

Received March 10, 2020, accepted March 28, 2020, date of publication April 1, 2020, date of current version April 16, 2020.

Digital Object Identifier 10.1109/ACCESS.2020.2984777

# Unsupervised Deep Domain Adaptation Based on Weighted Adversarial Network

XU JIA<sup>1</sup> AND FUMING SUN<sup>2</sup>

<sup>1</sup>School of Electronics and Information Engineering, Liaoning University of Technology, Jinzhou 121001, China

<sup>2</sup>School of Electronics and Communication Engineering, Dalian Minzu University, Dalian 116600, China

Corresponding author: Fuming Sun (sunfuming@dlnu.edu.cn)

This work was supported in part by the National Natural Science Foundation of China under Grant 61976042, Grant 61502216, and Grant 51679116, in part by the Natural Science Foundation of Liaoning Province under Grant 2019-ZD-0700 and Grant 2019-ZD-0702, and in part by the Overseas Training Project of Higher Education Institutions of Liaoning Province under Grant 2019GJWYB015.

**ABSTRACT** Recent studies indicate that adversarial learning can reduce distribution discrepancy between domains effectively, but when the samples belonged to different classes have similar characteristics in the domains, they may be incorrectly aligned to similar classes after domain adaption, which gives rise to negative transfer. To prevent such misalignment, we propose a weighted adversarial network based unsupervised domain adaptation method. Its contributions are mainly reflected in the following two aspects: 1) according to the similarity of features between classes, the different weights are given to the corresponding domain discriminators, which means that we will focus on the alignment of the classes with similar characteristics; 2) all domain discriminators will be given a certain weight again based on the entropy of true or pseudo label vectors, that is, the clearer the sample classification result, the greater its credibility during domain discriminators learning. Experimental results on several cross-domain benchmark data sets show that our newly proposed approach outperforms state of the art methods.

**INDEX TERMS** Adversarial network, domain adaptation, image classification, transfer learning, unsupervised learning.

## I. INTRODUCTION

In many computer visual tasks, a large amount of labeled training samples are needed for traditional machine learning methods, however, annotating datasets for every task are extremely expensive and time-consuming, so when the number of annotated samples is limited, how to achieve accurate image classification and recognition has become a research hotspot. Transfer learning relaxes the restriction that the training and testing data are drawn from the same distribution, and adopts domain adaptation to reduce the data distribution difference between the samples in the source and target domains, where the source and target domains contain labeled and unlabeled samples respectively, thus the unlabeled samples can be recognized using the knowledge which is migrated from the well-labeled source domain [1].

Many previous deep domain adaptation methods achieve the alignment of the source and target domains well, but it is rarely considered whether all classes in the domain are

aligned exactly. In other words, if the samples belonged to different classes are similar enough, they are likely to be misaligned after adaptation. To address this challenge, we propose a novel weighted adversarial network based domain adaptation method, where the first contribution is that we weights the domain discriminators to varying degrees according to the feature similarity between the classes, and achieve better alignment of all classes in the source and target domains. For example, it is assumed that there are several classes of samples in the source domain such as “bicycle”, “motorcycle”, “dog”, etc, since the classes of “bicycle” and “motorcycle” are strongly related to each other, we need to pay more attention to the alignment of these two classes as shown in Fig.1, that is, give their domain discriminators greater weights. In addition, the second contribution of our proposed method is to assign the weights to all domain discriminators according to the true or pseudo labels. For instance, if the prediction result of one sample is clear during training as shown in Fig.2(a), it will play a more important role in the training of domain discriminators; In contrast, the more confused the prediction result, the less weight the

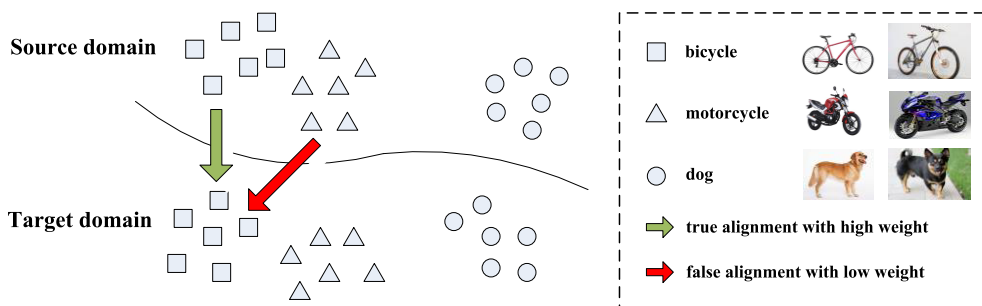


FIGURE 1. Schematic diagram of misaligned similar classes after domain adaptation.

