Data Augmentation in Emotion Classification using Generative Adversarial Networks

Xinyue Zhu\textsuperscript{1,2}, Yifan Liu\textsuperscript{2}, and Zengchang Qin\textsuperscript{*2}

\textsuperscript{1}School of Electronic Engineering
Beijing University of Posts and Telecommunications, Beijing 100876, China
\textsuperscript{2}Intelligent Computing and Machine Learning Lab, School of ASEE
Beihang University, Beijing 100191, China
\textsuperscript{*}zxysee@bupt.edu.cn,
\{yifan_liu, zcqin\}@buaa.edu.cn

Abstract. It is a difficult task to classify images with multiple labels only using a small number of labeled samples and especially, with unbalanced distribution. In this paper we propose a data augmentation method using generative adversarial networks (GAN), that can complement and complete the data manifold, assist the classifier better to find margins or hyper-planes of neighboring classes, and finally lead to a better performance in emotion classification task. Specifically, we design a pipeline containing a CNN model as classifier and a cycle-consistent adversarial networks (CycleGAN) to generate supplementary data from given classes. In order to avoid gradient vanishing, we apply a least-squared distance in least squares generative adversarial networks (LSGAN) to adversarial loss. We also propose several evaluation methods on three benchmark facial expression datasets to validate GAN's contribution in data augmentation. Qualitative evaluation indicates that data manifolds show a significant improvement in distribution integrity and margin clarity between classes. Quantitative comparisons show that we can obtain 5\%\textasciitilde10\% increase in the classification accuracy after employing our data augmentation technique.

Keywords: Data augmentation, Emotion classification, CycleGAN, LSGAN

1 Introduction

With the recent rise in high capacity of deep neural network, large labeled training datasets are becoming increasingly important. However, labeled datasets are hard to get. This is a general problem in machine learning and computer vision (CV) applications. In this case, synthesizing images to supplement training corpus and automatically obtain samples with specific features and given labels becomes a viable solution. Data augmentation is commonly applied means for enlarging image datasets, as training network with a large number of weights and variables would easily get over-fitting if insufficient training samples were provided. Traditional data augmentation methods such as geometric transformation
and RGB channels alteration [3][15][26] do greatly improve training performance of some datasets with inadequate data or imbalanced distribution. However, they contribute little to supplement the data manifold since only image-level samples are generated in this process. In our paper, a new method of data augmentation is proposed through Generative Adversarial Networks (GAN) in order to generate new samples from feature level, thus to supplement the data manifold from the true sense and lead to more clear margins of different distributed data.

As GANs have been developed to generate compelling natural images, we attempt to explore whether GAN-generated images can help enlarge original dataset and balance distribution as a way of data augmentation. GANs are used to generate images through an adversarial training procedure that learns the real data distribution. This 'fooling' and 'generating’ network is frequently applied in manipulating images for computer vision applications [4][13][16] but few works have been done in classification tasks for data augmentation use. Here we propose a simulated + semi-supervised learning approach, whose goal is to transfer the unlabeled data to the labeled domain.

More specifically, we build a basic convolutional neural network (CNN) classifier for image classification and train a CycleGAN model [30] with least-squared loss [21] to achieve image-to-image transformation. Our aim is to explore the effect of data augmentation using GAN, so we build a relatively shallow CNN model rather than an extremely powerful one, which is only requested to extract general features for each class and have a certain ability to distinguish among them. Contrary to this, much effort is paid to constructing GAN model and improving its performance in generating images of specific classes. CycleGAN is used because it is a proper model for image translation between two unpaired domains, which is in line with our purpose: to generate target images with insufficient samples from large reference samples.

In our research, we first train a classifier using original samples as our baseline. After that, we select one or more classes as our to-be-generated classes. In order to take advantage of existing samples, we choose a class as our reference class which has a large sample size. After successfully training the CycleGAN model, we export the graph and add generated images to original dataset before retraining the classifier.

