Hindawi
Computational Intelligence and Neuroscience
Volume 2018, Article ID 7208794, 10 pages
https://doi.org/10.1155/2018/7208794

Research Article

Unsupervised Domain Adaptation for Facial Expression Recognition Using Generative Adversarial Networks

Xiaoqing Wang[1,2] **Xiangjun Wang**[1,2] **and Yubo Ni**[1,2]

[1]*State Key Laboratory of Precision Measuring Technology and Instruments, Tianjin University, 300072, China*
[2]*Key Laboratory of MOEMS of the Ministry of Education, Tianjin University, 300072, China*

Correspondence should be addressed to Xiangjun Wang; tjuxjw@126.com

Received 14 April 2018; Accepted 19 June 2018; Published 9 July 2018

Academic Editor: António D. P. Correia

In the facial expression recognition task, a good-performing convolutional neural network (CNN) model trained on one dataset (source dataset) usually performs poorly on another dataset (target dataset). This is because the feature distribution of the same emotion varies in different datasets. To improve the cross-dataset accuracy of the CNN model, we introduce an unsupervised domain adaptation method, which is especially suitable for unlabelled small target dataset. In order to solve the problem of lack of samples from the target dataset, we train a generative adversarial network (GAN) on the target dataset and use the GAN generated samples to fine-tune the model pretrained on the source dataset. In the process of fine-tuning, we give the unlabelled GAN generated samples distributed pseudolabels dynamically according to the current prediction probabilities. Our method can be easily applied to any existing convolutional neural networks (CNN). We demonstrate the effectiveness of our method on four facial expression recognition datasets with two CNN structures and obtain inspiring results.

1. Introduction

Facial expressions recognition (FER) has a wide spectrum of application potentials in human-computer interaction, cognitive psychology, computational neuroscience, and medical healthcare. In recent years, convolutional neural networks (CNN) have achieved many exciting results in artificial intelligent and pattern recognition and have been successfully used in facial expression recognition [1]. Jaiswal et al. [2] present a novel approach to facial action unit detection using a combination of Convolutional and Bidirectional Long Short-Term Memory Neural Networks (CNN-BLSTM), which jointly learns shape, appearance, and dynamics in a deep learning manner. You et al. [3] introduce a new data set, which contains more than 3 million weakly labelled images of different emotions. Esser et al. [4] develop a model for efficient neuromorphic computing using the Deep CNN technique. H-W.Ng et al. [5] develop a cascading fine-tuning approach for emotion recognition. Neagoe et al. [6] propose a model for subject independent emotion recognition from facial expressions using combined CNN and DBN. However, these CNN models are often trained and tested on the same dataset, whereas the cross-dataset performance is less concerned. Although the basic emotions defined by Ekman and Friesen [7], anger, disgust, fear, happy, sadness, and surprise, are believed to be universal, the way of expressing these emotions can be quite diverse across different cultures, ages, and genders [8]. As a result, a well-trained CNN model, having high recognition accuracy on the training dataset, usually performs poorly on other datasets. In order to make the facial expression recognition system more practical, it is necessary to improve the generalization ability of the recognition model.

In this paper, we aim at improving the cross-dataset accuracy of a CNN model on facial expression recognition. One way to solve this problem is to rebuild models from scratch using large-scale newly collected samples. Large amounts of training samples, such as the dataset ImageNet [9] containing over 15 million images, can reduce the overfitting problem and help to train a reliable model. However, for facial expression recognition, it is expensive and sometimes even impossible to get enough labelled training data. Therefore, we proposed an unsupervised domain adaptation method, which is especially suitable for unlabelled small

target datasets. Domain adaptation aims at learning knowledge from one dataset (source dataset) and transferring the knowledge to a related but not identical dataset (target dataset). Recent progress involves deep neural networks into the domain adaptation. Long et al. [10] propose a Deep Adaptation Network (DAN) architecture, which generalizes deep convolutional neural network to the domain adaptation scenario. Ganin et al. [11] introduce an unsupervised domain adaptation in deep architectures that can be trained on large amount of labelled data from the source domain and large amount of unlabelled data from the target domain. Tzeng et al. [12] propose a new CNN architecture to exploit unlabelled and sparsely labelled target domain data. Our method also uses CNN as the basic structure. But unlike [11], which needs large-scale unlabelled data from the target domain, our method works well with small-size unlabelled target dataset by using GAN generated samples.

The generative adversarial networks (GAN) have two subnetworks: a generator and a discriminator. The discriminator decides whether a sample is generated or real, while the generator produces samples to cheat the discriminator. The GAN is first proposed by Goodfellow et al. [18]. DCGAN [19] combines GAN with CNN and provides techniques to improve the training stability. InfoGAN [20] learns interpretable representations by introducing latent codes. WGAN [21] introduces Wasserstein distant to replace the KL divergence, which solves the model collapse problem in GAN and produces GAN samples with higher diversity. In this work, we use a GAN model similar to the one used in the WGAN to generate unlabelled samples from the target data and these samples will help the baseline CNN to gain a better knowledge of the target distribution. Actually, we are not the first to introduce GAN generated samples into CNN training. Odena [22] treats GAN samples as a new class during a semisupervised training. Zheng et al. [23] also focus on semisupervised training and assign a uniform label distribution over all the existing classes to GAN samples. Unlike these methods, we give GAN samples distributed pseudolabels, which have different weights with different classes and we change the weights dynamically according to the CNN prediction probability vector during training.