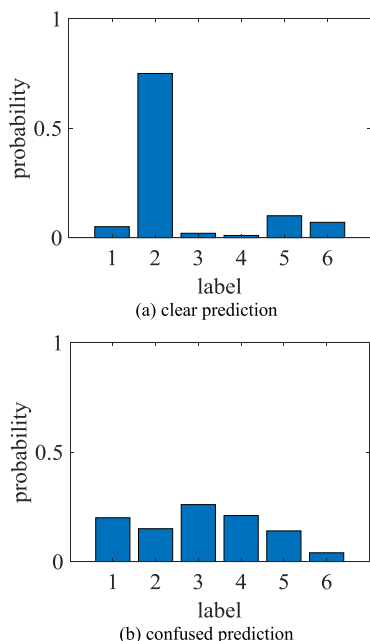


FIGURE 2. Probability distribution of prediction results.

sample will be given during training. Experimental results demonstrate that the proposed method can obtain high recognition accuracy on standard domain adaptation datasets.

The remainder of this paper is organized as follows. In section II, some related works are addressed. In section III, we give the details of the proposed domain adaptation method. We describe the experiments that were conducted on standard cross-domain recognition datasets to prove the effectiveness of the proposed method in section IV. Finally, conclusions are drawn in section V.

## II. RELATED WORK

At present, the existed deep domain adaptation methods can be roughly divided into the following three categories: the discrepancy-based, adversarial-based and reconstruction-based methods [2].

1) Discrepancy. The discrepancy-based methods is to reduce the shift between the source and target domains by fine-tuning the deep network. In supervised and semi-supervised domain adaptations, some feature similarity

metric learning based methods are proved to be effective [3]–[6], where the features are extracted from the different layers of deep network. In addition, there are two major techniques to deal with unsupervised domain adaptation, one is still based on feature similarity metric learning, unlike to supervised domain adaptations, the unlabeled samples are annotated by pseudo labels [7]–[9]; the other is aligning the statistical distribution of two domains by using some distribution discrepancies, where the commonly used distribution discrepancies include Maximum Mean Discrepancy (MMD) [10], [11], Joint Maximum Mean Discrepancy (JMMD) [12], Weighted Maximum Mean Discrepancy [8], Wasserstein distance [13], orthogonal discrepancy [14] and correlation alignment [15], [16].

2) Adversarial. In the adversarial-based methods, some domain discriminators are added to the deep network model, and we hope that at the same time as correctly classifying samples into the source and target domains, the learned features can confuse the domain discriminators as much as possible [17]. For this purpose, a gradient reversal layer is embedded to the deep network model [18]. Subsequently, some improved adversarial-based deep network models are continuously proposed, for example, multiple domains discriminators are used to classify different levels of features [19] or to align all categories of the source domain [20]–[21], and when the label space of target domain is a subset of the label space of source domain, a adversarial nets-based partial domain adaptation method is proposed [22].

3) Reconstruction. The existing reconstruction-based methods can be roughly divided into two categories, one is using encoder and decoder networks to implement feature extraction and image reconstruction respectively [23]–[25], and the other is based on different adversarial networks [26], [27].

In summary, from experimental results the features learned by the above methods have good transfer effect for most categories. However, for the classes with strong similarities, such as “cat” and “dog”, the recognition accuracy still needs to be further improved. Therefore, we assign greater weights to the domain discriminators of these classes than others, and reduce the impact of false pseudo labels during domain adaptation, so that all classes of the source and target domains are aligned exactly as much as possible.

### III. WEIGHTED ADVERSARIAL NETWORK BASED DOMAIN ADAPTATION

#### A. NETWORK MODEL

In domain adaptation, the source domain is given as  $D_s = \{(\mathbf{x}_i^s, \mathbf{y}_i^s)\}_{i=1}^{n_s}$ ,  $\mathbf{y}_i^s \in L_s$ ,  $d_i^s = 0$ , where  $\mathbf{x}_i^s$  and  $\mathbf{y}_i^s$  represent the sample and its label,  $n_s$ ,  $L_s$  and  $K_s$  are the number of samples, the set of labels and the number of labels in the source domain,  $d_i^s$  represents the domain label of the source domain; in addition, the target domain is defined as  $D_t = \{(\mathbf{x}_i^t)\}_{i=1}^{n_t}$ ,  $d_i^t = 1$ , where  $\mathbf{x}_i^t$  represents the sample,  $n_t$ ,  $L_t$  and  $K_t$  are the number of samples, the set of labels and the number of labels in the target domain,  $d_i^t$  represents the domain label of the target domain. The goal of our proposed method is to reduce the shifts across domains through learning of transfer feature  $\mathbf{f} = G_f(\mathbf{x})$  and classifier  $\mathbf{y} = G_y(\mathbf{f})$ .

To match the source and target domains upon the structures better, there are two issues that need to be considered: 1) If the prediction result of the training sample is not clear, it is difficult to determine its class, so using this sample to adjust the network may lead to negative transfer; 2) If we do not focus on the alignment of the classes with similar features, the samples belonged to these classes may be misclassified after domain adaptation. In response to the above issues, we have improved the deep learning network from the following two points.