The main contributions can be summarized as follows.

− We propose a pipeline for data augmentation by using GAN to generate supplementary data in emotion classification task, and classification accuracy is significantly improved compared to the baseline.
− We combine least-squared loss from LSGAN with original adversarial loss in CycleGAN to avoid the problem of vanishing gradients, and this application performs well during the training process.
− We analyze the problem of imbalanced distribution in classification task, and CycleGAN’s positive role to resolve this problem.
− We show the GAN’s ability of supplementing data manifold from the true sense, that is considered better than traditional data augmentation methods.
By possessing a more complete data manifold, the classifier can learn to find margins or hyper-planes between neighboring classes better.

2 Related Work

2.1 Learning with Imbalanced Emotion Datasets

Facial expression recognition has attracted much attention in computer vision in past few decades. Current techniques related to facial expression mainly focus on recognizing seven prototypical emotions (neutral, happy, surprised, fear, angry, sad, and disgusted), which are considered basic and universal emotions for human. Such recognition or classification are sometimes very difficult since there is only a slight difference between different emotions, which requires an efficient and subtle feature extractor to be trained. Moreover, in [23], the author pointed out the unbalanced distribution among emotion classes, which leads to low accuracy in classes with fewer samples. To deal with unbalanced datasets or the class imbalance problem, many methods were proposed such as undersampling [18], synthesizing minorities [2], creating 'box' around minorities [8] and etc. Different from these works, we aim to resolve this problem by generating minorities from low-dimensional manifold, which improves the data distribution from feature level.

2.2 Generative Adversarial Networks

Generative Adversarial Networks provide a way to learn deep representations through a competitive process involving a pair or pairs of networks. From the first model of GAN [9] in 2014, variant models based on GAN were proposed including conditional GAN(CGAN) [22], deep convolutional GAN(DCGAN) [24], adversarial learned inference(ALI) [7], Wasserstein GAN(WGAN) [1], improved Wasserstein GAN(WGAN-GP) [11], cycle-consistent GAN(CycleGAN) [30], least squares GAN(LSGAN) [21], Triple GAN [17]. Generative adversarial nets are now widely used in several image tasks such as single image super-resolution [16], image manipulation [29] and synthesis [4], image-to-image translation [13] and etc. In our paper, instead of continuing to work on these applications, we focus on data augmentation using GANs, whose generator is able to produce additional data given specific labels by learning a mapping from low-dimensional manifold to high-dimensional space.

2.3 Data Augmentation

In the field of deep learning, where the scale of dataset has a great influence on the final outcome, data augmentation is often used to expand the training corpus. As for the existing techniques of data augmentation, they can be grouped into two main types: a) geometric transformation which is relatively generic and computationally cheap and b) task-specific or guided-augmentation
methods which are able to generate synthetic samples given specific labels [6]. In the case of image classification, the first group of data augmentation methods always focus on generating image data through label-preserving linear transformations (translation, rotation, scaling, horizontal shearing) such as Affine [3], elastic deformations [26], patches extraction, RGB channels intensities alteration [15] and etc. However, if we look deeper into these methods, they only lead to an image-level transformation through depth and scale and actually not helpful for dividing a clear boundary of data manifold. Such data augmentation does not improve data distribution which is determined by higher-level features. Within the second group, more complex manually-specified augmentation schemes are proposed. For instance, in [12] authors proposed an approach to learn multivariate normal distribution of each class in the whole mean manifold and researchers in [6] designed an attribute-guided augmentation in feature space. And in the field of 3D motion capture, 2D images are used for generating 3D ones such as [25]. Our approach aims to solve similar task in [12] but is very different from all these methods above. In this paper, new training corpus is generated from advanced Generative Adversarial Networks, which are different from original images but remain high-level features extracted from them.

3 Data Augmentation using CycleGAN

In our pipeline (Fig.1) doing data augmentation using GAN (DAG), a CycleGAN model with least-squared loss is used to generate synthetic images as supplementary data, which is added with original data to complete classification task.