In this paper, the dataset that is used to train the baseline CNN is referred to as the source dataset, and the dataset being tested on for cross-dataset performance is referred to as target dataset. Our method uses samples generated by a generative adversarial network (GAN) to make up for the shortage of samples in the target dataset. More specifically, we apply our method on two widely used CNN structures, Alexnet [24] and VGG11 [25], and we train a CNN model as baseline using the source dataset, but our method is not limited with these two models and can be easily applied to other CNN models as well. Then we train a GAN using the target dataset to generate GAN samples. These unlabelled newly generated samples are combined with the source dataset to fine-tune the CNN baseline model to help the model to get a better recognition accuracy on the target domain. During fine-tuning, a distributed pseudolabel (DPL) is given to the newly generated sample according to its current prediction probabilities. We evaluate our method on the cross-dataset facial expression recognition task with four datasets. Experiments have shown that our method obtains state-of-the-art results on cross-dataset FER. The main contributions of this paper are as follows:

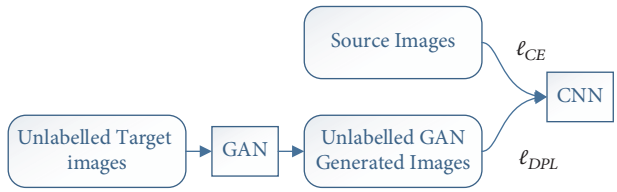

Figure 1: Overall training structure of the domain adaptation, ℓ_{CE} is the cross-entropy loss used for the images in the source dataset and ℓ_{DPL} is the distributed pseudolabel loss used for the unlabelled GAN images.

(i) Introducing an unsupervised domain adaptation method using GAN generated samples.

(ii) Proposing a distributed pseudolabel method for samples generated by GAN.

(iii) Improving the cross-dataset accuracy of baseline CNN in facial expression recognition using the proposed method.

2. Proposed Method

2.1. Overall Architecture. The overall architecture of our unsupervised domain adaptation is shown in Figure 1. We first train a facial expression recognition CNN with the source images. After training a CNN with the source dataset, we want to improve its cross-dataset performance without the ground truth label information of the target dataset. To this end, we must deal with the limited number of samples in the target dataset. GAN provides a solution for us. By training a GAN with the target dataset, we can generate more images that follow the same distribution as the target dataset. Then the CNN model can be fine-tuned with the combination of these generated images and the source dataset. Here we include the source dataset during fine-tuning because we find that the experiment results are better compared with those of fine-tuning with only the generated samples. The generated images, however, cannot be directly used to train a CNN because they are unlabelled. Inspired by Szegedy et al.'s [26] label smoothing regularization (LSR) used for supervised learning, we propose a distributed pseudolabel method (DPL) for our unlabelled GAN generated samples.

2.2. Distributed Pseudolabel. In a supervised training task, it is classic to use cross-entropy loss during training. Let $k = \{1, 2, \ldots, N\}$ be the predefined classes, where N is the number of classes. The cross-entropy loss function is as follows:

$$\ell = -\sum_{k=1}^{N} q(k) \log \frac{\exp(x_k)}{\sum_{i=1}^{N} \exp(x_i)} = -\sum_{k=1}^{N} q(k) \log(p(k)) \quad (1)$$

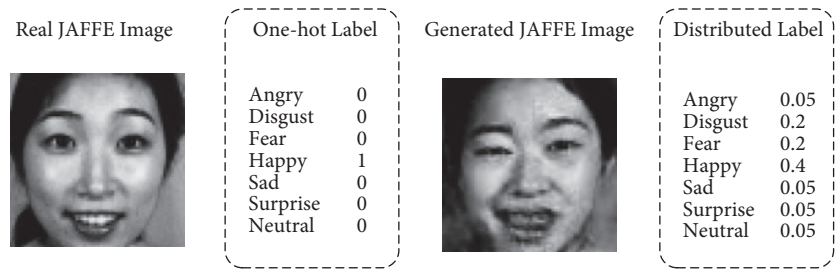

Figure 2: The one-hot label of real JAFFE Image and the distributed label of a generated JAFFE image used in our DPL method.

where $x_i, i \in \{1, 2, \ldots, N\}$ is the output of the CNN's last fully connected layer. $p(k)$ is the prediction probability of the input image belonging to class k, which is the normalized value of x_k. $q(k)$ is the label distribution of the input image.

Let k_{label} be the ground truth class; the one-hot label style $q(k)$ is defined as follows:

$$q(k) = \begin{cases} 1 & k = k_{label} \\ 0 & k \neq k_{label} \end{cases} \tag{2}$$

With (2), the loss function in (1) becomes

$$\ell_{CE} = -\log\left(p\left(k_{label}\right)\right) \tag{3}$$

Unlike supervised training, our GAN generated images has no ground truth class. Therefore, before using them to fine-tune the CNN model, we need to give them proper labels. We proposed our method based on an observation that each of the generated samples is a mixture of different people and different emotions from the target dataset, for example, a mouth from a happy person A, an eye from a sad person B, and another eye from a fear person C. This is because these GAN generated images randomly contain features learned from the target dataset, on which the GAN was trained. In the meantime, a generated image will be more similar to a certain emotion than the others. The generated image in Figure 2, for example, is more similar to an original happy face than other emotions. As a result, instead of giving them one-hot labels, we use distributed pseudolabels with different weights to different classes. This idea is inspired by the label smoothing method [26], in which $q(k)$ is defined as follows:

$$q(k) = \begin{cases} 1 - \varepsilon + \dfrac{\varepsilon}{N} & k = k_{label} \\ \dfrac{\varepsilon}{N} & k \neq k_{label} \end{cases} \tag{4}$$

where ε is a hyperparameter set to 0.1.

