 doi:10.20944/preprints202104.0651.v1

Speech Emotion Recognition using Data Augmentation Method by Cycle-Generative Adversarial Networks

Arash Shilandari [1, *], Hossein Marvi [2] and Hossein Khosravi [3]

[1, *] Department of Electrical and Computer Engineering, Ph.D. Student of Shahrood University of Technology, Shahrood, Iran; Shilandari@shahroodut.ac.ir
[2] Department of Electrical and Computer Engineering, Associate Professor of Shahrood University of Technology, Shahrood, Iran; h.marvi@shahroodut.ac.ir
[3] Department of Electrical and Computer Engineering, Associate Professor of Shahrood University of Technology, Shahrood, Iran; hosseinkhosravi@shahroodut.ac.ir

Abstract: Nowadays, and with the mechanization of life, speech processing has become so crucial for the interaction between humans and machines. Deep neural networks require a database with enough data for training. The more features are extracted from the speech signal, the more samples are needed to train these networks. Adequate training of these networks can be ensured when there is access to sufficient and varied data in each class. If there is not enough data; it is possible to use data augmentation methods to obtain a database with enough samples. One of the obstacles to developing speech emotion recognition systems is the Data sparsity problem in each class for neural network training. The current study has focused on making a cycle generative adversarial network for data augmentation in a system for speech emotion recognition. For each of the five emotions employed, an adversarial generating network is designed to generate data that is very similar to the main data in that class, as well as differentiate the emotions of the other classes. These networks are taught in an adversarial way to produce feature vectors like each class in the space of the main feature, and then they add to the training sets existing in the database to train the classifier network. Instead of using the common cross-entropy error to train generative adversarial networks and to remove the vanishing gradient problem, Wasserstein Divergence has been used to produce high-quality artificial samples. The suggested network has been tested to be applied for speech emotion recognition using EMODB as training, testing, and evaluating sets, and the quality of artificial data evaluated using two Support Vector Machine (SVM) and Deep Neural Network (DNN) classifiers. Moreover, it has been revealed that extracting and reproducing high-level features from acoustic features, speech emotion recognition with separating five primary emotions has been done with acceptable accuracy.

Keywords: speech processing, data augmentation, speech emotion recognition, generative adversarial networks

1. Introduction

The Data sparsity problem is known as one of the critical challenges in speech emotion recognition systems, which can be examined from three aspects: 1- The first problem is the unreality of emotions in emotion databases. Often these samples are recorded by professional actors and do not contain real emotions. This is because of legal and moral issues. [1]. 2- Another essential matter is annotation. Since the expressed emotions are different, annotation is always necessary. Annotation means an auxiliary instrument that helps guess or understand the emotion of the speaker through his speech. To analyze emotions, two discrete and continuous models are used. Some limited labels are used to index different

Preprints (www.preprints.org) | NOT PEER-REVIEWED | Posted: 26 April 2021 doi:10.20944/preprints202104.0651.v1

emotions in the discrete emotion model. For example, in EMODB which is based on the emotion discrete model, one label corresponding to one of the seven emotions of angry, happy, unhappy, fear, hate, tiredness, and neutral has been allocated to each sentence. The limited number of labels in the discrete model causes problems in expressing different emotions. For example, when a sentence is recognized with a happy or fearful label, the severity of these emotions is not known in these labels. Furthermore, the number of speeches with neutral emotions is the most in the sentences of a speech [1]. However, a balanced information bank is needed to train an emotion classifier network better.

DNNs require a wealth of data for training to achieve acceptable performance. Data amplification is a common way to increase the size of training sets, but data amplification in a classical way is only for specific tasks. Some standard data amplification techniques in processing images like transfer and rotation [2] are not used for processing text or speech. Synonymous substitution [3], which is mainly used to process text, is difficult for classifying and recognizing emotion from speech. Similarly, traditional data reinforcement methods for a speech like change in voice and change in acoustic signal velocity [4] are also inappropriate for images or texts. In contrast, the data augmentation method based on generative adversarial networks is focused on learning and simulating real data distribution and is independent of the duties and so an experienced taken from one work may also be used in other works.

Recent studies based on end-to-end and automatic methods (feature extraction and connected classification) are used for speech emotion recognition [5]-[6]. The input in these systems is feature vectors and the output is class labels. In [7], The features extracted by convolution filters.

With the development of DNNs in speech emotion recognition, various data augmentation methods have been explored [8]-[9]. Transfer learning Seems like a solution to the data sparsity problem [10]. The success of this method in image processing led researchers to use this method in speech processing [11]. Dang and colleagues proposed a feature-learning transfer method in which source domain data was transmitted to the target domain [8].

One of the effective methods to reinforce and augment data is the generative adversarial network introduced by Goodfellow and colleagues in 2014 [12]. Today, generative adversarial networks are recognized as a successful technique for increasing data. These networks have three main characteristics [12]: 1- they learn well the probability distribution in the complicated problems of the real world. 2- They are also taught by noisy and without label data. 3- They enjoy multinodular outputs; that is, they can produce several correct and different answers for a problem and increase the diversity of the produced samples. These networks consist of a generator network and a discriminator network (both are deep neural networks) and compete with each other. The generator network learns the desired pattern of the data and creates fake data to confuse the discriminator network, and the discriminator network is trained to determine if an imported sample of the original data distribution exists in the database. Data augmentation techniques based on generative adversarial networks help improve image recognition performance [13]. Zhang and colleagues introduced a generative adversarial network to produce high-dimensional data and showed that data augmentation by generative adversarial network acts better than the typical data augmentation techniques [14].

