Institut für Maschinelle Sprachverarbeitung

Universität Stuttgart
Pfaffenwaldring 5B
D–70569 Stuttgart

Masterarbeit

Improving Speech Emotion Recognition via Generative Adversarial Networks

Fang Bao

Studiengang:	Informatik
Prüfer/in:	Prof. Dr. Ngoc Thang Vu & Dr. Antje Schweitzer
Betreuer/in:	Michael Neumann M.Sc. & Prof. Dr. Ngoc Thang Vu
Beginn am:	15. November 2018
Beendet am:	15. Mai 2019

Abstract

Speech emotion recognition (SER) is a significant research topic in human-computer interaction. One of the major problems in SER is data scarcity. This master's thesis aims to investigate a novel data augmentation method based on cycle consistent adversarial networks (CycleGANs). It transfers feature vectors extracted from a unlabeled speech corpus into the domains of target emotions. Furthermore, the CycleGAN framework is extended with a classification loss which improves the discriminability between the generated data. The quality of the synthetic data is evaluated on both within-corpus and cross-corpus experiments of SER. Both show an improvement of classification performance with augmented data. Additionally, two meaningful problems met in our training process are discussed and analyzed.

Kurzfassung

Sprachliche Emotionserkennung (SER) ist ein bedeutendes Forschungsthema in der Menschen-Computer-Interaktion. Eines der Hauptprobleme der SER ist der Mangel an Daten. Ziel dieser Masterarbeit ist es, eine neue Methode für Datenergänzung zu untersuchen, die auf CycleGANs (cycle consistent adversarial networks) beruht. Mit dieser Methode werden die Merkmalsvektoren, die vom unannotierten Sprachkorpus extrahiert sind, in die Domänen der Zielemotionen übertragen. Darüber hinaus wird das CycleGAN-Framework mit einem Klassifizierungsverlust erweitert, damit sich die generierten Daten leichter voneinander unterscheiden lassen. Die Qualität der synthetischen Daten wird durch Experimente innerhalb eines Korpus und korpusübergreifend evaluiert. Die beiden Experimente zeigen eine Verbesserung der Klassifizierungsergebnisse wenn die synthetischen Daten ergänzt werden. Zusätzlich werden zwei im Trainingsprozess auftretende Probleme diskutiert und analysiert.

Contents

1 Introduction

> *“The robot had no feelings, only positronic surges that mimicked those feelings. (And perhaps human beings had no feelings, only neuronic surges that were interpreted as feelings.)"*
>
> — Isaac Asimov*, The Robots of Dawn [Asi94]*

Speech emotion recognition (SER) refers to automatically recognizing human emotion and affective states from speech [Sch18]. As an important and rapidly growing research field, SER has great potential to improve natural voice-based human-computer interaction, such as in-car board system to perceive emotional state of drivers, call center applications to detect annoyance of users and diagnostic tools for therapists, among others[EKK11].

1.1 Motivation

Data scarcity has been acknowledged as one of major challenges in SER, which is mainly reflected in three aspects:

- The first problem is a lack of large naturalistic emotional speech corpora. Very few benchmark databases can be shared among researchers. In particular, speech data that are collected from real life situations are not available for public use due to some legal and moral issues [EKK11]. Moreover, emotional speech utterances in most public databases are produced by actors who act as if they were in a required emotion state. As a result, the emotional expression might be exaggerated compared with real world situations [EKK11].
- The next crucial issue is annotation. Since expressed emotion and felt emotion are different, external annotation is always neccessary, even for acted emotions. However, the annotation job is very time-consuming. Due to high subjectivity and uncertainty, usually five or more annotators are required to form the basis of the construction of target labels [Sch18]. Further, an assessment is made to eliminate outliers and get labels by majority vote for categorical annotation or by average for dimensional annotation such as arousal and valence [Sch18].
- Finally, speech utterances in most databases are unbalancedly distributed over emotions. In general, the number of utterances with neutral emotion is the largest in speech corpora [EKK11]. However, for evaluating classification accuracy, a balanced database is preferred. In addition, if one sentence is recorded with different emotions, the human judgement on the perceived emotion could be solely based on the emotional content of the sentence without the influence of its lexical content [EKK11].

One effective method for data augmentation is generative adversarial networks (GANs) introduced by Goodfellow et al. [GPM+14] in 2014. In recent years, GANs have been recognized as one of the most successful approaches for sample generation. By means of an adversarial game between a discriminator and a generator, GANs are trained to generate samples that are indistinguishable from real data. Furthermore, they have following properties [GPM+14]:

- GANs can learn high-dimensional probability distributions in complex real-world problems.
- GANs can be trained with missing data, which is suitable for semi-supervised learning, where labels of many samples are missing.
- GANs have multi-modal outputs, which means they can produce multiple different correct answers and increase the diversity of generated samples.

The goal of this thesis is to evaluate the performance of SER when real training data are augmented in feature space with synthetic data generated by GANs. Specifically, it requires to design a GAN-based model which generates synthetic feature vectors of different emotional utterances, such that a neural network classifier used in SER can be improved when trained with a combination of real and synthetic feature vectors.

1.2 Contributions

To achieve this goal, we propose a method based on cycle-consistent adversarial networks (CycleGANs), one of GAN variants, to find a mapping that can be used to transfer feature vectors of an external large unlabeled speech corpus into the domain of each target emotion in SER. Thus, a large and balanced synthetic dataset can be built. The contributions of this thesis can be summerized as follows:

- We introduce a method based on emotion transfer to generate synthetic feature vectors for the purpose of classification.
- We propose a novel CycleGAN-based architecture that ensures similarity between real and generated samples on the one hand and provides emotional discriminability among generated samples on the other hand.
- We show that in both within-corpus and cross-corpus experiments, a neural network classifier trained on the combination of real and synthetic feature vectors achieves better classification performance than the classifier trained only on the real feature vectors.

1.3 Outline

Chapter 2 provides background information on GAN-based data augmentation for SER and focuses primarily on the introduction to basic GANs and three relevant GAN variants. In addition, a short overview about emotion representations is given. Chapter 3 describes our proposed method, in which we extend the concept of style transfer to emotion transfer and adapt CycleGANs to our problem. Chapter 4 presents the datasets and features for our experiments, as well as the experimental setup for the GAN training and four experiments that examine the quality of the synthetic samples generated by our method. The experiments aim to verify the similarity between the real and generated samples, check the improvements of SER performance contributed by data augmentation for both within-corpus and cross-corpus evaluation, as well as investigate the usage of the synthetic samples for feature extraction. The results of each experiment are reported separately. Chapter 5 discusses two problems we met in the training process. Chapter 6 concludes this thesis and illustrates possibilities for future work.

2 Background

Deep neural networks often require large amounts of training data to achieve good performance. Data augmentation is an often-used technique to enlarge the size of training dataset. Classical data augmentation applies small transformations to existing data in datasets, which is usually limited to specific tasks. Some common image data augmentation techniques, e.g. translation and rotation [WP17], are not applicable to text or speech processing. Synonym replacement [ZL15], often used in text processing, is hard to implement for speech-related tasks. Similarly, traditional augmentation methods for speech like voice transformation, changing the speed of the audio signal [KPPK15], are also not suitable for images or text. By contrast, GAN-based data augmentation focuses on the simulation of real data distribution, which is independent from tasks, and therefore the experience learned from one task can be utilized in others.

This chapter starts with a brief introduction to GANs. Furthermore, three GAN variants are explained, two of them have been applied for speech emotion recognition in previous research, the last one has achieved great success in image processing, which inspired us to investigate its application on speech emotion recognition. In addition, a short overview about emotion representations is given.

2.1 Generative Adversarial Networks

Generative models refer to any model that takes a set of training samples drawn from a distribution and learns to represent an estimate of that distribution [Goo16]. They can be divided into explicit models and implicit models. Explicit models compute the density function of the distribution directly while implicit models focus on generating samples from the distribution represented by the model. GANs are often used for sample generation [Goo16]. This section describes (1) the basic architecture of GANs, (2) the cost functions of GANs in a minimax game and a non-saturating heuristic game, (3) the convergence problem of GANs and its possibile solutions.

2.1.1 Architecture

A GAN is a generative model based on game theory that pits two adversaries against each other: a discriminator D and a generator G which are typically represented by a deep neural network with parameters $\theta^{(D)}$ and $\theta^{(G)}$, respectively [Goo16]. Figure 2.1 shows the architecture of a basic GAN, which is also called vanilla GAN. The generator takes a latent variable $\boldsymbol{z}$ sampled from a random noise distribution $p_{\boldsymbol{z}}(\boldsymbol{z})$ as input and outputs synthetic data $G(\boldsymbol{z}; \theta^{(G)})$.[1] The goal of the generator is to create samples that are indistinguishable from real ones. The discriminator receives on the one

[1] The semicolon in the function definition is used to separate input variables from parameters. In the following, $G(\boldsymbol{z})$ is used instead for simplicity. Similar to $D(\boldsymbol{x})$ and $D(G(\boldsymbol{z}))$.

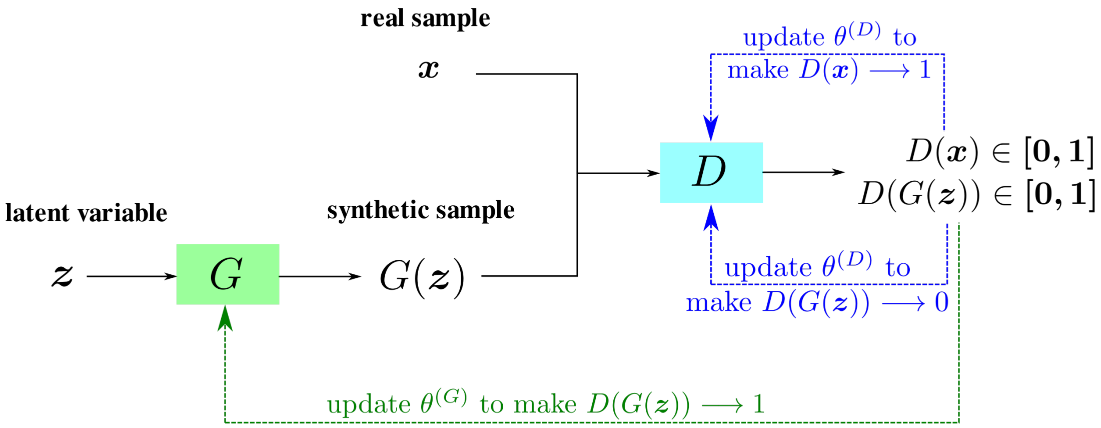

real sample
$\boldsymbol{x}$
latent variable
$\boldsymbol{z}$
G
synthetic sample
$G(\boldsymbol{z})$
D
update $\theta^{(D)}$ to make $D(\boldsymbol{x}) \longrightarrow 1$
$D(\boldsymbol{x}) \in [\mathbf{0}, \mathbf{1}]$
$D(G(\boldsymbol{z})) \in [\mathbf{0}, \mathbf{1}]$
update $\theta^{(D)}$ to make $D(G(\boldsymbol{z})) \longrightarrow 0$
update $\theta^{(G)}$ to make $D(G(\boldsymbol{z})) \longrightarrow 1$

Figure 2.1: Architecture of a basic GAN.

hand real samples $\boldsymbol{x}$ and on the other hand synthetic samples $G(\boldsymbol{z})$. The output $D(\boldsymbol{x})$ or $D(G(\boldsymbol{z}))$ is a scalar value which indicates the probability that a received sample came from the real training dataset rather than from the generator. The generator strives to make $D(G(\boldsymbol{z}))$ approach one (green dotted line in Figure 2.1) while the discriminator strives to make $D(G(\boldsymbol{z}))$ approach zero and make $D(\boldsymbol{x})$ approach one (blue dotted line in Figure 2.1) [Goo16].

2.1.2 Cost Functions

Suppose we have an input dataset $(s_i, y_i)_{i=1}^{N}$ in which half of data are from the real samples $\boldsymbol{x}$ and half are from the synthetic samples $G(\boldsymbol{z})$. Each training sample s_i corresponds to a label y_i. All real samples are labeled as one and all synthetic samples as zero. Since the discriminator's task can be considered a binary classification task, it's cost function can be defined as a binary cross-entropy loss [Goo16]

$$\begin{aligned} J^{(D)}(D, G) &= H((s_i, y_i)_{i=1}^{N}, D) \\ &= -\frac{1}{N}\sum_{i=1}^{N}[y_i \log D(s_i) + (1 - y_i)\log(1 - D(s_i))] \end{aligned} \tag{2.1}$$

If we substitute y_i with one for $s_i = \boldsymbol{x}$ and zero for $s_i = G(\boldsymbol{z})$, as well as replace the averages with expectations, we get the **discriminator's cost function** as follows [Goo16]:

$$J^{(D)}(D, G) = -\mathop{\mathbb{E}}_{\boldsymbol{x} \sim p_{\text{data}}(\boldsymbol{x})}[\log D(\boldsymbol{x})] - \mathop{\mathbb{E}}_{\boldsymbol{z} \sim p_{\boldsymbol{z}}(\boldsymbol{z})}[\log(1 - D(G(\boldsymbol{z})))] \tag{2.2}$$

where p_{data} is the data distribution over real sample $\boldsymbol{x}$. In a minimax (also called zero-sum) game, the sum of all players' cost is always zero, which means the generator's cost is the opposite of $J^{(D)}$. However, when we compute gradient descent with respect to G, only the second term in Eq. 2.2 matters. Therefore, the **generator's cost function in a minimax game** can be defined as follows [Goo16]:

$$J^{(G)}(G) = \mathop{\mathbb{E}}_{\boldsymbol{z} \sim p_{\boldsymbol{z}}(\boldsymbol{z})}[\log(1 - D(G(\boldsymbol{z})))] \tag{2.3}$$

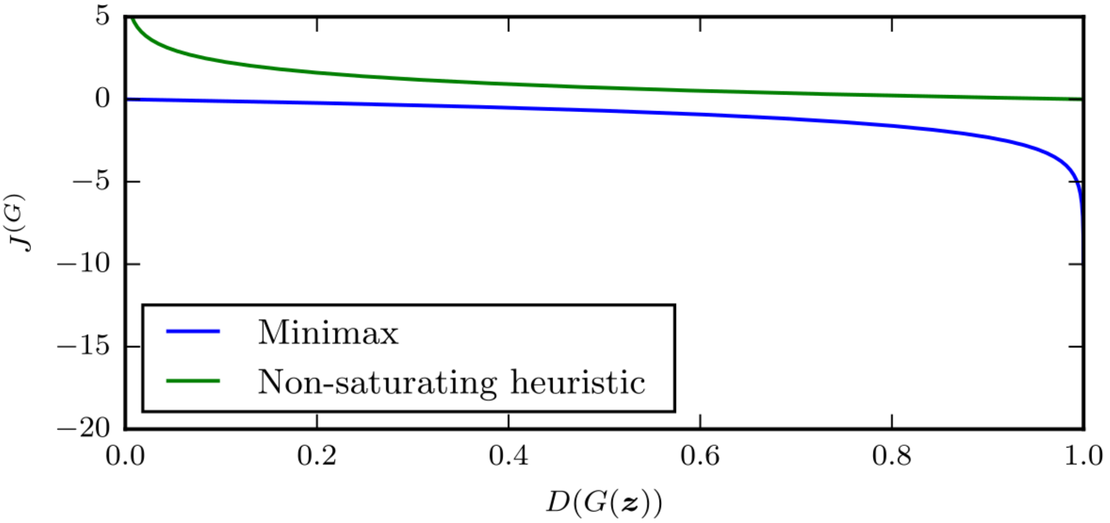


Figure 2.2: Comparison of generator's cost in a minimax game and a non-saturating heuristic game. The generator'cost $J(G)$ for generating a sample $G(\boldsymbol{z})$ depends on the probability $D(G(\boldsymbol{z}))$ that the discriminator assigns to the sample being real. The higher this probability is, the less cost the generator receives [Goo16]

.