By applying (4) to (1), the cross-entropy loss function evolves to

$$\ell_{LSR} = -(1 - \varepsilon)\log\left(p\left(k_{label}\right)\right) - \frac{\varepsilon}{N}\sum_{k=1}^{N}\log\left(p\left(k\right)\right) \tag{5}$$

Label smoothing assigns small values to the non-ground truth classes instead of 0. This strategy discourages the network to be tuned toward the ground truth class and thus reduces the chances of overfitting. As for the distributed label for our unlabelled GAN samples, we feed a generated sample into the CNN model and use the prediction probability vector to help us decide which class gets a higher weight. Since the baseline CNN model is trained on the source dataset, its prediction on our generated samples, which are generated from the target dataset, cannot be highly reliable. Therefore, instead of giving higher weight to only one class, we give higher weights to three classes which have the top 3 maximum prediction probabilities. We refer to this method as distributed pseudolabel method (DPL).

Let k_{top1}, k_{top2}, and k_{top3} be the classes that have the top 3 maximum prediction probabilities after a generated image passes through the CNN. Our distributed pseudolabel for this generated image is

$$q(k) = \begin{cases} \lambda_1 & k = k_{top1} \\ \lambda_2 & k = k_{top2} \\ \lambda_3 & k = k_{top3} \\ \dfrac{1 - \lambda_1 - \lambda_2 - \lambda_3}{N - 3} & otherwise \end{cases} \tag{6}$$

where λ_1, λ_2, and λ_3 are hyperparameters set to 0.4, 0.2, and 0.2 in our experiments.

During training, each time when a GAN generated image passes through the CNN, we assign a new distributed pseudolabel to it according to the current prediction, so the label of the generated image can change dynamically. With DPL, the entropy loss function for GAN generated images changes to (7), whereas the images from the source dataset still use the one-hot labels during training and their loss function is (3). Since the GAN images have no ground truth labels and their prediction results from CNN based on the source dataset cannot be highly accurate, DPL gives them the top 3 most likely classes they belong to and encourages the network to be tuned toward these classes with major consideration and thus help the network to gain a better recognition accuracy on the target dataset. The fine-tuning process using DPL is illustrated in Algorithm 1.

$$\ell_{DPL} = -\lambda_1 \log\left(p\left(k_{top1}\right)\right) - \lambda_2 \log\left(p\left(k_{top2}\right)\right)$$

Input:
M: pretrained CNN model
X_s: source images
X_g: GAN generated images
$X_{tr} = \{X_s, X_g\}$: training images
Functions:
$p \longleftarrow M(x)$: probability output of M given $x \in X_{tr}$
$loss \longleftarrow \ell_{DPL}(p)$: calculate the loss using DPL loss function
$loss \longleftarrow \ell_{CE}(p)$: calculate the loss using cross-entropy loss function
Training:
(1) **for** each epoch:
(2) **for** each $x \in X_{tr}$:
(3) $p \longleftarrow M(x)$
(4) **if** $x \in X_s$:
(5) $loss \longleftarrow \ell_{CE}(p)$
(6) **else if** $x \in X_g$:
(7) $loss \longleftarrow \ell_{DPL}(p)$
(8) Update M with $loss$
(9) **end**
(10) **end**

Algorithm 1: Fine-tuning process using DPL.

Figure 3: Sample images from four datasets.

$$- \lambda_3 \log\left(p\left(k_{top3}\right)\right) - \frac{1 - \lambda_1 - \lambda_2 - \lambda_3}{N - 3} \sum_{k=1(k \neq k_{top1}, k_{top2}, k_{top3})}^{N} \log\left(p\left(k\right)\right) \tag{7}$$

3. Implementation Details

3.1. Datasets. We use four FER datasets and seven emotions in our experiments. Figure 3 shows sample images from these datasets.

FER-2013 is a large-scale FER dataset used in the ICML 2013 workshop's facial expression recognition challenge [27]. The dataset has seven expressions including anger, disgust, fear, happy, sad, surprise, and neutral, and it comprises three parts: the training data (FER-TRA), which consists of 28709 images, the Public test data (FER-PUB), which consists of 3859 images, and the private test data (FER-PRI), which also consists of 3859 images. The images of FER-2013 were collected from the Internet and the faces greatly vary in age, pose and occlusion, thus resulting the accuracy of human recognition is only approximately $65 \pm 5\%$ [28]. As a powerful machine learning tools, the CNN can now surpass human beings on the FER-2013 task, and the state-of-the-art accuracy on FER-2013 is 75.42% by combining CNN extracted features and handcrafted features for training [29]. In this paper, we apply our method to two less fancy yet more commonly used CNN architectures, namely, Alexnet [24] and VGG11 [25]. When we use the FER-TRA for training and use the FER-PRI