3.1 Cycle-Consistent Adversarial Networks

In our work, CycleGAN [30] is used to realize unpaired image-to-image translation, learning mapping functions between images of reference class and of target class, namely domain $R$ and $T$. We use generator $G$ and $F$ to achieve domain transfer $G: R \rightarrow T$, $F: T \rightarrow R$, and discriminator $D_R$ and $D_T$, where $D_R$ aims to distinguish between images $R$ and translated images $F(T)$ and $D_T$ ditto. Although the generation from target images to reference ones is not required for our task, this bidirectional mapping is helpful to prevent the mode collapse problem since additional restriction is added in this process. Same with [30], the objective contains two terms: an adversarial loss for distribution matching and a cycle-consistency loss to guarantee the cycle-consistency. As for adversarial loss, $G$ tries to generate $G(r)$ which is so similar to $t$ that can fool the discriminator $D_T$, therefore the loss related to $G$ and $D_T$ is:

$$L(G, D_T, R, T) = E_{t \sim p_{data}(t)}[log D_T(t)] + E_{r \sim p_{data}(r)}[log(1 - D_T(G(r)))]$$

(1)

However, this log form makes training and convergence difficult since it is likely to cause gradient vanishing problem [1]. Here we apply a least-squared loss proposed
Fig. 1: Our pipeline of data augmentation using CycleGAN (DAG). A CNN classifier(top) and a CycleGAN model(bottom) make up two main components of the pipeline. Both reference images and target images are collected from the original data and flow into the CycleGAN work as Domain $R$ and $T$ respectively. Supplementary data is generated through generator $G$. After that, a CNN classifier is trained using original data and supplementary data as input. In CycleGAN model, $L_R$ is the LSGAN loss relative to Reference domains and $L_T$ is the LSGAN loss relative to Target domains. Besides, a cycle loss, namely $L_{cyc}$, is calculated to keep cycle consistency of the whole model.

In LSGAN [21] to avoid this phenomenon and maintain the same function as adversarial loss in original CycleGAN. For Domain $R$:

$$L_{LSGAN}(G, D_R, T, R) = E_{r \sim p_{data}(r)}[(D_R(r) - 1)^2] + E_{t \sim p_{data}(t)}[D_R(G(t))^2] \quad (2)$$

And for Domain $T$:

$$L_{LSGAN}(G, D_T, R, T) = E_{t \sim p_{data}(t)}[(D_T(t) - 1)^2] + E_{r \sim p_{data}(r)}[D_T(G(r))^2]$$

Therefore, the final loss is:

$$L(G, F, D_S, D_R) = L_R + L_T + L_{cyc}$$

$$= L_{LSGAN}(G, D_R, S, R) + L_{LSGAN}(F, D_S, R, S) + \lambda L_{cyc}(G, F)$$

The cycle consistency loss, namely $L_{cyc}$ in the full objective, is defined as:

$$L_{cyc}(G, F) = E_{r \sim p_{data}(r)}[\|F(G(r)) - r\|_1] + E_{t \sim p_{data}(t)}[\|G(F(t)) - t\|_1]$$
where $||\cdot||_1$ is the L1 norm. With these loss functions, the final functions we aim to solve is:

$$G^*, F^* = \arg \min_{F,G} \max_{D_T, D_R} L(G, F, D_T, D_R)$$

Details of CycleGAN can be referred to [30]

3.2 Class Imbalance and Data Manifold

When the classes have imbalanced distribution, the classifier prone to learn biased separators between classes. Take a binary classification in one-dimensional sample as an example. In Fig.2.a, class 1 and 2 are both generated from Gaussian distribution with a standard deviation of one, and has the mean of $\mu_1$ and $\mu_2$ respectively. Ideally, the separating line $x=(\mu_1+\mu_2)/2$, namely $S_t$ can best distinguish between these two classes. However, an imbalanced distribution in two classes will result in a biased separating line $S_r$ moving towards the minor class, since given samples are insufficient to form a correct margin with minimized loss. Some detailed discussions can be referred to [27].