The entire game can be summarized with a **value function** which is defined as the discriminator's payoff [Goo16]:

$$\min_{G}\max_{D} V(D,G) = \mathop{\mathbb{E}}_{\boldsymbol{x}\sim p_{\text{data}}(\boldsymbol{x})}[\log D(\boldsymbol{x})] + \mathop{\mathbb{E}}_{\boldsymbol{z}\sim p_{\boldsymbol{z}}(\boldsymbol{z})}[\log(1 - D(G(\boldsymbol{z})))] \tag{2.4}$$

where the generator tries to minimize it while the discriminator tries to maximize it.

However, the generator's cost in a minimax game does not perform well in practice, because when the generator maximizes the same cross-entropy that the discriminator minimizes, which makes the discriminator easily detect that the generated samples are fake and reject them with high confidence [Goo16]. As a result, the generator's gradient vanishes. Instead of mimizing the log-probability of the discriminator being correct in Eq. 2.3, the generator in a non-saturating heuristic game attempts to maximize the log-probability of the discriminator being mistaken. The **generator's cost function in a non-saturating heuristic game** is defined as follows [Goo16]:

$$J^{(G)}(G) = -\mathop{\mathbb{E}}_{\boldsymbol{z}\sim p_{\boldsymbol{z}}(\boldsymbol{z})} \log[D(G(\boldsymbol{z}))] \tag{2.5}$$

Figure 2.2 from [Goo16] shows the difference of generator's cost function in the minimax and non-saturating heuristic game. The horizontal axis describes the probability that a synthetic sample is regarded as real by the discriminator. The higher this value is, the less cost the generator receives. The left portion of the graph, where $D(G(\boldsymbol{z}))$ is near to zero, appears often at the start of a training process. At this time, the discriminator can often easily distinguish whether a sample is real or syntheitc, because the generator starts to sample from the random noise distribution $p_{\boldsymbol{z}}(\boldsymbol{z})$ with random parameters. It can be seen that the minimax curve is very flat at the left end, which means the generator has very little gradient. Using gradient descent optimization, the generator has already stopped to improve its output at the initial stage. By constrast, the curve of the non-saturating heuristic game loses its gradient at the right end, where the training process arrives at the optimality and the generated samples are capable of fooling the discriminator. Therefore, the generator's cost function in the non-saturating game is often used in practice while the minimax game version is rather used for theoretical analysis [Goo16].

2.1.3 Convergence Problem

GANs are usually trained with simultaneous gradient descent on the two players' cost[2]. If G and D have enough capacity[3] and training time, they will reach a unique Nash equilibrium $(\theta^{(\mathrm{G})}_{\mathrm{opt}}, \theta^{(\mathrm{D})}_{\mathrm{opt}})$ at which each player can't reduce their cost without changing the other's parameters [FRL+18]. At this equilibrium point, the generated samples converge to a good estimate of p_{data} and the discriminator fails to distinguish the real samples from the synthetic ones, i.e.

$$\begin{aligned} D(\boldsymbol{x}; \theta^{(D)}_{\text{opt}}) &= \frac{1}{2} \qquad \forall \boldsymbol{x} \sim p_{\text{data}}(\boldsymbol{x}) \\ D(G(\boldsymbol{z}; \theta^{(G)}_{\text{opt}}); \theta^{(D)}_{\text{opt}}) &= \frac{1}{2} \qquad \forall \boldsymbol{z} \sim p_{\boldsymbol{z}}(\boldsymbol{z}) \end{aligned} \tag{2.6}$$

In practice, GANs often face the problem of non-convergence. It can be inferred from Eq. 2.4 that our objective is to find a point that is a minimum with respect to $\theta^{(G)}$ and meanwhile a maximum with respect to $\theta^{(D)}$. Therefore, the equilibrium for the adversarial game is not a local minimum, but a saddle point of the value function V, as shown in Figure 2.3. The red and blue arrows characterize the trajectory of G and D in simultaneous gradient descent, respectively. As we can see, G tries to go downhill and D uphill. The non-convergence of GANs could happen when D and G go into a stable circular obit instead of arriving at the saddle point [GBC16], as illustrated with solid lines in Figure 2.3. Even worse, although the current value is already near the equilibrium point, every following gradient update still might cause large oscillation [Goo16], as illustrated with dotted lines.

To tackle the convergence problem, Saliman et al. have proposed some tricks to make GAN training easily convergent, including feature matching, minibatch discrimination, one-sided label smoothing and virtual batch normalization [SGZ+16]. Here, we list some tips that are quite useful during our training process: (1) The simultaneous updates of D and G should be balanced, such that neither of them overpowers the other. (2) Decaying learning rates can prevent the cost function from oscillating and diverging. (3) The architecture and loss of the basic (vanilla) GAN can be adjusted to improve adversarial regularization. (4) Side information like labels should be incorporated if they exist. (5) Non-saturating heuristic cost functions are favored in practical use [Goo16].

[2] The algorithm can be found in section 4 in [GPM+14].

[3] The capacity means that the discriminator can reach its optimum given G and $\max_D V(G, D)$ is convex in $\theta^{(G)}$ [GBC16; GPM+14].

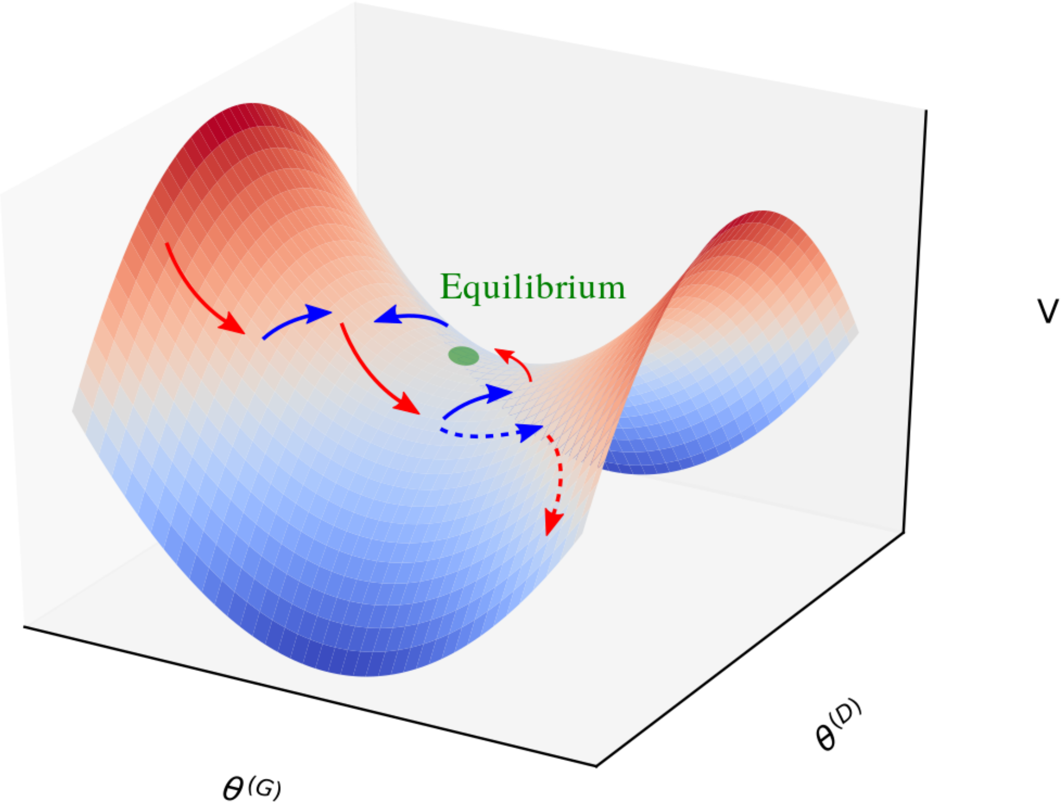


Figure 2.3: Convergence problems of GANs. The equlibrium point for a minimax game is a saddle point of the value function, which is hard to reach due to the risk of circulation and oscillation.

2.2 GAN Variants for SER

In recent years, GANs have also been used for speech emotion recognition. For instance, Chang and Scherer utilized a deep convolutional GAN (DCGAN) to learn a discriminative representation of emotional speech in a semi-supervised way [CS17], Han et al. proposed a conditional adversarial training framework to predict arousal and valence from speech signals [HZR+18]. Here, we focus on the application of GANs for synthetic data generation, which aims to synthesize more data that capture the distribution of real data to improve the performance of SER. This section contains three GAN variants for this purpose: adversarial autoencoder, conditional GAN and CycleGAN.

2.2.1 Adversarial Autoencoder

Autoencoder is a type of neural network which consists of an encoder and a decoder. The encoder learns to compress input data into a bottleneck layer (called code vector) and the decoder learns to uncompress the code vector to reconstruct the original input data. Autoencoder was originally invented in the 1980s for sparse representation and dimensionality reduction.

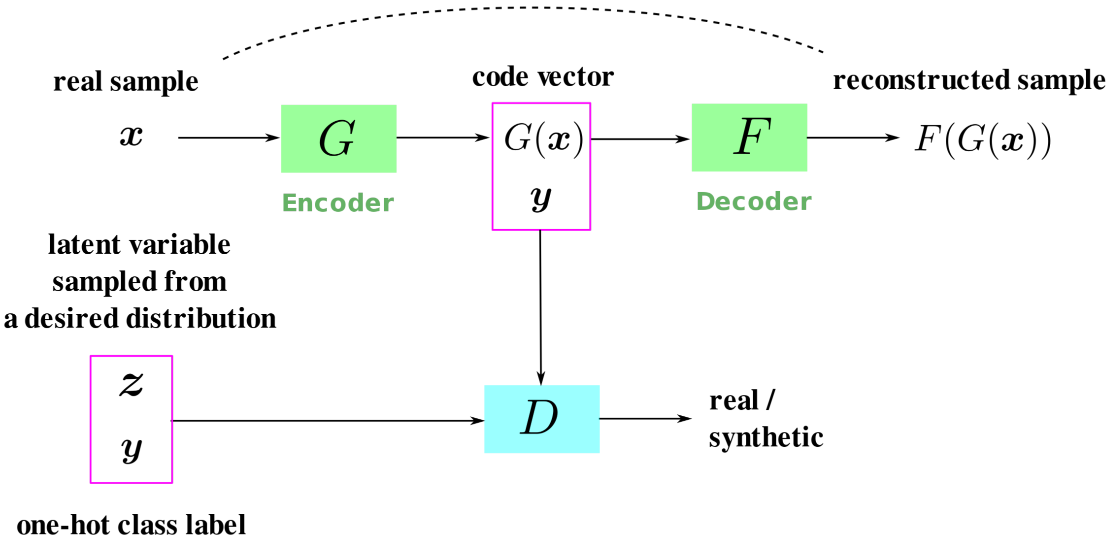


Figure 2.4: Archtitecture of adversarial autoencoder.

Recently, Makhazani et al. proposed adversarial autoencoder (AAE) that turns an autoencoder into a generative model [MSJ+16]. The architecture of AAE is shown in Figure 2.4. The encoder G works as generator to produce code vectors which are similar to a desired distribution $p(\boldsymbol{z})$. The code vectors can be considered a compressed representation of real samples $\boldsymbol{x}$. The reconstructed samples $F(G(\boldsymbol{x}))$ from the decoder F are the synthetically generated samples that can be used to improve the classification. The discriminator D tells apart the generated code vectors $G(\boldsymbol{x})$ from the latent variable $\boldsymbol{z}$ directly sampled from the desired distribution $p(\boldsymbol{z})$. For a multiclass classification task, $p(\boldsymbol{z})$ is usually an N component Gaussian Mixture Model (GMM), where N is the number of classes.

For a semi-supervised training procedure, label information can be incorporated in adversarial regularization [MSJ+16]. In the AAE for SER, each component in the GMM corresponds to one of the emotion labels [SGS+17]. These labels are encoded as one-hot vectors and added to the input of the discriminator, as shown in Figure 2.4.