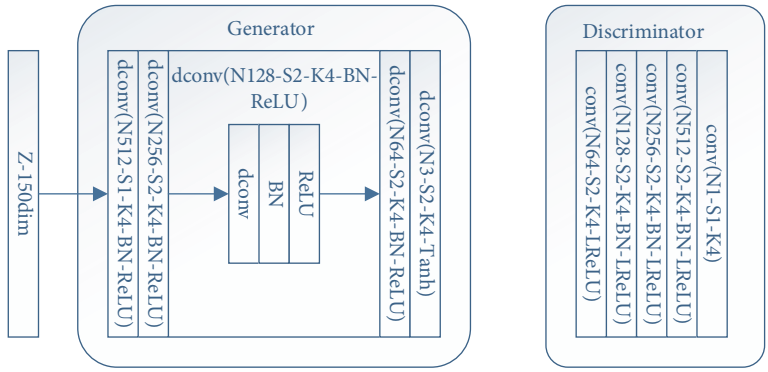

Figure 4: Network structure of GAN. The convolutional layer is denoted as conv and the transposed convolutional layer is denoted as dconv. N stands for neurons (channels), S stands for stride, and K stands for kernel size. LReLU means leaky ReLU nonlinearity and BN means batch normalization.

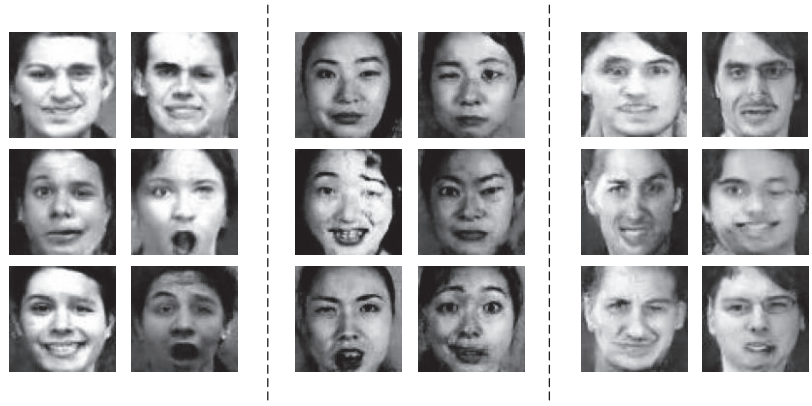

Figure 5: From left to right, GAN generated images from CK+, JAFFE, and MMI.

for testing, the recognition accuracy we get is 66.37% and 65.03% respectively.

JAFFE dataset consists of 213 facial expression images from 10 Japanese females [30, 31]. They posed seven basic expressions (anger, disgust, fear, happy, sad, surprise, plus neutral expression). We use all of the images in JAFFE, either as source dataset or as target dataset in different experiments.

CK+ dataset consists of 593 sequences from 123 subjects, among which 327 sequences have emotion labels [32]. The dataset contains seven expressions including anger, disgust, fear, happy, sad, surprise, and contempt. We only chose the peak frame from the sequences labelled with the first six expressions. In addition, we chose the first frame from some of the sequences as neutral samples. In total we use 363 images from CK+. The CK+ dataset is used as source dataset as well as target dataset.

MMI dataset consists of over 2900 videos and high resolution still images of 75 subjects, in which 235 videos have emotional labels [33, 34]. We chose the peak frame of each video with the six basic emotions and the first frame as neutral emotion. In total we use 242 images from MMI. The MMI dataset was only used as target dataset in our experiments.

3.2. Network Structure. The Alexnet [24] and VGG11 [25] architectures are used as the CNN for expression recognition in our experiment, and, with each model, we modify the last fully connected layer to have 7 neurons to predict the seven emotion classes. We detect and crop the faces out of JAFFE, CK+, and MMI but we do not crop the FER2013 because the original images are too small (48 × 48). All the images are resized into 224 × 224 before training. We train the CNNs on the source dataset and test them on the target dataset. The cross-dataset accuracy of these models is used as our experiment baseline, which are shown in Table 1. And these CNN models are used as the pretrained models for fine-tuning in the domain adaptation process. During training, we use stochastic gradient descent with the learning rate set to 0.00001 and the momentum set to 0.9. We train the Alexnet for 50 epochs and the VGG11 for 100 epochs. The CNN models are all trained on Pytorch.

We resize the face-cropped images of CK+, JAFFE, and MMI to 64 × 64 for GAN training. The GAN structure is showed in Figure 4. Following [21], we use Wasserstein distance to calculate the loss during training. The input vector z is set to 150-dim, and for each GAN model we train 5000 epochs. The GAN models are also trained on Pytorch. Figure 4 shows the GAN structure used in our experiment. Figure 5 shows some of the samples of our GAN generated images.

In the domain adaptation training process, we use 2k GAN images in each experiment. The weights of the top 3

Table 1: Experiment results of the recognition accuracy on the target dataset.

Model	Source Dataset	Target Dataset	Source Only	Our Result
VGG11	FER-2013	JAFFE	44.60%	59.62%
Alexnet	FER-2013	JAFFE	50.70%	54.46%
Alexnet	FER-2013	MMI	58.14%	61.86%
Alexnet	FER-2013	CK+	71.90%	76.58%
Alexnet	CK+	JAFFE	46.94%	51.64%
Alexnet	JAFFE	CK+	60.33%	65.01%

Table 2: Comparison with other published methods.