1) If the label vector of sample  $\mathbf{x}_i^s$  in the source domain is  $\mathbf{y}_i^s$ , where  $\mathbf{y}_i^s = [y_i^s(1) \ y_i^s(2) \ \dots \ y_i^s(K_t)]^T$ ,  $y_i^s(j) = 1$ ,  $y_i^s(k) = 0$ ,  $k = 1, 2, \dots, j-1, j+1, \dots, K_t$ , it is reasonable to align the  $j$ th class using the sample  $\mathbf{x}_i^s$ , and the weight of the sample  $\mathbf{x}_i^s$  should be set to 1 in the alignment process; If the prediction result  $\hat{\mathbf{y}}_i^t$  of sample  $\mathbf{x}_i^t$  in the target domain is clear as shown in Fig.2(a), i.e.,  $\hat{y}_i^t(j) \approx 1$ ,  $\hat{y}_i^t(k) \approx 0$ ,  $k = 1, 2, \dots, j-1, j+1, \dots, K_t$ ,  $\mathbf{x}_i^t$  is very likely to belong to the  $j$ th class, and it can be considered to play an important role in the alignment process, so the sample  $\mathbf{x}_i^t$  should be given a large weight; on the contrary, if the label prediction result  $\hat{\mathbf{y}}_i^t$  is not clear as shown in Fig.2(b), such prediction result will make the sample  $\mathbf{x}_i^t$  untrusted in the alignment process, so its weight should be as small as possible. From the above analysis, we proposed a new weight function according to the probability distribution entropy of the label vector as shown in (1), which is located between the prediction result and the domain discriminators in the network as shown in Fig.3,

$$h(\hat{\mathbf{y}}_i) = 1 - \frac{1}{K_s \log K_s} \sum_{k=1}^{K_s} (\hat{y}_i(k) \log \hat{y}_i(k)) \quad (1)$$

where if the sample  $\mathbf{x}$  belongs to the source domain,  $\hat{\mathbf{y}}_i$  is the true label vector, i.e.,  $\hat{\mathbf{y}}_i = \mathbf{y}_i^s$ ; else  $\hat{\mathbf{y}}_i$  is the pseudo label vector.

2) From the analysis above, we want to achieve the alignment of all classes between the source and target domains by using the true or pseudo labels of the training samples. However, because of the similarity differences between classes, if we align all classes using the same weight, the classes with similar features will most likely be misaligned, such as ‘‘cat’’ and ‘‘dog’’, ‘‘bicycle’’

and ‘‘motorcycle’’. Therefore, in order to avoid that the classes which have strong similarity to others are misaligned, we should pay more attention to the alignments of these classes in the domain adaptation process, i.e., give greater weights to these alignments. It is well known that there have been many effective functions which can measure feature similarity during clustering [28], [29], where linear discriminant analysis (LDA) model can be considered suitable to measure the differences between classes well. Therefore, in the improved network we add a new weight function  $s(\mathbf{f})$  as shown in (2) based on the idea of LDA, which is used to measure the similarity between different classes.

$$s_j(\mathbf{f}) = \max_{k=1,2,\dots,K_s, k \neq j} \left\{ \frac{\left\langle \frac{1}{c_j} \sum_{\mathbf{y}_i^s(j)=1} \mathbf{f} \mathbf{x}_i^s, \frac{1}{c_k} \sum_{\mathbf{y}_i^s(k)=1} \mathbf{f} \mathbf{x}_i^s \right\rangle}{\left\| \frac{1}{c_j} \sum_{\mathbf{y}_i^s(j)=1} \mathbf{f} \mathbf{x}_i^s \right\| \cdot \left\| \frac{1}{c_k} \sum_{\mathbf{y}_i^s(k)=1} \mathbf{f} \mathbf{x}_i^s \right\|} \right\} \quad (2)$$

where  $c_j$  is the number of the samples with  $j$ -label during one iteration, and  $\mathbf{f} \mathbf{x}_i^s$  represents the feature of the sample  $\mathbf{x}_i^s$  in the source domain,  $i = 1, 2, \dots, n_s$ .

From the above analysis, we can conclude that the sample with less feature unique should be given greater weight during class alignment.

In summary, the network architecture of the proposed method is shown in Fig.3,

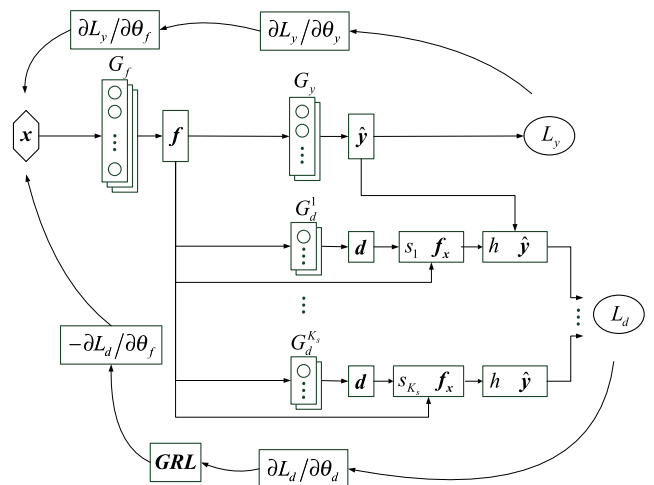


FIGURE 3. The proposed network architecture.