The losses of AAE are composed of reconstruction loss and adversarial loss. In the AAE for SER, the reconstruction loss $\mathcal{L}^{\mathrm{REC}}$ is defined as the Mean Squared Error (MSE) between the input $\boldsymbol{x}$ and the reconstruction $F(G(\boldsymbol{x}))$. Weights of the encoder and the decoder are updated to minimize this value. The adversarial loss can be further divided into two cross-entropy losses: the cross-entropy loss $\mathcal{L}^{\mathrm{G}}$ for the generated code vectors $G(\boldsymbol{x})$ to be labeled as one, and the cross-entropy loss $\mathcal{L}^{\mathrm{D}}$ for $G(\boldsymbol{x})$ to be labeled as zero and meanwhile for $\boldsymbol{z}$ to be labeled as one. Weights of the encoder are updated to minmize $\mathcal{L}^{\mathrm{G}}$, weights of the encoder and the discriminator are updated to minimize $\mathcal{L}^{\mathrm{D}}$ [SGS+17]. During the training process, the losses $\mathcal{L}^{\mathrm{REC}}$, $\mathcal{L}^{\mathrm{D}}$ and $\mathcal{L}^{\mathrm{G}}$ are minimized with alternating Stochastic Gradient Descent (SDG).

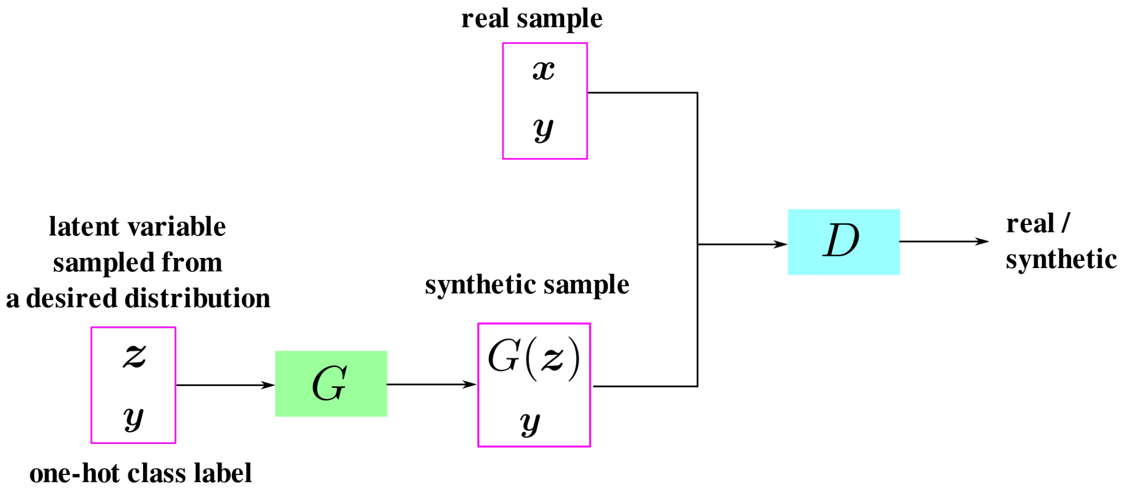


Figure 2.5: Architecture of conditional GAN.

2.2.2 Conditional GAN

Conditional GAN is an extension of vanilla GAN which learns the parameters of the discriminator and the generator conditioned on labels [MO14]. Given a set of data points $\boldsymbol{x}$ and their corresponding labels $\boldsymbol{y}$, a vanilla GAN models the distribution $p(\boldsymbol{x})$ while a conditional GAN learns the conditional distribution $p(\boldsymbol{x}|\boldsymbol{y})$ [SGE18].

Sahu et al. investigated the usage of vanilla GAN and conditional GAN in synthetic data generation for SER [SGE18]. The generator G is required to generate high-dimensional (1582-d) feature vectors $G(\boldsymbol{z})$ of speech signals from a low-dimensional (2-d) distribution $p_{\boldsymbol{z}}(\boldsymbol{z})$. It has been proved that a vanilla GAN cannot reach convergence and conditional GAN is capable of arriving at convergence when trained with special schemes [SGE18].

Instead of sampling the latent variable $\boldsymbol{z}$ from a random noise distribution in vanilla GAN, the conditional GAN for SER chose $p_{\boldsymbol{z}}(\boldsymbol{z})$ to be an N component GMM like the desired distribution in AAE. As shown in Figure 2.5, for each real sample $\boldsymbol{x}$ with label $\boldsymbol{y}$, the latent variable $\boldsymbol{z}$ is sampled from the corresponding component of the GMM distribution. The label $\boldsymbol{y}$ is added to the input of the discriminator in the form of one-hot vector for both the real and synthetic sample.

Moreover, the parameters of the generator have to be initialized with decoder weights of a pre-trained AAE, otherwise the adversarial loss would still fail to reach convergence [SGE18]. This indicates the difficulty of learning real data distribution in a high-dimensional space, which makes the discriminator to easily overpower the generator. Therefore, Sahu et al. also kept the generator's learning rate much higher than the discriminator's (0.001 vs 0.0001 respectively) and trained the generator for five iterations for every iteration of discriminator training [SGE18].

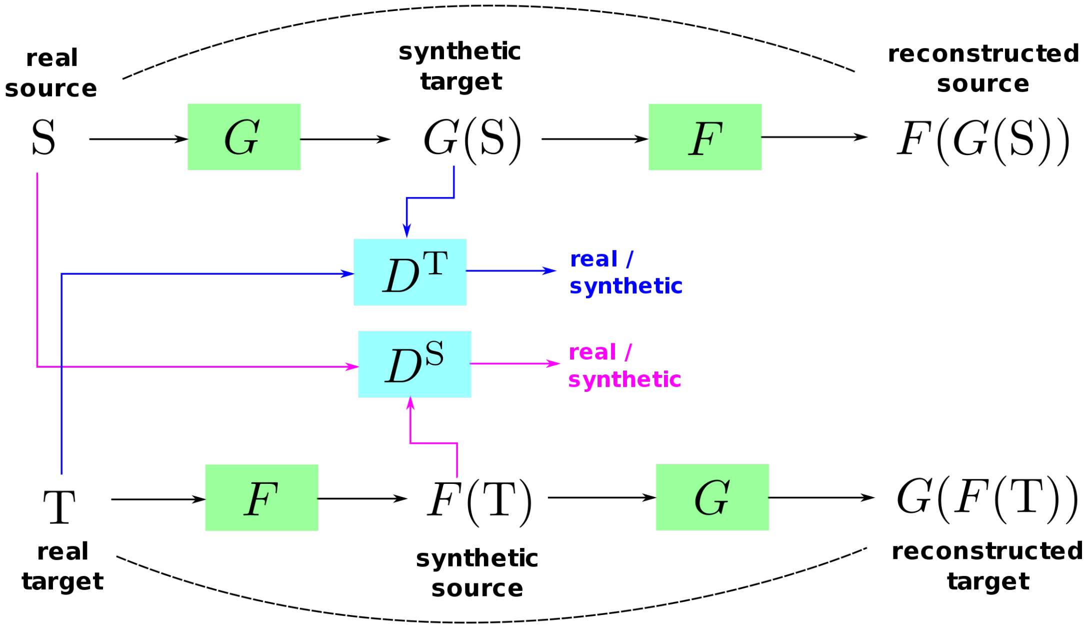


Figure 2.6: Architecture of a CycleGAN.

2.2.3 Cycle-Consistent Adversarial Networks

Cycle-consistent adversarial networks (CycleGAN) are known as one successful approach to solve the image-to-image translation problem[4] with unpaired datasets. The bijective mapping learned by a CycleGAN can capture special characteristics of one image collection and figure out how these characteristics could be translated into the other image collection [ZPIE17]. The great success of CycleGANs in image style transfer inspired us to generate synthetic data for SER with emotion transfer.

Figure 2.6 shows the architecture of a CycleGAN. It consists of two mapping functions G and F. G learns to translate samples from a source domain S to a target domain T. F is an inverse mapping of G. Both mapping functions G and F can be regarded as generators for target data generation and source data generation, respectively. Besides that, there are two adversarial discriminators D^{T} and D^{S}. As the adversary against G in target data generation, D^{T} distinguishes real target T from synthetic target $G(\mathrm{S})$. Similarly, D^{S} distinguishes real source S from synthetic source $F(\mathrm{T})$. In order to ensure that the generated samples can be translated back to the original samples, a CycleGAN reconstructs its source and target such that $F(G(\mathrm{S})) \approx$ S and $G(F(\mathrm{T})) \approx$ T, which is therefore called cycle-consistent [ZPIE17].

[4] Image-to-image translation refers to converting an image from one representation of a given scene to another, e.g. grayscale to color, image to semantic labels, edge-map to photograph, and so on [ZPIE17].

The losses of a CycleGAN consist of adversarial loss and cycle consistency loss. The adversarial loss can be further divided into a target data generation part and a source data generation part. The adversarial loss for target data generation is defined as follows [ZPIE17]

$$\mathcal{L}^{\mathrm{GAN}}(G, D^{\mathrm{T}}, \mathrm{S}, \mathrm{T}) = \mathop{\mathbb{E}}_{\mathrm{t} \sim p_{\mathrm{t}}} [\log D^{\mathrm{T}}(\mathrm{t})] + \mathop{\mathbb{E}}_{\mathrm{s} \sim p_{\mathrm{s}}} [\log(1 - D^{\mathrm{T}}(G(\mathrm{s}))] \tag{2.7}$$

Note that the adversarial loss is expressed here in the form of value function. Therefore, its objective is $\min\limits_{G} \max\limits_{D^{\mathrm{T}}} \mathcal{L}^{\mathrm{GAN}}(G, D^{\mathrm{T}}, \mathrm{S}, \mathrm{T})$. Similarly, the adversarial loss for source data generation is $\min\limits_{F} \max\limits_{D^{\mathrm{S}}} \mathcal{L}^{\mathrm{GAN}}(F, D^{\mathrm{S}}, \mathrm{T}, \mathrm{S})$.

Since the mapping functions are represented by deep neural networks with large amounts of parameters, the learned mapping functions are not unique. To further reduce the space of possible mapping functions, the cycle consistency loss is essential. Zhu et al. defined the cycle consistency loss as follows [ZPIE17]:

$$\mathcal{L}^{\mathrm{cyc}}(G, F) = \mathop{\mathbb{E}}_{\mathrm{t} \sim p_{\mathrm{t}}} [\|(G(F(\mathrm{t})) - \mathrm{t})\|_1] + \mathop{\mathbb{E}}_{\mathrm{s} \sim p_{\mathrm{s}}} [\|F(G(\mathrm{s})) - \mathrm{s}\|_1] \tag{2.8}$$

where they mentioned that the L1 norm in this loss can be replaced with other metrics. The overall loss for CycleGAN is:

$$\mathcal{L}(G, F, D^{\mathrm{T}}, D^{\mathrm{S}}) = \mathcal{L}^{\mathrm{GAN}}(G, D^{\mathrm{T}}, \mathrm{S}, \mathrm{T}) + \mathcal{L}^{\mathrm{GAN}}(F, D^{\mathrm{S}}, \mathrm{T}, \mathrm{S}) + \lambda \mathcal{L}^{\mathrm{cyc}}(G, F) \tag{2.9}$$

where λ controls the relative importance of the two losses [ZPIE17].

Interestingly, a CycleGAN can also be seen as a combination of two adversarial autoencoders. Instead of showing a compressed representation of real samples, the code vectors here are a translation of the real samples into another domain [ZPIE17].

2.3 Representations of Emotions

Emotions can be represented in discrete categorical labels or continuous dimensional labels [NV17]. In general, categorical labels contain basic emotions such as anger, disgust, fear, happiness, sadness and surprise. For comparison, neutral speech utterances are also included in almost all emotional databases. Other than categorical labels, dimensional labels describe affective experience in several dimensions. Most dimensional models incorporate arousal and valence. Arousal refers to whether an event is exicting or calming, valence codes emotional events as positive or negative [Ken04]. It has been proved that combining both representations of emotions can improve prediction results [NV17].

3 Proposed Method

Style transfer is a heated topic in image processing. It aims at synthesizing a novel image by combining the content of one image with the style of another image [ZPIE17]. If we take emotions as an affective "style" of speech, emotion transfer can be used as a method to generate new samples of emotional speech.

In general, there is a distinction between example-guided style transfer, in which the target style comes from a single example, and collection style transfer, in which the target style is defined by a collection of images [HLBK18].[1]. For instance, example-guided style transfer learns to mimic the style of "The Starry Night", a single piece of Van Gogh, while collection style transfer learns to generate images in the style of an entire collection of Van Gogh [ZPIE17]. In terms of emotion transfer, collection style transfer is preferred, because it can capture the emotional characteristics that all utterances in the collection have in common with and get rid of the influences of non-emotional characteristics that some single utterance contains.

Collection style transfer can be regarded as a task of image-to-image translation which can also be applied to other tasks, such as object transfiguration[2], image to semantic labels[3], etc. The "pix2pix" framework proposed by Isola et al. [IZZE17] was the first GAN-based framework for image-to-image translation [HLBK18]. But it requires paired training data which are difficult and expensive to obtain. Since the image-to-image translation problem without supervision is inherently ill-posed and requires additional constraints [HLBK18], Zhu et al. use cycle-consistent loss as constraint in CycleGANs [ZPIE17] and achieve compelling results in image style transfer even with unpaired training data. Due to lack of pararell training data in most emotional speech databases, CycleGANs are the most appropriate image style transfer method that can be adapted to emotion transfer.

In this chapter, we first provide an overview of our proposed method which involves multiple research fields, and its relationship with the representative research works in these fields. Furthermore, the architecture and loss functions of our method are explained in detail, including the adjustments we made on CycleGANs for SER. In addition, the reasons of using unlabeled data as real source are given.

[1] Example-guided style transfer learns a representation to separate and recombine the image content and style [GEB16]. Collection style transfer focuses on learning the mapping between two collections [ZPIE17]

[2] Object transfiguration aims to transform a particular type of object in an image to another type of object without influencing the background regions [CXYT18].

[3] Image labeling is a problem in image segmentation. It aims to identify homogeneous regions in an image by associating each pixel in the image with a label denoting a semantically meaningful part [SCCL06].

