Improving Feature's Capability of Carrying Category-specific Information for Adversarial Domain Adaptation

Yundong Li*
School of Information Science and Technology
North China University of Technology
Beijing, China
liyundong@ncut.edu.cn

Chen Lin
School of Information Science and Technology
North China University of Technology
Beijing, China
cimmerri@163.com

Wei Hu
School of Information Science and Technology
North China University of Technology
Beijing, China
huwei1603@126.com

Han Dong
School of Information Science and Technology
North China University of Technology
Beijing, China
dh1040760693dh@outlook.com

***Abstract*—Recent research has shown that generative adversarial networks (GANs) have been successfully applied in aligning features for domain adaptation. However, the extracted features might lose category-specific information, because they are distinguished as either a source or a target during the adversarial training. To address this issue, a two-stage training framework consisting of two sets of GANs and a dedicated classifier is proposed in this study. In the pretraining stage, we use an encoder–decoder–classifier structure to obtain discriminative and representative features of the source domain and use it as a reference in the subsequent training. In the adversarial training stage, two sets of GANs are used to align target-domain features with those of the source domain and transfer samples of the target domain to the source domain simultaneously. A dedicated classifier is trained along with the adversarial loss to force the generated features of the target domain to carry category-specific information, which significantly improves classification performance. The features of the source domain stay intact in the adversarial training stage. Thus, our approach can alleviate the training burden of GANs. The proposed method has been validated on digital datasets and office31 datasets. Experimental results demonstrate that an average accuracy of 96.5% and 95.1% is achieved, which leads to superior or comparable performance to state-of-the-art results.**

Keywords—deep learning, domain adaptation, generative adversarial network, transfer learning.

I. INTRODUCTION

Deep learning has recently been dominant for diverse machine learning problems and applications. However, the impressive performance of deep learning is greatly attributed to a large number of labeled samples [1]. For a novel target task, manual annotations of samples are often prohibitive, which hinders the rapid deployment of deep learning **Error! Reference source not found.**. An intuitive idea to alleviate the burden of data annotations is to directly use the off-the-shelf classifier from the source domain for the target task. However, this idea usually fails to generalize well given a shift between source and target domains [38]. As a branch of transfer learning[2], domain adaptation (DA) aims to reduce the distribution discrepancy between source and target domains. In this study, we focus on the issue of unsupervised DA, that is, leveraging sufficient labeled samples of source domain and unlabeled data in the target domain to improve classification performance over target test data.

DA algorithms based on feature representation are widely discussed [4] . Data of source and target domains are mapped into a latent feature space, in which the feature distribution distance is shortened by optimizing a loss function along with a mathematical constraint that forces the feature distributions close to one another [5]. In comparison with handcrafted features, the hierarchical features extracted by convolutional neural networks (CNNs) [6] are more discriminative and representative. Therefore, CNNs are widely used in various DA diagrams as feature extraction networks in combination with maximum mean discrepancy (MMD) [7] to achieve domain invariance [15].

Recent works have shown that adversarial losses have been successfully applied to unsupervised DA. Generative adversarial networks (GANs) [8] were originally defined to generate images conditioned on noise from a specific distribution [9]. Adversarial training can be conducted in two ways when GAN is applied to DA, that is, image style translation at the pixel level and alignment at the feature level [12]. In the pixel-level methods, GAN is used to generate fake images approximating that of the target domain conditioned on images of the source domain, and the generated fake images are then used along with label information in the source domain to train a classifier for the target task [11,22,23]. However, pixel-level approaches can only achieve good results when the image foregrounds between the source and target domains are insignificantly different [37]. An alternative solution is to extract features of the target domain by using a generator and make it indistinguishable to those of the source domain by training a domain classifier. A problem in this method is that the domain classifier can only

Dr. Yundong Li's research work was supported by Beijing Natural Science Foundation (4182020) and Key Laboratory of Large Structure Health Monitoring and Control, Shijiazhuang. (Corresponding Author)

978-1-7281-6926-2/20/$31.00 ©2020 IEEE

distinguish features as either a source or a target, and category-specific information of both domains is not considered [29]. In addition, DA algorithms based on GAN are difficult to train [36]. A step-by-step training strategy is utilized to alleviate the complexity of training [12].

To address the aforementioned issues, we propose a two-stage GAN training framework for DA. In the first stage, the pipeline is composed of an encoder–decoder–classifier structure. The labeled samples of the source domain are used to train a classifier and retrieve reconstructions of the input data. The encoder–decoder–classifier structure ensures that the extracted high-level features are discriminative and representative for the source domain. Discrimination is critical for the features to obtain accurate predictions and is ensured by the supervised learning of the classifier. Representation means that the inputs can be recovered from the extracted features and is ensured by the reconstruction loss of autoencoders. In the second stage, we use one set of GAN to extract features of the target domain and align it with those of the source domain and another set of GAN to reconstruct data of the target domain from the extracted features and align them with the data of the source domain simultaneously. A classifier is also placed into the second discriminator to make the extracted features sensitive to category information. The main contributions of this study are as follows. (1) We propose a two-stage training framework in which the features of the source domain from an off-the-shelf classification task can be reused, and the features are kept intact in the sequential training process. Thus, the training time and complexity are mitigated. (2) We apply two sets of GANs for DA at feature and pixel levels simultaneously, which can properly transfer the features of the target domain to the feature space of the source domain. Samples of the target domain can also be transferred into the image style of the source domain while remaining the foreground information. (3) Motivated by the hypothesis that source and target domains share the same category information [39], we provide a solution to make the extracted features of the target domain contain category-specific information by using an additional classifier in the second discriminator in combination with reconstructions of the target domain. The lack of labeled samples in the target domain is a considerable challenge for unsupervised DA. We use samples of the source domain and reconstructions from source-domain features to train a classifier along with the adversarial loss to make the generated features of the target domain remain category-specific information.