![Fig. 2: (a) is a binary classification in one-dimensional sample. (b) and (c) are data manifolds and according margins with imbalanced (a) and balanced (b) distribution.](image)

Now back to our facial expression classification task. Under the assumption that image samples lie on a submanifold in a high-dimensional space, image classification task is actually a task to explore the underlying geometric structure of data distribution, thus to find best-split hyper-planes in this space. These hyper-planes divide the space into several parts according to margins, each represents a clustering of a specific class. (Fig.2.c) This process is similar to learn a separator to properly split spaces into two parts in the scenario introduced above.

When the dataset is imbalanced, it is much likely to form an incomplete manifold since in the same space, minorities are distributed more sparcely in their regions. In this case, biased margins or hyper-planes are learned, making it a difficult task for classifier to predict correct labels for given instances. (Fig.2.b) Although some data augmentation techniques mentioned in 2.1 and 2.3 can
alleviate this problem from several aspects, the most essential solution is to further complement and complete the data manifold.

So what is the role of CycleGAN in doing data augmentation? While the original GAN [9] learns a mapping from low-dimensional manifold (directly determined by noise $z$) to high-dimensional data spaces, the CycleGAN, as a tool for translation between two domains of both high-dimensional data, need to learn a low-dimensional manifold and also the parameters to map it back to high-dimensional space. Here we use $P_r$ and $P_t$ to represent real distribution of domain $R$ and $T$, and $M$ is the low-dimensional manifold. As domain $R$ has only a small number of training samples, when it is projected into low dimension, sparse distribution cannot form a complete $M$ with efficient feature information. A CycleGAN is then introduced. Sufficient samples in $P_r$ lead to a more complete $M$ and $G_\theta$ is learned through minimizing distance between $G(r)$ and $T$ (e.g., $\|G(r) - T\|^2$) to ‘pull’ $M$ to $P_t$. If we again project $P_t$ into manifold $M$, they will form a meaningful feature-level manifold thanks to the generated samples.

4 Experimental Studies

4.1 Toy Data Experiment

Before doing experiments on facial expression datasets, we first validate GAN’s role in completing data manifold on a toy dataset. We use different two-dimensional Gaussian distributions $(x, y) \sim N_m(\mu_m, \sigma_m), m \in \{1, 2, 3\}$ to simulate the distribution of three classes of data, where $\mu_1=[0, 6], \mu_2=[6.5, 7], \mu_3=[2, 2]$ and the covariance matrix is $([[2, 1], [1, 2]])$ for all three distributions. Imbalanced dataset is artificially created by randomly sampling 1000, 1000 and 100 points from each class respectively for training and 3*100 points for testing. A simple support vector machine (SVM) with linear kernal [14] is applied to classification task, which learns margins between neighboring classes. After that, we train a CycleGAN to generate 900 target class (minority class) from reference class (one of majority classes) and these supplementary samples are added to the original dataset and trained on the same SVM classifier.

We draw two figures (Fig.2.b and c) to show the data distribution and learned margins before and after adding the CycleGAN-generated samples. The original biased margins in imbalanced dataset (b) show a clear movement to more correct ones in (c). Moreover, we calculate the prediction accuracy on 300 testing corpus based on these two models and obtain an increase from 93.3% to 98.0%. Although the distribution and dimension of data in this toy experiment is much simpler than real image data, the results are stronger enough to support the positive role of GAN in improving data manifold in imbalanced datasets by doing data augmentation.