Method	Source Dataset	Target Dataset	Recognition Accuracy on The Target Dataset
Meguid et al. [13]	Bu-3DFE	JAFFE	41.96%
Wen et al. [14]	FER2013	JAFFE	50.70%
Gu et al. [15]	CK	JAFFE	55.87%
Zhu et al. [16]	FEED	JAFFE	**61.97**%
Our Method	CK+	JAFFE	51.64%
Our Method	FER2013	JAFFE	59.62%
Mayer et al. [17]	CK	MMI	60.30%
Mayer et al. [17]	FEED	MMI	58.90%
Our Method	FER2013	MMI	**61.86**%
Gu et al. [15]	JAFFE	CK+	54.05%
Mayer et al. [17]	FEED	CK+	56.60%
Wen et al. [14]	FER2013	CK+	76.05%
Our Method	JAFFE	CK+	65.01%
Our Method	FER2013	CK+	**76.58**%

classes (λ_1, λ_2, and λ_3) in DPL are set as 0.4, 0.2, and 0.2, respectively.

4. Experiment Results and Discussion

4.1. Experiment Results. We conduct a series of experiments over different datasets. During training, the source dataset and its label information is used to train the CNN, whereas the target dataset is only used to generated GAN samples without the label information. The label information of the target dataset is only used for testing. First, the relatively large dataset, FER-2013, is used as source dataset. When using Alexnet as the CNN structure, we take JAFFE, MMI, and CK+ as target dataset and obtain 3.76%, 3.72%, and 4.41% recognition accuracy improvement on the target dataset, respectively. We also train on VGG11 with FER-2013 as the source dataset and JAFFE as the target dataset to examine our method on a different CNN structure, and the recognition accuracy increases by 15.02%. Then we use smaller dataset as source dataset to further test our method. When we use CK+ as the source dataset, we get 4.70% improvement of recognition accuracy on JAFFE and 4.68% improvement when using JAFFE as source dataset and CK+ as target dataset. The experiment results have shown that our method can improve the CNN model's recognition accuracy on the target dataset with different datasets as well as different CNN structures.

4.2. Comparison with Other Published Method. We compare our experiment results with other published cross-dataset recognition accuracy results in Table 2, and Table 2 shows that our method outperforms most of the published results. When JAFFE is the target dataset, Zhu et al. achieve higher accuracy (61.97%) than our method (59.62%), but they use part of the JAFFE datasets with ground truth labels for transfer learning, whereas our method requires no ground truth labels from the target data at all.

4.3. Comparison of Confusion Matrix. We compare the confusion matrix of our result with the baseline CNN trained only with the source dataset to see the recognition accuracy changes of each class of the emotions. In Tables 3 and 4, the source dataset is FER-2013, the target dataset is JAFFE, and the CNN structure is VGG11. Table 3 shows that the baseline CNN trained on FER2013 performs poorly on JAFFE and has a tendency of classifying the JAFFE images as Neutral. More specifically, it misclassifies all the Anger images as Neutral and has low classification accuracy on Disgust and Sad, only 13.79% and 16.13% respectively, whereas 55.5% and 80.65% of these two emotions are misclassified as Neutral. Table 4 shows that, after applying our method, the Angry, Disgust, and Sad accuracy improves to 23.33%, 58.62%, and 32.26%, respectively, which certifies that our method can effectively improve the baseline CNN's understanding of the target domain.

Table 3: The target dataset recognition accuracy (%) confusion matrix of baseline CNN, FER-2013⟶JAFFE.

	Angry	Disgust	Fear	Happy	Sad	Surprise	Neutral
Angry	**0.00**	0.00	0.00	0.00	0.00	0.00	100.00
Disgust	0.00	**13.79**	3.45	0.00	27.59	0.00	55.17
Fear	0.00	0.00	**34.38**	3.13	0.00	9.38	53.13
Happy	0.00	0.00	0.00	**64.52**	0.00	3.23	32.26
Sad	3.23	0.00	0.00	0.00	**16.13**	0.00	80.65
Surprise	0.00	0.00	0.00	6.67	0.00	**83.33**	10.00
Neutral	0.00	0.00	0.00	0.00	0.00	0.00	**100.00**

Table 4: The target dataset recognition accuracy (%) confusion matrix of our method, FER-2013⟶JAFFE.

	Angry	Disgust	Fear	Happy	Sad	Surprise	Neutral
Angry	**23.33**	13.33	0.00	0.00	10.00	0.00	53.33
Disgust	6.90	**58.62**	3.45	0.00	17.24	0.00	13.79
Fear	9.38	31.25	**50.00**	0.00	0.00	6.25	3.13
Happy	0.00	0.00	0.00	**96.77**	0.00	0.00	3.23
Sad	25.81	9.68	16.13	3.23	**32.26**	3.23	9.68
Surprise	0.00	0.00	6.67	13.33	0.00	**80.00**	0.00
Neutral	6.67	0.00	0.00	6.67	6.67	3.33	**76.67**

Table 5: Comparison with two other methods using GAN generated images, FER-2013⟶JAFFE.

Method	Recognition Accuracy on Target Dataset	
	Alexnet	VGG11
Baseline	50.70%	44.60%
Pseudo-label	51.17%	42.72%
LSRO	53.99%	57.75%
DPL	**54.46**%	**59.62**%

4.4. Comparison with Two Other GAN-Based Domain Adaptation Methods. We compare DPL with two alternative methods using GAN generated images, the pseudolabel [35], and the LSRO [23].