The remainder of this paper is organized as follows. Section 2 discusses recent research on DA. Section 3 explains the model structure and training strategy of the proposed method. Section 4 discusses the experimental results and the effects of parameters on the results. Section 5 concludes the study.

II. Related Work

DA provides a transfer learning solution to facilitate classification issues in the target-domain leveraging knowledge learned from the source domain. DA, in combination with deep learning, has recently become a focus of research [5]. Research has indicated that hierarchical features of the lower layers are general, whereas the higher features are task-specific [40]. Several unsupervised DA methods have accordingly been investigated. Tzend et al. presented a deep domain confusion (DDC) method, in which an adaptation layer was added before the last layer of Alexnet [41]. An MMD loss was minimized over the adaptation layers of source and target domains during training. Long et al. proposed deep adaptation networks (DANs) to extend the marginal distribution adaptation of DDC from one layer to multiple task-specific layers [15]. On the basis of DAN, joint adaptation networks align the joint distributions of multiple domain-specific layers by optimizing the joint maximum mean discrepancy criteria [21]. Different from DDC, deep correlation alignment is proposed to minimize the domain shift by aligning the second-order statistics of source and target distributions instead of the MMD criteria [24]. Ghifary et al. presented deep reconstruction classification networks (DRCNs) for unsupervised DA, and these networks consist of two pipelines: one classifier for labeled data in the source domain and the other for unlabeled data reconstruction in the target domain [14]. In their method, the encoder parts of both pipelines share the same weights to ensure domain invariance.

GAN has been widely used for DA. Existing GAN-based approaches are broadly divided into pixel translation methods and feature alignment methods [12]. In the pixel translation method, samples can be transferred to other domains and then be used to train a classifier. Bousmalis et al. proposed a method called PixelDA to learn the transformation from one domain to the other domains in pixel space in an unsupervised manner [11]. They used a model based on GAN to make the source domain images seem to be drawn from the target domain. Judy et al. proposed a DA method of CyCADA [24] based on a previous work of CycleGAN [10]. Its innovation is to ensure semantic consistency and provide additional interpretability in the two domains in the feature space. Lv et al. proposed a TarGAN model [25]. This model collaborates with high mutual information constraints and weight-sharing mechanisms and then uses GAN to generate target-domain samples corresponding to the source-domain labels. Researchers have also been looking for ways to align features instead of image translations using GAN [26]. Tzeng et al. proposed the adversarial discriminative DA (ADDA) method [27]. They used GAN to extract features of the target domain and align them with those of the source domain. Hu et al. proposed a DA model based on GAN with two discriminators called DupGAN [28]. DupGAN can obtain features with domain invariance and maintain classification information to achieve conversion between two domains. It uses a classifier in the intermediate feature to predict labels of the target domain. Hence, the features are discriminative and can be further used in the discriminator. Long et al. proposed conditional domain adversarial networks (CDANs), which can align the joint distribution of features and classification [19]. They used multilinear conditioning methods to capture multimodal structures of the data and increase the discernibility of source and target domains. Entropy conditioning was added to the objective function to make the classifier have good mobility. Kuniaki et al. proposed a method of using task-specific decision boundaries called MCD to align the distribution of source and target domains [29]. The main idea of this method is to maximize the difference between outputs of two classifiers to detect target-domain samples that are farther away from the source domain. The feature generator minimizes this difference to generate

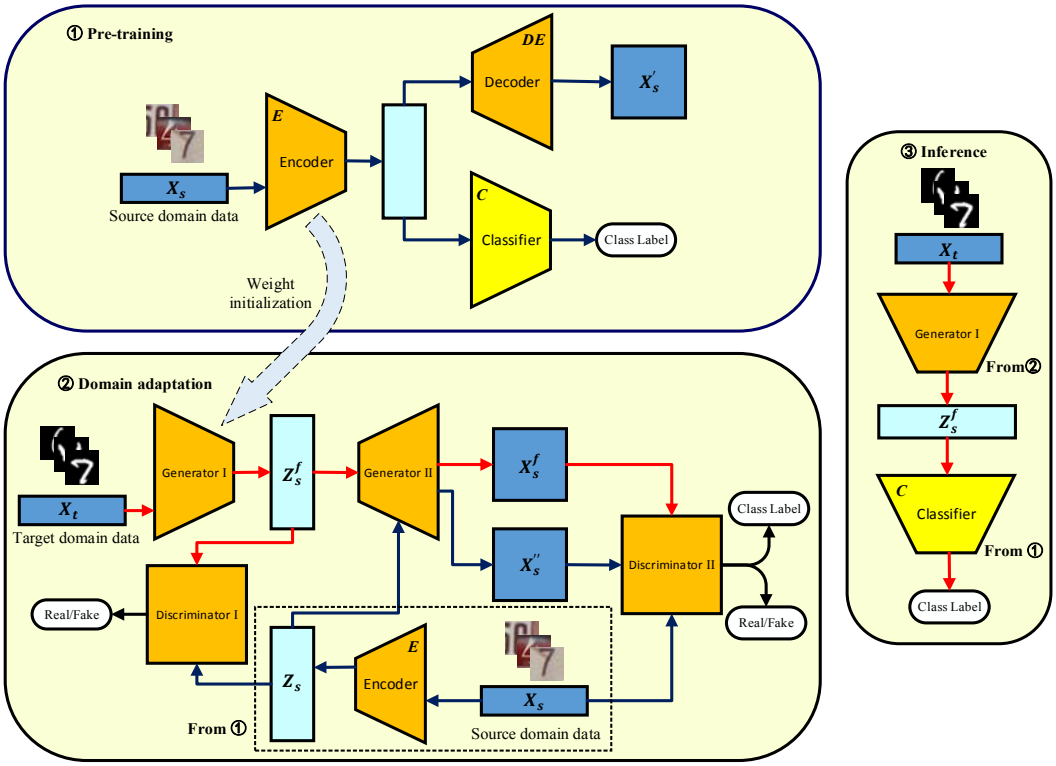


Fig.1. Diagram of the proposed DA approach.

target-domain features that are close to those of the source domain.