The present study has represented a cycle generative adversarial network for data augmentation existing in EMODB and then two classifier networks for speech emotion recognition. This network produces samples like actual data and provides a database with more samples to train the emotion classifier network. Also, the effectiveness of the generative adversarial network will be discussed, and standard cross-entropy error will be substituted with Wasserstein Divergence to train generative adversarial network. EMODB was used to do experiments, and data were analyzed. The

Preprints (www.preprints.org) | NOT PEER-REVIEWED | Posted: 26 April 2021 doi:10.20944/preprints202104.0651.v1

results show that the suggested method of cycle generative adversarial network in this study can be used for improving the performance of a speech emotion recognition system in EMODB [15].

Section 2 reviews standard solutions for the data sparsity problem. Section 3 describes the suggested network design and represents theoretical analysis. Section 4 introduces experiment details, including data description, features, experimental regulations, and evaluation protocols. Section 5 represents and analyzes experimental results. Finally, section 6 represents the conclusion and future works.

2. The Related Works

2.1. Related work done

Data sparsity in each class or imbalance database may prevent a deep neural network from being able to learn the distribution of data, or overfitting happens. Regularization can be an effective technique to solve [16]. The following solution is to reduce the size and limit the scatter in the data [17]. Nevertheless, this method is suitable when there are many features in the data because it may remove helpful information from the data.

When we have a data sparsity problem with a lack of data in the database, we can expand the database with data augmentation methods. Other methods have usually changed primary data and cause problems like rotation, adding noise, echoing to speech, and clipping signals [18]. Advanced data augmentation methods based on generative adversarial networks and their types are conditional generative adversarial network and or cycle generative adversarial network. Hu and colleagues used a deep convolutional neural network to produce extra features to train acoustic models and understood that data augmentation helps speech recognition systems a lot [19]. Sahu and colleagues [20] synthesized feature vectors using automatic adversarial encoders using Gaussian mixed noise in the generator network. They showed that the synthesized samples increase the performance of the classifier network, but the generated samples tend to follow the desired distribution instead of following the distribution of the database data. Sahu and colleagues [9] also made a model based on a conditional generative adversarial network to generate artificial feature vectors. Several tricks, including generator initialization with detector weights and automatic adversarial encoder and several times generator weights updating before updating detector network weights in each training course, were applied to train conditional generative adversarial network.

One fundamental problem in training generative adversarial networks is making sure of balance in training generator and detector. dynamic alternating training [14] can be used so that the number of training epochs in the generator network and the discriminator network can not be equal. The ultimate goal is not the number of training epochs but the amount of training in each network. For example, the generator network can be trained three times, but the discriminator network can be trained once in each epoch. Instead of learning a predefined distribution, the detector network in the suggested cycle generative adversarial network in this study learns the distribution of data produced by the generator simultaneously, and both networks grow together.

2.2. Generative Adversarial Network

As mentioned before, generative adversarial networks consist of two deep neural networks. The generator network produces fake data, and the discriminator network separates the accurate data from the fake data. The general purpose of generative adversarial networks can be expressed as follows [21]:

$$\min_{G}\max_{D} V(D, G)=\mathop{E}_{x\sim p_{data}(x)}[\log D(x)]+\mathop{E}_{z\sim p_z(z)}[\log (1-D (G(z)))] \quad (1)$$

Preprints (www.preprints.org) | NOT PEER-REVIEWED | Posted: 26 April 2021 doi:10.20944/preprints202104.0651.v1

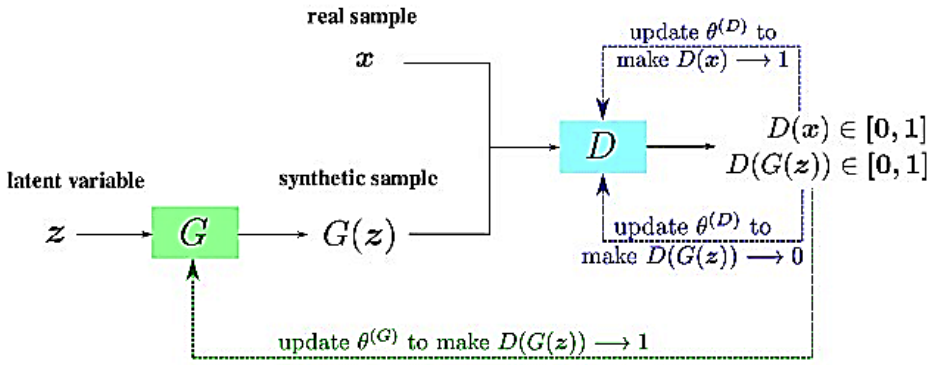


Figure 1: the structure of a generative adversarial network

Practically, according to [21], G is trained to maximize log D(x), instead of training G to minimize log (1-D (G(z))). This objective function may produce a stronger gradient which reduces the vanishing gradient problem without overcoming this equilibrium point of G and D.

$$J^{(D)}(D, G) = -\underset{x \sim p_{data}(x)}{E}[\log D(x)] - \underset{z \sim p_z(z)}{E}[\log(1 - D(G(z)))] \quad (2)$$

$$J^{(G)}(G) = -\underset{z \sim p_z(z)}{E}[\log(D(G(z)))] \quad (3)$$

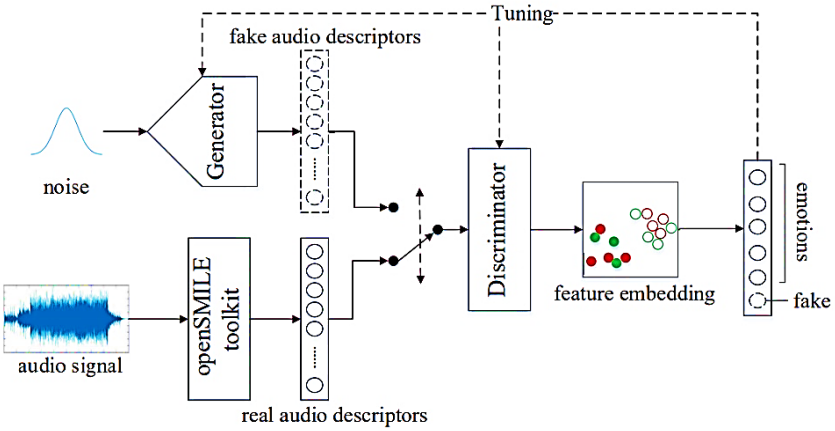


Figure 2: Diagram of a speech emotion generative adversarial network

According to figure 2, during training these networks, The initial weights are randomly selected, and both networks are trained in competition with each other. However, the network can be pre-trained and use better weights to get started. First, a batch of the training set and a batch of generator output are taken, and the weights of the discriminator are updated using them. Then, the weights of the discriminator are locked, and a batch of generator output is given to the discriminator and this network updates generator weights in the backpropagation method, and this process continues. The entire process of training a generative adversarial network is shown in algorithm 1:

Algorithm 1. training a generative adversarial network in vanishing gradient method

Preprints (www.preprints.org) | NOT PEER-REVIEWED | Posted: 26 April 2021 doi:10.20944/preprints202104.0651.v1

Repeated for the number of training repetitions:

Repeated for the number of k:

Sampled for the number m of the initial noise space pg(z).

$z = \{z^{(1)}, z^{(2)}, z^{(3)}, \ldots, z^{(m)}\}$

Sampled for m number of data initial distribution p data.

$x = \{x^{(1)}, x^{(2)}, x^{(3)}, \ldots, x^{(m)}\}$

By calculating the gradient, the loss of the discriminator network is calculated:

$$\nabla_{\theta_d} \frac{1}{m} \sum_i \left[\log D\left(x^{(i)}\right) + \log\left(1 - D(G(z^{(i)}))\right)\right]$$

The end of the second loop

Sampled for the number m of the initial noise space pg(z).

$z = \{z^{(1)}, z^{(2)}, z^{(3)}, \ldots, z^{(m)}\}$

Generator weights are updated in the gradient descent method as follows:

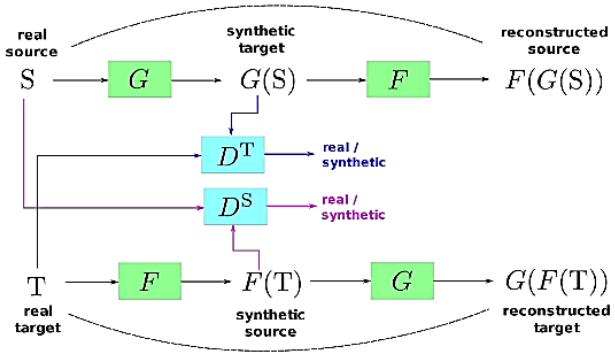
$$\nabla_{\theta_g} \frac{1}{m} \sum_i \left[\log\left(1 - D(G(z^{(i)}))\right)\right]$$

The end of the first loop

2.3. Generative Adversarial Network

Cycle generative adversarial networks are known as a successful method of image-to-image transfer for non-paired databases. For example, grey to colored, image to semantic label, etc. Image transfer which is learned by a generative adversarial network may record the features of transfer from one set of images and recognizes how to use these features for another image set transfer [22]. The great success of these networks in image transfer has made the researchers use them for emotional data augmentation.

Figure 3 shows the architecture of a data augmentation cycle adversarial network. This network includes two transfer functions F, and G. G learns how to transfer samples from one source S to target source T. F is a structure unlike G. both transfer functions F and G may be considered as a generator to produce target data and also to produce source data. Moreover, two adversarial discriminator networks DT and DS, exist, which are targeted as an enemy against G in data generation. DT discriminates real target from the artificial target, and DS discriminates real source from an artificial source. The cycle generative adversarial network can return the generated samples to the original samples. This network sets its target so that $F(G(\mathrm{S})) \approx \mathrm{S}$ and $G(F(\mathrm{T})) \approx \mathrm{T}$, and so it is called cycle-GAN [23].



Preprints (www.preprints.org) | NOT PEER-REVIEWED | Posted: 26 April 2021 doi:10.20944/preprints202104.0651.v1

Figure 3: the structure of a cycle generative adversarial network

Cycle adversarial network losses include losing opposite network adversity and, as a result, data overfitting and losing cycle consistency. Removing adversity may be transformed into a part of target data production and a part of source data production. The loss function for target data production is as follows [23]:

$$L^{GAN}(G, D^T, S, T)=\underset{t\sim p_t}{E}[\log D^T(t)]+\underset{s\sim p_x}{E}[\log(1-D^T(G(s))] \quad (4)$$

Losses are expressed as value functions. So, in the production process, the goal is Min G Max DT $\mathcal{L}$GAN(G, DT, S, T), and to reproduce real data, the objective is Min F Max DS $\mathcal{L}$GAN(F, DS, T, S).

Transmission functions in deep neural networks due to the large amounts of parameters are not unique. Zou and colleagues have defined cycle losing as follows [23]:

$$L^{cyc}(G, F)=\underset{t\sim p_t}{E}[\|(G(F(t))-t)\|_1]+\underset{s\sim p_x}{E}[\|F(G(s))-s\|_1] \quad (5)$$

Since they have mentioned that L1 may be substituted with other criteria in these losses, total losses for cycle generative adversarial network are as follows:

$$L(G, F, D^T, D^S)=L^{GAN}(G, D^T, S, T)+L^{GAN}(F,D^S, T, S)+\lambda L^{cyc}(G, F) \quad (6)$$

Where λ controls the relative importance of both losses [23].

