applied sciences

MDPI

Article

Unsupervised Domain Adaptation with Coupled Generative Adversarial Autoencoders

Xiaoqing Wang and Xiangjun Wang *

State Key Laboratory of Precision Measuring Technology and Instruments, Tianjin University, Tianjin 300072, China; tjuwxq@tju.edu.cn
* Correspondence: tjuxjw@126.com; Tel.: +86-022-2740-3395

Received: 9 November 2018; Accepted: 5 December 2018; Published: 7 December 2018

Abstract: When large-scale annotated data are not available for certain image classification tasks, training a deep convolutional neural network model becomes challenging. Some recent domain adaptation methods try to solve this problem using generative adversarial networks and have achieved promising results. However, these methods are based on a shared latent space assumption and they do not consider the situation when shared high level representations in different domains do not exist or are not ideal as they assumed. To overcome this limitation, we propose a neural network structure called coupled generative adversarial autoencoders (CGAA) that allows a pair of generators to learn the high-level differences between two domains by sharing only part of the high-level layers. Additionally, by introducing a class consistent loss calculated by a stand-alone classifier into the generator optimization, our model is able to generate class invariant style-transferred images suitable for classification tasks in domain adaptation. We apply CGAA to several domain transferred image classification scenarios including several benchmark datasets. Experiment results have shown that our method can achieve state-of-the-art classification results.

Keywords: unsupervised domain adaptation; generative adversarial networks; autoencoder

1. Introduction

Large-scale well-annotated datasets such as Microsoft COCO [1], ImageNet [2] and KITTI [3] have played a vital role in the recent success of deep learning based models on computer vision tasks such as image classification, target detection, semantic segmentation and so on. However, models trained with large datasets still cannot generalize well to novel datasets when these datasets have different feature distributions. The typical solution is to further fine-tune these models on the task specific datasets. However, creating such datasets can be expensive and time-consuming. Unsupervised domain adaptation offers a solution to this problem by learning a mapping between a labeled dataset (source domain) and an unlabeled dataset (target domain) or by learning domain invariant features. Conventional domain adaptation approaches for image classification are usually developed in two separate steps: designing and extracting fixed features and then training models to reduce their differences in either the marginal distributions or the conditional distributions between domains [4–7]. Recent deep learning based domain adaptation approaches avoid the difficulty of feature design by extracting features automatically through convolutional neural networks [8–13].

Among all kinds of deep neural network based domain adaptation approaches, generative adversarial network (GAN) [14] has become a popular branch. A typical GAN trains a generator and a discriminator to compete against each other. The generator is trained to produce synthetic images as real as possible, whereas the discriminator is trained to distinguish the synthetic and real images. When applying GAN to domain adaptation for image classification, there are two major types of approaches. The first type trains a GAN to generate unlabeled target domain images, thus enlarging the data

volume to train a more robust image classifier [15–17]. In these methods, the training strategy of the final classifier need to be carefully designed since the newly generated images have no label. The other type of approaches generate labeled target domain images directly by transferring the source domain images into target domain style and have achieved some state-of-the-art results, such as CoGAN [18] and UNIT [19]. These methods are based on the shared latent space assumption, which assumes that the differences of the source domain and the target domain are primarily low-level, and that the two domains share a common high-level latent space. This assumption works well for simple scenarios such as digits adaptation between MNIST [20] and USPS [21] but faces challenges when the semantic features are more complex. When shared high-level latent space in different domains does not exist or such latent space is not as ideal as assumed, these methods will fail [18].

In this paper, we propose an unsupervised domain adaptation method for image classification by combining generative adversarial networks with autoencoders. We call our proposed network architecture Coupled Generative Adversarial Autoencoders (CGAA). Our work is perhaps most similar to CoGAN and UNIT, but we try to solve the aforementioned shortcomings of these methods by the following designs: CGAA consists of a pair of generative adversarial networks (GAN) and a domain adaptive classifier. The architecture of the generator in GAN is designed based on the autoencoder. During training, part of the layers in the generators are forced to share their weights, which gives our model the ability to learn the domain transformation in an unsupervised manner and generate synthetic target domain images with label. By decoupling the highest level layer, we give our model the capacity to tolerant the differences of high-level features between the domains. The classifier provides a class-invariant loss to help the generator produce more suitable images for the classification task in domain adaptation. The main contributions of this work are:

- We propose an unsupervised domain adaptation method for image classification. Our method trains a pair of coupled generative adversarial networks in which the generator has an encoder-decoder structure.
- We force part of the layers in the generator to share weights during training to generate labeled synthetic images, and make the highest level layer decoupled for different high-level representations.
- We introduce a class consistent loss into the GAN training, which is calculated from the output of a stand-alone domain adaptive classifier. It can help the generator to generate more suitable images for domain adaptation.

2. Related Work

The goal of unsupervised domain adaptation is to transfer knowledge from a labeled source dataset to a target dataset where labeled data is not available. Recent studies have tried to learn transferable features with deep neural networks. The DDC method [11] learned domain invariant representations by introducing an adaptation layer and a Maximum Mean Discrepancy (MMD) domain confusion loss. The work in [22] extended the MMD to jointly mitigate the gaps of marginal and conditional distributions between source and target domain. The DAN method [9] embedded task-specific layers in a reproducing kernel Hilbert space to enhance the feature transferability. The DANN method [8,23] suggested that the features suitable for domain adaptation should be both discriminative and domain-invariant and added a domain classifier at the end of the feature extractor to learn domain invariant features. CAN [24] suggested that some characteristic information from target domain data may be lost after learning domain-invariant features with DANN. Therefore, CAN introduced a set of domain classifiers into multiple blocks to learn domain-informative representations at lower blocks and domain-uninformative representations at higher blocks. The work of [25] proposed to learn a representation that transfered the semantic structure from a well labeled source domain to the sparsely labeled target domain by adding a domain classifier and a domain confusion loss. The DRCN [12] proposed a model which had two pipelines: The first was label prediction for the source domain and the second was data reconstruction for the target domain. ADDA [26] learns

the representation of the source domain and then maps the target data to the same space through a domain-adversarial loss.

Other works have attempted to use GANs [14] into image-to-image translation and domain adaptation. The "pix2pix" framework [27] used a conditional generative adversarial network to learn a mapping from input to output images with paired images. CycleGAN [28] learned the mapping without paired training examples using a cycle consistency loss. The method in [29] used GAN to translate unpaired images between domains while remain high level semantic information aligned by introducing attention consistent loss. CoGAN [18] learned a joint distribution of images without corresponding supervision by training two GANs to generate the source and target images respectively given the same noise input and tying the high-level layer parameters of the two GANs. Instead of generating images from noise vectors, PixelDA [30] generated style-transferred images conditioned on the source images. CoGASA [31] integrated a stacked autoencoder with the CoGAN, and UNIT [19] proposed an image-to-image translation framework based on CoGAN and VAE [32].

3. Proposed Approach

In this section, we introduce the model structure of CGAA and explain our training strategy. As illustrated in Figure 1, CGAA contains seven sub-networks. Two image encoders ENC_S and ENC_T, two image decoders DEC_S and DEC_T, two adversarial discriminators D_S and D_T, and a classifier C.