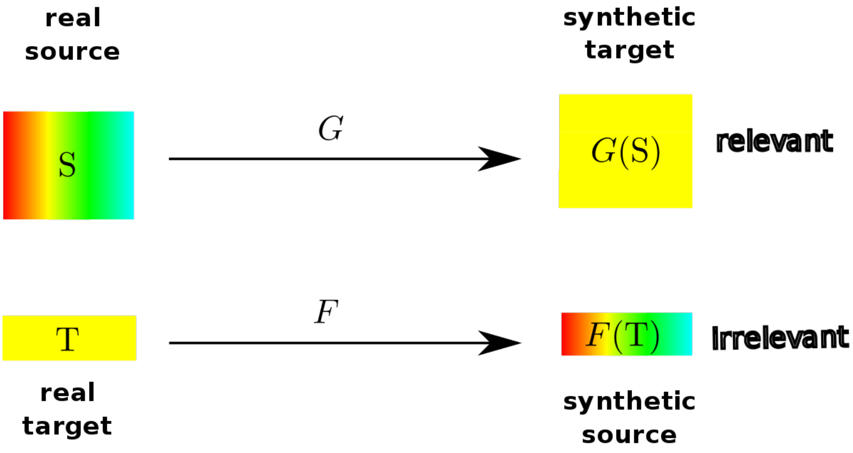


Figure 3.1: Concept of emotion transfer to avoid the necessity of building mappings for each pair of emotion classes. The functions G and F map the real source S and real target T to the synthetic target $G(\mathrm{S})$ and synthetic source $F(\mathrm{T})$, respectively. Each color stands for a type of emotion. We are only interested in the synthetic target $G(\mathrm{S})$ which is a large synthetically generated dataset with the target emotion. The synthetic source $F(\mathrm{T})$ will be ignored.

3.1 GAN-based Emotion Transfer for Data Augmentation

Since a CycleGAN model learns one-to-one mappings between a source and a target domain [ARS+18; CCK+18], for a labeled speech dataset with N emotion classes, we need to establish a mapping between each pair of them, i.e. $\frac{N(N-1)}{2}$ mappings, which is quite expensive. In our method, the labeled data of each emotion type are used as the target domain while the source domain is a large external unlabeled dataset. Figure 3.1 shows the concept of emotion transfer in our method. As mentioned in section 2.2.3, a CycleGAN maps its real source and real target to a synthetic target and a synthetic source, respectively. Therefore, we can generate a synthetic target dataset which is as large as the real source and has the same emotion as the real target dataset. The synthetic target will be further used for the data augmentation for SER, while the synthetic source is in the domain of the unlabeled dataset and will be ignored. Instead of training N CycleGANs separately, we incorporate the N CycleGANs into a whole framework to associate the generated samples of each target emotion with each other, this framework will be explained in section 3.2.

In addition, our method generates synthetic samples in feature space, which means the input of the CycleGANs-framework is not a raw speech signal, but feature vectors used for classification, which are extracted by openSMILE [EWGS13][4]. The reason is that our primary purpose is to improve the classification performance of SER instead of speech synthesis. However, sample generation in feature space has both advantages and disadvantages. The advantage is that we can focus on the investigation of simulating data distribution via GANs without considering speech synthesis. The

[4] OpenSMILE is an open-source software for automatic extraction of features from audio signals and for classification of speech and music signals. Section 4.1.2 describes details about the extracted features in our method.

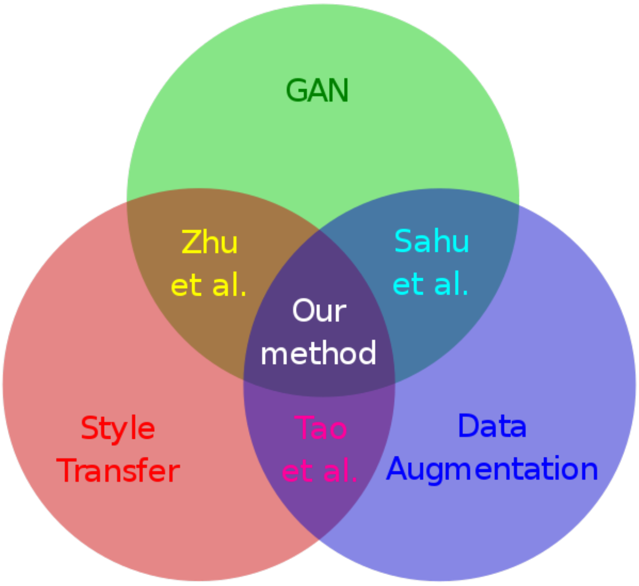


Figure 3.2: Involved research fields in our method and its relationship with related works in these fields: Zhu et al. [ZPIE17] proposed CycleGANs that can be used in image style transfer, which is however too expensive for data augmentation because of one-to-one mapping; Sahu et al. [SGE18] utilized conditional GAN to generate synthetic data for SER from a low-dimensional GMM rather than via style transfer; Tao et al. [TKL06] transferred neutral speech to emotional speech with other methods than GANs to enlarge emotional speech corpora.

disadvantage is that the emotion of generated samples lacks an explict form for human perceptual evaluation, but we are still capable of testing their emotional characteristics by comparing their similarity with real data samples. The details will be explained in section 4.3.

Figure 3.2 shows the involved research fields in our method, i.e. GAN, style transfer and data augmentation. To the best of our knowledge, we are the first to combine all the three fields to synthesize samples for improving classification performance of SER. It can be shown from a few representative research works in these fields that missing anyone of these three fields would limit the quality, quantity or scope of generated samples.

- The CycleGANs proposed by Zhu et al. [ZPIE17] combine the use of GAN and style transfer. However, a single CycleGAN establishes a one-to-one mapping between two emotions each time, which neglects their correlation with other emotions. As a result, the generated samples of different emotions cannot capture the discriminability between each other.
- Sahu et al. [SGE18] use conditional GAN for data augmentation. They generate synthetic samples from a 2-dimensional GMM distribution rather than via emotion transfer. It requires label information of the training data to learn the conditional distribution. In this way, the generated dataset can only be as large as the real training dataset.

- The prosody conversion proposed by Tao et al. [TKL06] is a traditional method of style transfer for data augmentation in speech synthesis. However, it focuses only on the transfer from neutral speech to emotional speech, which constraints the possibilites of conversion between speech of different emotions.

3.2 Adapting CycleGANs to Data Augmentation

Our method can be seen as an adaptation of CycleGANs to data augmentation. This section provides a mathematical formulation of our adjustments to the architecture and loss functions of CycleGANs. They are composed of three parts: building CycleGANs as components of a whole framework, introducing a classifiation loss to distinguish between generated samples and using large unlabeled data as real source.

3.2.1 CycleGANs as Components

Given a labeled dataset X with N emotion classes, we generate synthetic samples for each emotion i using one CycleGAN. As shown in the upper part of Figure 3.3, the CycleGAN establishes a bijective mapping between a source domain S and a target domain T_i, where S is an external unlabeled dataset and T_i represents the samples of emotion i in the labeled dataset X. The two mapping functions G_i and F_i are used for translating from source to target and from target to source, respectively. The adversarial discriminator D_i^T encourages G_i to generate synthetic targets indistinguishable from real samples. According to Eq. 2.7, we define the adversarial loss for G_i and D_i^{T} as $\mathcal{L}_i^{\mathrm{GAN}}(G_i, D_i^{\mathrm{T}}, \mathrm{S}, \mathrm{T}_i)$. Similarly, for the generator F_i and the discriminator D_i^{S} we have the adversarial loss $\mathcal{L}_i^{\mathrm{GAN}}(F_i, D_i^{\mathrm{S}}, \mathrm{S}, \mathrm{T}_i)$. The total adversarial loss is defined as

$$\begin{aligned}\mathcal{L}_i^{\mathrm{GAN}}(G_i, F_i, D_i^{\mathrm{T}}, D_i^{\mathrm{S}}, \mathrm{S}, \mathrm{T}_i) = {} & \mathcal{L}_i^{\mathrm{GAN}}(G_i, D_i^{\mathrm{T}}, \mathrm{S}, \mathrm{T}_i) \\ & + \mathcal{L}_i^{\mathrm{GAN}}(F_i, D_i^{\mathrm{S}}, \mathrm{S}, \mathrm{T}_i)\end{aligned} \tag{3.1}$$

where the generators G_i and F_i try to minimize it while the discriminators D_i^{T} and D_i^{S} try to maximize it.

In addition, a CycleGAN regularizes the adversarial training with a cycle consistency loss. It translates the synthetic target $G_i(\mathrm{S})$ back to the source domain and computes the MSE between the real source S and reconstruction $F_i(G_i(\mathrm{S}))$. The same is done for T_i and the reconstructed target $G_i(F_i(\mathrm{T}_i))$. Consequently, the total cycle consistency loss is defined as follows:

$$\begin{aligned}\mathcal{L}_i^{\mathrm{cyc}}(G_i, F_i, \mathrm{S}, \mathrm{T}_i) = {} & \mathop{\mathbb{E}}_{\mathrm{s} \sim p_{\mathrm{s}}}[\|(F_i(G_i(\mathrm{s})) - \mathrm{s})\|_2^2] \\ & + \mathop{\mathbb{E}}_{\mathrm{t} \sim p_{\mathrm{t}}}[\|G_i(F_i(\mathrm{t})) - \mathrm{t}\|_2^2]\end{aligned} \tag{3.2}$$

where we use L2 norm to replace L1 norm in Eq. 2.8. This is because it is explictly stated in the original CycleGAN paper [ZPIE17] that the measurement of cycle consistency loss can be different from tasks. In our case, L2 norm achieves a better performance.

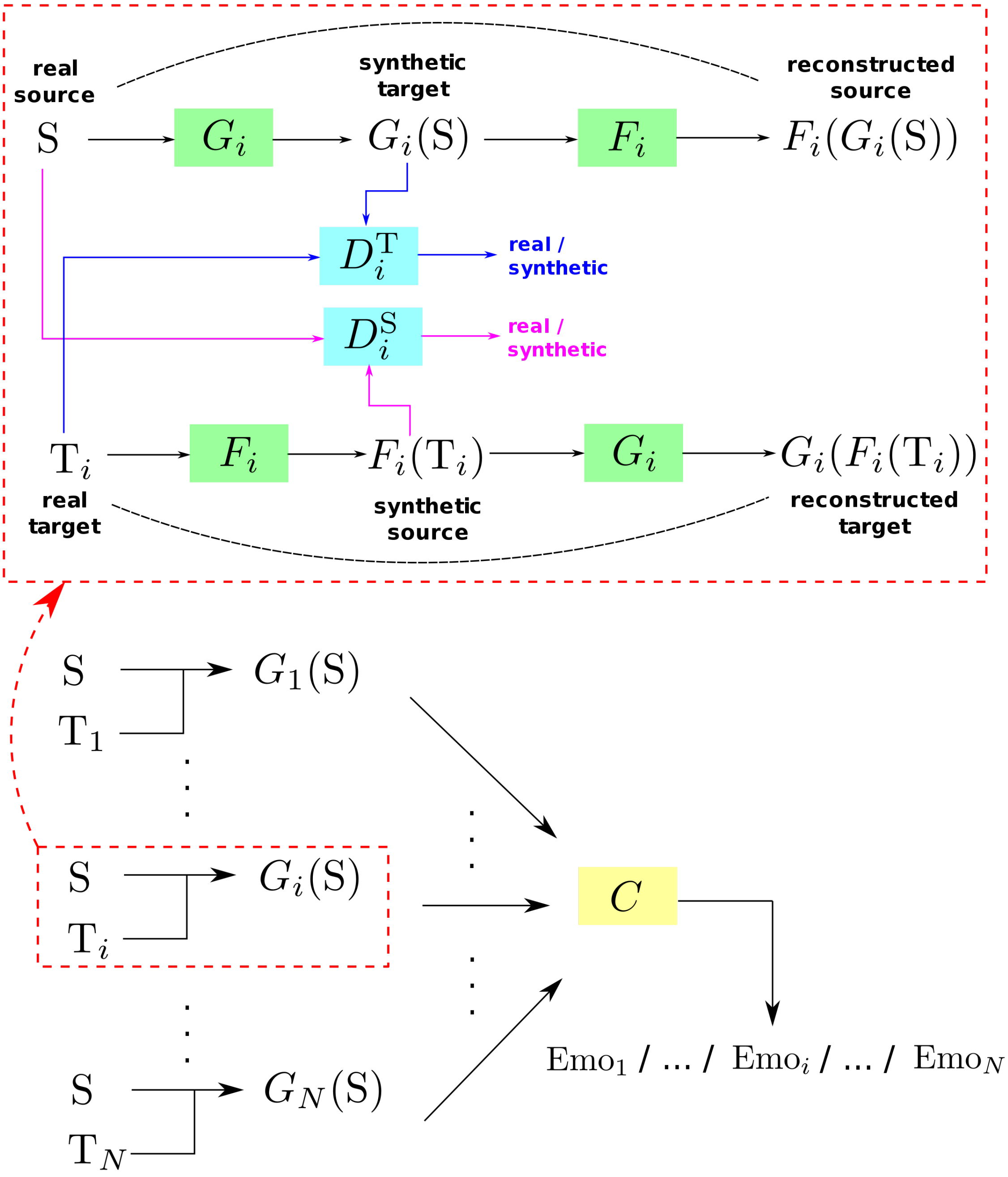


Figure 3.3: Architecture of our model.