3. Methodology

3.1. The Suggested Method

For a dataset with X label and N emotional class, artificial samples are made for each emotion I using a cycle generative adversarial network. According to figure 4, cycle generative adversarial network transfers between one source S and one target domain Ti, where S is a dataset without label and Ti shows emotional samples in the labeled dataset. Discriminator networks DTi and DSi are used to produce artificial target which is not recognizable from real samples. Generator loss and discriminator loss are introduced by $\mathcal{L}$GANi (Gi, Ti, S, Ti) and $\mathcal{L}$GANi (Fi, DTi, S, Ti), respectively and:

$$L_i^{GAN}(G_i, F_i, D_i^T, D_i^S, S, T_i)=L_i^{GAN}(G_i, D_i^T, S, T_i)+L_i^{GAN}(F_i, D_i^S, S, T_i) \quad (7)$$

As mentioned before, generators try to minimize it while discriminators try to maximize it.

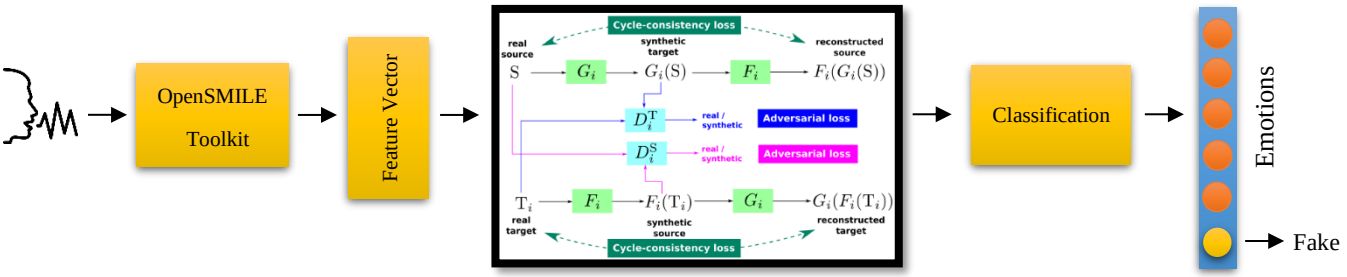


Figure 4: the architecture of the suggested structure

Moreover, the cycle generative adversarial network regulates the end of its cycle by regulating the loss function, which is considered for that. To do so, artificial target Gi(S) is taken back to primary data, and Mean Squared Error (MSE) is calculated between real data S and reconstructed data Fi (Gi(S)). This is similarly done for Ti and reconstructed target data (Gi(S)). As a result, the total loss function to exit from the cycle will be as follows:

Preprints (www.preprints.org) | NOT PEER-REVIEWED | Posted: 26 April 2021 doi:10.20944/preprints202104.0651.v1

$$L_i^{cyc}(G_i, F_i, S, T_i)=\underset{s\sim p_x}{E}[||(F_i(G_i(s))-s)||_2^2]+\underset{t\sim p_t}{E}[||G_i(F_i(t))-t)||_2^2] \quad (8)$$

Since cycle generative adversarial network learns data image as one to one function between a noise source and a target sample, it is necessary to generate vector transfer between each pair of them that is N(N-1)/2 for each speech dataset with N emotional class, but this is too much and complicates the calculations. This study has used labeled data of each emotional class as target data domain while source data domain is an extensive unlabeled dataset. This cycle generative adversarial network images real data and its target data to artificial data and one artificial database. Therefore, an artificial target database is generated that is the size of a real database with the same emotions. This artificial database is used to merge with a real database to increase its data. Instead of training the N cycle generative adversarial network separately, these networks are placed in a complete framework, and the samples produced by each emotion are related to each other. This framework will be explained later.

It should be noted that synthetic samples are generated in the feature space and feature vectors extracted by OpenSMILE software [24]. Its advantage is that regardless of speech synthesis, studies have been focused on simulating data distribution through cycle generative adversarial network, and its disadvantage is that the produced emotions are devoid of a prominent figure to evaluate humans perceptually. Nevertheless, it is still possible to test their emotional features compared with their similarity with actual data samples.

3.2. Overcoming Gradient Descent Problem in Training Cycle Generative Adversarial Networks Process

Cycle adversarial data augmentation using Wasserstein Distance has been suggested in this study to overcome gradient vanishing and gradient descent problems. Extreme gradient descent practically stops the process of weight modification and training generators and discriminators. Considering two probability distributions Pr and Pg, Wasserstein Distance is defined as follows:

$$W_1(\mathbb{P}_r, \mathbb{P}_g) = \sup_{\|f\|_L\leq 1} \underset{x\sim\mathbb{P}_r}{\mathbb{E}}\{f(x)\} - \underset{\tilde{x}\sim\mathbb{P}_g}{\mathbb{E}}\{f(\tilde{x})\} \quad (9)$$

Where fL≤1 shows that f satisfies the 1-Lipschitz limitation. If weights are more or less than the expected limit in the weight clipping method, they will be changed into minimum or maximum of a specific value, and in gradient penalty method, gradient penalty is based on Lipschitz, which derived from this fact that if gradients are at most 1 everywhere, they are 1-Lipschitz functions. Their square difference from one is used as a gradient penalty. According to [25], weight clipping may lead to a non-optimal solution. Gradient penalty was also applied to overcome weight clipping limitations [26]. However, if there is a data sparsity problem, the satisfying k- Lipschitz limitation is difficult for the whole data domain. Accordingly, Wu and colleagues [25] suggested a new divergence for Wasserstein Divergence, which can calculate Wasserstein Distance without applying Lipschitz as follows:

$$L_D = \underset{x\sim P_r}{E}\{f(x)\} - \underset{\tilde{x}\sim P_g}{E}\{f(\tilde{x})\} + \lambda \underset{\tilde{x}\sim P_u}{E}[\|\nabla f(\tilde{x})\|^p] \quad (10)$$

Where λ controls the effect of gradient modification on target function, Pu is measuring Radon probability, and p is related to Lp space for function f. Also, as mentioned in [25], it must $\lambda > 0$ and $p > 1$ for LDIV to be symmetric divergence. Finally, the loss function in generator and discriminator is as follows:

$$\mathcal{L}_G^{(WC-GAN)} = \mathbb{E}_{p(x,y,z)}\left\{D(G(z,y)) - a\sum_{k=1}^{K} y_{emo}^{(k)} \log C(G(z,y))_k\right\} \quad (11)$$

Preprints (www.preprints.org) | NOT PEER-REVIEWED | Posted: 26 April 2021 doi:10.20944/preprints202104.0651.v1

$$\mathcal{L}_D^{(WC-GAN)} = \mathbb{E}_{p(x,z,\tilde{x},y)}\{D(E(x) - D(G(z,y)) + \lambda[\|\nabla_{\tilde{x}}\|^p]\} \quad (12)$$

The structure of the cycle generative adversarial network in the Wasserstein method is like in figure 3. The difference between these two networks is that the discriminator uses Sigmoid Activation Function, while Wasserstein cycle adversarial data augmentation uses the linear activation function in the final layer.