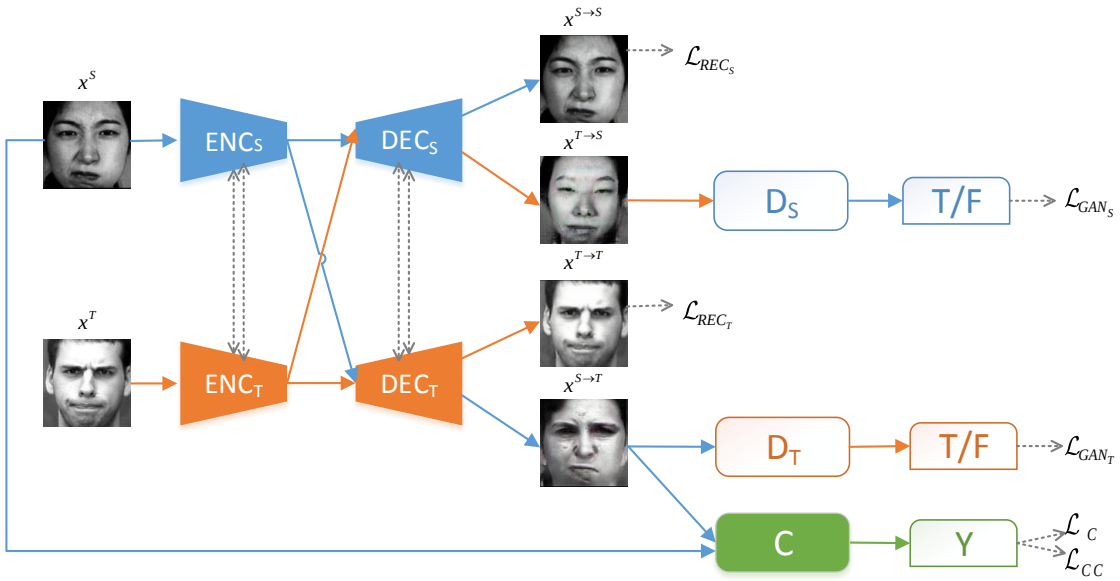


Figure 1. Overview of our model architecture. x^S and $x^T$are images from the source and target domain. The encoder ENC_S and ENC_T are two sequences of convolution layers (including Resnet blocks [33]) that map images to a code in a higher level latent space, DEC_S and DEC_T are two sequences of de-convolution layers (including Resnet blocks) that generate images from the outputs of the encoder. The Discriminator D_S and D_T determine whether an image is real or synthesized. During training, we share the weights of the two encoders except for the first and the last layer. Similarly, the weights of decoders are also tied except the first and the last layer. $DEC_S\left(ENC_S\left(x^S\right)\right) \to x^{S\to S}$ and $DEC_T\left(ENC_T\left(x^T\right)\right) \to x^{T\to T}$ are reconstructed images. $DEC_S\left(ENC_T\left(x^T\right)\right) \to x^{T\to S}$ and $DEC_T\left(ENC_S\left(x^S\right)\right) \to x^{S\to T}$ are style-transferred images. C is the classifier trained by the source images and the style-transferred source images.

3.1. Image Reconstruction and Autoencoder

The encoder ENC_S and decoder DEC_S constitute an autoencoder for the source domain X^S . The ENC_S maps an input image $x^S \in X^S$to a code in a latent space and based on this code, the DEC_S reconstructs the input image as $x^{S\to S}$. Similarly, ENC_T and DEC_T constitute an autoencoder for the

target domain X^T . The aim of these two autoencoders is to reconstruct images as similar as possible to their input images in each domain. We use the mean squared error as the loss function to penalize the differences between inputs and outputs:

$$\mathcal{L}_{REC_S}(ENC_S, DEC_S) = \mathbb{E}_{X^S}[\frac{1}{k}\|x^S - DEC_S(ENC_S(x^S))\|_2^2] \tag{1}$$

$$\mathcal{L}_{REC_T}(ENC_T, DEC_T) = \mathbb{E}_{X^T}[\frac{1}{k}\|x^T - DEC_T(ENC_T(x^T))\|_2^2] \tag{2}$$

where k is the number of pixels in input x^S and $\|\cdot\|$ is the squared L_2-norm.

3.2. Style Transfer and GAN

Style-transferred synthetic images can be generated by changing the combination of encoders and decoders. More specifically, let DEC_T take the output of ENC_S, and let DEC_S take the output of ENC_T, thus we are able to change the style of images between domains.

When training an autoencoder, element level penalties such as squared error, is the classic choice. However, as discussed in [34], they are actually not ideal for image generation, and the generated images are always blurred. Therefore, we combine autoencoder with GAN in our method. By jointly training an autoencoder and a GAN, we can generate better images with the feature level metric expressed by the discriminator. In our method CGAA, the ENC_S, DEC_T and D_T constitute a generative adversarial network. During training, DEC_T takes the output of ENC_S, mapping an input source domain image x^S into a target domain style synthetic image $x^{S\to T}$, and discriminator D_T is trained to distinguish between synthetic images $x^{S\to T}$ and real images x^T from the target domain. Similarly, ENC_T and DEC_S generate synthetic source-style images $x^{T\to S}$ conditioned on the target domain images x^T and D_S is trained to distinguish between real source domain images x^S and synthetic images $x^{T\to S}$. With this pair of GANs, our goal is to minmax the following object:

$$\begin{aligned}\min_{ENC_S,ENC_T,DEC_S,DEC_T,C} \max_{D_S,D_T} \quad & \alpha\mathcal{L}_{GAN_S}(ENC_T, DEC_S, D_S) + \beta\mathcal{L}_{REC_S}(ENC_S, DEC_S) \\ & \alpha\mathcal{L}_{GAN_T}(ENC_S, DEC_T, D_T) + \beta\mathcal{L}_{REC_T}(ENC_T, DEC_T)\end{aligned} \tag{3}$$

where α and β are weights that balance the GAN loss and the reconstruction loss. $\mathcal{L}_{GAN_S}$ and $\mathcal{L}_{GAN_T}$ represent the GAN loss:

$$\mathcal{L}_{GAN_S}(ENC_T, DEC_S, D_S) = \mathbb{E}_{X^S}[\log D_S(x^S)] + \mathbb{E}_{X^T}[\log(1 - D_S(DEC_S(ENC_T(x^T))))] \tag{4}$$

$$\mathcal{L}_{GAN_T}(ENC_S, DEC_T, D_T) = \mathbb{E}_{X^T}[\log D_T(x^T)] + \mathbb{E}_{X^S}[\log(1 - D_T(DEC_T(ENC_S(x^S))))] \tag{5}$$

3.3. Weight Sharing

Previous shared latent space assumption based methods such as CoGAN [18] and UNIT [19] are able to conduct the domain transfer training without paired images in different domains by sharing weights in the generators. They assume that images from different domains only have low-level semantic differences due to noise, resolution, illumination and color, etc. Furthermore, a pair of corresponding images in two domains share the same high-level concepts. Therefore, layers responsible for high level representation are forced to share their weights. However, these methods are based on the existence of shared high-level representations in the two domains. If the high-level semantic features are complex and such shared representations do not exist or are hard to find, these methods will not work out well. To this end, our method extends the previous works by only sharing part of the high-level layers and decoupling the rest. More specifically, we do not share the weights of last layer in the encoder, the first layer in decoder, and the last two layers in the discriminator, as shown

is Figure 2. Under this structure, the generative models will, to some extent, tolerate different high level representations in different domains.