(i) Pseudolabel takes the class which has the highest predicted probability as the unlabelled image's one-hot pseudolabel and updates the pseudolabel each time when the unlabelled image is fed into the network.

(ii) LSRO is a regularization method used for GAN samples generated from a person re-ID dataset; they presume that the generated samples do not belong to any of the person predefined and should be labelled with a uniform distribution $q(k) = 1/k$ over all the classes.

This experiment is conducted using both two CNN architectures, the Alexnet and the VGG11, with 2k GAN images. The source dataset is FER-2013 and the target dataset is JAFFE. Table 5 shows that pseudolabel does not work well on the FER task and even decreases the cross-dataset accuracy on VGG11. LSRO does improve the model's accuracy on the target dataset, but our method has the best results on both networks.

Table 6: Comparison with real images, FER-2013⟶JAFFE.

Method	Recognition Accuracy on Target Dataset	
	Alexnet	VGG11
Sour-only	50.70%	44.60%
Real-213	51.64%	56.34%
GAN-213	52.11%	55.87%
GAN-2k	54.46%	59.62%
GAN-2k+Real-213	55.40%	60.09%

4.5. Comparison with Real Images. In previous experiments, we only fine-tune the CNN with GAN generated samples; now we want to investigate how our method performs with real images from the target dataset. We use FER-2013 as the source dataset and train a CNN with it. And we treat the JAFFE as unlabelled target images to fine-tune the CNN with DPL. Since the JAFFE dataset has only 213 images, we fine-tune a CNN with 213 generated images for comparison. Table 6 shows that our method works on unlabelled real images as well, and the real-world images actually achieve better cross-dataset accuracy on VGG11 (56.34%) compared with the result trained with the same amount of GAN generated samples (55.87%). But limited by the total number of images, the results with real 213 images are far below our results with 2k generated samples. Then we combine 213 real images and 2k GAN generated images to fine-tune the CNN. This strategy slightly outperforms our best results with 2k generated images by a margin of 0.94% and 0.47% on Alexnet and VGG11, respectively. These experiment results indicate that our DPL method not only can be used on domain adaptations with GAN generated samples but also can be used on unsupervised learning tasks with real-world images.

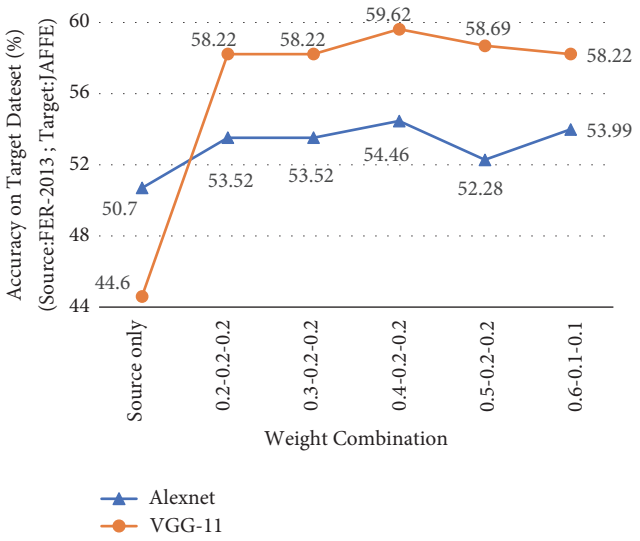

Figure 6: The experiment result with different weight combinations (λ_1-λ_2-λ_3).

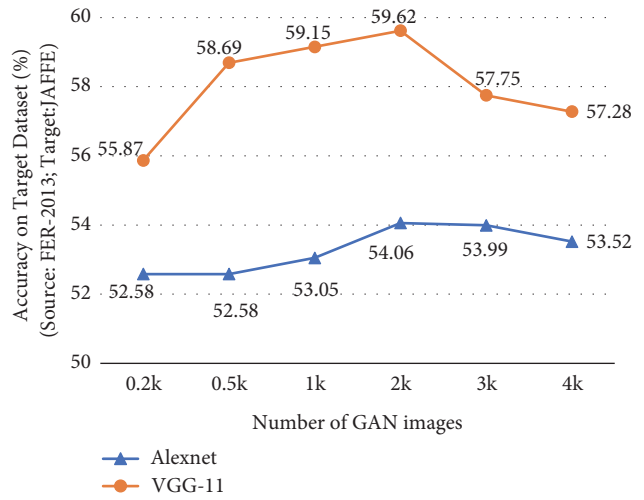

Figure 7: The experiment result using different number of GAN images.

4.6. Weight Parameters for DPL. The weights of the top 3 classes (λ_1, λ_2, and λ_3) are hyperparameters in DPL. Figure 6 shows the experiment result with different weight combinations. In this experiment, the source dataset is FER-2013, and the target dataset is JAFFE. During training, we combine 2k GAN images with the source dataset to fine-tune the CNN baseline model. We set the learning rate to 0.000001 and stop fine-tuning at 10 epochs. Figure 6 shows that the 0.4-0.2-0.2 combination achieves the best result with the target dataset on both models.