3.3. The Advantages of Cycle Generative Adversarial Network

Cycle adversarial data augmentation networks use Jensen-Shannon Divergence as a divergence criterion. According to [27], if two data distributions are less overlapped and or they are not overlapped, Jensen-Shannon Divergence will be constant, which leads to a gradient vanishing problem. The method proposed in this study can solve this problem. In the first training, S and T distribution are much overlapped, which makes problems for the discriminator to separate and discriminate these two vector groups; therefore, the discriminator network faces many cross-entropy errors, and the generator network receives a gradient error.

Moreover, adversarial data augmentation networks can easily use other divergence methods like Wasserstein divergence for gradient descent. In comparison with Jensen-Shannon Divergence, the advantage of Wasserstein Divergence is that even if data is not overlapped with each other, it may measure the distance between two data distributions. The hidden space generated by adversarial data augmentation networks also makes learning emotional information more straightforward and more accessible due to vectors' lower dimensions. Additionally, practical programs [26]-[28] have shown that models produced by Wasserstein Divergence are better than other divergence models like Jensen-Shannon Divergence and maximum mean discrepancy. Therefore, it seems that Wasserstein adversarial data augmentation network may produce more meaningful emotional vectors.

3.4. Recognizing between Samples Produced by Cycle Adversarial Data Augmentation Network

Figure 5 shows that imaging data by cycle adversarial data augmentation network causes similarity between real and artificial data distribution. a classification loss function is defined between fake data to make sure that it is correctly allocated to target emotions class. classification loss has been defined as cross-entropy error:

$$L^{cls} = -\sum_i y_i log\ (C(G_i(S))) \quad (13)$$

Where yi is the label of target emotions. The total loss is as follows:

$$L=\sum_i L_i^{GAN}+\lambda^{cyc}\sum_i L_i^{cyc}+\lambda^{cls}L^{cls} \quad (14)$$

λcyc and λcls parameters are weights to lose cycle and to lose classification.

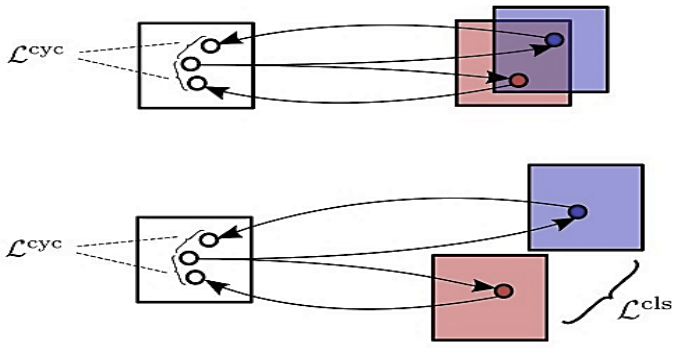


Figure 5: Difference between two mapping samples without losing classification and with losing classification

4. Experiments

Preprints (www.preprints.org) | NOT PEER-REVIEWED | Posted: 26 April 2021 doi:10.20944/preprints202104.0651.v1

4.1. Dataset

Experiments in this study have been done on EmoDB [29]. EmoBD is a small dataset including 800 sentences and has been divided into seven emotional classes. All speeches have been recorded by ten professional actors in German. This database includes emotions like anger, happiness, unhappiness, fear, hate, and tiredness and neutral that five emotions were used to perform the experiments.

4.2. Feature Extraction

As mentioned earlier, OpenSMILE, which is an open-source software to extract features from acoustic and speech signals, has been used to extract feature and form feature vectors [30]. These features have been defined in Paralinguistic Challenge Interspeech 2010 [31] and include 1582 features. Also, features with zero frequency are deleted, and other features are normalized independently by z-norm. Zhang and colleagues have demonstrated that [32] z normalization will improve error minimization for the classifier. Python, Keras Library, and Tensorflow have been used to train and test networks.

4.3. Regulations to Do Experiments

Since there are five emotions happy, unhappy, angry, fear, and neutral, for classification, the suggested model has consisted of five generators, five discriminators, and a classifier that is all implemented by forwarding neural networks. As it is challenging to train generators to learn considering expansive dimensions of feature vectors and their high distribution, both Gi and Fi generators pre-trained based on reconstruction error between S and Fi(Gi(S)) and also reconstruction error between Ti and Gi(Fi(Ti)).

As mentioned, DNN with two hidden layers and 800 hidden neurons was used in cycle adversarial data augmentation networks. Also, DNN and SVM networks were used as classifiers, and Leaky ReLU was applied to all layers. The linear kernel used in the SVM classifier. Also, Xavier Algorithm [33] and Adam Optimizer [34] with 0.0002 learning rate and reduced every 50 courses linearly with 0.8 coefficient used for DNN network initialization and training them, respectively. DNNs implemented using Tensorflow (v 2.1) in Python, while SVMs implemented using Scikit-Learn Package.

At first, the suggested model trained using five cycle generative adversarial networks in a parallel form with pre-trained weights for the generator. Table 1 shows the other parameters.