Different from the previous methods such as AFA [12] and ADDA [27], we have proposed effective methods at each stage of the algorithm. We add an autoencoder into the pretraining model to ensure that features not only contain classification information but also representative data information. Two sets of GANs are used in the DA stage. The target-domain features are aligned with the source-domain features using the first set of GAN. The second set of GAN is used to reconstruct inputs of the target domain and align them with those of the source domain. A dedicated classifier is added to the second discriminator to force the generated features of the target domain to carry category-specific information.

III. Methodology

We define the source-domain data and its labels as $X_s = \{x_s^i\}_{i=1}^{N_s}$ and $Y_s = \{y_s^i\}_{i=1}^{N_s}$ and the target-domain data as $X_t = \{x_t^i\}_{i=1}^{N_t}$. The proposed framework consists of three steps, namely, pretraining, DA, and inference, as shown in Fig.1. In this section, we elucidate the method from three aspects.

A. Pretraining

The purpose of pretraining is to obtain features of the source-domain data. The pretraining scheme uses an encoder–decoder–classifier structure. After the source-domain data X_s are entered into the encoder E to obtain the features Z_s, the decoder DE and classifier C are used to generate reconstructed data X_s' and predicted labels, respectively.

The classifier model for training source-domain data uses a cross-entropy function, which is defined as follows:

$$L_{cls,src} = \min_{E,C} \mathbb{E}_{(x_s,y_s)\sim(X_s,Y_s)} C(E(x_s), y_s) \quad (1)$$

A mean square error loss is utilized in the training of the autoencoder and is defined as follows:

$$L_{rec} = \min_{E,DE} \sum\nolimits_{(x_s,x_s')\sim(X_s,X_s')} \left\| x_s - DE\left(E(x_s)\right) \right\|^2 \quad (2)$$

The final loss of the pretraining model is the sum of Equations (1) and (2), as shown as follows:

$$L_{src} = L_{cls,src} + L_{rec} \quad (3)$$

The pretraining model can extract discriminative and representative features of the source domain by optimizing the loss shown in Equation (3). The features contain category-specific and representative information of the original images in the source domain.

B. DA

1) Training of GAN I

The proposed DA algorithm is mainly composed of two GANs. The original objective function of GAN is

$$\min_G \max_D V(D,G) = \mathbb{E}_{x\sim P_{data}(x)}\left[log\left(D(x)\right)\right] + \mathbb{E}_{z\sim P_z(z)}\left[log\left(1 - D\left(G(z)\right)\right)\right] \quad (4)$$

where x is the real sample, and z is the random vector. The individual losses of the generator and discriminator are respectively described as follows.

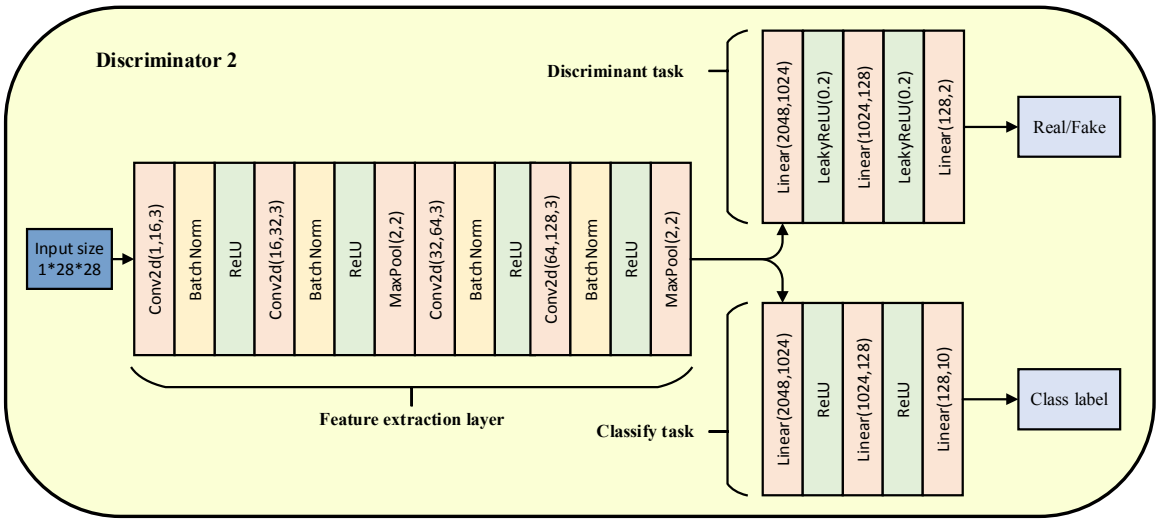


Fig.2. Model structure of discriminator II. The size of the input images is $1 \times 28 \times 28$. The parameters of input and output channels and the convolution kernel size of the convolution layers are labeled in brackets, and stride is set to 1. The parameters of kernel size and stride of the max pool layers are also annotated in the brackets.

$$\min_G V(D,G) = \mathbb{E}_{z\sim p_z(z)}\left[\log\left(1-D\left(G(z)\right)\right)\right] \quad (5)$$

$$\max_D V(D,G) = \mathbb{E}_{x\sim p_{data}(x)}\left[log\left(D(x)\right)\right] + \mathbb{E}_{z\sim p_z(z)}\left[\log\left(1-D\left(G(z)\right)\right)\right] \quad (6)$$

GAN is conducted in a minimum–maximum approach, in which the discriminator attempts to distinguish inputs as true or fake, whereas the generator tries to fool the discriminator by generating samples similar to the true one as possible. We attempt to align interdomain features by using the generator to produce fake features similar to those of the source domain.