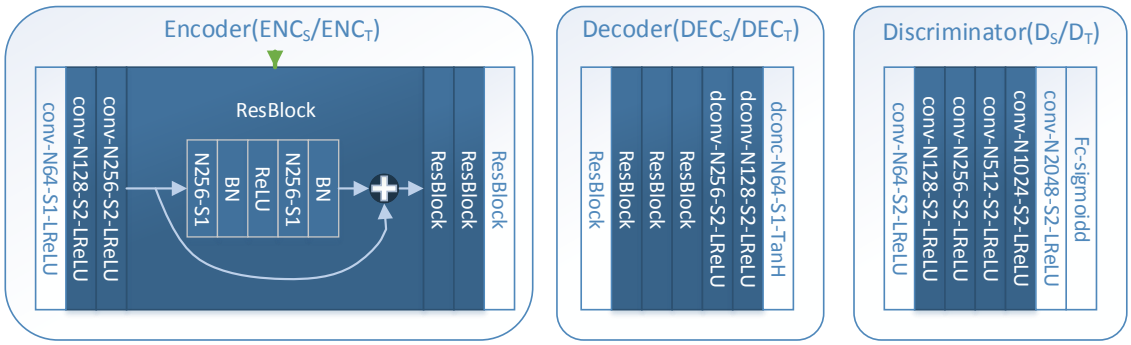


Figure 2. The network architecture. The ENC_S and ENC_T are of the same structure as the encoder shown in this figure, so are the two decoders and the two discriminators. The convolution layer is denoted as conv, the transposed convolution layer (deconvolution layer) is denoted as dconv and the residual block is denoted as ResBlock. N means neurons (channels), S means stride, and LReLU means leaky ReLU. BN stands for batch normalization layer and Fc stands for fully connected layer. We share the weights of the dark-color layers in the coupled models during training.

3.4. Domain Adapted Classifier

The focus of the unsupervised domain adaptation method described in this paper is to extend a classifier's generalization ability on two domains, originally trained on the source domain that generalizes to the unlabeled target domain. To this end, we train a classifier C with the source domain images and the synthetic target domain images generated by $\{ENC_S, DEC_T\}$. Unlike some other domain adaptation works where the discriminator is modified as classifier, our classifier has a stand-alone structure, shown in Figure 1, which is easy to be detached from the whole network for future training. We do not describe the detailed architecture of the classifier in Figure 2 because it is task specific. During training, we use the typical cross-entropy loss to optimize C:

$$\mathcal{L}_C(ENC_S, DEC_T, C) = \mathbb{E}_{x^S, y^S}[-y^S \log C(DEC_T(ENC_S(x^S))) - y^S \log C(x^S)] \tag{6}$$

In addition, the classifier C has another function in CGAA, that is being a part of the optimization of the generator $\{ENC_S, DEC_T\}$ with a class-consistency loss. When training the generator, C assigns a label $\hat{y}$ the generated image $x^{S\to T}$, and the class-consistency loss is defined as:

$$\mathcal{L}_{CC}(ENC_S, DEC_T, C) = \mathbb{E}_{x^S, y^S}[-y^S \log C(DEC_T(ENC_S(x^S)))] \tag{7}$$

where y^S is the class label of the input x^S. The class-consistency loss makes sure the output image $x^{S\to T}$ remains class-invariant, which is essential for the classification task in domain adaptation. With $\mathcal{L}_C$ and $\mathcal{L}_{CC}$, our final optimization object becomes:

$$\begin{aligned} \min_{ENC_S, ENC_T, DEC_S, DEC_T, C} \quad \max_{D_S, D_T} \quad & \alpha\mathcal{L}_{GAN_S}(ENC_T, DEC_S, D_S) + \beta\mathcal{L}_{REC_S}(ENC_S, DEC_S) \\ & \alpha\mathcal{L}_{GAN_T}(ENC_S, DEC_T, D_T) + \beta\mathcal{L}_{REC_T}(ENC_T, DEC_T) + \gamma\mathcal{L}_{CC}(ENC_S, DEC_T, C) \\ & \mathcal{L}_C(ENC_S, DEC_T, C) \end{aligned} \tag{8}$$

The minmax optimization of Equation (8) is achieved by two alternative steps. During the first step, we keep the discriminators and the classifier fixed, optimize the generators and at the same time, minimize the reconstruction losses and the class consistent loss. During the second step, we keep the generators fixed and optimize the discriminators and the classifier.

4. Experiment Results and Evaluation

To evaluate our method, we conduct experiments on various domain adaptation scenarios and compare our results with other recently reported methods.

4.1. Facial Expression Recognition

We first evaluate our method on cross domain facial expression recognition task with three publicly available facial expression datasets: JAFFE, MMI and CK+. The images in these datasets have different resolutions and illuminations, and the subjects vary in gender, age and cultural background. Figure 3 shows some of the sample images from these datasets.

JAFFE dataset [35,36] contains 213 facial expression images. These images are from 10 Japanese females with seven expressions (angry, disgust, fear, happy, sad, surprise and neutral). We use all of the images in JAFFE in our experiments.

MMI dataset [37,38] consists of over 2900 videos as well as still images of 75 subjects, in which 235 videos have emotional labels. We choose the peak frame of each video that has the six basic emotions (angry, disgust, fear, happy, sad and surprise) and the first frame of these videos as neutral emotion images. In total we use 242 images from MMI.

CK+ dataset [39] consists of 593 image sequences from 123 subjects, 327 sequences of which have emotional labels. The dataset labels seven expressions including angry, disgust, fear, happy, sad, surprise, and contempt. We only choose the peak frame from the sequences labelled with the first six expressions. In addition, we choose the first frame from some of the sequences as neutral samples. In total we use 363 images from CK+.

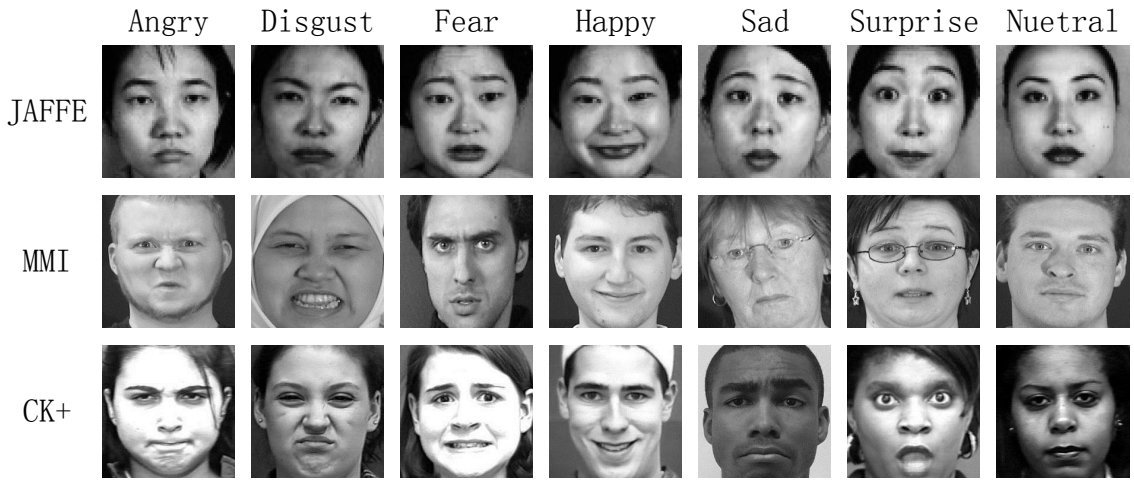


Figure 3. Sample images from facial expression recognition datasets.