4.7. The Number of GAN Samples. Here we look into the impact of the number of GAN generated images used for DPL on the experiment results. We take FER-2013 as source dataset and JAFFE as target dataset. The 0.4-0.2-0.2 weights combination is used for DPL and the learning rate is set to 0.000001. We stop fine-tuning after 10 epochs. From Figure 7 we can see that, at first, on both VGG11 and Alexnet, the recognition accuracy of the target dataset increases with the number of the generated samples. But after it peaks at 2k images, the accuracy improvement falls again. This is because, when the GAN images are too few, it is inadequate to fine-tune a CNN model toward the target dataset, whereas when the GAN images are too many, the CNN will give too much effort to classify those generated images with pseudolabels. The pseudolabels are not as trustworthy as the ground truth labels and we do not want a CNN to take them too seriously.

5. Conclusion

In this paper, we propose an unsupervised domain adaptation method, a method using GAN generated samples to improve the cross-dataset performance of facial expression recognition. When training the CNN with unlabelled GAN generated samples, we introduce a distributed pseudolabel method (DPL). With our method, domain adaptation can be achieved with limited target data without ground truth labels. Experiments have shown that our method outperforms other GAN-based domain adaptation methods and can get state-of-the-art cross-dataset recognition accuracy. When using FER-2013 as the source dataset, we obtain 15.02%, 3.76%, 3.72%, and 4.41% recognition accuracy improvement on the target dataset JAFFE (VGG11), JAFFE (Alexnet), MMI, and CK+, respectively. When using CK+ as the source dataset, we obtain 4.70% improvement of recognition accuracy on JAFFE and 4.68% improvement when using JAFFE as source dataset and CK+ as target dataset. Future work may extend the unsupervised DPL to a semisupervised version since the real-world samples with ground truth label in target dataset might provide better estimation of the target data. Also, it will be intriguing to apply our method to other domain adaptation tasks.

Data Availability

The datasets used during the current study are available in the following repository: http://www.kasrl.org/jaffe.html https://mmifacedb.eu/ https://www.kaggle.com/c/challenges-in-representation-learning-facial-expression-recognition-challenge http://www.pitt.edu/~emotion/ck-spread.htm.

Conflicts of Interest

The authors declare that there are no conflicts of interest regarding the publication of this paper.

Acknowledgments

This work was supported in part by National Natural Science Foundation of China, Grant no. 51575388.

References

[1] C. P. Latha and M. Priya, "A Review on Deep Learning Algorithms for Speech and Facial Emotion Recognition," *APTIKOM Journal on Computer Science and Information Technologies*, vol. 1, pp. 88–104, 2016.

[2] S. Jaiswal and M. Valstar, "Deep learning the dynamic appearance and shape of facial action units," in *Proceedings of the IEEE Winter Conference on Applications of Computer Vision, WACV 2016*, USA, March 2016.

[3] Q. You, J. Luo, H. Jin, and J. Yang, "Building a large scale dataset for image emotion recognition: The fine print and the benchmark," in *Proceedings of the 30th AAAI Conference on Artificial Intelligence, AAAI 2016*, pp. 308–314, USA, February 2016.

[4] S. K. Esser, P. A. Merolla, J. V. Arthur et al., "Convolutional networks for fast, energy-efficient neuromorphic computing," *Proceedings of the National Acadamy of Sciences of the United States of America*, vol. 113, no. 41, pp. 11441–11446, 2016.

[5] H.-W. Ng, V. D. Nguyen, V. Vonikakis, and S. Winkler, "Deep learning for emotion recognition on small datasets using transfer learning," in *Proceedings of the ACM International Conference on Multimodal Interaction, ICMI 2015*, pp. 443–449, USA, November 2015.

[6] V.-E. Neagoe, A. Barar, N. Sebe, and P. Robitu, "Deep Learning Approach for Subject Independent Emotion Recognition from Facial Expressions," *Recent Advances in Image, Audio and Signal Processing*, pp. 978-960, 2013.

[7] P. Ekman, W. V. Friesen, M. O'Sullivan et al., "Universals and cultural differences in the judgments of facial expressions of emotion," *Journal of Personality and Social Psychology*, vol. 53, no. 4, pp. 712–717, 1987.

[8] M. N. Dailey, C. Joyce, M. J. Lyons et al., "Evidence and a Computational Explanation of Cultural Differences in Facial Expression Recognition," *Emotion*, vol. 10, no. 6, pp. 874–893, 2010.

[9] O. Russakovsky, J. Deng, H. Su et al., "Imagenet large scale visual recognition challenge," *International Journal of Computer Vision*, vol. 115, no. 3, pp. 211–252, 2015.

[10] M. Long, Y. Cao, J. Wang, and M. I. Jordan, "Learning transferable features with deep adaptation networks," in *Proceedings of the 32nd International Conference on Machine Learning, ICML 2015*, pp. 97–105, France, July 2015.

[11] Y. Ganin and V. Lempitsky, "Unsupervised domain adaptation by backpropagation," in *Proceedings of the 32nd International Conference on Machine Learning, ICML 2015*, pp. 1180–1189, France, July 2015.

[12] E. Tzeng, J. Hoffman, T. Darrell, and K. Saenko, "Simultaneous deep transfer across domains and tasks," in *Proceedings of the 15th IEEE International Conference on Computer Vision, ICCV 2015*, pp. 4068–4076, Chile, December 2015.

[13] M. K. Abd El Meguid and M. D. Levine, "Fully automated recognition of spontaneous facial expressions in videos using random forest classifiers," *IEEE Transactions on Affective Computing*, vol. 5, no. 2, pp. 141–154, 2014.