where  $G_f$ ,  $G_y$  and  $G_d$  represent the feature extractor, the feature classifier and the domain discriminator respectively, whose parameters are  $\theta_f$ ,  $\theta_y$  and  $\theta_d^i$ ,  $i = 1, 2, \dots, K_s$ , and **GRL** represents the gradient reversal layer.

Then, in order to obtain good model parameters through training, we need to design a reasonable loss function, which needs to meet the following three requirements as much as possible:

1) Minimize the label prediction loss  $L_y$  of the samples in the source domain by seeking the parameters  $\theta_y$  and  $\theta_f$ ;

2) Minimize the domain prediction loss  $L_d$  of the samples in the source and target domains by seeking  $\theta_d^i$ ,  $i = 1, 2, \dots, K_s$ ;

3) Maximize the domain prediction loss  $L_d$  of the samples in the source and target domains by seeking  $\theta_f$ , where the purpose is to make the feature distributions of the source and target domains as similar as possible.

From the above analysis, the loss function can be written as (3),

$$\begin{aligned}
 L & \left( \theta_f, \theta_y, \theta_d^1, \theta_d^2, \dots, \theta_d^{K_s} \right) \\
 & = \frac{1}{n_s} \sum_{i=1,2,\dots,N; d_i=0} L_y \left( G_y \left( G_f \left( \mathbf{x}_i \right) \right), \mathbf{y}_i \right) \\
 & \quad - \frac{\lambda}{n_s + n_t} \sum_{i=1,2,\dots,N} \sum_{k=1}^{K_s} L_d \left( G_d \left( s_k \left( G_f \left( \mathbf{x}_i \right) \right) h \left( \hat{\mathbf{y}}_i \right) \hat{\mathbf{y}}_i \left( k \right) \right. \right. \\
 & \quad \left. \left. \times G_f \left( \mathbf{x}_i \right) \right), d_i \right) \quad (3)
 \end{aligned}$$

and its optimization problem is to seek the best parameters  $\hat{\theta}_y$ ,  $\hat{\theta}_f$  and  $\hat{\theta}_d^i$ ,  $i = 1, 2, \dots, K_s$ , which satisfy (4) and (5).

$$\left( \hat{\theta}_f, \hat{\theta}_y \right) = \arg \min_{\theta_f, \theta_y} L \left( \theta_f, \theta_y, \theta_d^1, \theta_d^2, \dots, \theta_d^{K_s} \right) \quad (4)$$

$$\left( \hat{\theta}_d^1, \hat{\theta}_d^2, \dots, \hat{\theta}_d^{K_s} \right) = \arg \max_{\theta_d^1, \theta_d^2, \dots, \theta_d^{K_s}} L \left( \theta_f, \theta_y, \theta_d^1, \theta_d^2, \dots, \theta_d^{K_s} \right) \quad (5)$$

## B. OPTIMIZATION OF PARAMETERS BASED ON BATCH GRADIENT DESCENT

In the loss function of our proposed deep network, the function  $s_i(\mathbf{f})$ , which depends on the samples of the source domain, represents the similarities between classes, and the function  $h(\hat{\mathbf{y}})$  represents the credibility of label vector, which depends on the samples of both the source and target domains. Therefore, it is necessary to select a certain number of samples from the source and target domains at each iteration during training, then use batch gradient descent method to optimize the model parameters, which is summarized below in Algorithm 1.

## IV. EXPERIMENTS

### A. DATASETS

The proposed method is evaluated on three commonly used datasets, which include Office-31 dataset [30] and ImageCLEF-DA dataset [31], and Office-Caltech-10 dataset [32].

1) The Office-31 dataset consists of the following three domains: ‘‘Amazon’’ (A31), ‘‘Webcam’’ (W31) and ‘‘DSL’’ (D31), and each domain contains 31 categories of objects, where some samples are shown in Fig. 4.