3.2.2 Discriminability between Generated Samples

The bijective mapping of CycleGANs ensures similarity between the distribution of real and synthetic data. However, the real data are not perfectly separable, because they're not only influenced by emotions, but also by other factors such as speakers, scenarios, languages, etc. Therefore, we impose a classification loss between the generated data to ensure that they can be properly assigned to their target emotion class. The lower part of Figure 3.3 shows the classifier C. The classification loss is defined as a softmax cross-entropy loss:

$$\mathcal{L}^{\text{cls}} = -\sum_{i} y_i \text{log}(C(G_i(\text{S}))) \tag{3.3}$$

where y_i is the label of the target emotion i. The overall loss for our method is defined as

$$\mathcal{L} = \sum_{i} \mathcal{L}_i^{\text{GAN}} + \lambda^{\text{cyc}} \sum_{i} \mathcal{L}_i^{\text{cyc}} + \lambda^{\text{cls}} \mathcal{L}^{\text{cls}} \tag{3.4}$$

The parameters λ^{cyc} and λ^{cls} are weights for cycle-consistent loss and classification loss, respectively. They affect the similarity of generated feature vectors to real data samples and the emotional discriminability between generated samples. As a result, our model learns a generalized distribution from real data samples instead of merely reconstructing the exact same distribution of these samples.

Figure 3.4 shows the difference between the mappings without and with the classification loss. The enhanced emotional discriminability can reduce the influence from those non-emotional factors, which makes the generated samples also applicable to cross-corpus classification. An extensive discussion about the discriminability can be found in section 5.1.2.

3.2.3 Usage of Large Unlabeled Dataset

Synthetic data generation is a process of mapping source data to a target emotion. While the target emotion is one of the labeled classes, the source samples are not limited to labeled data. We use a large unlabeled external dataset as source due to the following reasons: (1) unlabeled data are plentiful and easy to obtain, (2) since the external corpus has different content from the labeled data, the generated samples contain potentially useful new information, and (3) when the same unlabeled data is transferred to each of the target emotions, the synthetic dataset is composed of balanced samples which only differ in emotions. As a result, the classification can be more concentrated on emotions and independent from content.

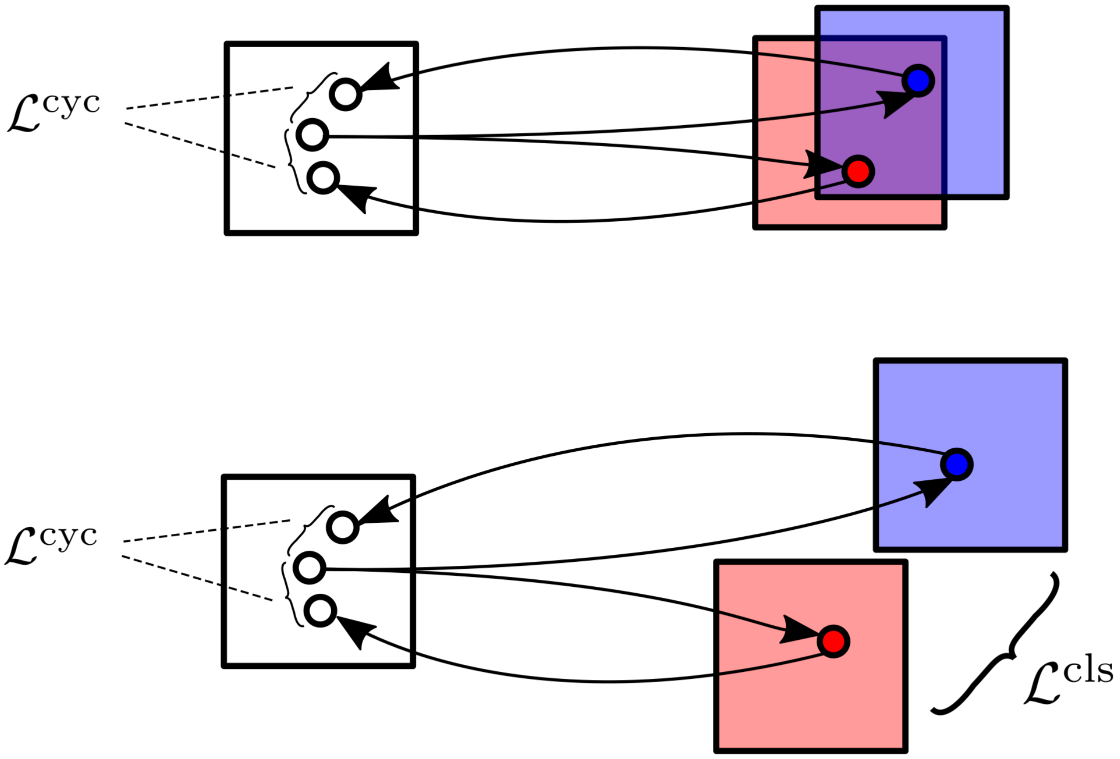

$\mathcal{L}^{\text{cyc}}$
$\mathcal{L}^{\text{cyc}}$
$\mathcal{L}^{\text{cls}}$

Figure 3.4: Difference between the mappings without and with classification loss.

4 Experiments and Results

In this chapter, we first describe our datasets and extracted features, then provide details about the GAN training process for generating synthetic feature vectors. With these synthetic data, we conduct four experiments to: (1) see how similar they are distributed compared with real data samples in target domains and how separable they are between each other over emotions, (2) examine if the classification performance of SER can be improved with these augmented synthetic feature vectors both in within and (3) cross corpus evaluation, (4) check if the synthetic feature vectors can be used for feature selection and the classification performance can therefore be further improved.

4.1 Data and Features

Since the classification results in [SGE18] are used as a baseline, we follow their configurations for labeled data and feature extraction to ensure comparability of SER performance.

4.1.1 Datasets

Three databases are used in our implementation. Two of them act as real source and target domain in our GAN training, respectively. The third one is utilized as test data for the cross corpus experiment.

- **IEMOCAP.** The Interactive Emotional Dyadic Motion Capture (IEMOCAP) [BBL+08] is an annotated emotional speech database which is used as target data in our GAN training. It contains five sessions of English dyadic conversations (one female and one male actor in each session). The entire corpus contains 10,039 utterances annotated with 10 emotion classes. As in [SGE18], we use four of them: angry, happy, sad and neutral, where the happy class also incorporates the samples labeled as excited, resulting in 5,531 samples in total.[1]
- **TEDLIUM.** Acting as unlabeled source data in our GAN training, the Tedlium corpus (release 2) [RDE14] contains 1,495 Ted talks comprising 207 hours of English speech. The talks have been segmented according to the timing information in the transcripts, resulting in 92,973 segments.
- **MSP-IMPROV.** As test data of our cross-corpus evaluation, the MSP-IMPROV database [BPB+17] contains English dyadic interactions between actors. It consists of 7,798 samples from 12 speakers across the same four emotion classes.[2] Due to its class imbalance, MSP-IMPROV database is used as test dataset [SGE18].

[1] Distribution: 1,103 angry, 1,636 happy, 1,708 neutral, 1,084 sad

[2] Distribution: 792 angry, 2,644 happy, 3,477 neutral, 885 sad

Pre-training	
Learning rate	2×10^{-4}
Number of epochs	10k
Minibatch size	64
Dropout	0.2
Layer size	[1582, 1000, 500, 1000, 1582]
Optimizer	Adam
Activation function	Leaky ReLU
GAN Training	
Weight decay	0.8
Learning rate	2×10^{-4}
Number of epochs	2k
Minibatch size	64
Dropout (generator)	0.2
Dropout (discrimintor)	0.2
Dropout (domain classifier)	0.5
Layer size (generator)	[1582, 1000, 500, 1000, 1582]
Layer size (discriminator)	[1582, 1000, 1000, 1]
Layer size (domain classifier)	[1582, 100, 100, 4]
Optimizer	Adam
Activation function	Leaky ReLU

Table 4.1: Hyperparameters of pre-training and GAN training

4.1.2 Features

We use the openSMILE toolkit [EWGS13] to extract acoustic features for each utterance. These features are defined in the ‘emobase2010’ reference feature set, i.e. the Interspeech 2010 Paralinguistic Challenge feature set [SSB+10]. It consists of 1,582 features which are multiple functionals computed from a set of acoustic low level descriptors.

4.2 Experimental Settings for GAN Training

Since there are four emotions to be classified, our model consists of four generators, four discriminators and one classifier. They are all implemented by feed-forward neural networks. Due to the difficulty for generators to learn a high-dimensional distribution, we pre-train each pair of the generators G_i and F_i based on the reconstruction loss between S and $F_i(G_i(\mathrm{S}))$ as well as the reconstruction loss between T$_i$ and $G_i(F_i(\mathrm{T}_i))$ (annotated with dotted curve in Figure 3.3).

Initialized with the pre-trained weights for generators, our model is trained with four parallel CycleGANs which transfer the unlabeled data to each of the target emotions individually. To reduce loss oscillation, the initial learning rate is set to 0.0002 and is linearly decayed every 50 epochs by a factor of 0.8. The other hyperparameters of the pre-training and the GAN training are listed in Table 4.1.

To balance the generators and discriminators, we update the generators twice and the discriminators once at each iteration. Besides that, we use one-sided label smoothing as introduced by [SGZ+16].

Our experiments are implemented with TensorFlow (v 1.10.0) [ABC+16]. In terms of preprocessing, min-max normalization is used for synthetic features generation. For classification we scale the features on each dataset with z-normalization[3] separately, because Zhang et. al. [ZWWS11] have shown that z-normalization yields an improvement over min-max normalization for cross-corpus classification.

4.3 Experiment 1: Emotion Transfer

In this experiment, we first test the feasibility of adapting CycleGANs to emotion transfer in feature space, i.e. whether the synthetically generated feature vectors are capable of preserving the distribution of real target samples when the classification loss is not introduced. Furthermore, we also specify how the classification loss changes the distribution of the synthetic data.

4.3.1 Experimental Setup

We generate four synthtetic emotional datasets in feature space with different values of λ^{cls} from 0 to 3, where λ^{cyc} is set to 5 for all setups. Below we indicate the synthetic datasets generated with $\lambda^{\text{cls}} = i$ with the notation “syn_i”. Obviously, the classification loss has the largest impact on syn_3 and no impact on syn_0.

In order to compare complex distributions in high-dimensional space, we first introduce a measure of overlap of individual feature values [HB00]:

$$f = \frac{(\mu_1 - \mu_2)^2}{\sigma_1^2 + \sigma_2^2} \tag{4.1}$$

where μ_1, μ_2, σ_1, σ_2 are the means and standard deviations of the two datasets for a specific feature dimension. The larger this value is, the less overlapping area between the two datasets exists. For multi-dimensional problems, we have two options: average or maximum over all the feature dimensions.

For our problem, we need to measure: (1) the overlap between the target emotional dataset and each synthetic dataset; (2) the overlap of different emotions within each synthetic dataset. The first goal is to check the similarity between the synthetic and target datasets, where an average over all the feature dimensions is preferred, because a similarity between two distributions on one dimension doesn’t mean the two distribution are similar over all the dimensions. By contrast, the second goal is to examine the discriminability between the generated samples, where the maximum over all the feature dimensions is more appropriate, because as long as there exists one highly discrimninating feature between two distributions, they can be easily separated, no matter how the differences on the other dimensions look like.

[3] Also known as standard normalization or z-scores.

4.3.2 Results

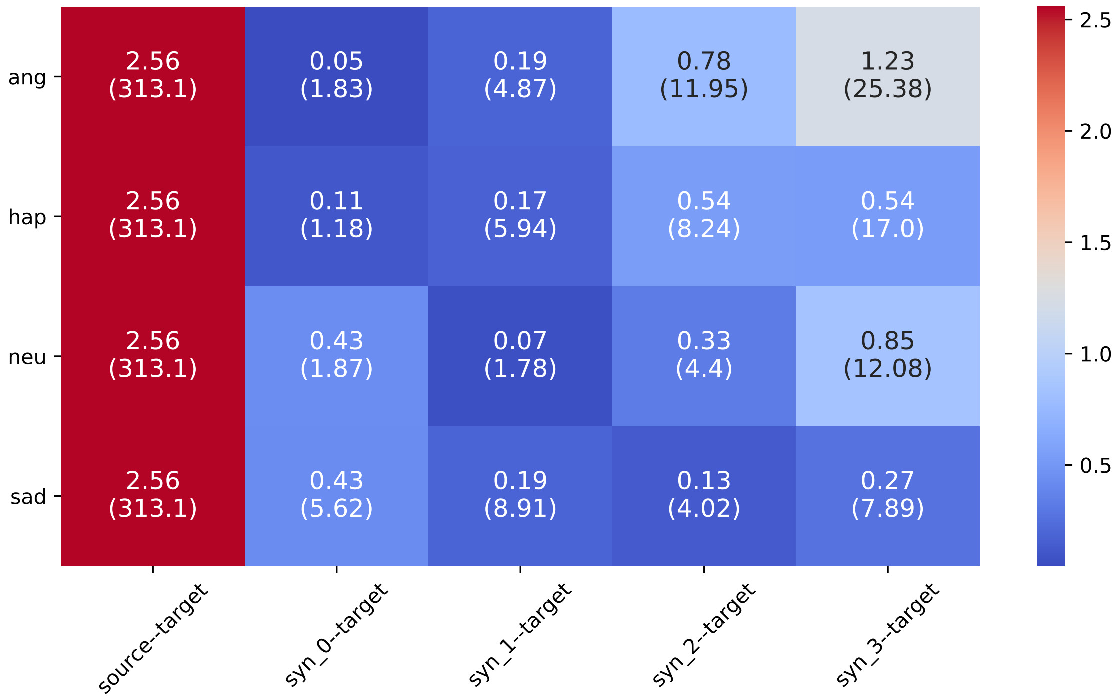


Figure 4.1: Overlap measures between datasets. The larger a value is, the less overlap two databases for an emotion have. The maximum values over all the feature dimensions are given in brackets. The corresponding average is given above the maximum.

Figure 4.1 and 4.2 illustrate the results of the two measures. The color differences in both heatmaps are based on the average over all the feature dimensions, although a maximum value is more appropriate for the second heatmap to show the separability between emotions. This is because we compare not only the values within each heatmap, but also the values between the two heatmaps. Therefore, a unified statistical value is required, but we also provide the maximum over all the feature dimensions in brackets.