4.2 Datasets

In our experiment, three benchmark datasets are selected: Facial Expression Recognition Database(FER2013) [10], Static Facial Expressions in the Wild
(SFEW) [5] and The Japanese Female Facial Expression (JAFFE) Database [19]. All these datasets contain 7 types of face emotion including ‘angry’, ‘disgust’, ‘fear’, ‘happy’, ‘sad’, ‘surprise’, and ‘neutral’ (labeled 0~7 during training and testing process). Samples from FER2013 database are shown in Fig.3(left) as an example. The distribution of this dataset is imbalanced and in order to fully utilize this imbalanced dataset, several schemes for training and evaluation are provided. We sample the images in equal proportions by 20% for each class in FER2013 since training on an oversized datasets is not our original intention. SFEW and JAFFE, though have basically average distribution among classes, only have a small number of samples, about 50~200 per class. During the training process of CycleGAN, we choose ‘neutral’ class as our reference class and the other six are regarded as target ones, since it is natural to generate faces with emotion from non-emotional ones.

![Generated images](image)

Fig. 3: The original samples and generated samples of each classes. The left two column is original datasets and the rest is generated one. The neutral class, as reference class, has no generated samples in our experiment.

4.3 Results

We first train a CNN model based on original FER2013 datasets (20% sampled) as our baseline and the result is shown in Table.1

In order to get the most intuitive result, we choose class ‘disgust’ and ‘sad’ from FER2013 as our target classes, which are much smaller than the other classes and as a result, cannot obtain sufficient learning and optimizing, thus reach a
relatively low accuracy when trained on the baseline. (See Table.1, baseline) In this case, two CycleGANs are trained to generate ‘disgust’ and ‘sad’ images respectively (See Fig.3), and then are filled into the original datasets to balance the distribution and complete the data manifold. See Table.1 for testing results.

| Class   | Accuracy-2000(%) | Accuracy-4000(%) |
|---------|------------------|------------------|
|         | baseline +disgust +sad | baseline +disgust +sad |
| All     | 91.04 94.25 94.65 | 90.77 93.82 94.32 |
| angry   | 93.70 93.71 93.05 | 93.47 93.36 92.89 |
| disgust | 73.91 91.30 95.65 | 79.62 88.89 94.44 |
| fear    | 90.88 92.18 94.46 | 90.38 91.43 94.58 |
| happy   | 91.87 96.34 93.70 | 91.75 96.37 94.21 |
| sad     | 87.86 93.61 97.44 | 89.22 93.26 94.61 |
| surprise| 94.27 99.12 96.48 | 93.46 97.09 96.85 |
| neutral | 89.55 91.94 94.63 | 88.24 93.06 94.48 |

Table 1: Accuracy of both baseline model (CNN) and our pipeline (CNN+CycleGAN). ‘2000’ and ‘4000’ after ‘Accuracy’ represent the number of all testing samples. Besides, ‘+disgust’ or ‘+sad’ represents adding generated samples of class ‘disgust’ or ‘sad’ into the baseline.

From the Table 1, it is clear that (a) the accuracy of whole classes is improved and (b) accuracy of target class raise greatly and it is worth mentioning that (c) the accuracy of reference class ‘neutral’ also increase. Therefore, we can intuitively prove the ability of CycleGAN to generate reliable images, which is helpful in enlarging minorities. Furthermore, this data augmentation of one class also improves accuracy of other classes, since by generating new samples, the data manifold is further supplemented and becomes more completed, thus make more clearly the margins between classes.

In order to provide more powerful verification that this data augmentation indeed contributes to the shape of data manifold, we apply a t-distributed stochastic neighbor embedding (t-SNE) algorithm [20] to visualize the distribution of training samples by reducing high-dimensional data (48*48) to 2D plane. (Fig.4) Compared to the baseline, where sample size of ‘disgust’ and ‘sad’ is too small to form a clear margin with other classes, 2 and 3 in Fig.4 shows great improvement in enlarging the sample size, supplementing the data manifold and completing data distribution. Picture 4 is a much stronger validation where both two classes stand out to improve data manifold.