2) The ImageCLEF-DA dataset consist of three domains: ‘‘Caltech-256’’ (C), ‘‘ImageNet ILSVRC 2012’’ (I) and

### Algorithm 1 Model Parameters Optimization Process

1. Input  $D_s, D_t, \lambda$  and  $\omega$ , where  $D_s = \left\{ \left( \mathbf{x}_i^s, \mathbf{y}_i^s \right) \right\}_{i=1}^{n_s}, D_t = \left\{ \left( \mathbf{x}_i^t \right) \right\}_{i=1}^{n_t}$ ;
2. Initialize  $\theta_y^{(0)}, \theta_f^{(0)}$  and  $\theta_d^{i(0)}, i = 1, 2, \dots, K_s, c = 0$ ;
3. **for**  $epoch = 1, 2, \dots, \omega$ , **do**
4. Solve  $s_j(\mathbf{f}), j = 1, 2, \dots, K_s$ ;
5. **for**  $batch = 1, 2, \dots$ , **do**
6. Solve  $h(\mathbf{y})$ , where each batch contains  $2K_s$  samples;
7. Update  $\theta_y^{(c)}, \theta_f^{(c)}$  and  $\theta_d^{i(c)}, i = 1, 2, \dots, K_s$ ;
8.  $\theta_f^{(c+1)} \leftarrow \theta_f^{(c)} - \mu \left( \frac{\partial L_y^{(c)}}{\partial \theta_f} - \lambda \frac{\partial L_d^{(c)}}{\partial \theta_f} \right)$
9.  $\theta_y^{(c+1)} \leftarrow \theta_y^{(c)} - \mu \frac{\partial L_y^{(c)}}{\partial \theta_y}$
10.  $\theta_d^{i(c+1)} \leftarrow \theta_d^{i(c)} - \mu \frac{\partial L_d^{(c)}}{\partial \theta_d^i}, i = 1, 2, \dots, K_s$
11. **end for**
12. **end for**
13. Output  $\hat{\theta}_y, \hat{\theta}_f$  and  $\hat{\theta}_d^i, i = 1, 2, \dots, K_s$ .








FIGURE 4. Some samples of Office-31 dataset.



FIGURE 5. Some samples of ImageCLEF-DA dataset.

‘‘Pascal VOC 2012’’ (P), each domain contains 12 categories of objects, where some samples are shown in Fig. 5.

TABLE 1. The classified results of some easily misclassified samples.

The samples and these label					
	aircraft	boat	motorcycle	car	aircraft
Network 3	✓	bottle	bicycle	bus	car
Network 2	✓	✓	bicycle	✓	car
Network 1	✓	✓	✓	✓	✓

3) The Office-Caltech-10 dataset consists of four different domains: “Amazon” (A), “Webcam” (W), “DSLR” (D) and “Caltech” (C), each domain contains 10 categories of objects, where “Amazon”, “Webcam” and “DSLR” are selected from Office-31 dataset, “Caltech” is added as a new domain, and some samples are shown in Fig.6.

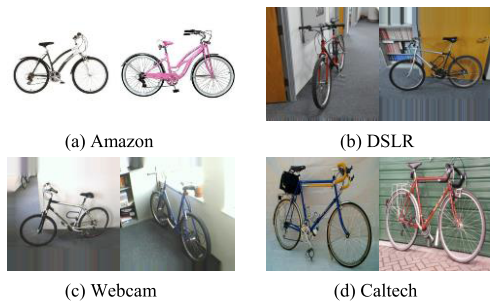


FIGURE 6. Some samples of Office-Caltech-10 dataset.

**B. MODEL ARCHITECTURES AND PARAMETERS SETTING**

The used datasets in our experiments are not very large, if we train the network from scratch, the over fitting is easily to appear during training, so it is suitable to fine-tune the deep network which has been pre-trained on ImageNet, where the AlexNet [23] and the three fully connected layers ( $f \rightarrow 1024 \rightarrow 1024 \rightarrow 2$ ) [20] can be used for feature extractor and domain discriminator, and the initial model parameters are selected from the literature [7] in the proposed method. In addition to designing the network architecture, there are two parameters, i.e., the learning rate  $\mu$  and the balance factor  $\lambda$ , to be set according to the experimental results, where the optimal parameters can be acquired by maximizing the average recognition accuracy  $Ara$  on the above three datasets as shown in (6). Fig.7 demonstrates the variation of the average recognition accuracies when  $\lambda \in \{0.01, 0.05, 0.1, 0.5, 1, 2\}$  and  $\mu \in \{0.1, 0.01, 0.001\}$ .

$$\lambda^* = \arg \max_{\lambda} Ara(\lambda) \tag{6}$$

From Fig.7, we can see that when the parameters  $\lambda$  and  $\mu$  are set as 0.5 and 0.01 respectively, the highest average recognition accuracy can be obtained.

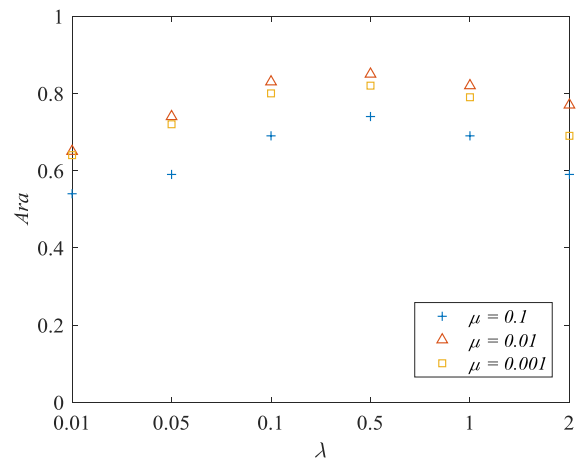


FIGURE 7. The average recognition accuracies under different  $\lambda$  and  $\mu$ .