In Figure 4.1, the horizontal axis stands for the datasets to be compared, the vertical axis displays the four emotions in the target and synthetic datasets. As we can see, the first column has the same value for each emotion. The reason is that the source dataset is unlabeled, therefore we compute the overlap between the whole source and target dataset, rather than for each emotion. It can be observed that the large gap between source and target is narrowed by each synthetic dataset, which means CycleGANs are capable of mapping the data from the source to the four target domains to some extent. But there exists large imbalance between the four mappings. For instance, in the case without classification loss, syn_0 has a much more similar distribution to the target for the emotions “angry” and “happy” than for “neutral” and “sad”. Furthermore, with the increase of the weight for classification loss, the synthetic and target datasets have generally less overlapping area, although some exceptions can be found, especially in the emotion “sad”.

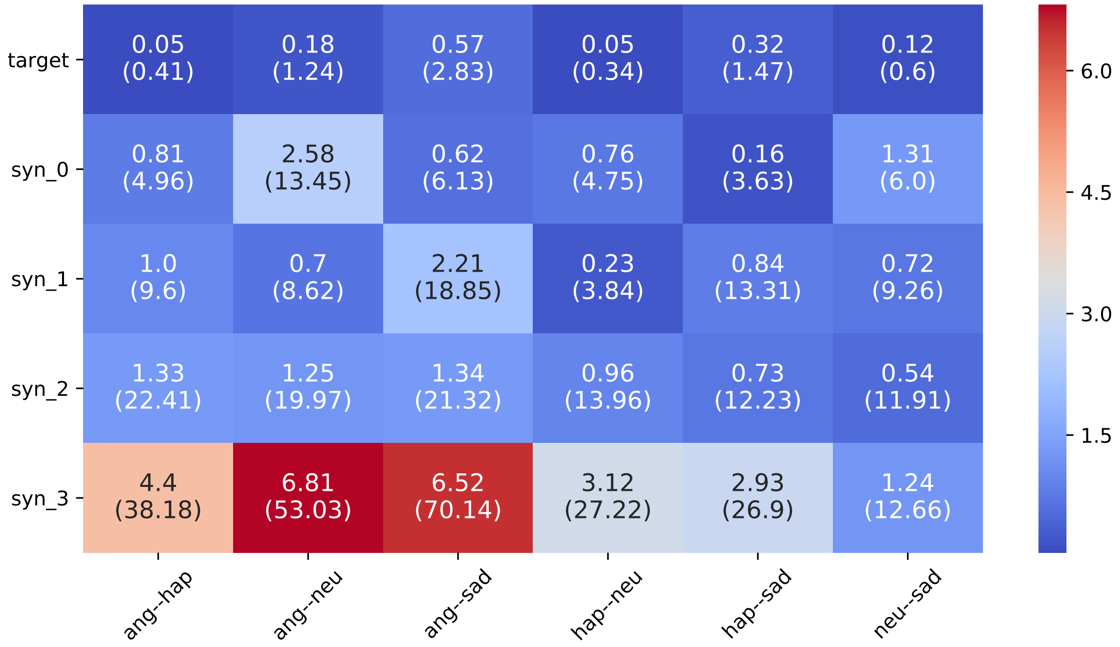


Figure 4.2: Overlap measures between emotions. The larger a value is, the less overlap two emotion classes within a database have. The maximum values over all the feature dimensions are given in brackets. The corresponding average is given above the maximum.

In Figure 4.2, the horizontal axis stands for the two emotions to be compared with, the vertical axis displays the target and four synthetic datasets. It can be seen that the larger the value of λ^{cls} is, the less overlapping area between two emotions exists, and therefore the more easily separable over emotions the datasets are. All the emotion pairs listed in Figure 4.2 have this property.

Another interesting observation is that for any synthetic dataset syn_i, the values in its column of the first heatmap are generally larger than the values in its row of the second heatmap, which means the difference between emotions is often larger than the difference between datasets. The large values in the row of syn_3 (Figure 4.2) reflect that the classification loss imposes the generated emotional data to be far away from the overlapping regions. Therefore, data with different emotions occur more separately.

4.4 Experiment 2: Within-corpus Evaluation

In this experiment, we augment the real training dataset with the synthetically generated datasets to check if the classification performance of SER can be improved within the IEMOCAP corpus.

	Real	**Syn.**	**Real + Syn.**
Learning rate	1×10^{-5}		5×10^{-6}
Number of epochs	70	5	30
Minibatch size	64	256	
Dropout	0.2	0.5	
Hidden layer size	[100, 100]	[200, 200]	[1000, 1000]
Optimizer	Adam		
Activation function	Leaky ReLU		

Table 4.2: Hyperparameters of within-corpus classification

4.4.1 Experimental Setup

We build three feed-forward neural network classifiers which are trained on: (1) only real samples taken from IEMOCAP, (2) only synthetic features and (3) the combination of both. We perform leave-one-session-out cross-validation on IEMOCAP to ensure that results are speaker-independent. Table 4.2 lists the hyperparameters for the three classifiers.

We report unweighted average recall (UAR) as performance measure. Since the neural networks are initialized with random weights, we repeat all experiments five times and report mean and standard deviation of the results.

4.4.2 Results

Table 4.3 shows our results for the cross-validation evaluation. For comparison, we use the results reported by Sahu et al. [SGE18] as a baseline which has the same experimental settings as we have. It can be seen that our classification result based only on real samples is comparable to [SGE18], although different types of classifiers are used[4].

	Real	**Syn.**	**Real + Syn.**
Sahu et al. [SGE18]	59.42	34.09	60.29
$\lambda^{\text{cls}} = 0$	**59.48 ± 0.71**	**51.57 ± 0.60**	58.79 ± 0.77
$\lambda^{\text{cls}} = 2$		46.59 ± 0.75	**60.37 ± 0.70**

Table 4.3: Results for cross-validation evaluation on IEMOCAP.

Using only synthetically generated samples as training data, we observe a significantly higher performance on the test set than reported in [SGE18]. For $\lambda^{\text{cls}} = 0$, the UAR arrives at 51.57%, which implies that our approach generates feature vectors that are closer to the underlying distribution of real data samples.

[4] Sahu et al. use SVMs which are not appropriate for our case, because the synthetic data generated by our method are not located in the overlapping regions between classes. We use neural network classifers.

We notice that the UAR for the setting with $\lambda^{\text{cls}} = 2$ is notably lower than for $\lambda^{\text{cls}} = 0$ when using only synthetic data as training samples. To gain a deeper understanding of the performance differences, we visualize the compressed representation of IEMOCAP data (Figure 4.3) and compare the confusion matrices for the classificaiton using only synthetic data (left-hand sides of Figure. 4.4b- 4.4c). It can be seen from Figure 4.3 that in the direction of the most principal component of the IEMOCAP dataset, the classes "angry" and "sad" are more easily separated than the other two classes[5]. Therefore, the synthetic "angry" and "sad" data generated by our method are more likely to be located on the "edges" of these two classes or even outside, while the synthetic "happy" and "neutral" data are more likely to be far away from them due to the classification loss. A classifier trained only on this synthetic dataset has rarely seen that "happy" data occur in the "angry" region and "neutral" data occur at the "sad" region. As a result, these "happy" and "neutral" data in the test set are easily mistaken as "angry" and "sad", respectively. That might also be the reason why the synthetic confusion matrix in Figure 4.4b shows a strong bias towards the class "angry" and "sad" [BNV19].

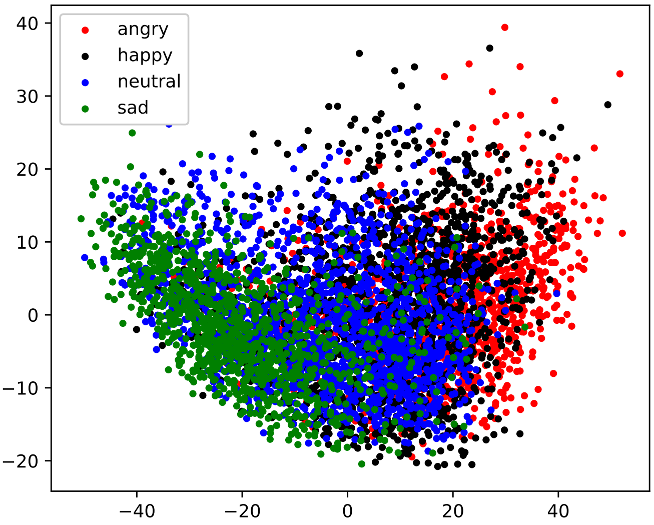
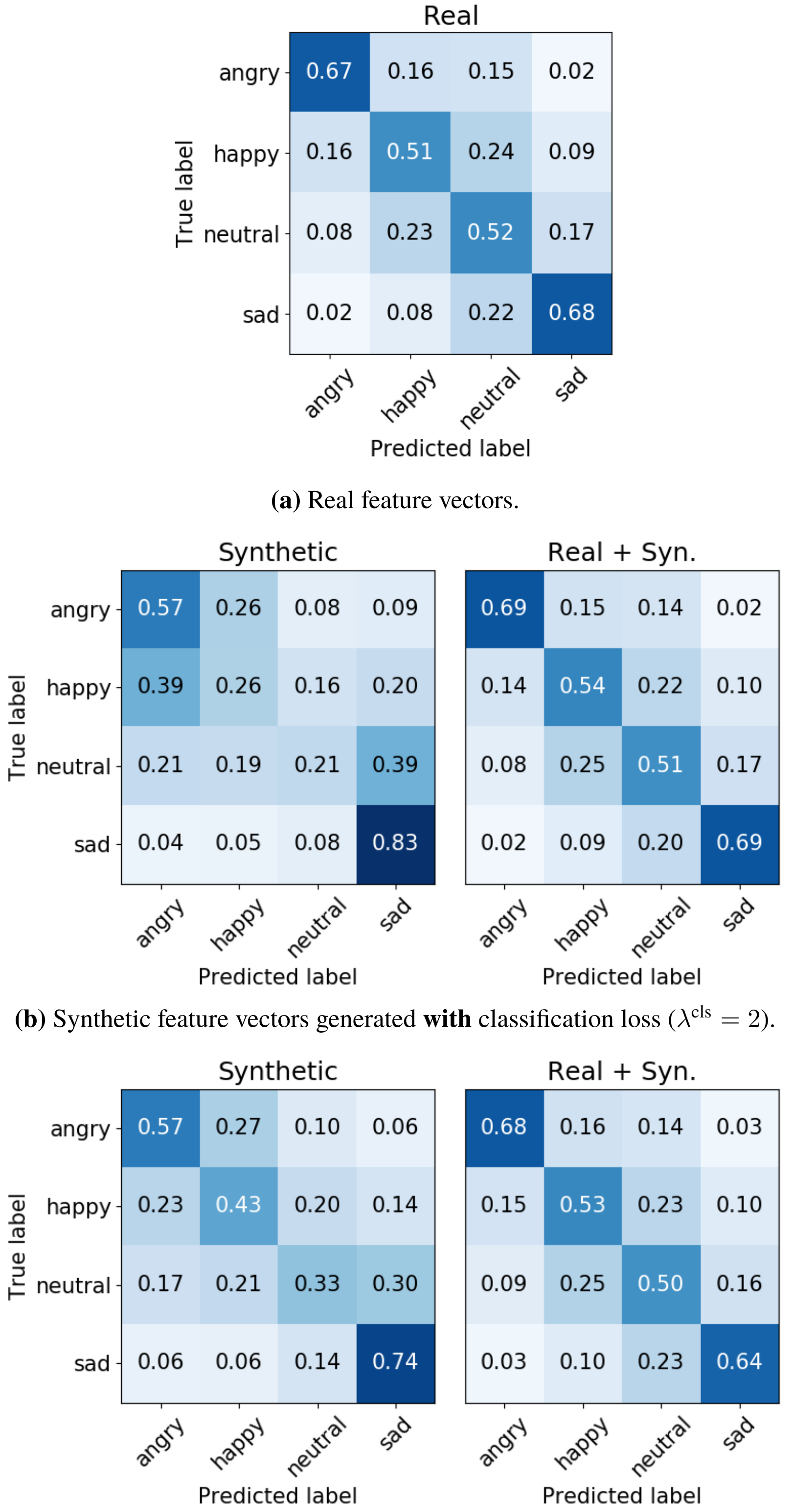


Figure 4.3: PCA representation of IEMOCAP data.

Augmenting the real dataset with the synthetic feature vectors generated with classification loss ($\lambda^{\text{cls}} = 2$), we achieve not only an improvement over the result based only on real dataset, but also slightly outperform the baseline [SGE18]. By contrast, the combination of the real data and the synthetic data generated without classification loss decreases the UAR to some extent.

It can be seen from the confusion matrices that the predictions and error patterns based on the augmented dataset (right-hand sides of Figure 4.4b and 4.4c) are similar to those based on the real dataset (Figure 4.4a). For the setting *with* classification loss (Figure 4.4b), we observe improvements for the three classes "angry", "happy" and "sad" – whereas in the setting *without* classification loss (Figure 4.4c), the result for the class "sad" drops below the corresponding value in Figure 4.4a [BNV19].

[5] This can be also verified when we compare the overlaps between different emotions in the target dataset (First line in Figure 4.2).



(a) Real feature vectors.

(b) Synthetic feature vectors generated **with** classification loss ($\lambda^{\text{cls}} = 2$).

(c) Synthetic feature vectors generated **without** classification loss ($\lambda^{\text{cls}} = 0$).

Figure 4.4: Averaged confusion matrices on IEMOCAP.

	Real	**Syn.**	**Real + Syn.**
Learning rate	1×10^{-5}		5×10^{-6}
Number of epochs	70	5	30
Minibatch size	64	256	
Dropout	0.2	0.5	
Hidden layer size	[50, 50]	[200, 200]	[200, 200]
Optimizer	Adam		
Activation function	Leaky ReLU		

Table 4.4: Hyperparameters of cross-corpus classification

These findings show that the proposed classification loss in our CycleGAN framework can in fact improve classification results for SER, but could potentially introduce a bias towards certain categories [BNV19]. In addition, we have recognized a strong overfitting problem when training *only* on synthetically generated feature vectors, which will be discussed in section 5.2.