As shown by Step 2 in Fig.1, the generator is initialized using the encoder parameters in the pretraining stage. Samples of the target domain are fed into generator I to obtain fake features $Z_s^f = G_1(X_t)$. The discriminator then determines the authenticity of the features. The encoder E is not involved in the adversarial training stage, and the parameters of encoder E remain unchanged during the training of GANs. The loss functions of generator I and discriminator I are defined as follows:

$$L_{G_1,tgt} = \min_{G_1} \mathbb{E}_{x_t\sim X_t}\, log\left(1-D_1\left(G_1(x_t)\right)\right) \quad (7)$$

$$L_{D_1,src} = \max_{D_1} \mathbb{E}_{x_s\sim X_s}\, log\left(D_1\left(E(x_s)\right)\right) + \mathbb{E}_{x_t\sim X_t}\, log\left(1-D_1\left(G_1(x_t)\right)\right) \quad (8)$$

2) Training of GAN II

The second set of GAN is further used to reconstruct data of the target domain from the features extracted by generator I. The reconstructions can be represented as $X_s^f = G_2\left(Z_s^f\right)$. We align the reconstructions with the data of the source domain instead of those of the target domain. Thus, an image style similar to that of the source domain is expected for the generated images, whereas the foreground is retained because these images are generated from the extracted high-level features.

As mentioned in the Introduction section, target-domain features extracted by GAN cannot include sufficient category-specific information because the discriminator only distinguishes them as either a source or a target. The lack of category information prevents target-domain samples from being accurately classified. To address this issue, we add a classifier in discriminator II, as shown in Fig.2. This classifier is trained not only by using source-domain data X_s but also by reconstructions X_s'' generated by generator II using source-domain features Z_s as inputs.

The additional classifier is trained in a min–max manner along with the adversarial loss of GAN 2 and involved into the training of generator II and discriminator II. The loss function of the classifier related to generator II training is defined as Equation (9), and Equation (10) shows the classification loss in the training of discriminator II.

$$L_{cls,G_2} = \min_{G_2} \mathbb{E}_{(x_s,y_s)\sim(X_s,Y_s)} D_{2_{cls}}\left(G_2\left(E(x_s)\right),y_s\right) \quad (9)$$

$$L_{cls,D_2} = \max_{D_2} \mathbb{E}_{(x_s,y_s)\sim(X_s,Y_s)} D_{2_{cls}}(x_s,y_s) + \mathbb{E}_{(x_s,y_s)\sim(X_s,Y_s)} D_{2_{cls}}\left(G_2\left(E(x_s)\right),y_s\right) \quad (10)$$

The classification loss also affects the training of generator I, and the loss function related to generator I is defined as

$$L_{cls,G_1} = \min_{G_1} \mathbb{E}_{(x_s,y_s)\sim(X_s,Y_s)} D_{2_{cls}}\left(G_2\left(E(x_s)\right),y_s\right) \quad (11)$$

This additional classifier improves the accuracy of target classification task due to the following facts: The reconstruction X_s^f of the target domain and reconstruction X_s'' of the source domain are generated using the same network structure; source-domain data X_s and reconstruction X_s'' are used to train the dedicated classifier; reconstruction X_s^f is aligned to the source-domain data X_s; this structure and training strategy make generator II sensitive to category information; and force feature Z_s^f is extracted from target domain to contain additional category-specific information.

In addition to classification loss, the loss of GAN 2 also includes two other parts, namely, adversarial loss between the reconstruction of target domain and the source domain, and adversarial loss between the reconstruction of source domain and the source domain, which are shown in Equations (12)−(15).

Adversarial loss function between X_s^f and X_s:

$$L_{G_2,tgt} = \min_{G_2} \mathbb{E}_{x_t \sim X_t} log\left(D_2\left(G_2\left(G_1(x_t)\right)\right)\right) \quad (12)$$

$$L_{D_2,tgt} = \max_{D_2} \mathbb{E}_{x_t \sim X_t} log\left(1 - D_2\left(G_2\left(G_1(x_t)\right)\right)\right) + \mathbb{E}_{x_s \sim X_s} log\left(D_2(x_s)\right) \quad (13)$$

Adversarial loss function between X_s'' and X_s :

$$L_{G_2,src'} = \min_{G_2} \mathbb{E}_{x_s \sim X_s} log\left(D_2\left(G_2\left(E(x_s)\right)\right)\right) \quad (14)$$

$$L_{D_2,src'} = \max_{D_2} \mathbb{E}_{x_s \sim X_s} log\left(1 - D_2\left(G_2\left(E(x_s)\right)\right)\right) + \mathbb{E}_{x_s \sim X_s} log\left(D_2(x_s)\right) \quad (15)$$

We adopt an alternative training strategy to optimize the proposed DA framework. The detailed training procedure is shown in Algorithm 1. α and β are tradeoff coefficients, which are helpful for network convergence.