**C. RESULTS AND ANALYSIS**

To verify the effectiveness of the innovations, we will compare the feature transfer abilities based on the following three network structures: 1) network 1: the proposed network; 2) network 2: the network without  $h(\hat{y})$  layer, 3) network 3: the network without  $h(\hat{y})$  and  $s_i(f)$  layers, where the compare results are visualized in Fig.7 by using t-SNE.

First, From Fig.8(b), we can see that the recognition accuracy of the network without the proposed functions  $h(\hat{y})$  and  $s_i(f)$  is 0.786, where some images of different classes with similar features are easily misclassified; Then, when the domain discriminators are weighted differently based on the feature similarities between classes, the recognition accuracy is significantly improved to 0.825 as shown in Fig.8(d); At last, we add the label credibility weight function  $h(\hat{y})$  to the network, and the recognition accuracy is further improved to 0.840. It can be seen that the proposed two weight functions have a positive effect on correct classification, especially for the classes with similar features as shown in Fig.9.

As we know, the features of the classes “Person” and “Aircraft” are significantly different with each other, it is not difficult to distinguish the two classes, therefore, In Fig.9(d) and Fig.9(e), we can see that the proposed models is almost the same as some common used models in terms of the recognition accuracy. However, the discrimination between

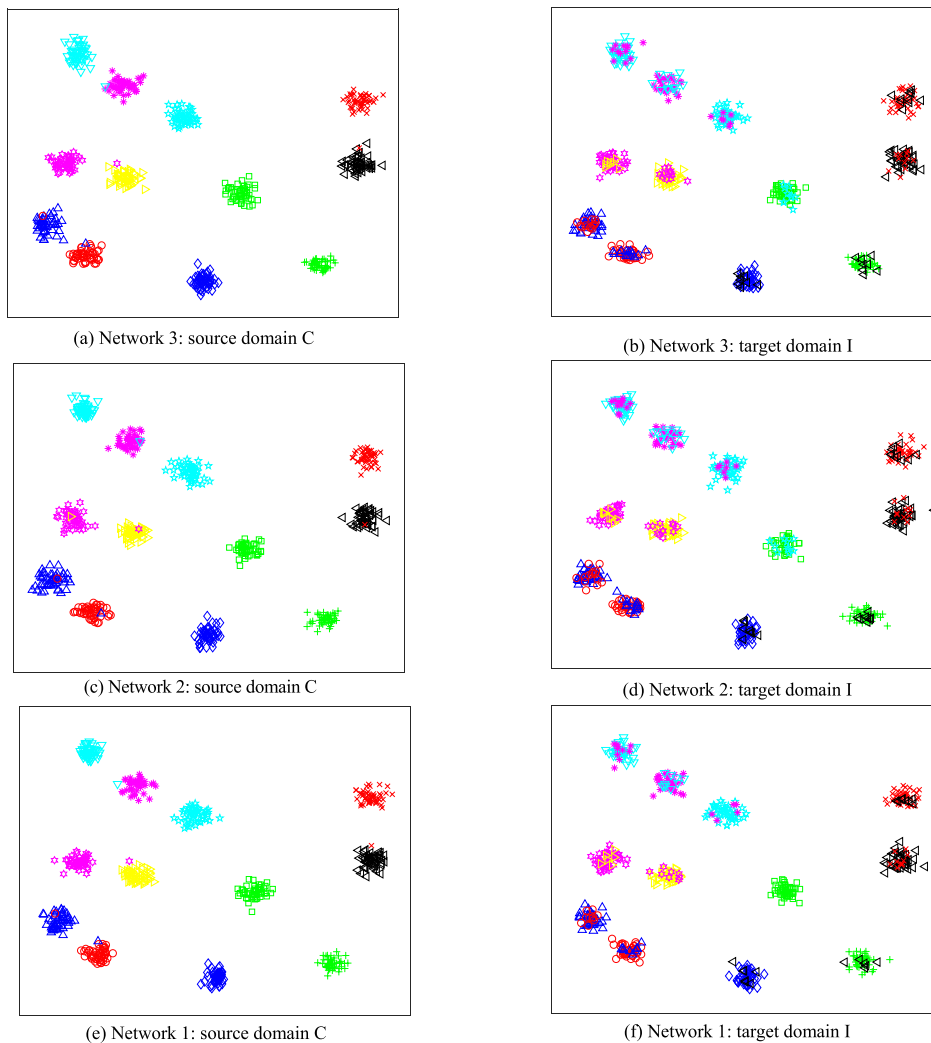


FIGURE 8. The t-SNE visualization of deep features on the ImageCLEF-DA dataset.