Table 1: pre-training and training parameters of cycle adversarial data augmentation network

Pre-training of cycle adversarial data augmentation network	
Layer size	[1582, 1000, 500, 1000,1582]
Number of epochs	5000
Dropout	0.2
Minibatch size	128
Cycle adversarial data augmentation network training	
Number of epochs	2000
Weight decay	0.8
Minibatch	128

Preprints (www.preprints.org) | NOT PEER-REVIEWED | Posted: 26 April 2021 doi:10.20944/preprints202104.0651.v1

To balance training G and D, generator weights were updated two times per epoch, and discriminator weights were updated one time per epoch. Moreover, unilateral label preprocessing [35] has been used.

4.4. Regulations to Do Experiments

There are ten speakers in EmoDB. Nine speakers and the rest of the data have been used in each training course to test the network. LOSO-CV shows that there has been no training data in data augmentation. Weighted accuracy and unweighted average recall have been used to compare performance as follows:

$$\text{UAR} = \frac{1}{K}\sum_{k=1}^{K}\frac{\text{true}-\text{positives}_k}{\text{total}-\text{positives}_k} \tag{15}$$

$$\text{WA} = \frac{\sum_{k=1}^{K}\text{true}-\text{positives}_k}{\sum_{k=1}^{K}\text{total}-\text{positives}_k} \tag{16}$$

5. Results

Since EmoBD is a small database, it is expected to be an extensive and powerful database after data augmentation. The augmented data were gradually and randomly added to the original data, and two DNN and SVM classifiers were used for speech emotion recognition. L2 regulation was used to train deep neural networks, and each experiment was repeated three times, and the mean performance was reported as the absolute accuracy. Figure 6 shows the results of the SVM and DNN classification in the EmoBD emotional database with real data.

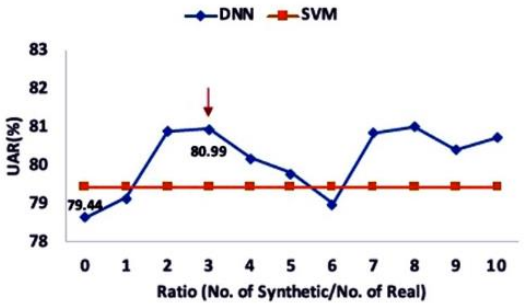


Figure 6: comparing data classifiers results with real samples

To highlight the effectiveness of the cycle adversarial data augmentation method, its performance compared with some standard data augmentation techniques like sample reproduction, add random noise to feature vectors and artificial sampling SMOTE [36].

Augmenting the primary data by adding some noise to the feature vectors is a way to generate new data for network training, but its success depends on the amount of data noise, and fluctuations in the result are always possible. Figure 7 shows this result.

Preprints (www.preprints.org) | NOT PEER-REVIEWED | Posted: 26 April 2021 doi:10.20944/preprints202104.0651.v1

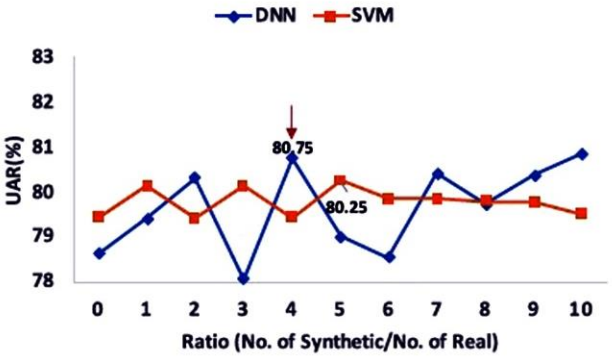


Figure 7: comparing data classification results with real data and samples augmented to primary data using Gaussian noise

Making fake data similar to primary samples helps deep neural networks learn data distribution better, but repetitive samples will not lead to network better training. The SMOTE method is designed to augment samples in one class and can not be used to augment samples in all classes and has a relatively stable performance [36]. Figure 8 shows the results of this method.

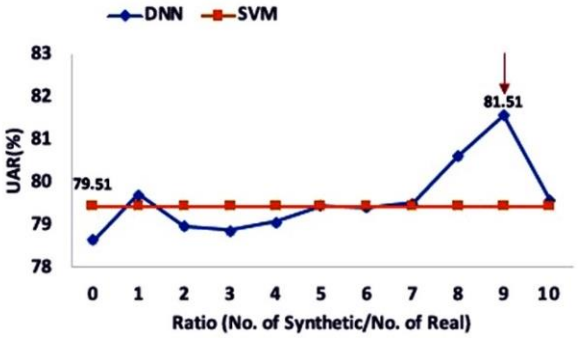


Figure 8: comparing data classification results with SMOTE method

The method based on cycle adversarial data augmentation can train dynamic classifiers better by adding more artificial data to the training set. Surprisingly, the cycle adversarial data augmentation method may lead to the improvement of SVM performance. The results show that augmenting artificial data in this method helps SVM better recognize metadata in feature space and classify them with better performance. Figure 9 shows the performance of two classifiers by combing read and augmented data based on a cycle generative adversarial network.

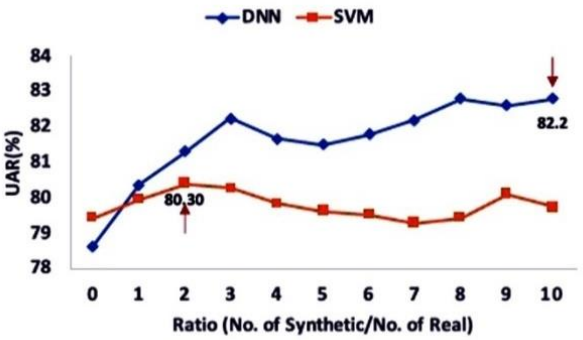


Figure 9: comparing data classification results with cycle generative adversarial network

According to figure 10, it is possible to improve performance by data augmentation approach based on Wasserstein Distance introduced in sections 3-5. The unweighted average recall is gradually augmented by adding artificial samples

Preprints (www.preprints.org) | NOT PEER-REVIEWED | Posted: 26 April 2021 doi:10.20944/preprints202104.0651.v1

to the training set. In decimal data validation, when the number of augmented sets is more than five, one of them achieves 100% outstanding results. These results show that data augmentation based on cycle generative adversarial network may generate new and meaningful emotional vectors which help the performance of emotion recognition classifier.