4.5 Experiment 3: Cross-corpus Evaluation

In this experiment, we investigate whether the synthetic samples generated by our method can be also used to train a model for predicting emotions on another dataset. The cross-corpus evaluation requires more generalization ability of classfication models.

4.5.1 Experimental Setup

As in section 4.4, we train three neural network classifiers based on real data, synthetic data and the combination of both. The whole IEMOCAP dataset is used as training data and the MSP-IMPROV dataset as test data. Table 4.4 shows the hyperparameters for training the three classifers, they are almost the same as in Table 4.2, only the hidden layer size is reduced[6]. Since the training and test dataset in cross-corpus evaluation are more different than in within-corpus evaluation, a more generalized model for emotion classification should be learned. Otherwise, the model learned on the training data would be too complex to fit the test data and easily cause an overfitting problem.

4.5.2 Results

Table 4.5 shows the cross-corpus experimental results of the baseline [SGE18] and our method which varies the value of λ^{cls} from 0 to 3. It can be seen that all of our classfiers trained on only synthetic data have a higher UAR than the baseline, where the classifier with $\lambda^{\text{cls}} = 1$ performs the best. For the classifiers trained on the combination of both real and synthetic data, the UAR

[6] [BNV19] uses a more standard way for hyperparameter tuning based on the split of development and test dataset.

	Real	**Syn.**	**Real + Syn.**
Sahu et al. [SGE18]	45.14	33.96	45.40
$\lambda^{\text{cls}} = 0$	**45.58** ± **0.40**	39.74 ± 0.70	42.38 ± 0.30
$\lambda^{\text{cls}} = 1$		**41.59** ± **0.80**	44.58 ± 0.50
$\lambda^{\text{cls}} = 2$		40.68 ± 0.70	**46.00** ± **0.50**
$\lambda^{\text{cls}} = 3$		39.26 ± 0.70	45.29 ± 0.51

Table 4.5: Results for cross-corpus evaluation on MSP-IMRPOV.

for $\lambda^{\text{cls}} = 2$ is the best and outperforms the baseline. It can be concluded that the introduced classification loss is beneficial for cross-domain scenarios when assigned with a proper value to its weight λ^{cls}.

4.6 Experiment 4: Feature Selection

In this experiment, we aim to find the features that are most relevant with emotions and then train a classifier based on those selected features. If the redundant information that is not related with emotions can be filtered out, then the classification performance for the cross-corpus experiment can be further improved.

As mentioned in section 4.3, the synthetic emotional data generated by our method are likely to occur at the regions where they can be more certainly assigned to their emotion. Therefore, the difference between the synthetic data and the target data (IEMOCAP) stands for the changes required to enhance the emotional part of the data. The largest dimensions of the difference vector contain the most emotional information.

4.6.1 Experimental Setup

We calculate the difference vector between the syn_2 and the IEMOCAP dataset, then we extract the n largest feature dimensions of the difference vector. The difference is computed as overlap measures defined in Eq. 4.1. Besides that, we train two neural network classifiers based on the n selected features of (1) only real data and (2) the combination of real and synthetic data, respectively. The hyperparameters are the same as in Table 4.4. We set n to different values to observe the relationship between the classification performance and the number of selected features.

4.6.2 Results

Figure 4.5 shows the UAR of both classifiers. As we can see, the classifier based on the combination of real and synthetic data reaches its maximum (46.26%) when $n = 1000$, and the classifier based on only real data arrives at its maximum (45.41%) when $n = 1300$. It can be inferred that the last 582 features for the "real+syn" classifier and the last 282 features for the "real" classifier contain

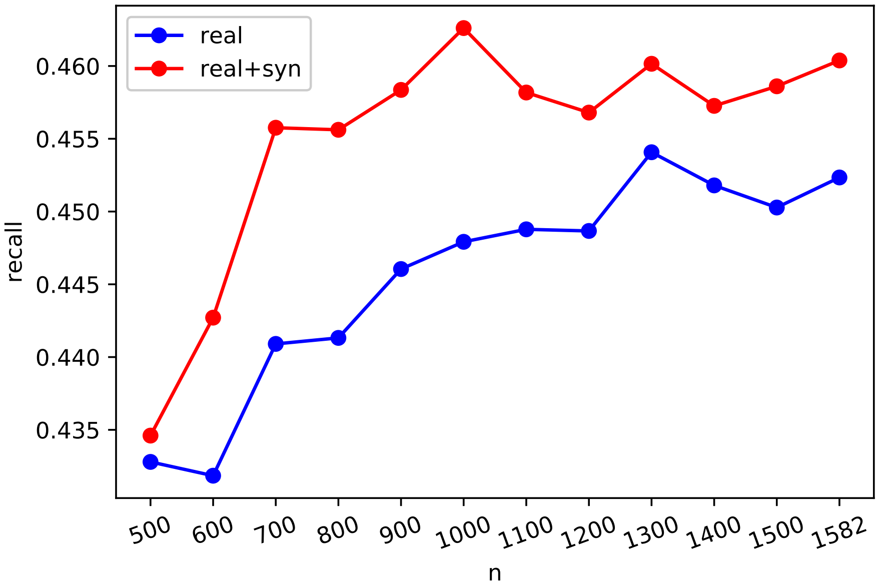


Figure 4.5: Classification results based on selected features.

possibly very little information about emotions. Therefore, we have proved that the synthetic data can not only be used for data augmentation but also for feature selection to improve the classification performance.

5 Discussion

In this chapter, we discuss two interesting problems met in the experiments above. The first problem is how to balance the similarity and discriminability in the process of synthetic sample generation. The second problem is why we have a strong overfitting when we train a classifier based only on synthetic samples.

5.1 Balance between Similarity and Discriminability

Our method has two goals for generating synthetic samples: (1) the synthetic samples should have a similar distribution to real data samples; (2) the synthetic samples should be separable. These two goals are also reflected on our overall loss function defined in Eq. 3.4, where the CycleGAN part (adversarial loss and cycle consistency loss) corresponds to the goal of similarity while the classification loss corresponds to the goal of discriminability. Therefore, the weight λ^{cls} controls which goal is more dominant.

It is clear that similarity and discriminality cannot be improved at the same time, because the distribution of real data samples is usually not separable. In the following subsections, we analyze two cases where only one of the two goals is considered.

5.1.1 Similarity without Discriminability

First, we only consider the similarity to real data samples. Although we have already the synthetic dataset syn_0 without classification loss, to further simulate the distribution of real data samples, we train an extra synthetic dataset based on the CycleGAN loss and a mean-std-loss. The mean-std-loss is defined as follows:

$$\mathcal{L}^{\text{mean-std}}(\mu, \sigma) = (\mu^{\text{T}} - \mu)^2 + (\sigma^{\text{T}} - \sigma)^2 \tag{5.1}$$

where μ and σ are the mean and standard deviation of the generated data, and μ^{T} and σ^{T} are the mean and standard deviation of the target dataset (IEMOCAP). We annotate the generated dataset as “syn_-1” for simplicity, although it uses mean-std-loss instead of setting classification loss to minus one.

Figure 5.1 shows the overlap measures between the target and two synthetic datasets (syn_-1 and syn_0). It can be seen that syn_-1 is very similar to the target dataset for each emotion. Compared with syn_0, syn_-1 has more overlaps with the target dataset.

We conduct the cross-corpus experiment described in section 4.5 on syn_-1 with the same setup. The experimental results are shown in Table 5.1.

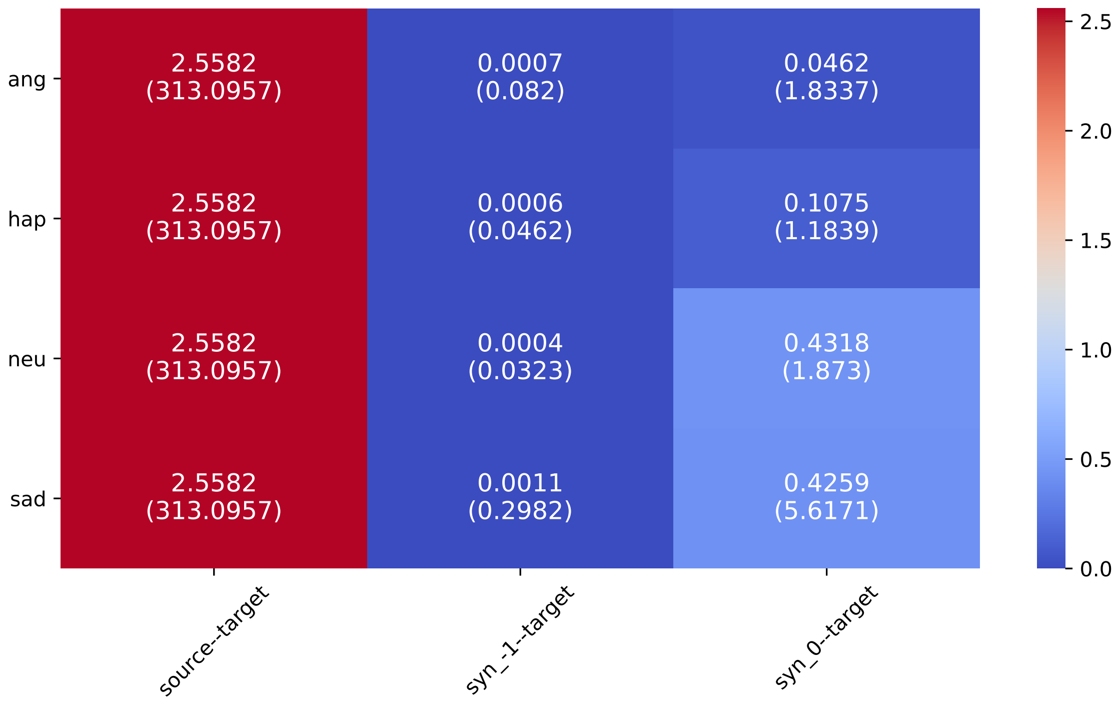


Figure 5.1: Overlap measures between target dataset and the generated data based on mean-std-loss. It is obvious that syn_-1 has more overlaps with the target than syn_0.

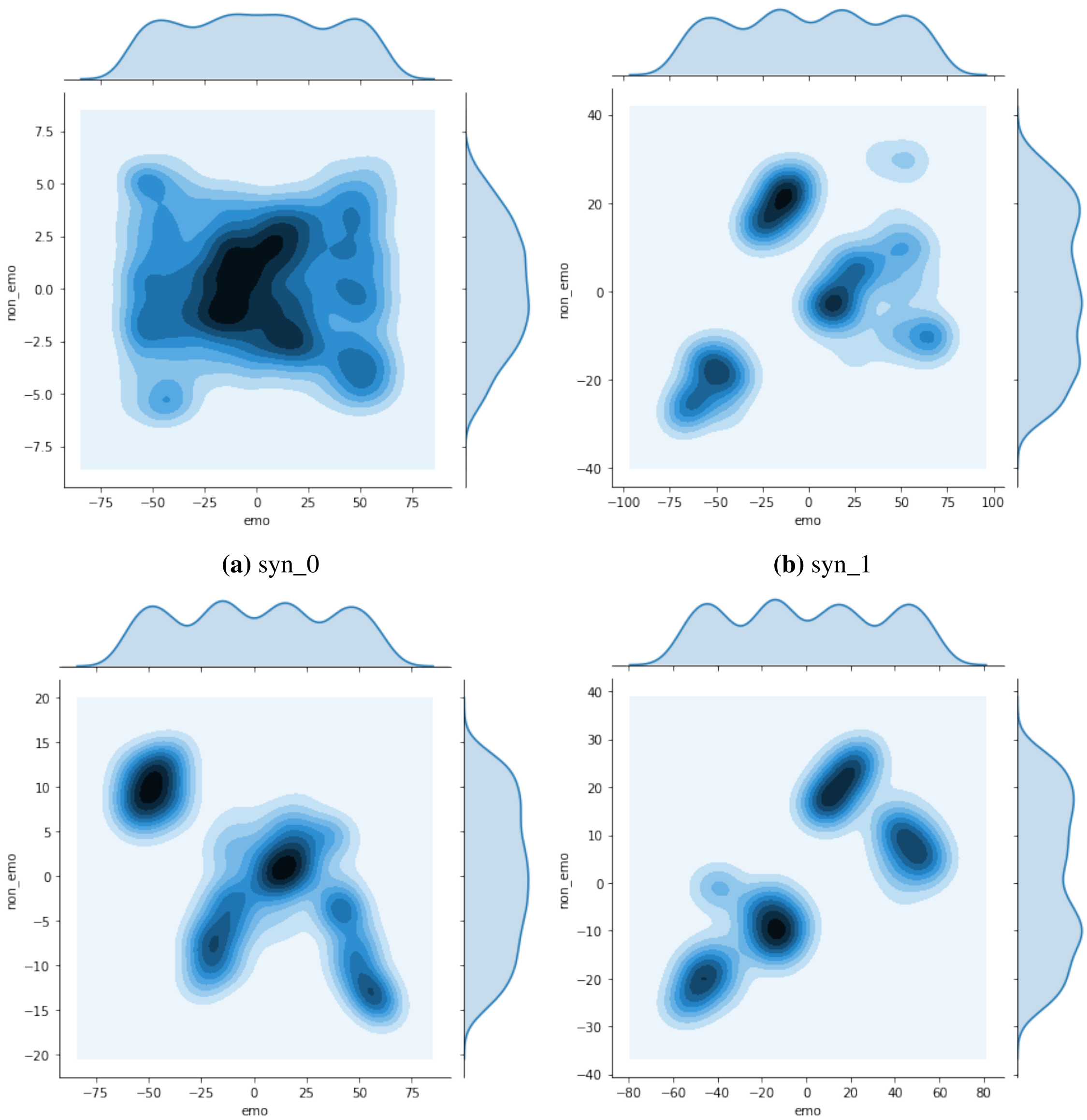
Syn.	Real + Syn.
42.82 ± 0.43	41.22 ± 0.46

Table 5.1: Results for cross-corpus evaluatioin on synthetic data generated with mean-std-loss.