[14] G. Wen, Z. Hou, H. Li, D. Li, L. Jiang, and E. Xun, "Ensemble of Deep Neural Networks with Probability-Based Fusion for Facial Expression Recognition," *Cognitive Computation*, vol. 9, no. 5, pp. 597–610, 2017.

[15] W. Gu, C. Xiang, Y. V. Venkatesh, D. Huang, and H. Lin, "Facial expression recognition using radial encoding of local Gabor features and classifier synthesis," *Pattern Recognition*, vol. 45, no. 1, pp. 80–91, 2012.

[16] R. Zhu, T. Zhang, Q. Zhao, and Z. Wu, "A transfer learning approach to cross-database facial expression recognition," in *Proceedings of the 8th IAPR International Conference on Biometrics, ICB 2015*, pp. 293–298, Thailand, May 2015.

[17] C. Mayer, M. Eggers, and B. Radig, "Cross-database evaluation for facial expression recognition," *Pattern Recognition and Image Analysis*, vol. 24, no. 1, pp. 124–132, 2014.

[18] I. J. Goodfellow, J. Pouget-Abadie, M. Mirza et al., "Generative adversarial nets," in *Proceedings of the 28th Annual Conference on Neural Information Processing Systems 2014, NIPS 2014*, pp. 2672–2680, Canada, December 2014.

[19] A. Radford, L. Metz, and S. Chintala, "Unsupervised representation learning with deep convolutional generative adversarial networks," 2015, https://arxiv.org/abs/1511.06434.

[20] X. Chen, Y. Duan, R. Houthooft, J. Schulman, I. Sutskever, and P. Abbeel, "InfoGAN: Interpretable representation learning by information maximizing generative adversarial nets," in *Proceedings of the 30th Annual Conference on Neural Information Processing Systems, NIPS 2016*, pp. 2180–2188, Spain, December 2016.

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

[22] A. Odena, "Semi-supervised learning with generative adversarial networks," 2016, https://arxiv.org/abs/1606.01583.

[23] Z. Zheng, L. Zheng, and Y. Yang, "Unlabeled Samples Generated by GAN Improve the Person Re-identification Baseline in Vitro," in *Proceedings of the 2017 IEEE International Conference on Computer Vision (ICCV)*, pp. 3774–3782, Venice, October 2017.

[24] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "Imagenet classification with deep convolutional neural networks," in *Proceedings of the 26th Annual Conference on Neural Information Processing Systems (NIPS '12)*, pp. 1097–1105, Lake Tahoe, Nev, USA, December 2012.

[25] K. Simonyan and A. Zisserman, "Very deep convolutional networks for large-scale image recognition," 2014, https://arxiv.org/abs/1409.1556.

[26] C. Szegedy, V. Vanhoucke, S. Ioffe, J. Shlens, and Z. Wojna, "Rethinking the inception architecture for computer vision," in *Proceedings of the 2016 IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2016*, pp. 2818–2826, July 2016.

[27] I. J. Goodfellow, D. Erhan, P. L. Carrier et al., "Challenges in representation learning: A report on three machine learning contests," in *Proceedings of the International Conference on Neural Information Processing*, pp. 117–124, 2013.

[28] P. Giannopoulos, I. Perikos, and I. Hatzilygeroudis, "Deep learning approaches for facial emotion recognition: A case study on FER-2013," in *Advances in Hybridization of Intelligent Methods*, vol. 85, pp. 1–16, Springer, Berlin, Germany, 2018.

[29] M. I. Georgescu, R. T. Ionescu, and M. Popescu, "Local Learning with Deep and Handcrafted Features for Facial Expression Recognition," 2018, https://arxiv.org/abs/1804.10892.

[30] M. Lyons, S. Akamatsu, M. Kamachi, and J. Gyoba, "Coding facial expressions with Gabor wavelets," in *Proceedings of the 3rd IEEE International Conference on Automatic Face and Gesture Recognition, FG 1998*, pp. 200–205, Japan, April 1998.

[31] M. J. Lyons, "Automatic classification of single facial images," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 21, no. 12, pp. 1357–1362, 1999.

[32] P. Lucey, J. F. Cohn, T. Kanade, J. Saragih, Z. Ambadar, and I. Matthews, "The extended Cohn-Kanade dataset (CK+): a complete dataset for action unit and emotion-specified expression," in *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition Workshops (CVPRW '10)*, pp. 94–101, IEEE, San Francisco, Calif, USA, June 2010.

[33] M. Pantic, M. Valstar, R. Rademaker, and L. Maat, "Web-based database for facial expression analysis," in *Proceedings of the IEEE International Conference on Multimedia and Expo (ICME '05)*, pp. 317–321, Amsterdam, Netherlands, July 2005.

[34] M. Valstar and M. Pantic, "Induced disgust, happiness and surprise: an addition to the mmi facial expression database," in *Proceedings of the 3rd Intern. Workshop on EMOTION (satellite of LREC): Corpora for Research on Emotion and Affect*, p. 65, 2010.

[35] D.-H. Lee, "Pseudo-label: The simple and efficient semi-supervised learning method for deep neural networks," in *Proceedings of the Workshop on Challenges in Representation Learning, ICML*, vol. 3, p. 2, 2013.