Algorithm 1 Iterative training procedure of DA

Training iterations = N

Input:

Source-domain data: $X_s = \{x_s^i\}_{i=1}^{N_s}$, Source-domain label: $Y_s = \{y_s^i\}_{i=1}^{N_s}$.

Target-domain data: $X_t = \{x_t^i\}_{i=1}^{N_t}$.

For i in 1:N **do**:

Update discriminator 2:

$$L_{D_2} = L_{cls,D_2} + L_{D_2,src'} + L_{D_2,tgt} \quad (16)$$

Update generator 2, $\alpha = 0.5$:

$$L_{G_2} = L_{cls,G_2} + L_{G_2,src'} + \alpha L_{G_2,tgt} \quad (17)$$

Update discriminator 1:

$$L_{D_1} = L_{D_1,src} \quad (18)$$

Update generator I, $\beta = 0.1$:

$$L_{G_1} = L_{G_1,tgt} + \beta L_{cls,G_1} \quad (19)$$

end for

C. Inference

The inference procedure is straightforward. As shown in Step 3 in Fig.1, the target-domain data are entered into generator I that is trained in Step 2 to obtain domain-aligned features, and then enter the source-domain classifier trained in Step 1 to predict the labels.

IV. Experimental Results and Discussion

Two groups of experiment are conducted to demonstrate the effectiveness of the proposed approach. In the first group, we validate the algorithm using popular digital datasets and compare it with state-of-the-art approaches. The second group of experiment discusses the effects of model structure and parameter selection on classification performance.

A. Digital Dataset Experiment

This experiment uses three types of digital dataset consisting of 10 classes of digits, namely, SVHN [30] (73212 training images, 26032 testing images, size: 3 × 32 × 32), MNIST **Error! Reference source not found.** (60000 training images, 10000 testing images, size: 1 × 28 × 28), and USPS [31] (7291 training images, 2007 testing images, size: 1 × 16 × 16), as shown in Fig.3. Subjective observations indicate that the USPS and MNIST datasets are similar, whereas differences between SVHN and the other two datasets are significant. Related studies have shown that small similarity among data leads to high difficulty in transferring. Therefore, transfer from SVHN to MNIST is challenging.

Images of MNIST dataset are in grayscale, and USPS and

Fig.3. Examples of datasets used in the experiment.

SVHN are RGB images; thus, all of the images are converted into grayscale with a resolution of 28 × 28. Three transfer tasks are performed in the experiment, as shown as follows.

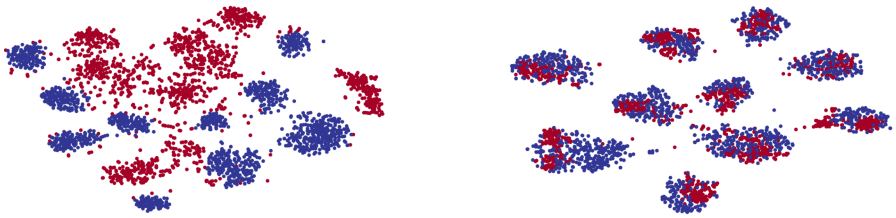
MNIST → USPS: MNIST is used as labeled source domain and USPS servers as unlabeled target domain. A total of 2000 images from MNIST and 1800 images from USPS are randomly sampled to maintain consistency with relevant research [42].

USPS → MNIST: USPS is used as the labeled source domain and MNIST as unlabeled target domain. A total of 2000 images from MNIST and 1800 images from USPS are randomly sampled to maintain consistency with relevant research.

SVHN → MNIST: SVHN is used as labeled source domain and MNIST as unlabeled target domain. All of the training samples are used.

All test codes are implemented under Pytorch [33] framework. In the pretraining stage, Adam [34] optimizer is used as the optimization tool. The learning rate is set to 0.0002, β1 is 0.5, β2 is 0.999, and the batch size is set to 128 for each domain. In the DA stage, a stochastic gradient descent (SGD) optimizer is used as the optimization function. The momentum parameter is set to 0.9, the weight decay is set to 0.000025, the learning rate is 0.0002, and the batch size is 128 images for each domain.

An experimental comparison with the state-of-the-art methods in terms of accuracy is shown in Table I. The proposed approach outperforms other methods on the task of MNIST → USPS. Our method also achieves a comparable result with other approaches on the task of USPS → MNIST. Although the transfer performance of the task of SVHN → MNIST is not satisfied compared with MCD, DeepJDOT, and TarGAN, our approach still achieves highest scores over the state-of-the-art methods in terms of average accuracy on the three digital datasets.



(a) (b)

Fig.4. Visualization of DA process of the transfer task SVHN → MNIST. (a) Distributions of features extracted by the pretrained model. The blue dots indicate features of the source domain, which are well classified into 10 classes. The red dots illustrate features of the target domain, which are classified poorly by the pretrained model. (b) Feature distributions of target domain extracted by the well-trained generator I.

Feature distributions of source and target domains are visualized using the t-SNE [35] algorithm to demonstrate the feature alignment effect of the proposed approach, as shown in Fig.4. The features of the target domain are well classified into 10 categories after DA, and each category is well aligned with that of the source domain. The experimental result in Fig.4 shows that the category-specific information contained in the extracted features is significantly enhanced by the proposed approach.