In this experiment, the network structure of our method is shown in Figure 2. Since the facial expression datasets are rather small, to avoid over-fitting, we use the Alexnet model pre-trained on ImageNet as the base model of the classifier and fine-tune it in our experiment. Table 1 shows experiment results of the classifier's accuracy tested on the target domain. In Table 1, the source model is trained with only the labeled source dataset. As for the adapted model, to evaluate the effectiveness of our proposed method, we train with three different settings. In all three settings, the parameters of the low-level layers in encoders, decoders and discriminators are not shared, which are the first layer of the encoder, the last layer of the decoder and the first layer of the discriminator. As for the high-level layers, the first experiment shares the weights of all of these layers, which is a similar structure to UNIT described in [19]. The second experiment has decoupled high-level layers in ENC and DEC, which means we do not share the last layer of the two encoders and the first layer of the two decoders. The last one is to have decoupled high-level layers in ENC, DEC as well as D, which is the same setting

described in Figure 2. A decoupled D means we do not share the last two layers in the discriminators. Figure 4 shows examples of the style-transferred images generated by UNIT and the last experiment setting of CGAA.

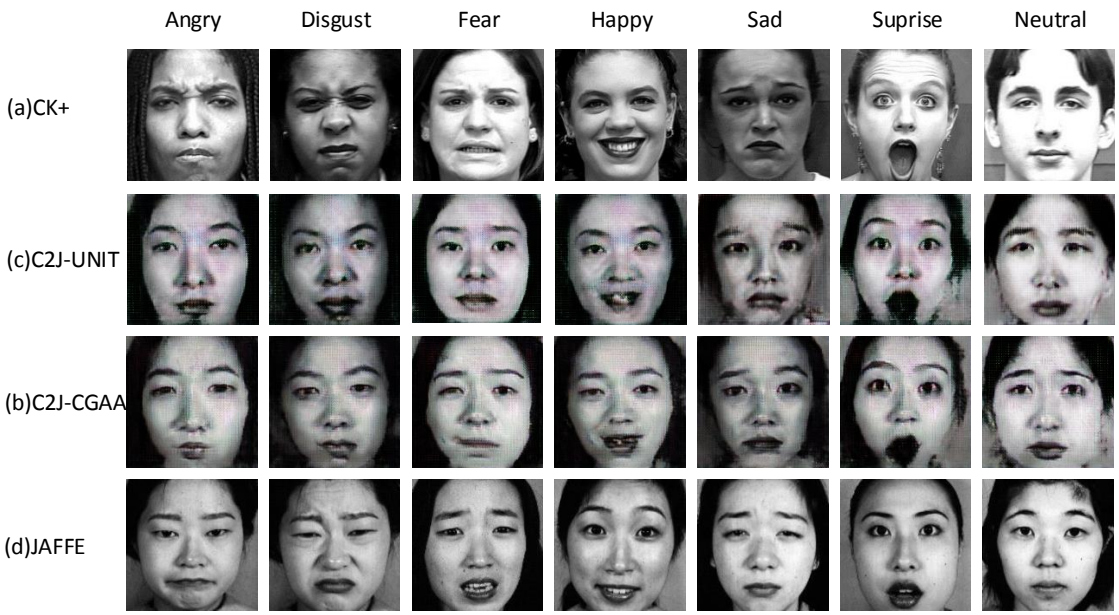
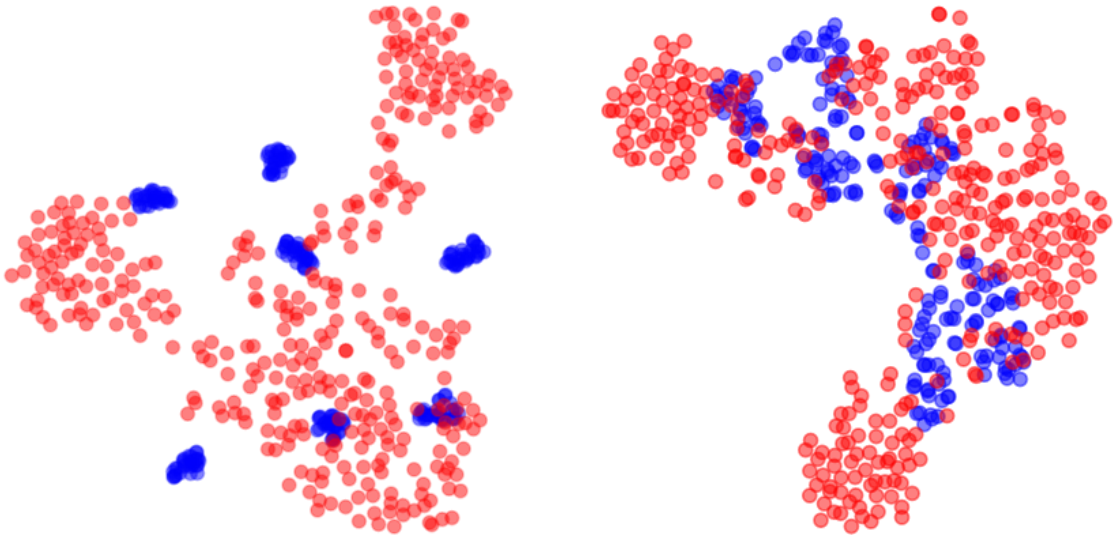


Figure 4. The style-transferred images generated by our model when trained to adapt CK+ to JAFFE. (**a**) The Source images from CK+. (**b**) The generated samples when adapt CK+ images in (**a**) into JAFFE style using UNIT. (**c**) The generated samples when adapt CK+ images in (**a**) into JAFFE style using our method CGAA. (**d**) Random target images from the JAFFE.

Table 1. Recognition accuracy evaluation for domain adaptation on facial expression datasets. JAFFE(J), MMI(M), CK+(C). J→M means J is the source and M is the target. Bold numbers are the best results.