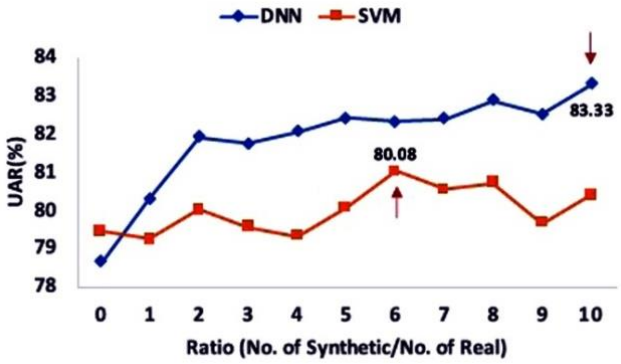


Figure 10: comparing data classification results with data augmentation based on cycle generative adversarial network and Wasserstein Distance

Table 2 shows the highest percent of WA and UAR with different methods. These results have been achieved using different values of augmented data. This table shows that by augmenting data based on the cycle adversarial data augmentation method, the classifier network is better trained for emotion recognition. The results are better than [37] shown by the SVM using the handmade features for speech emotion recognition. Unweighted Average Recall is higher in this method than Chen and colleagues [38] that used 3D CRNNs to produce features.

Table 2: comparing the results of different data augmentation and speech emotion recognition techniques

Method	Classifier	WA%	UAR%
Add noise	DNN	82.06	80.75
Add noise	SVM	81.12	80.25
SMOTE	DNN	82.43	81.51
SMOTE	SVM	80.83	79.51
Cycle generative adversarial network	DNN	83.55	82.50
Cycle generative adversarial network	SVM	81.50	80.30
Cycle generative adversarial network + Wasserstein Distance	DNN	84.49	**83.33**
Cycle generative adversarial network + Wasserstein Distance	SVM	81.07	80.08
2D-ACRNN [38]	DNN		79.38
3D-ACRNN [38]	DNN		82.82

6. Conclusion

Data sparsity is a critical problem in the training of deep neural networks and causes the speech emotion recognition system not to achieve acceptable results in applications. Typically, sparse data for training leads to overfitting and network structure complications. This study presents a new network for data augmentation to produce artificial samples in EmoBD, which places the generated samples in the primary data space. Instead of vectors containing

emotion features in space with high dimensions, the suggested method, by producing synthetic samples, creates a cloud of artificial data in the space of each emotional class that completely covers the leading data space. Also, the results showed that this method could overcome the Vanishing Gradient Problem during the training process and make the training process continue intelligently. The results showed that the added samples improved the function of speech emotion recognizing and included in the space of actual samples. Additionally, the Wasserstein loss function was added to the network architecture to train cycle adversarial data augmentation network and showed that the produced artificial samples would be more separable by the classifier.

This study only investigated a simple item where the data augmentation network is emotional vectors extracted by OpenSMILE. However, the suggested model still has some problems. For example, the produced samples are similar to the samples used in the training network and follow them. As a result, if the test data distribution is different from training data distribution, the augmented data will not be helpful. Future researches are needs to use a method to generalize this method.

References

1. M. El Ayadi, M. S. Kamel, F. Karray. "Survey on speech emotion recognition: Features, classification schemes, and databases." in: Pattern Recognition 44.3 (2011), pp. 572–587.
2. J. Wang, L. Perez. "The effectiveness of data augmentation in image classification using deep learning." in: Convolutional Neural Networks Vis. Recognit (2017).
3. X. Zhang, Y. LeCun. "Text understanding from scratch." in: arXiv preprint arXiv:1502.01710 (2015).
4. T. Ko, V. Peddinti, D. Povey, S. Khudanpur. "Audio augmentation for speech recognition." in: Sixteenth Annual Conference of the International Speech Communication Association. 2015.
5. G. Trigeorgis et al., "Adieu features? End-to-end speech emotion recognition using a deep convolutional recurrent network," in Proc. IEEE Int. Conf. Acoust., Speech Signal Process. (ICASSP), Mar. 2016, pp. 5200–5204.
6. 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, Sep. 2018, pp. 3683–3687.
7. P. Li, Y. Song, I. McLoughlin, W. Guo, and L. Dai, "An attention pooling based representation learning method for speech emotion recognition," in Proc. Interspeech, Sep. 2018, pp. 3087–3091.
8. J. Deng, Z. Zhang, E. Marchi, and B. Schuller, "Sparse autoencoder based feature transfer learning for speech emotion recognition," in Proc. Humaine Assoc. Conf. Affect. Comput. Intell. Interact., Sep. 2013, pp. 511–516.
9. S. Sahu, R. Gupta, and C. Espy-Wilson, "On enhancing speech emotion recognition using generative adversarial networks," 2018, arXiv:1806.06626. [Online]. Available: http://arxiv.org/abs/1806.06626
10. S. J. Pan and Q. Yang, "A survey on transfer learning," IEEE Trans. Knowl. Data Eng., vol. 22, no. 10, pp. 1345–1359, Oct. 2010.
11. M. Wang and W. Deng, "Deep visual domain adaptation: A survey," Neurocomputing, vol. 312, pp. 135–153, Oct. 2018.
12. I. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. Courville, Y. Bengio. "Generative adversarial nets." in: Advances in neural information processing systems. 2014.
13. A. Antoniou, A. Storkey, and H. Edwards, "Data augmentation generative adversarial networks," 2017, arXiv:1711.04340. [Online]. Available: http://arxiv.org/abs/1711.04340
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 J. Biomed. Health Informat., vol. 24, no. 1, pp. 300–310, Jan. 2020.
15. F. Burkhardt, A. Paeschke, M. Rolfes, W. F. Sendlmeier, and B. Weiss, "A database of German emotional speech," in Proc. 9th Eur. Conf. Speech Commun. Technol., 2005, pp. 1–4.
16. S. N. Negahban, P. Ravikumar, M. J. Wainwright, and B. Yu, "A unified framework for high-dimensional analysis of M-estimators with decomposable regularizers," Stat. Sci., vol. 27, no. 4, pp. 538–557, Nov. 2012.
17. H. Zou, T. Hastie, and R. Tibshirani, "Sparse principal component analysis," J. Comput. Graph. Statist., vol. 15, no. 2, pp. 265–286, 2006.
18. T. DeVries and G. W. Taylor, "Dataset augmentation in feature space," 2017, arXiv:1702.05538. [Online]. Available: https://arxiv.org/abs/1702.05538
19. H. Hu, T. Tan, and Y. Qian, "Generative adversarial network-based data augmentation for noise-robust speech recognition," in Proc. IEEE Int. Conf. Acoust., Speech Signal Process. (ICASSP), Apr. 2018, pp. 5044–5048.
20. S. Sahu, R. Gupta, G. Sivaraman, W. Abdalmageed, and C. Espy-Wilson, "Adversarial auto-encoders for speech-based emotion recognition," in Proc. Interspeech, Aug. 2017, pp. 1243–1247.
21. I. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. Courville, and Y. Bengio, "Generative adversarial nets," in Proc. Adv. Neural Inf. Process. Syst., 2014, pp. 2672–2680.