TABLE 2. Comparison of recognition accuracy using Office-31 dataset.

Method	$A \rightarrow W$	$D \rightarrow W$	$W \rightarrow D$	$A \rightarrow D$	$D \rightarrow A$	$W \rightarrow A$	Average
TCA [32]	59.0±0.0	90.2±0.0	88.2±0.0	57.8±0.0	51.6±0.0	47.9±0.0	65.8
GFK [31]	58.4±0.0	93.6±0.0	91.0±0.0	58.6±0.0	52.4±0.0	46.1±0.0	66.7
DDC [11]	61.0±0.5	95.0±0.3	98.5±0.3	64.9±0.4	47.2±0.5	49.4±0.6	69.3
DAN [7]	68.5±0.3	96.0±0.1	99.0±0.1	66.8±0.2	50.0±0.4	49.8±0.3	71.7
RevGrad [18]	73.0±0.6	96.4±0.4	99.2±0.3	-	-	-	-
RTN [10]	73.3±0.3	96.8±0.2	99.6±0.1	71.0±0.2	50.5±0.3	51.0±0.1	73.7
MADA [20]	78.5±0.2	99.8±0.1	100.0±0.0	74.1±0.1	56.0±0.2	54.5±0.3	77.1
Ours	80.2±0.1	99.8±0.1	99.8±0.1	76.8±0.1	57.5±0.2	57.2±0.2	78.6

“Bicycle” and “Motorcycle” is not great, which look more similar comparing with “Person” and “Aircraft”, so they are more likely to be misclassified for some networks. For example, the true label of the test image in Fig.9(a) is “Motorcycle”, since network 3 does not pay enough attention to these similar classes during training, the image is incorrectly recognized as shown in Fig.9(d). On the contrary,

the proposed method solve the problem well using two novel weighted functions. The three networks based classification results of some samples which are easily misclassified are shown in Table 1.

From Table 1, we can see that some classes are not very related to each other, for example, “Aircraft” and “Car”, but due to the factors such as shooting angle, these images may

TABLE 3. Comparison of recognition accuracy using ImageCLEF-DA dataset.

Method	$I \rightarrow P$	$P \rightarrow I$	$I \rightarrow C$	$C \rightarrow I$	$C \rightarrow P$	$P \rightarrow C$	Average
DAN	67.3±0.2	80.5±0.3	87.7±0.3	76.0±0.3	61.6±0.3	88.4±0.2	76.9
RTN	67.4±0.3	82.3±0.3	89.5±0.4	78.0±0.2	63.0±0.2	90.1±0.1	78.4
RevGrad	66.5±0.5	81.8±0.4	89.0±0.5	79.8±0.5	63.5±0.4	88.7±0.4	78.2
MADA	68.3±0.3	83.0±0.1	91.0±0.2	80.7±0.2	63.8±0.2	92.2±0.3	79.8
Ours	69.7±0.2	85.2±0.1	91.3±0.2	81.9±0.2	65.6±0.3	93.3±0.2	81.2

TABLE 4. Comparison of recognition accuracy using Office-Caltech-10 dataset.

Method	$A \rightarrow C$	$A \rightarrow D$	$A \rightarrow W$	$C \rightarrow A$	$C \rightarrow D$	$C \rightarrow W$	$D \rightarrow A$	$D \rightarrow C$	$D \rightarrow W$	$W \rightarrow A$	$W \rightarrow C$	$W \rightarrow D$	Average
TCA	81.2	82.8	84.4	92.1	87.9	88.1	90.4	79.6	96.6	85.6	75.5	99.4	87.0
GFK	76.2	86.0	89.5	90.7	77.1	78.0	89.8	77.9	97.0	88.5	77.1	98.1	85.5
DDC	83.5	88.4	83.1	91.9	88.8	85.4	89.0	79.2	98.1	84.9	73.4	100	87.1
DAN	84.1	91.7	91.8	92.0	89.3	90.6	90.0	80.3	98.5	92.1	81.2	100	90.1
RTN	88.1	95.5	95.2	93.7	94.2	96.9	93.8	84.6	99.2	92.5	86.6	100	93.4
Ours	89.2	95.5	96.2	94.4	96.1	96.9	95.2	86.4	99.2	93.6	88.2	100	94.2