After generating specific classes to validate GAN’s positive role in data augmentation, we make further experiments on our pipeline based on all three datasets mentioned in 4.2. During this process, a baseline model and a model using our data augmentation pipeline (pre-train+fine-tune) is trained respectively. In our pipeline, all classes except ‘neutral’ are generated from Cycle-GAN and then added as supplementary training corpus for training classifica-
Fig. 4: Data manifold of four types of training samples using t-SNE algorithm: baseline(1), adding generated ‘disgust’ samples(2) or ‘sad’ samples(3), and samples of both two classes(4) to original datasets.

After applying our pipeline of data augmentation using GAN, accuracy of all the three datasets has visibly improved. As for FER2013 database which has obvious unbalanced distribution among classes, our data augmentation technique is able to complete data manifold, especially for those which have much smaller samples. And for small datasets like SFEW and JAFFE, our technique can generate feature-level synthetic images from existing samples to enlarge the original datasets and form clear margins or hyper-planes between neighboring classes.
Datasets | accuracy | baseline | DAG:pre-train+fine-tune
--- | --- | --- | ---
FER2013 | 91.04(7%) 90.77(14%) | 94.71(7%) 94.35(14%) | 94.71(7%) 94.35(14%) |
FER+SFEW | 31.92 | 39.07 | |
FER+JAFFE | 93.87 | 95.80 | |

Table 2: Testing accuracy of baseline and our pipeline (DAG). In the column ‘DAG:Pre-train + Fine-tune’, ‘Pre-train’ represents the first 10k steps training on generated images from all six classes and ‘Fine-tune’ represents another 10k fine-tuning steps training on original datasets. SFEW and JAFFE datasets are trained based on the FER2013 model.

5 Conclusions and Discussions

In this paper, we explored GAN’s possible role and advantage in data augmentation of emotion classification task and the results are positive.

We propose a pipeline for data augmentation by using GANs to generate auxiliary data in emotion classification task. It is worth mentioning that no extra data is utilized during the process, so that this data augmentation is free for external data as traditional ones. During the process of training CycleGAN model, a least-squared loss is combined with original adversarial loss from CycleGAN to avoid possible gradient vanishing. Besides, we show the GAN’s ability of supplementing data manifold from the true sense, which is considered better than traditional data augmentation methods. Because of possessing a more complete data manifold, the classifier can better learn to find margins or hyper-planes of neighboring classes. Experiments on three benchmark datasets indicate that both qualitative evaluation on improvement in distribution integrity and margin clarity between classes and quantitative comparisons with the baseline show exciting results.

Still, the work has some limitations. For instance, the datasets we select is limited to emotions and only CycleGAN is used in our model to prove our point of view. Therefore, the future work contains applying our model to general datasets for image classification, and use as many as possible GANs model to implement data augmentation to provide more stronger validations.