Training Method	Decoupled Layers ENC	DEC	D	J→M	M→J	C→M	M→C	J→C	C→J
Source				0.330	0.362	0.428	0.697	0.634	0.437
UNIT	–	–	–	0.507	0.470	0.567	0.719	0.733	0.526
CGAA	√	√	–	**0.521**	0.460	**0.581**	**0.736**	0.744	0.559
	√	√	√	**0.521**	**0.498**	**0.581**	**0.736**	**0.769**	**0.573**

In each domain adaptation, we use all available source examples and target examples and resize them to 224 × 224 pixels to train the generative adversarial networks. Only the label information of the source dataset is used to train the classifier. Optimization is done on Pytorch using Adam with the learning rate set as 0.0001 and the weight decay as 0.0001. The α, β and γ in Equation (8) are set as 1.0, 1.0 and 0.1. Figure 4 shows some of the style-transferred images generated in the C→J domain adaptation under our last kind of network setting. The experiment results in Table 1 show that our CGAA model with partially-decoupled high level layers outperforms the model with all the high level layers tied-up in all six domain adaptations. In addition, we find that decoupling the encoder-decoder can lead to a significant increase of recognition accuracy, whereas decoupling the discriminator has only a small impact on the experiment result. Therefore, in other experiments described in this paper, we use the last setting in Table 1 as CGAA for evaluation. We visualize the feature distribution of the two domains before and after the adaptation (J→C), as shown in Figure 5. Figure 5 proves that our model can make the distribution of the features from the two domains much closer, which brings about a higher accuracy in the classification.

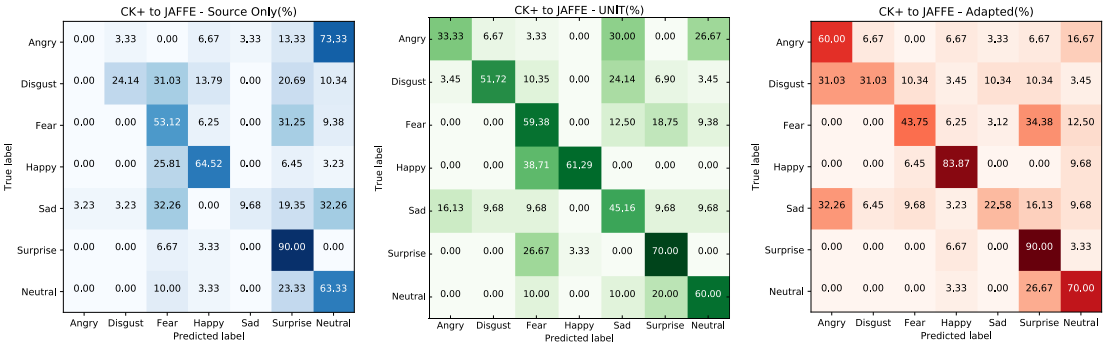


(a) Source-only J → C (b) Adapted J → C

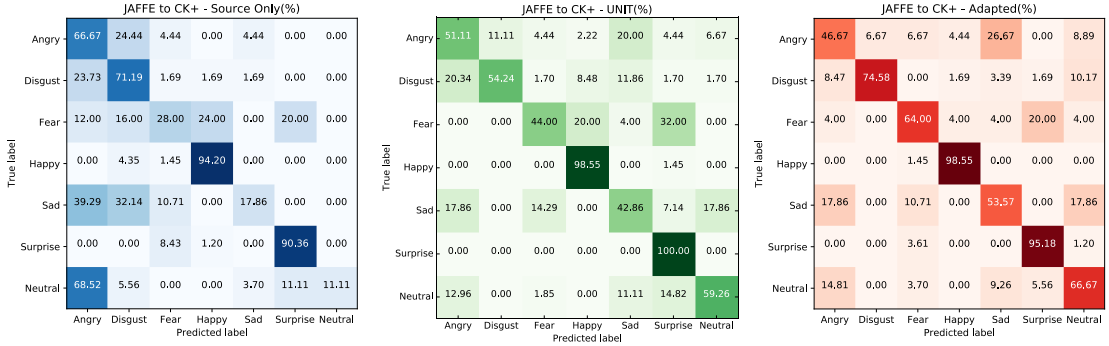
Figure 5. The effect of our adaptation depicted by t-SNE [40] visualizations of the extracted features from the last hidden layer of the classifier in the JAFFE to CK+ scenario. Blue dots are examples from the source dataset JAFFE, red dots are examples from the target dataset CK+. (**a**) is when only the source dataset is used for training. (**b**) is when our adaptation procedure is done. The adaptation of our method makes the distribution of the features from the two datasets much closer.

To further evaluate the effectiveness of our method, we compare the confusion matrices of the class-wise classification accuracy on target domain before and after adaptation.

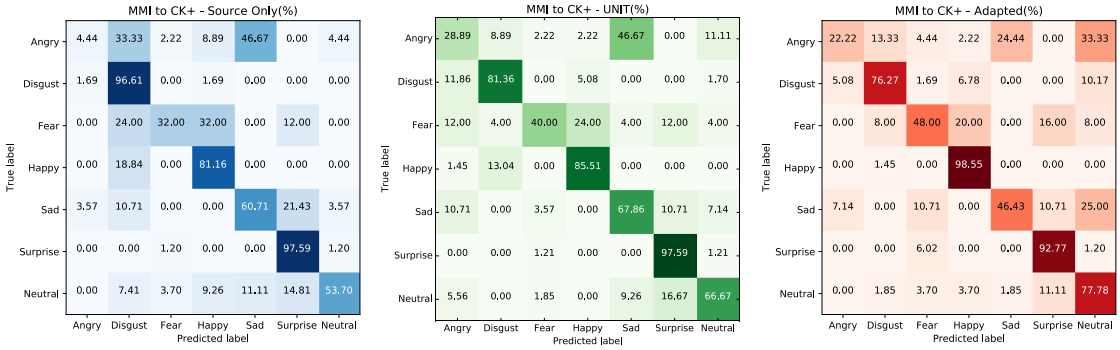
As shown in Figure 6, the blue ones are the confusion matrices when only source domain images are used for training, the green ones are matrices when UNIT is used for domain adaption and the red ones are matrices when our method is used for domain adaptation. When trained on source domain only, the model have difficulties separating *Angry* and *Neutral* between CK+ and JAFFE (see Figure 6a,b, and also cannot seperate *Angry* and *Sad* between MMI and CK+ (see Figure 6c,d). When trained on MMI and tested on JAFFE, the model misclassifies a lot of images as *Surprised* (see Figure 6e) whereas when trained on JAFFE and tested on MMI, the model misclassifies most of the images as *Angry* (see Figure 6f). These misclassifications are caused by the semantic gap between domains. Figure 6 shows that our domain adaptation method can help the model to cross the semantic gap between domains and increase the class-wise classification accuracies.



