintala, “Unsupervised representation learning with deep convolutional generative adversarial networks,” *arXiv preprint arXiv:1511.06434*, 2015.

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

[5] J. Chang and S. Scherer, “Learning representations of emotional speech with deep convolutional generative adversarial networks,” in *Acoustics, Speech and Signal Processing (ICASSP), 2017 IEEE International Conference on*. IEEE, 2017, pp. 2746–2750.

[6] M. Mirza and S. Osindero, “Conditional generative adversarial nets,” *arXiv preprint arXiv:1411.1784*, 2014.

[7] X. Wang and A. Gupta, “Generative image modeling using style and structure adversarial networks,” in *European Conference on Computer Vision*. Springer, 2016, pp. 318–335.

[8] P. Isola, J.-Y. Zhu, T. Zhou, and A. A. Efros, “Image-to-image translation with conditional adversarial networks,” *arXiv preprint*, 2017.

[9] J. Li, W. Monroe, T. Shi, S. Jean, A. Ritter, and D. Jurafsky, “Adversarial learning for neural dialogue generation,” *arXiv preprint arXiv:1701.06547*, 2017.

[10] 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.

[11] Y. Kim and E. M. Provost, “Emotion recognition during speech using dynamics of multiple regions of the face,” *ACM Transactions on Multimedia Computing, Communications, and Applications (TOMM)*, vol. 12, no. 1s, pp. 25, 2015.

[12] 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*. ACM, 2013, pp. 835–838.

[13] F. Eyben, F. Weninger, M. Wöllmer, and B. Schuller, “open-source media interpretation by large feature-space extraction,” .

[14] C. Busso, S. Parthasarathy, A. Burmania, M. AbdelWahab, N. Sadoughi, and E. M. Provost, “Msp-improv: An acted corpus of dyadic interactions to study emotion perception,” *IEEE Transactions on Affective Computing*, vol. 8, no. 1, pp. 67–80, 2017.

[15] I. Goodfellow, “Nips 2016 tutorial: Generative adversarial networks,” *arXiv preprint arXiv:1701.00160*, 2016.

[16] R. Gupta, C.-C. Lee, and S. Narayanan, “Classification of emotional content of sighs in dyadic human interactions,” in *Acoustics, Speech and Signal Processing (ICASSP), 2012 IEEE International Conference on*. IEEE, 2012, pp. 2265–2268.

[17] R. Gupta, P. G. Georgiou, D. C. Atkins, and S. S. Narayanan, “Predicting client’s inclination towards target behavior change in motivational interviewing and investigating the role of laughter,” in *Fifteenth Annual Conference of the International Speech Communication Association*, 2014.

[18] R. Gupta and S. S. Narayanan, “Predicting affective dimensions based on self assessed depression severity.,” in *INTERSPEECH*, 2016, pp. 1427–1431.