22. M. El Ayadi, M. S. Kamel, F. Karray. “Survey on speech emotion recognition: Features, classification schemes, and databases.” in: Pattern Recognition 44.3 (2011), pp. 572–587.
23. J.-Y. Zhu, T. Park, P. Isola, A. A. Efros. “Unpaired image-to-image translation using cycle-consistent adversarial networks.” in: Proceedings of the IEEE International Conference on Computer Vision. 2017, pp. 2223–2232.
24. F. Eyben, F. Weninger, F. Gross, B. Schuller. “Recent developments in openSMILE, the Munich open-source multimedia feature extractor”. In: Proc. of the21st ACM international conference on Multimedia. ACM. 2013.
25. J. Wu, Z. Huang, J. Thoma, D. Acharya, and L. Van Gool, “Wasserstein divergence for GANs,” in Proc. Eur. Conf. Comput. Vis. (ECCV), Sep. 2018, pp. 653–668.
26. I. Gulrajani, F. Ahmed, M. Arjovsky, V. Dumoulin, and A. C. Courville, “Improved training of Wasserstein GANs,” in Proc. Adv. Neural Inf. Process. Syst., I. Guyon, U. V. Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, and R. Garnett, Eds. New York, NY, USA: Curran Associates, 2017, pp. 5767–5777.
27. M. Arjovsky, S. Chintala, and L. Bottou, “Wasserstein generative adversarial networks,” in Proc. Int. Conf. Mach. Learn., Aug. 2017, pp. 214–223.
28. J. Shen, Y. Qu, W. Zhang, and Y. Yu, “Wasserstein distance guided representation learning for domain adaptation,” in Proc. AAAI Conf. Artif. Intell. 2018, pp. 4058–4065.
29. F. Burkhardt, A. Paeschke, M. Rolfes, W. F. Sendlmeier, and B. Weiss, “A database of German emotional speech,” in Proc. 9th Eur. Conf. Speech Commun. Technol., 2005, pp. 1–4.
30. F. Eyben, F. Weninger, F. Gross, B. Schuller. “Recent developments in openSMILE, the Munich open-source multimedia feature extractor.” In: Proc. of the 21st ACM international conference on Multimedia. ACM. 2013.
31. B. Schuller, S. Steidl, A. Batliner, F. Burkhardt, L. Devillers, C. A. Müller, S. S. Narayanan, et al. “The InterSpeech 2010 paralinguistic challenge.” In: InterSpeech. Vol. 2010. 2010.
32. Z. Zhang, F. Weninger, M. Wöllmer, B. Schuller. “Unsupervised learning in cross-corpus acoustic emotion recognition.” in: 2011 IEEE Workshop on Automatic Speech Recognition & Understanding. IEEE. 2011, pp. 523–528.
33. X. Glorot and Y. Bengio, “Understanding the difficulty of training deep feedforward neural networks,” in Proc. 13th Int. Conf. Artif. Intell. Statist., 2010, pp. 249–256.
34. D. P. Kingma and J. L. Ba, “Adam: A method for stochastic optimization,” in Proc. 3rd Int. Conf. Learn. Represent. (ICLR), 2015, pp.1–15.
35. T. Salimans, I. Goodfellow, W. Zaremba, V. Cheung, A. Radford, X. Chen. “Improved techniques for training GANs.” in: Advances in Neural Information Processing Systems.2016.
36. N. V. Chawla, K. W. Bowyer, L. O. Hall, and W. P. Kegelmeyer, “SMOTE: Synthetic minority over-sampling technique,” J. Artif. Intell. Res., vol. 16, pp. 321–357, Jun. 2002.
37. I. Luengo, E. Navas, and I. Hernaez, “Feature analysis and evaluation for automatic emotion identification in speech,” IEEE Trans. Multimedia, vol. 12, no. 6, pp. 490–501, Oct. 2010.
38. M. Chen, X. He, J. Yang, and H. Zhang, “3-D convolutional recurrent neural networks with attention model for speech emotion recognition,” IEEE Signal Process. Lett., vol. 25, no. 10, pp. 1440–1444, Oct. 2018.