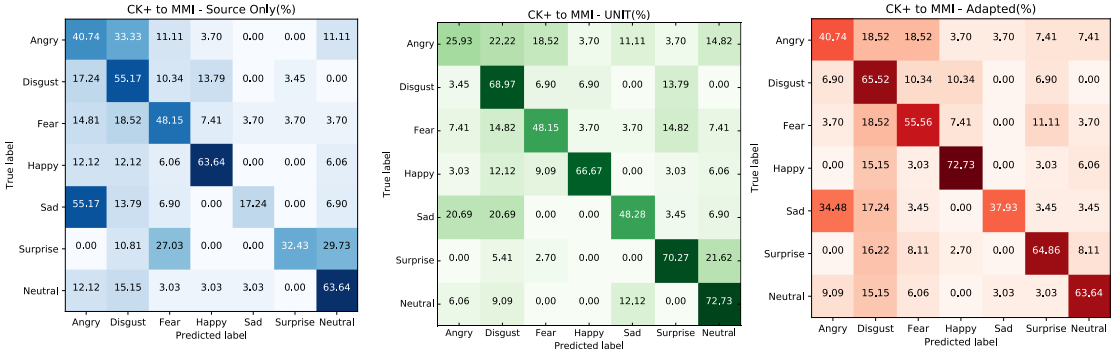
(**a**) CK+ to JAFFE



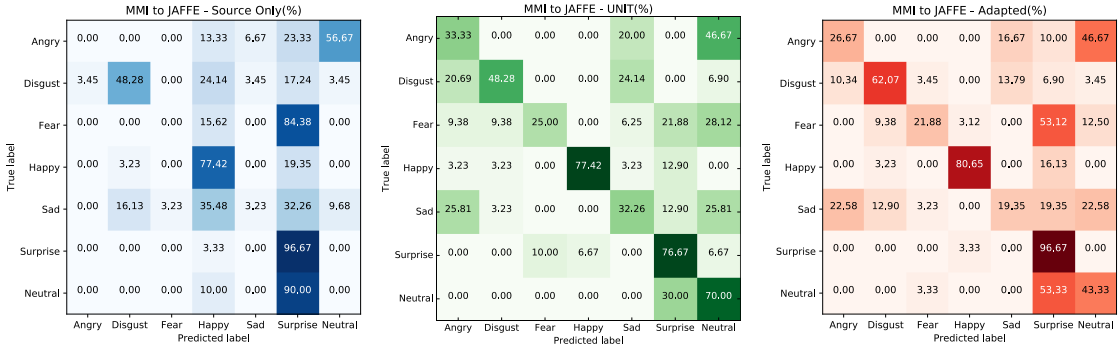
(**b**) JAFFE to CK+



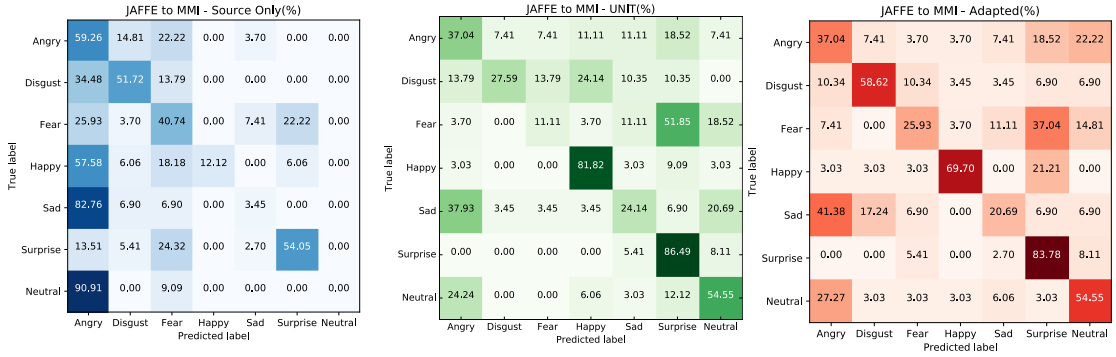
(**c**) MMI to CK+



(**d**)CK+ to MMI



(**e**) MMI to JAFFE



(**f**) JAFFE to MMI

Figure 6. Confusion matrices of the classification accuracy on target domain in facial expression recognition experiment.

4.2. Office Dataset

In this experiment, we evaluate our method on the Office dataset [41]. This is the most popular benchmark dataset for object recognition in the domain adaptation field. This dataset has 4410 images across 31 classes of everyday objects in three domains: amazon (A), webcam (W), and dslr (D). The amazon contains product pictures with no background from the Amazon website, and images in webcam and dslr contains similar real-world objects with different resolution. Following previous domain adaptation work [26], we use ResNet-50 as the model structure for the classifier. Other sub-parts of the model are the same as those shown in Figure 2. We adopt the common "fully-transductive" training protocol [8,9,26], using all available labeled source examples and unlabeled target examples. Optimization is done on Pytorch using Adam for 10 epochs with the learning rate set as 0.0001 and the weight decay set as 0.0001. The α, β and γ in Equation (8) are 1.0, 1.0 and 0.1. We evaluate our method on all six domain adaptations and compare our method with other reported domain adaptation approaches (some of them only have experiment results on three domain adaptations). We also implement two other methods on Pytorch based on share-latent space assumption, which are CoGAN and UNIT. Note that in the original papers of these two methods, the classifier is gained by attaching a softmax layer to the last hidden layer of the discriminator. Whereas in our implementation of these methods, we train a stand-alone classifier with the same structure of our method for a fair comparison. The experiment results in Table 2 show that our method is a competitive method and achieve state-of-the-art results compared with previously-reported methods, except for D→W. The method proposed in this paper aims to solve the problem of domain adaptation when the high-level features in the two domains are different and the shared high-level latent space cannot be established. As shown in Figure 7, the images of webcam (W) and dslr (D) are actually very similar, only having differences in the illumination and the image resolution. In other words, their high-level features are the same. Therefore, our method did not achieve better results than other methods in this particular task, but obtained better results in other more challenging tasks with obvious high-level feature differences, such as W→A and D→A.

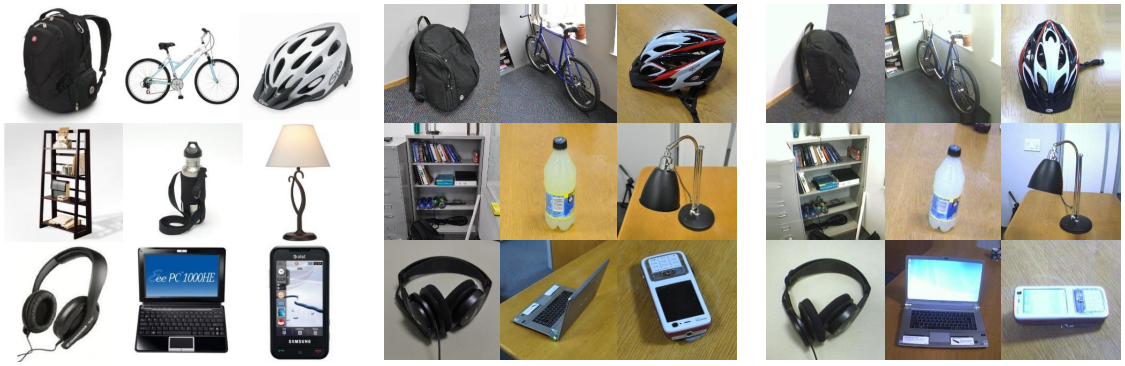


Figure 7. Sample images from Office dataset.

Table 2. Recognition accuracy evaluation for domain adaptation on the Office dataset. Amazon (A), webcam (W), dslr (D). A→W indicates A is the source dataset and W is the target dataset. Bold numbers are the best results.