When a classifier is trained only on synthetic data, syn_-1 achieves a better UAR than any other synthetic data in Table 4.5. However, when a classifier is trained on the combination of both real and synthetic data, syn_-1 has the lowest UAR, which is even lower than the performance of the classifier trained only on itself. Besides that, from their training and test loss, we see overfittings for both classifications, which means syn_-1 and the target dataset are so similar that there is less variety between the two datasets. Therefore, the classifier trained on the combination of them has no generalization ability to predict on other emotional databases.

5.1.2 Discriminability without Similarity

Now we only consider the discriminability between generated samples. Suppose the distribution of our emotional datasets can be considered a joint distribution of two high-level features, the one is emotional feature f_{emo} which contains all the emotional information, and the other is non-emotional feature $f_{\text{non\_emo}}$ which is irrelavant for emotions. When we increase λ^{cls}, it means we are trying to enhance the impact of f_{emo} and relatively weaken $f_{\text{non\_emo}}$.



(a) syn_0

(b) syn_1

(c) syn_2

(d) syn_3

Figure 5.2: 2d Joint distribution of f_{emo} and $f_{\mathrm{non\_emo}}$

In order to understand this more intuitively, we take syn_0, syn_1, syn_2 and syn_3 as examples. First of all, we use the overlap measures defined in Eq. 4.1 to compute their difference with target dataset (IEMOCAP). Then we extract the 30 largest feature dimensions and the 30 smallest ones. As mentioned in section 4.6, the difference between the target and the synthetic data generated by our method can be used to reflect emotional information. Therefore, the 30 largest dimensions act now as f_{emo} and the 30 smallest ones act as $f_{\text{non\_emo}}$. We use t-Distributed Stochastic Neighbour Embedding (t-SNE)[1] to reduce the dimension of both feature vectors from 30 to 1 individually. Then we plot their 2d joint distribution, as shown in Figure 5.2. The one-dimensional distribution above each subfigure is the distribution of f_{emo}, and the one at the right side of each subfigure is the distribution of $f_{\text{non\_emo}}$.

As we can see, the distribution of f_{emo} is close to a four component Gaussian Mixture Model, which represents the four emotion classes in the target dataset. With the increase of λ^{cls}, the variance of the Gaussians tends to be smaller, which means they are more easily separated from each other. The joint distribution is heavily impacted by the dominant dimension. It can be seen that the joint distribution in Figure 5.2d can be easily separated in four classes. Therefore, the larger the value λ^{cls} is, the easier classifier we need to train for the synthetic data. However, a too simple classifier may have a large bias problem on any test input dataset.

5.1.3 How to Reach Balance

The balance problem between similarity and discriminability can be regarded as the trade between variance and bias.

When we only consider to generate synthetic data with a similar distribution to the real data samples, a classifier trained on these synthetic data has a high variance, because it can be only applied to the datasets that are very similar to the target, otherwise it might suffer from overfitting problems.

When synthetic samples are generated only with classification loss, then a classifier trained on these synthetic data has a high bias, because the classifier can only learn a very simple model, which is also called underfitting problems.

Therefore, a balance between similarity and discriminality is neccessary. To reach the balance, we need to tune the weight λ^{cls}. It relates to the distribution of both training and test datasets. If the training data have a severe overlapping between classes, we should set a bigger λ^{cls} to prevent the high variance of a classifier trained on the synthetic data. If the training data don't have severe overlapping between classes, we can set a smaller value to λ^{cls} or even don't use the classification loss at all. The distribution of test data is unknown. However, when we know this is a within-corpus test, then the test data usually have a more similar distribution to the training data. Therefore, a smaller λ^{cls} is preferred. By contrast, for cross-corpus test, we need a bigger λ^{cls} to increase the generalization ability of classification models. In practice, a proper value of the weight is always different from task to task. Therefore, parameter tuning is requried. In our case, the balance is reached when $\lambda^{\text{cls}} = 2$.

[1] t-SNE is a dimension reduction approach that preserves the local distances between points in high dimensional space.

5.2 Overfitting Problem

A strong overfitting has been observed when we use only synthetic data to train a classification model. To specify the reason, we need to compare the distribution of the synthetic data and test data. Due to the classification loss, the synthetic data are easily separable, which makes the training error very small. The test data are however from complex real-world and therefore not easily separable.

To narrow the gap between the training and test data, we need to generate data which follow a more complex distribution. Since we have only one single labeled dataset as target, the only complex distribution we can learn from is this target dataset. However, as we mentioned in section 5.1.1, we cannot learn the target dataset exactly, otherwise the classfication model trained on the synthetic data would lose its generalization ability on other datasets that are not similar to the target dataset. This is also the motivation why we introduce the classification loss.

Therefore, to learn a complex emotional data distribution but not exactly the distribution of the single target dataset, we need data from more different sources to ensure the diversity of distribution between training data.

At last, we need to point out that although there exists a large gap between training and test errors, the UAR of our method based on only synthetic data has a significant improvement over previous work. So our method is still applicable to data augmentation for SER, especially when only one single labeled database is available.

6 Conclusion

Data augmentation via generative adversarial networks is a relevant topic for SER. In contrast to previous methods which generate synthetic feature vectors from a low-dimensional space, we propose a CycleGAN-based method to transfer unlabeled data into different target emotions. Our experiments have shown a considerable similarity between the distribution of synthetic and target feature vectors. Furthermore, we introduced a classification loss to the network architecture to enable the synthetic samples to be distinguishable. Experiments on IEMOCAP and MSP-IMPROV have shown improvements in classification performance over previous methods when training on synthetic features as well as on the combination of real and synthetic samples. In addition, we investigate the balance problem between similarity and discriminability as well as the overfitting problem in the training process using only synthetic data. For future work, possible directions are utilizing several speech emotion corpora as target data for CycleGAN training and using the proposed method to generate synthetic samples in data space rather than in feature space.

Bibliography

[ABC+16] M. Abadi, P. Barham, J. Chen, Z. Chen, A. Davis, J. Dean, M. Devin, S. Ghemawat, G. Irving, M. Isard, et al. “Tensorflow: a system for large-scale machine learning.” In: *OSDI*. Vol. 16. 2016 (cit. on p. 31).

[ARS+18] A. Almahairi, S. Rajeswar, A. Sordoni, P. Bachman, A. Courville. “Augmented cyclegan: Learning many-to-many mappings from unpaired data”. In: *arXiv preprint arXiv:1802.10151* (2018) (cit. on p. 22).

[Asi94] I. Asimov. *The robots of dawn*. Spectra, 1994 (cit. on p. 7).

[BBL+08] C. Busso, M. Bulut, C.-C. Lee, A. Kazemzadeh, E. Mower, S. Kim, J. N. Chang, S. Lee, S. S. Narayanan. “IEMOCAP: Interactive emotional dyadic motion capture database”. In: *Language resources and evaluation* 42.4 (2008) (cit. on p. 29).

[BNV19] F. Bao, M. Neumann, N. T. Vu. “CycleGAN-based emotion style transfer as data augmentation for speech emotion recognition”. In: *Manuscript submitted for publication* (2019) (cit. on pp. 35, 37).

[BPB+17] C. Busso, S. Parthasarathy, A. Burmania, M. AbdelWahab, N. Sadoughi, E. M. Provost. “MSP-IMPROV: An acted corpus of dyadic interactions to study emotion perception”. In: *IEEE Transactions on Affective Computing* 8.1 (2017) (cit. on p. 29).

[CCK+18] Y. Choi, M. Choi, M. Kim, J.-W. Ha, S. Kim, J. Choo. “Stargan: Unified generative adversarial networks for multi-domain image-to-image translation”. In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. 2018, pp. 8789–8797 (cit. on p. 22).

[CS17] J. Chang, S. Scherer. “Learning representations of emotional speech with deep convolutional generative adversarial networks”. In: *2017 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE. 2017, pp. 2746–2750 (cit. on p. 15).

[CXYT18] X. Chen, C. Xu, X. Yang, D. Tao. “Attention-GAN for Object Transfiguration in Wild Images”. In: *The European Conference on Computer Vision (ECCV)*. Sept. 2018 (cit. on p. 21).

[EKK11] 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 (cit. on p. 7).

[EWGS13] 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 (cit. on pp. 22, 30).

[FRL+18] W. Fedus, M. Rosca, B. Lakshminarayanan, A. M. Dai, S. Mohamed, I. Goodfellow. “Many paths to equilibrium: GANs do not need to decrease a divergence at every step”. In: *International Conference on Learning Representations (ICLR)* (2018) (cit. on p. 14).

[GBC16] I. Goodfellow, Y. Bengio, A. Courville. *Deep learning*. MIT press, 2016 (cit. on p. 14).

[GEB16] L. A. Gatys, A. S. Ecker, M. Bethge. “Image style transfer using convolutional neural networks”. In: *Proceedings of the IEEE conference on computer vision and pattern recognition*. 2016, pp. 2414–2423 (cit. on p. 21).

[Goo16] I. Goodfellow. “NIPS 2016 tutorial: Generative adversarial networks”. In: *arXiv preprint arXiv:1701.00160* (2016) (cit. on pp. 11–14).

[GPM+14] 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 (cit. on pp. 8, 14).

[HB00] T. K. Ho, M. Basu. “Measuring the complexity of classification problems”. In: *Proceedings 15th International Conference on Pattern Recognition. ICPR-2000*. Vol. 2. IEEE. 2000, pp. 43–47 (cit. on p. 31).

[HLBK18] X. Huang, M.-Y. Liu, S. Belongie, J. Kautz. “Multimodal unsupervised image-to-image translation”. In: *Proceedings of the European Conference on Computer Vision (ECCV)*. 2018, pp. 172–189 (cit. on p. 21).

[HZR+18] J. Han, Z. Zhang, Z. Ren, F. Ringeval, B. Schuller. “Towards conditional adversarial training for predicting emotions from speech”. In: *2018 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE. 2018, pp. 6822–6826 (cit. on p. 15).

[IZZE17] P. Isola, J.-Y. Zhu, T. Zhou, A. A. Efros. “Image-to-image translation with conditional adversarial networks”. In: *Proceedings of the IEEE conference on computer vision and pattern recognition*. 2017, pp. 1125–1134 (cit. on p. 21).

[Ken04] E. A. Kensinger. “Remembering emotional experiences: The contribution of valence and arousal”. In: *Reviews in the Neurosciences* 15.4 (2004), pp. 241–252 (cit. on p. 19).

[KPPK15] T. Ko, V. Peddinti, D. Povey, S. Khudanpur. “Audio augmentation for speech recognition”. In: *Sixteenth Annual Conference of the International Speech Communication Association*. 2015 (cit. on p. 11).

[MO14] M. Mirza, S. Osindero. “Conditional generative adversarial nets”. In: *arXiv preprint arXiv:1411.1784* (2014) (cit. on p. 17).

[MSJ+16] A. Makhzani, J. Shlens, N. Jaitly, I. Goodfellow, B. Frey. “Adversarial autoencoders”. In: *Proc. of International Conference on Learning Representations (ICLR)*. 2016 (cit. on p. 16).

[NV17] M. Neumann, N. T. Vu. “Attentive convolutional neural network based speech emotion recognition: A study on the impact of input features, signal length, and acted speech”. In: *arXiv preprint arXiv:1706.00612* (2017) (cit. on p. 19).

[RDE14] A. Rousseau, P. Deléglise, Y. Esteve. "Enhancing the TED-LIUM Corpus with Selected Data for Language Modeling and More TED Talks." In: *Proc. of the Ninth International Conference on Language Resources and Evaluation (LREC-2014)*. 2014 (cit. on p. 29).

[SCCL06] Y. Schnitman, Y. Caspi, D. Cohen-Or, D. Lischinski. "Inducing semantic segmentation from an example". In: *Asian conference on computer vision*. Springer. 2006, pp. 373–384 (cit. on p. 21).

[Sch18] B. W. Schuller. "Speech emotion recognition: Two decades in a nutshell, benchmarks, and ongoing trends". In: *Communications of the ACM* 61.5 (2018), pp. 90–99 (cit. on p. 7).

[SGE18] S. Sahu, R. Gupta, C. Espy-Wilson. "On Enhancing Speech Emotion Recognition Using Generative Adversarial Networks". In: *Proc. of Interspeech*. 2018 (cit. on pp. 17, 23, 29, 34, 35, 37, 38).

[SGS+17] S. Sahu, R. Gupta, G. Sivaraman, W. AbdAlmageed, C. Espy-Wilson. "Adversarial Auto-encoders for Speech Based Emotion Recognition". In: *Proc. of Interspeech*. 2017 (cit. on p. 16).

[SGZ+16] 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 (cit. on pp. 14, 31).

[SSB+10] 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 (cit. on p. 30).

[TKL06] J. Tao, Y. Kang, A. Li. "Prosody conversion from neutral speech to emotional speech". In: *IEEE Transactions on Audio, Speech, and Language Processing* 14.4 (2006), pp. 1145–1154 (cit. on pp. 23, 24).

[WP17] J. Wang, L. Perez. "The effectiveness of data augmentation in image classification using deep learning". In: *Convolutional Neural Networks Vis. Recognit* (2017) (cit. on p. 11).

[ZL15] X. Zhang, Y. LeCun. "Text understanding from scratch". In: *arXiv preprint arXiv:1502.01710* (2015) (cit. on p. 11).

[ZPIE17] 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 (cit. on pp. 18, 19, 21, 23, 24).

[ZWWS11] 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 (cit. on p. 31).

List of Figures

List of Tables

Erklärung

Ich versichere, diese Arbeit selbstständig verfasst zu haben. Ich habe keine anderen als die angegebenen Quellen benutzt und alle wörtlich oder sinngemäß aus anderen Werken übernommene Aussagen als solche gekennzeichnet. Weder diese Arbeit noch wesentliche Teile daraus waren bisher Gegenstand eines anderen Prüfungsverfahrens. Ich habe diese Arbeit bisher weder teilweise noch vollständig veröffentlicht. Das elektronische Exemplar stimmt mit allen eingereichten Exemplaren überein.

Ort, Datum, Unterschrift