References

1. Arjovsky, M., Chintala, S., Bottou, L.: Wasserstein gan. arXiv preprint arXiv:1701.07875 (2017)
2. Chawla, N.V., Bowyer, K.W., Hall, L.O., Kegelmeyer, W.P.: Smote: synthetic minority over-sampling technique. Journal of artificial intelligence research 16, 321–357 (2002)
3. Cireşan, D.C., Meier, U., Masci, J., Gambardella, L.M., Schmidhuber, J.: High-performance neural networks for visual object classification. arXiv preprint arXiv:1102.0183 (2011)
4. Denton, E.L., Chintala, S., Fergus, R., et al.: Deep generative image models using a laplacian pyramid of adversarial networks. In: Advances in neural information processing systems. pp. 1486–1494 (2015)
5. Dhall, A., Goecke, R., Lucey, S., Gedeon, T.: Static facial expression analysis in tough conditions: Data, evaluation protocol and benchmark. In: Computer Vision Workshops (ICCV Workshops), 2011 IEEE International Conference on. pp. 2106–2112. IEEE (2011)
6. Dixit, M., Kwitt, R., Niethammer, M., Vasconcelos, N.: Aga: Attribute guided augmentation. arXiv preprint arXiv:1612.02559 (2016)
7. Dumoulin, V., Belghazi, I., Poole, B., Lamb, A., Arjovsky, M., Mastropietro, O., Courville, A.: Adversarially learned inference. arXiv preprint arXiv:1606.00704 (2016)
8. Goh, S.T., Rudin, C.: Box drawings for learning with imbalanced data. In: Proceedings of the 20th ACM SIGKDD international conference on Knowledge discovery and data mining. pp. 333–342. ACM (2014)
9. Goodfellow, I., Pouget-Abadie, J., Mirza, M., Xu, B., Warde-Farley, D., Ozair, S., Courville, A., Bengio, Y.: Generative adversarial nets. In: Advances in neural information processing systems. pp. 2672–2680 (2014)
10. Goodfellow, I.J., Erhan, D., Carrier, P.L., Courville, A., Mirza, M., Hamner, B., Cukierski, W., Tang, Y., Thaler, D., Lee, D.H., et al.: Challenges in representation learning: A report on three machine learning contests. In: International Conference on Neural Information Processing. pp. 117–124. Springer (2013)
11. Gulrajani, I., Ahmed, F., Arjovsky, M., Dumoulin, V., Courville, A.: Improved training of wasserstein gans. arXiv preprint arXiv:1704.0028 (2017)
12. Hauberg, S., Freifeld, O., Larsen, A.B.L., Fisher, J., Hansen, L.: Dreaming more data: Class-dependent distributions over diffeomorphisms for learned data augmentation. In: Artificial Intelligence and Statistics. pp. 342–350 (2016)
13. Isola, P., Zhu, J.Y., Zhou, T., Efros, A.A.: Image-to-image translation with conditional adversarial networks. arXiv preprint arXiv:1611.07004 (2016)
14. Keerthi, S.S., Lin, C.J.: Asymptotic behaviors of support vector machines with gaussian kernel. Neural computation 15(7), 1667–1689 (2003)
15. Krizhevsky, A., Sutskever, I., Hinton, G.E.: Imagenet classification with deep convolutional neural networks. In: Advances in neural information processing systems. pp. 1097–1105 (2012)
16. Ledig, C., Theis, L., Huszár, F., Caballero, J., Cunningham, A., Acosta, A., Aitken, A., Tejani, A., Totz, J., Wang, Z., et al.: Photo-realistic single image super-resolution using a generative adversarial network. arXiv preprint arXiv:1609.04802 (2016)
17. Li, C., Xu, K., Zhu, J., Zhang, B.: Triple generative adversarial nets. arXiv preprint arXiv:1703.02291 (2017)
18. Liu, X.Y., Wu, J., Zhou, Z.H.: Exploratory undersampling for class-imbalance learning. IEEE Transactions on Systems, Man, and Cybernetics, Part B (Cybernetics) 39(2), 539–550 (2009)
19. Lyons, M.J., Akamatsu, S., Kamachi, M., Gyoba, J., Budynek, J.: The japanese female facial expression (jaffe) database. In: Proceedings of third international conference on automatic face and gesture recognition. pp. 14–16 (1998)
20. Maaten, L.v.d., Hinton, G.: Visualizing data using t-sne. Journal of Machine Learning Research 9(Nov), 2579–2605 (2008)
21. Mao, X., Li, Q., Xie, H., Lau, R.Y., Wang, Z., Smolley, S.P.: Least squares generative adversarial networks. arXiv preprint ArXiv:1611.04076 (2016)
22. Mirza, M., Osindero, S.: Conditional generative adversarial nets. arXiv preprint arXiv:1411.1784 (2014)
23. Ng, H.W., Nguyen, V.D., Vonikakis, V., Winkler, S.: Deep learning for emotion recognition on small datasets using transfer learning. In: Proceedings of the 2015 ACM on international conference on multimodal interaction. pp. 443–449. ACM (2015)
24. Radford, A., Metz, L., Chintala, S.: Unsupervised representation learning with deep convolutional generative adversarial networks. arXiv preprint arXiv:1511.06434 (2015)
25. Rogez, G., Schmid, C.: Mocap-guided data augmentation for 3d pose estimation in the wild. In: Advances in Neural Information Processing Systems. pp. 3108–3116 (2016)
26. Simard, P.Y., Steinkraus, D., Platt, J.C., et al.: Best practices for convolutional neural networks applied to visual document analysis. In: ICDAR. vol. 3, pp. 958–962 (2003)
27. Wallace, B.C., Small, K., Brodley, C.E., Trikalinos, T.A.: Class imbalance, redux. In: Data Mining (ICDM), 2011 IEEE 11th International Conference on. pp. 754–763. IEEE (2011)
28. Yu, Z., Zhang, C.: Image based static facial expression recognition with multiple deep network learning. In: Proceedings of the 2015 ACM on International Conference on Multimodal Interaction. pp. 435–442. ACM (2015)
29. Zhu, J.Y., Krähenbühl, P., Shechtman, E., Efros, A.A.: Generative visual manipulation on the natural image manifold. In: European Conference on Computer Vision. pp. 597–613. Springer (2016)
30. Zhu, J.Y., Park, T., Isola, P., Efros, A.A.: Unpaired image-to-image translation using cycle-consistent adversarial networks. arXiv preprint arXiv:1703.10593 (2017)