Training Method	A→W	W→A	W→D	D→W	A→D	D→A
Source	0.670	0.498	0.952	0.941	0.689	0.515
DDC	0.594	–	0.917	0.925	–	–
DAN	0.685	0.531	0.990	0.960	0.670	0.540
DAH	0.683	0.530	0.988	0.961	0.665	0.555
DRCN	0.687	0.549	0.990	0.964	0.668	0.560
DANN	0.730	–	0.992	0.964	–	–
RTN	0.733	0.510	**0.996**	0.968	0.710	0.505
ADDA	0.751	–	**0.996**	**0.970**	–	–
CoGAN	0.745	0.549	**0.996**	0.968	0.710	0.560
UNIT	0.751	0.566	0.992	0.968	0.715	0.568
CGAA	**0.752**	**0.575**	**0.996**	0.957	**0.723**	**0.572**

4.3. Office-Home Dataset

Finally, we test our model on Office-home dataset [13]. This is a newer, larger and more challenging dataset compared to the classic Office dataset. It has about 15,500 images cross 4 domains, with each domain containing images from 65 classes of everyday objects. As shown in Figure 8, the four domains are: Art, Clipart, Product and Real-world. Images in Art are artistic depictions of objects and Clipart contains clipart images. Product consists of images of objects without background and Real-World consists of images of objects captured with a camera. We conduct this experiment with the same setting as the classic Office dataset experiment. Table 3 shows that our results outperform competitors in all of the domain adaptations in this experiment.

Table 3. Recognition accuracy evaluation for domain adaptation on the office-home dataset. Art (A), Clipart (C), Product (P), Real-World (R). A→C indicates A is the source dataset and C is the target dataset. Bold numbers are the best results.

Method	A→C	A→P	A→R	C→A	C→P	C→R	P→A	P→C	P→R	R→A	R→C	R→P
DAN [1]	0.307	0.422	0.541	0.328	0.476	0.498	0.291	0.341	0.567	0.436	0.383	0.627
DANN [1]	0.333	0.430	0.544	0.322	0.491	0.498	0.305	0.381	0.568	0.447	0.427	0.647
DAH	0.316	0.408	0.517	0.347	0.519	0.528	0.299	0.396	0.607	0.450	0.451	0.625
CoGAN	0.399	0.545	0.672	0.471	0.570	0.579	0.478	0.406	0.635	0.580	0.489	0.728
UNIT	0.404	0.554	0.670	0.480	0.572	0.583	0.509	0.412	0.658	0.599	0.503	0.726
CGAA	**0.434**	**0.571**	**0.676**	**0.499**	**0.577**	**0.591**	**0.517**	**0.435**	**0.662**	**0.612**	**0.517**	**0.749**

[1] Results reproduced from [13].

Figure 8. Sample images from Office-home dataset.

5. Conclusions

In this paper, we proposed an unsupervised domain adaptation method called coupled generative adversarial autoencoders. The weight-sharing training strategy proposed in this paper extends the shared high-level latent space assumption and improves the tolerance of the model to the differences in high-level semantic features between domains. Under this training strategy, our model can generate style-transferred images with unpaired images in the two domains and domain adaptation is done by training a classifier with the target-style images generated from the source images. With this proposed method, we achieve state-of-the-art experiment results on various domain adaptation scenarios including popular benchmark datasets.

Author Contributions: X.W. (Xiaoqing Wang) performed the experiments, analyzed the data and wrote the paper; X.W. (Xiangjun Wang) contributed the GPU used in the experiments and modified the paper.

Acknowledgments: This work was supported in part by the National Natural Science Foundation of China: 51575388.

Conflicts of Interest: The authors declare no conflict of interest.

References

1. Lin, T.Y.; Maire, M.; Belongie, S.; Hays, J.; Perona, P.; Ramanan, D.; Dollár, P.; Zitnick, C.L. Microsoft coco: Common objects in context. In *European Conference on Computer Vision*; Springer: Cham, Switzerland, 2014; pp. 740–755.
2. Deng, J.; Dong, W.; Socher, R.; Li, L.J.; Li, K.; Fei-Fei, L. Imagenet: A large-scale hierarchical image database. In Proceedings of the 2009 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Miami, FL, USA, 20–25 June 2009; pp. 248–255.
3. Geiger, A.; Lenz, P.; Urtasun, R. Are we ready for autonomous driving? the kitti vision benchmark suite. In Proceedings of the 2012 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Providence, RI, USA, 16–21 June 2012; pp. 3354–3361.
4. Fernando, B.; Habrard, A.; Sebban, M.; Tuytelaars, T. Unsupervised visual domain adaptation using subspace alignment. In Proceedings of the 2013 IEEE International Conference on Computer Vision (ICCV), Sydney, NSW, Australia, 1–8 December 2013; pp. 2960–2967.
5. Gong, B.; Shi, Y.; Sha, F.; Grauman, K. Geodesic flow kernel for unsupervised domain adaptation. In Proceedings of the 2012 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Providence, RI, USA, 16–21 June 2012; pp. 2066–2073.
6. Long, M.; Wang, J.; Ding, G.; Sun, J.; Yu, P.S. Transfer feature learning with joint distribution adaptation. In Proceedings of the IEEE International Conference on Computer Vision, Sydney, Australia, 1–8 December 2013; pp. 2200–2207.
7. Pan, S.J.; Tsang, I.W.; Kwok, J.T.; Yang, Q. Domain adaptation via transfer component analysis. *IEEE Trans. Neural Netw.* **2011**, *22*, 199–210. [CrossRef] [PubMed]
8. Ganin, Y.; Ustinova, E.; Ajakan, H.; Germain, P.; Larochelle, H.; Laviolette, F.; Marchand, M.; Lempitsky, V. Domain-adversarial training of neural networks. *J. Mach. Learn. Res.* **2016**, *17*, 2030–2096.