TABLE I. ACCURACY (%) COMPARISON WITH STATE-OF-THE-ART METHODS ON DIGITAL DATASETS

Method	MNIST → USPS	USPS → MNIST	SVHN → MNIST	Avg
Source only [a]	76.1	57.1	61.4	64.9
DRCN [14]	91.8	73.7	82.0	82.5
DSN [18]	91.3	73.2	82.7	82.4
ADDA [27]	92.9	93.8	76.0	87.5
ATDA [16]	93.2	84.1	86.2	87.8
RAAN [20]	89.0	92.1	89.2	90.1
AFA [12]	96.2	89.7	89.7	91.9
GTA [13]	92.8	90.8	92.4	92.0
CyCADA [24]	95.6	96.5	90.4	94.2
CDAN [19]	95.6	98.0	89.2	94.3
MCD [29]	94.2	94.1	96.2	94.8
DeepJDOT [17]	95.0	96.4	96.7	96.0
DupGAN [28]	96.0	**98.8**	92.5	95.7
TarGAN [25]	93.8	94.1	**98.1**	95.3
Ours	**98.4**	98.0	93.2	**96.5**

[a] The "source only" is the baseline that uses a classifier trained by using only the source domain without DA to classify target data.

Fig.5. Comparison between reconstructed image and source-domain data.

Our method uses a two-GAN structure that can align features and transfer image style simultaneously. Another experiment is conducted to illustrate the image transfer effect, as shown in Fig. 5. In this experiment, samples of the target domain are fed into the well-trained generator I to obtain features, and the features are then decoded by the decoder in the pretrained model to reconstruct the inputs. The bottom row of Fig.5 presents the samples of the target domain. The reconstructions of the samples are shown in the middle row of Fig.5. The backgrounds are transferred to the style of the source domain shown in the top row of Fig.5, whereas the foregrounds remain unchanged. Thus, the category-specific information is well reserved.

B. *Office-31 Dataset Experiment*

The Office-31 dataset consists of three domains with 31 classes: AMAZON (2817 images), WEBCAM (795 images), and DSLR (498 images), as shown in Fig 6. The WEBCAM dataset and the DSLR dataset are consist of images of real scene. There are small differences between the two data sets, but the two datasets are quite different from the AMAZON dataset.

Fig.6. Examples of Office-31 datasets used in the experiment.

Due to the small amount of data in the dataset, it is difficult to train a well-converged model. So we use ResNet-50 [43] trained on ImageNet as the Encoder for this experiment.

In the pretrain stage, the size of the input data is $3 \times 224 \times 224$, batch size is 16. The optimization function of the encoder and classifier is SGD, momentum parameter is set to 0.9, and the learning rate is 0.001. The optimization function of the decoder is Adam, $\beta 1$ is 0.5, $\beta 2$ is 0.999, weight decay is set to 0.000025 and the learning rate is 0.0001. In the DA stage, Adam optimizer is used as the optimization function. The momentum parameter is set to 0.9, the weight decay is set to 0.000025, the learning rate is 0.00001, and the batch size is 8 images for each domain.

Three sets of experiments, AMAZON→WEBCAM(A→W), DSLR→WEBCAM(D→W), and WEBCAM→DSLR(W→D), were used to verify the performance of the domain adaptation algorithm under the office31 dataset, and compared with the mainstream domain adaptation algorithm, as shown in Table II.

TABLE II. ACCURACY (%) COMPARISON WITH STATE-OF-THE-ART METHODS ON OFFICE-31

Method	A → W	D →W	W → D	Avg
ResNet-50 [b]	68.4	96.5	99.6	88.2
DAN [15]	80.5	97.1	99.6	92.4
DANN [44]	82.5	96.9	99.1	92.8
RTN [45]	84.5	96.8	99.4	93.6
JAN [46]	85.4	97.4	99.8	94.2
MADA [47]	**90.0**	97.4	99.6	95.7
GTA [13]	89.5	97.9	99.8	**95.7**
Ours	87.3	**98.0**	**100.0**	95.1

[b] The "ResNet-50" is the baseline that uses a classifier trained by using only the source domain without DA to classify target data.

It can be noted from Table II that the method proposed in this paper has achieved good results in experiments D→W and W→B, but the accuracy of experiment A→W is lower than that of MADA and GTA. Fig. 7. Shows the transfer effect of the proposed method on AMAZON. The reconstructed image is similar to the source-domain image and corresponds to the classification information of the target domain. These results can prove that the algorithm also performs well in real scenarios.

Fig.7. Comparison between reconstructed image and source-domain data.

C. Discussion of Model Structure and Parameters

1) Effect of reconstruction loss in the pretraining model

TABLE III. ACCURACY (%) OF THE COMPARISON BETWEEN THE ORIGINAL STUDY AND THE ADDITION OF THE AUTOENCODER

Method	MNIST → USPS	USPS → MNIST	SVHN → MNIST	Avg
ADDA	92.9	93.8	76.0	87.5
ADDA_Rec	**95.8**	**96.1**	**82.0**	**91.3**

Our pretraining model is based on ADDA [27]. However, only a classification network exists in ADDA. We add an autoencoder structure to the pretraining model and optimize it by using a reconstruction loss along with the classification loss. A set of experiments is conducted to illustrate the effectiveness of reconstruction loss, as shown in Table III. In this table, ADDA indicates the baseline that only has a classifier to predict labels. ADDA_Rec indicates the method with a classifier and an autoencoder. A reconstruction loss can significantly improve classification accuracy. Consequently, features extracted by such networks contain additional representative information that can facilitate DA in the subsequent process.