A Training Details

CNN Model

In any stage of classification task, we all apply batch-size = 32, stable learning rate=1e-3 and training step = 20000. Adam optimizer is used whose parameter $\beta_1 = 0.5$. More detailed configurations are listed in Table. 3

CycleGAN

During training the CycleGAN model, we use batch-size=1, learning rate=2e-4 and 1e-4. Adam optimizer is used and $\beta_1$ is set to 0.5. Besides, the hyper-parameters of CycleGAN are 10 for both $\lambda_1$ and $\lambda_2$. More detailed configurations are listed in Table.4 and 5
Table 3: Configuration of the convolutional neural network, "s" represents stride. FC means fully connected operation and there are two FC layers in this network.

| Layer Type              | Configuration |
|-------------------------|---------------|
| Input image             | 48*48*1       |
| Convolution&ReLU 3x3, 1, 64 s=1 |               |
| Max-Pooling&Norm 1x3, 3, 1 s=2 |               |
| Convolution&ReLU 3x3, 64, 128 s=1 |             |
| Max-Pooling&Norm 1x3, 3, 1 s=2 |               |
| FC*2                    | 256           |
| Softmax                 | [256, 7]      |
| Output logits           | [7]           |

Table 4: Configuration of the generator in CycleGAN, "s" represents stride. Conv, BN, Deconv represent convolution, batch-normalization and deconvolution(matrix transpose) respectively. We apply 6 Resnet block in our network and each block has 2 convolution layers.

| Layer Type              | Configuration |
|-------------------------|---------------|
| Input 48*48*1           |               |
| Conv-BN-ReLU 4x4, 64, s=2 |              |
| Conv-BN-ReLU 4x4, 128, s=2 |             |
| Conv-BN-ReLU 4x4, 256, s=2 |            |
| Conv-BN-ReLU 4x4, 512, s=2 |            |
| Conv-BN-ReLU 4x4, 1, s=1 |               |
| Output 48*48*1          |               |

Table 5: Configuration of the discriminator in CycleGAN, "s" represents stride. Settings and representations are same as generator.

| Layer Type              | Configuration |
|-------------------------|---------------|
| Input 48*48*1           |               |
| Conv-BN-ReLU 4x4, 64, s=2 |              |
| Conv-BN-ReLU 4x4, 128, s=2 |             |
| Conv-BN-ReLU 4x4, 256, s=2 |            |
| Conv-BN-ReLU 4x4, 512, s=2 |            |
| Conv-BN-ReLU 4x4, 1, s=1 |               |
| Output 1                |               |