9. Long, M.; Cao, Y.; Wang, J.; Jordan, M.I. Learning transferable features with deep adaptation networks. In Proceedings of the 32nd International Conference on International Conference on Machine Learning, Lille, France, 6–11 July 2015; Volume 37, pp. 97–105.
10. Shen, C.; Song, R.; Li, J.; Zhang, X.; Tang, J.; Shi, Y.; Liu, J.; Cao, H. Temperature drift modeling of mems gyroscope based on genetic-elman neural network. *Mech. Syst. Signal Proc.* **2016**, *72–73*, 897–905.
11. Tzeng, E.; Hoffman, J.; Zhang, N.; Saenko, K.; Darrell, T. Deep domain confusion: Maximizing for domain invariance. *arXiv* **2014**, arXiv:1412.3474.
12. Ghifary, M.; Kleijn, W.B.; Zhang, M.; Balduzzi, D.; Li, W. Deep reconstruction-classification networks for unsupervised domain adaptation. In *European Conference on Computer Vision*; Springer: Cham, Switzerland, 2016; pp. 597–613.
13. Venkateswara, H.; Eusebio, J.; Chakraborty, S.; Panchanathan, S. Deep Hashing Network for Unsupervised Domain Adaptation. In Proceedings of the 2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Honolulu, HI, USA, 21–26 July 2017; pp. 5385–5394.
14. Goodfellow, I.; Pouget-Abadie, J.; Mirza, M.; Xu, B.; Warde-Farley, D.; Ozair, S.; Courville, A.; Bengio, Y. Generative adversarial nets. In Proceedings of the Advances in Neural Information Processing Systems, Montreal, QC, Canada, 8–13 December 2014; pp. 2672–2680.
15. Lee, D.H. Pseudo-label: The simple and efficient semi-supervised learning method for deep neural networks. In Proceedings of the ICML 2013 Workshop: Challenges in Representation Learning (WREPL), Atlanta, GA, USA, 21 June 2013; Volume 3, p. 2.
16. Zheng, Z.; Zheng, L.; Yang, Y. Unlabeled Samples Generated by GAN Improve the Person Re-Identification Baseline in Vitro. In Proceedings of the IEEE International Conference on Computer Vision, Venice, Italy, 22–29 October 2017; pp. 3754–3762.
17. Wang, X.; Wang, X.; Ni, Y. Unsupervised Domain Adaptation for Facial Expression Recognition Using Generative Adversarial Networks. *Comput. Intel. Neurosci.* **2018**, *2018*, 7208794. [CrossRef] [PubMed]
18. Liu, M.Y.; Tuzel, O. Coupled generative adversarial networks. In Proceedings of the Advances in Neural Information Processing Systems (NIPS 2016), Barcelona, Spain, 5–10 December 2016; pp. 469–477.
19. Liu, M.Y.; Breuel, T.; Kautz, J. Unsupervised image-to-image translation networks. In Proceedings of the Advances in Neural Information Processing Systems (NIPS 2017), Long Beach, CA, USA, 4–9 December 2017; pp. 700–708.
20. LeCun, Y.; Bottou, L.; Bengio, Y.; Haffner, P. Gradient-based learning applied to document recognition. *Proc. IEEE* **1998**, *86*, 2278–2324. [CrossRef]
21. Denker, J.S.; Gardner, W.; Graf, H.P.; Henderson, D.; Howard, R.; Hubbard, W.; Jackel, L.D.; Baird, H.S.; Guyon, I. Neural network recognizer for hand-written zip code digits. In Proceedings of the Advances in Neural Information Processing Systems (NIPS 1989), Denver, CO, USA, 27–30 November 1989; pp. 323–331.
22. Liu, J.; Li, J.; Lu, K. Coupled local–global adaptation for multi-source transfer learning. *Neurocomputing* **2018**, *275*, 247–254. [CrossRef]
23. Ganin, Y.; Lempitsky, V. Unsupervised Domain Adaptation by Backpropagation. In Proceedings of the International Conference on Machine Learning, Lille, France, 7–9 July 2015; pp. 1180–1189.
24. Zhang, W.; Ouyang, W.; Li, W.; Xu, D. Collaborative and Adversarial Network for Unsupervised domain adaptation. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Salt Lake City, Utah, 18–22 June 2018; pp. 3801–3809.
25. Tzeng, E.; Hoffman, J.; Darrell, T.; Saenko, K. Simultaneous deep transfer across domains and tasks. In Proceedings of the 2015 IEEE International Conference onComputer Vision (ICCV), Santiago, Chile, 7–13 December 2015; pp. 4068–4076.
26. Tzeng, E.; Hoffman, J.; Saenko, K.; Darrell, T. Adversarial Discriminative Domain Adaptation. In Proceedings of the 2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Honolulu, HI, USA, 21–26 July 2017; pp. 2962–2971.
27. Isola, P.; Zhu, J.Y.; Zhou, T.; Efros, A.A. Image-to-Image Translation with Conditional Adversarial Networks. In Proceedings of the 2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Honolulu, HI, USA, 21–26 July 2017; pp. 5967–5976.
28. Zhu, J.Y.; Park, T.; Isola, P.; Efros, A.A. Unpaired Image-to-Image Translation Using Cycle-Consistent Adversarial Networks. In Proceedings of the 2017 IEEE International Conference on Computer Vision (ICCV), Venice, Italy, 22–29 October 2017; pp. 2242–2251.

29. Mao, X.; Wang, S.; Zheng, L.; Huang, Q. Semantic invariant cross-domain image generation with generative adversarial networks. *Neurocomputing* **2018**, *293*, 55–63. [CrossRef]
30. Bousmalis, K.; Silberman, N.; Dohan, D.; Erhan, D.; Krishnan, D. Unsupervised Pixel-Level Domain Adaptation with Generative Adversarial Networks. In Proceedings of the 2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Honolulu, HI, USA, 21–26 July 2017; pp. 95–104.
31. Kiasari, M.A.; Moirangthem, D.S.; Lee, M. Coupled generative adversarial stacked Auto-encoder: CoGASA. *Neural Netw.* **2018**, *100*, 1–9. [CrossRef] [PubMed]
32. Kingma, D.P.; Welling, M. Auto-encoding variational bayes. *arXiv* **2013**, arXiv:1312.6114.
33. He, K.; Zhang, X.; Ren, S.; Sun, J. Deep residual learning for image recognition. In Proceedings of the 2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Las Vegas, NV, USA, 27–30 June 2016; pp. 770–778.
34. Larsen, A.B.L.; Sønderby, S.K.; Larochelle, H.; Winther, O. Autoencoding beyond pixels using a learned similarity metric. In Proceedings of the International Conference on Machine Learning, New York, NY, USA, 20–22 June 2016; pp. 1558–1566.
35. Lyons, M.J.; Budynek, J.; Akamatsu, S. Automatic classification of single facial images. *IEEE Trans. Pattern Anal. Mach. Intell.* **1999**, *21*, 1357–1362. [CrossRef]
36. Lyons, M.; Akamatsu, S.; Kamachi, M.; Gyoba, J. Coding facial expressions with gabor wavelets. In Proceedings of the 1998 IEEE International Conference on Automatic Face and Gesture Recognition, Nara, Japan, 14–16 April 1998; pp. 200–205.
37. Valstar, M.; Pantic, M. Induced disgust, happiness and surprise: An addition to the mmi facial expression database. In Proceedings of the 3rd International Workshop on EMOTION (Satellite of LREC): Corpora for Research on Emotion and Affect, Valletta, Malta, 23 May 2010; pp. 65–70.
38. Pantic, M.; Valstar, M.; Rademaker, R.; Maat, L. Web-based database for facial expression analysis. In Proceedings of the IEEE International Conference on Multimedia and Expo (ICME 2005), Amsterdam, The Netherlands, 6 July 2005; p. 5.
39. Lucey, P.; Cohn, J.F.; Kanade, T.; Saragih, J.; Ambadar, Z.; Matthews, I. The extended cohn-kanade dataset (ck+): A complete dataset for action unit and emotion-specified expression. In Proceedings of the 2010 IEEE Computer Society Conference on Computer Vision and Pattern Recognition Workshops (CVPRW), San Francisco, CA, USA, 13–18 June 2010; pp. 94–101.
40. Van Der Maaten, L. Accelerating t-SNE using tree-based algorithms. *J. Mach. Learn. Res.* **2014**, *15*, 3221–3245.
41. Saenko, K.; Kulis, B.; Fritz, M.; Darrell, T. Adapting visual category models to new domains. In *European Conference on Computer Vision*; Springer: Cham, Switzerland, 2010; pp. 213–226.

© 2018 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (http://creativecommons.org/licenses/by/4.0/).