2) Effect of classification in discriminator II

TABLE IV. ACCURACY (%) OF ADDING CLASSIFICATION DISCRIMINATION TO DISCRIMINATOR II

L_{cls}	MNIST → USPS	USPS → MNIST	SVHN → MNIST	Avg
×	98.2	95.2	86.7	93.4
√	**98.4**	**98.0**	**93.2**	**96.5**

Features of the target domain extracted by the generator might lose category-specific information during the alignment process by the discriminator. To alleviate this issue, we add a dedicated classifier to the discriminator networks. No labels exist in the target domain; hence, samples of the source domain are used to train the classifier. The reconstructions from the source-domain features are also involved in the training of the classifier to make the generator networks sensitive to category-specific information. A set of experiments is conducted to demonstrate the effectiveness of the additional classifier. Table IV shows the accuracy comparison, where the top row indicates the results without the classifier, and the bottom row denotes the results with the classifier. The additional classifier can significantly improve classification performance.

V. CONLUSION

In this study, we propose a two-stage training DA framework that can align domain feature and transfer image style simultaneously. The proposed approach consists of two sets of GANs, one for feature alignment and the other for reconstruction of the target domain. In the pretraining stage, discriminative and representative features of the source domain are extracted by an encoder–decoder–classifier structure. In the adversarial training stage, features of the target domain are extracted and aligned with features of the source domain. The target domain is reconstructed from the extracted features and aligned with the source domain. A dedicated classifier is also integrated into the second discriminator to retain the category-specific information of the generated features of the target domain. The experimental results show that the proposed approach achieves an average accuracy of 96.5% and 95.1% over the three digital datasets and the Office-31 datasets, which is superior or comparable to the state-of-the-art results. In future works, we will investigate the applications of the proposed approach in image generation and DA for highly complex images.

REFERENCES

[1] R. Sharma, and A. S. Bist, "MACHINE LEARNING: A SURVEY," International Journal of Engineering Sciences & Research Technology, 2015.

[2] Pan, S. Jialin, and Yang, Qiang, "A Survey on Transfer Learning," IEEE Transactions on Knowledge & Data Engineering, 22.10(2010):1345-1359.

[3] Wang, Mei, and Weihong Deng, "Deep visual domain adaptation: A survey," Neurocomputing pp.135-153, 2018.

[4] Ganin, Yaroslav, and V. Lempitsky, "Unsupervised domain adaptation by backpropagation," arXiv preprint arXiv:1409.7495, 2014.

[5] Venkateswara, Hemanth, S. Chakraborty, and S. Panchanathan, "Deep-Learning Systems for Domain Adaptation in Computer Vision: Learning Transferable Feature Representations," IEEE Signal Processing Magazine,:pp.117-129, 2017.

[6] A. Krizhevsky, I Sutskever, and G. Hinton, "ImageNet Classification with Deep Convolutional Neural Networks," in Advances in neural information processing systems, pp.1097-1105, 2012.

[7] Gretton, K. Borgwardt, M. Rasch, B. Schoelkopf, and A. Smola, "A Kernel Two-Sample Test," Journal of Machine Learning Research, pp.723-773, 2012.

[8] I. J. Goodfellow, et al, "Generative Adversarial Networks," in Advances in Neural Information Processing Systems, pp.2672-2680, 2014.

[9] Brock, Andrew, J. Donahue, and K. Simonyan, "Large scale gan training for high fidelity natural image synthesis," arXiv preprint arXiv:1809.11096 ,2018.

[10] Zhu, Jun-Yan, T. Park, P. Isola, and A. Efros, "Unpaired image-to-image translation using cycle-consistent adversarial networks," In Proceedings of the IEEE international conference on computer vision, pp. 2223-2232, 2017.

[11] K. Bousmalis, N. Silberman, D. Dohan, D. Erhan, and D. Krishnan, "Unsupervised Pixel-Level Domain Adaptation with Generative Adversarial Networks," in IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Honolulu, HI, 2017, pp. 95-104.

[12] Volpi, Riccardo, P. Morerio, S. Savarese, and V. Murino, "Adversarial feature augmentation for unsupervised domain adaptation," In IEEE Conference on Computer Vision and Pattern Recognition (CVPR), pp. 5495-5504, 2018.

[13] S. Sankaranarayanan, Y. Balaji, C. D. Castillo, and R. Chellappa, "Generate to adapt: Aligning domains using generative adversarial networks," In IEEE Conference on Computer Vision and Pattern Recognition (CVPR), pp. 8503-8512, 2018.

[14] M. Ghifary, W. B. Kleijn, Mengjie Zhang, D. Balduzzi, and Wen Li, "Deep reconstruction-classification networks for unsupervised domain adaptation," In European Conference on Computer Vision, pp. 597-613, Springer, Cham, 2016.

[15] Mingsheng Long, Yue Cao, Jianmin Wang, and M. I. Jordan, "Learning transferable features with deep adaptation networks," arXiv preprint arXiv:1502.02791,2015.

[16] Kuniaki Saito, Yoshitaka Ushiku and Tatsuya Harada, "Asymmetric tri-training for unsupervised domain adaptation," In Proceedings of the 34th International Conference on Machine Learning-Volume 70, pp. 2988-2997, JMLR. org, 2017.

[17] B. B. Damodaran, B. Kellenberger, R. Flamary, D. Tuia, and N. Courty, "Deepjdot: Deep joint distribution optimal transport for unsupervised domain adaptation," In Proceedings of the European Conference on Computer Vision (ECCV), pp. 447-463, 2018.

[18] K. Bousmalis, G. Trigeorgis, N. Silberman, D. Krishnan, and D. Erhan, "Domain separation networks," In Advances in Neural Information Processing Systems, pp. 343-351, 2016.

[19] Mingsheng Long, Zhangjie Cao, Jianmin Wang, and M. I. Jordan, "Conditional adversarial domain adaptation," In Advances in Neural Information Processing Systems, pp. 1640-1650, 2018.

[20] Qingchao Chen, Yang Liu, Zhaowen Wang, I. Wassell, and K. Chetty, "Re-weighted adversarial adaptation network for unsupervised domain adaptation," In IEEE Conference on Computer Vision and Pattern Recognition (CVPR), pp. 7976-7985, 2018.

[21] Mingsheng Long, Han Zhu, Jianmin Wang, and M. I. Jordan, "Deep Transfer Learning with Joint Adaptation Networks," in Proceedings of the 34th International Conference on Machine Learning, Sydney, Australia, PMLR 70, 2017.

[22] Mingyu Liu, T. Breuel, and J. Kautz, "Unsupervised image-to-image translation networks," In Advances in Neural Information Processing Systems, pp. 700-708, 2017.

[23] Mingyu Liu, and O. Tuzel, "Coupled generative adversarial networks," In Advances in neural information processing systems, pp. 469-477, 2016.

[24] J. Hoffman, E. Tzeng, and et al, "Cycada: Cycle-consistent adversarial domain adaptation," arXiv preprint arXiv:1711.03213.

[25] Fengmao Lv, Jun Zhu, Guowu Yang, and Lixin Duan, "TarGAN: Generating target data with class labels for unsupervised domain adaptation," *Knowledge-Based Systems*, 172, pp: 123-129, 2019.

[26] K. Saito, S. Yamamoto, Y. Ushiku, and T. Harada, "Open set domain adaptation by backpropagation," In Proceedings of the European Conference on Computer Vision (ECCV), pp. 153-168, 2018.

[27] E. Tzeng, J. Hoffman, K. Saenko, and T. Darrell, "Adversarial discriminative domain adaptation," In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 7167-7176, 2017.

[28] Lanqing Hu, Meina Kan, Shiguang Shan, and Xilin Chen, "Duplex generative adversarial network for unsupervised domain adaptation," In IEEE Conference on Computer Vision and Pattern Recognition (CVPR), pp. 1498-1507, 2018.

[29] K. Saito, K. Watanabe, Y. Ushiku, and T. Harada, "Maximum classifier discrepancy for unsupervised domain adaptation," In IEEE Conference on Computer Vision and Pattern Recognition (CVPR), pp. 3723-3732, 2018.

[30] Y. Netzer, T. Wang, and A. Coates, "Reading digits in natural images with unsupervised feature learning," in NIPS, 2011.

[31] Y. Lecun, L. Bottou, Y. Bengio and P. Haffner, "Gradient-based learning applied to document recognition," in Proceedings of the IEEE, vol. 86, no. 11, pp. 2278-2324, Nov. 1998.

[32] J. S. Denker，W. R. Gardner, and et al, "Neural network recognizer for hand-written zip code digits," In Advances in neural information processing systems, pp. 323-331, 1989.

[33] Pytorch. [Online]. Available: https://pytorch.org/

[34] D. P. Kingma, and J. Ba, "Adam: A method for stochastic optimization," arXiv preprint arXiv:1412.6980 ,2014.

[35] L. Maaten, and G. Hinton, "Visualizing data using t-SNE." Journal of machine learning research 9, no. pp. 2579-2605, Nov 2008:.

[36] M. Arjovsky, S. Chintala, and L. Bottou "Wasserstein GAN," In International conference on machine learning, pp. 214-223, 2017.

[37] P. Isola, Junyan Zhu, Tinghui Zhou, and A. A. Efros, "Image-to-image translation with conditional adversarial networks," In IEEE conference on computer vision and pattern recognition (CVPR), pp. 1125-1134, 2014.

[38] G. Cai, Y. Wang, M. Zhou, and L. He, "Unsupervised Domain Adaptation with Adversarial Residual Transform Networks," arXiv preprint arXiv:1804.09578.

[39] J. Ren, J. Yang, N. Xu, and D. J. Foran, "Factorized Adversarial Networks for Unsupervised Domain Adaptation," arXiv preprint arXiv:1806.01376.

[40] J. Yosinski, J. Clune, Y. Bengio, and H. Lipson, "How transferable are features in deep neural networks?" in Proc. Advances in Neural Information Processing Systems Conf, pp. 3320–3328, 2014.

[41] E. Tzeng, J. Hoffman, N. Zhang, K. Saenko, and T. Darrell, "Deep domain confusion: Maximizing for domain invariance," arXiv Preprint, arXiv:1412.3474,2014.

[42] M. Long, J.Wang, G. Ding, J. Sun, and P. S. Yu, "Transfer feature learning with joint distribution adaptation," In 2013 IEEE International Conference on Computer Vision, pp. 2200–2207, Dec 2013.

[43] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun, "Deep residual learning for image recognition," In Proceedings of the IEEE conference on computer vision and pattern recognition, pp. 770-778.

[44] Y. Ganin, E. Ustinova, and et al, "Domain-adversarial training of neural networks," The Journal of Machine Learning Research, 17(1), 2096-2030.

[45] Guanyu Cai, Yuqin Wang, Mengchu Zhou, and Lianghua He, "Unsupervised domain adaptation with residual transfer networks," In Advances in Neural Information Processing Systems, 2018, pp. 136-144.

[46] Mingsheng Long, Han Zhu, Jianmin Wang, M. I. Jordan, "Deep transfer learning with joint adaptation networks," In Proceedings of the 34th International Conference on Machine Learning, 2016, Volume 70, pp. 2208-2217. JMLR. org.

[47] Z. Pei, Z. Cao, M. Long, and J. Wang, "Multi-adversarial domain adaptation," In Thirty-Second AAAI Conference on Artificial Intelligence, 2018.nd