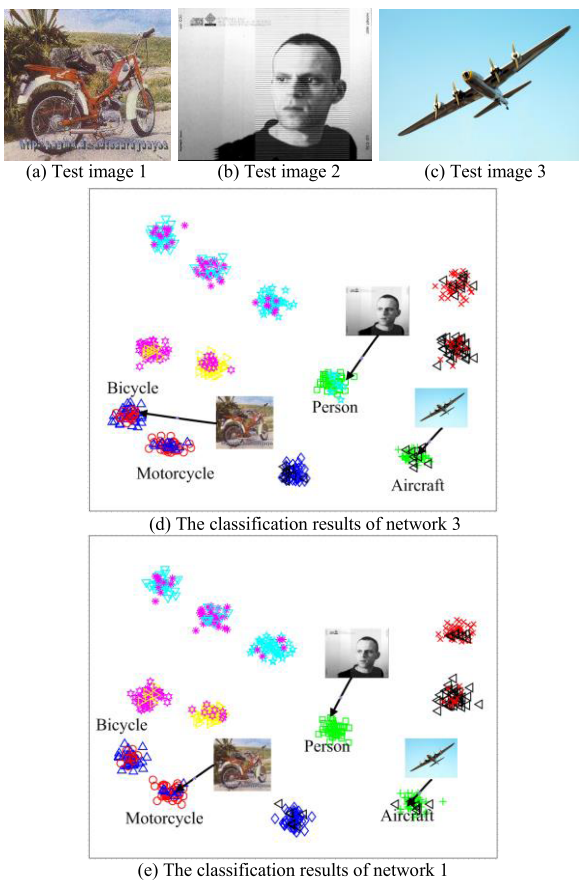


FIGURE 9. The testing images and their classification results using different networks.

still be misclassified. As shown in Table 1, the true labels of the first and last images are “Aircraft”, but they do not look very similar, where the last image is misclassified as “Car”. In this case, the proposed model can still recognize

these images accurately, and we can conclude that the two proposed functions are helpful for improving the robustness of the model.

Then, we will compare the proposed method with some existed models using Office-31 dataset, ImageCLEF-DA, dataset, and Office-Caltech-10 dataset, and the comparison results are shown in Table 2, Table 3, and Table 4.

In the three tables, we can see that the proposed method makes that all the classes in the source and target domains are better aligned after adaptation, especially for the classes that have a strong correlation with other classes, and achieve higher recognition accuracy than other methods, therefore, it can be concluded from the experimental results that our innovations are of great significance.

V. CONCLUSION

In this paper, an unsupervised domain adaptation method based on weighted adversarial network is proposed, which implements the alignment of all classes between domains according to feature similarity and label credibility. To the best of our knowledge, the weighted adversarial network based on feature similarity and label credibility has not been proposed, and its biggest innovation is to improve the effect of distinguishing the classes with similar features such as “bicycle” and “motorcycle”. The experimental results on the used two datasets showed the improvements in accuracy.

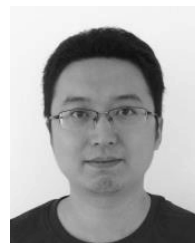
ACKNOWLEDGMENT

The authors would like to thank the reviewers for their helpful comments and suggestions.

REFERENCES

[1] Y. Pang, S. Wang, and Y. Yuan, “Learning regularized LDA by clustering,” *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 25, no. 12, pp. 2191–2201, Dec. 2014.

- [2] M. Wang and W. Deng, "Deep visual domain adaptation: A survey," *Neurocomputing*, vol. 312, pp. 135–153, Oct. 2018.
- [3] S. Gupta, J. Hoffman, and J. Malik, "Cross modal distillation for supervision transfer," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Las Vegas, NV, USA, Jun. 2016, pp. 2827–2836.
- [4] J. L. Hu, J. W. Lu, and Y. P. Tan, "Deep transfer metric learning," in *Proc. IEEE Conf. on Comput. Vis. Pattern Recognit.*, Boston, MA, USA, Sep. 2015, pp. 325–333.
- [5] X. C. Peng, J. Hoffman, X. Y. Stella, and K. Saenko, "Fine-to-coarse knowledge transfer for low-res image classification," in *Proc. IEEE Int. Conf. Image Process.*, Phoenix, AZ, USA, Sep. 2016, pp. 3683–3687.
- [6] E. Tzeng, J. Hoffman, T. Darrell, and K. Saenko, "Simultaneous deep transfer across domains and tasks," in *Proc. IEEE Int. Conf. Comput. Vis.*, Santiago, Chile, Jun. 2015, pp. 4068–4076.
- [7] M. Long, Y. Cao, J. Wang, and M. I. Jordan, "Learning transferable features with deep adaptation networks," in *Proc. 32nd Int. Conf. Mach. Learn.*, Ithaca, NY, USA, vol. 37, 2015, pp. 97–105.
- [8] H. Yan, Y. Ding, P. Li, Q. Wang, Y. Xu, and W. Zuo, "Mind the class weight bias: Weighted maximum mean discrepancy for unsupervised domain adaptation," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Honolulu, HI, USA, Jul. 2017, pp. 2272–2281.
- [9] X. Zhang, F. Xinnan Yu, S.-F. Chang, and S. Wang, "Deep transfer network: Unsupervised domain adaptation," 2015, *arXiv:1503.00591*. [Online]. Available: <http://arxiv.org/abs/1503.00591>
- [10] M. S. Long, H. Zhu, J. M. Wang, and M. I. Jordan, "Unsupervised domain adaptation with residual transfer networks," in *Proc. Adv. Neural Inf. Process. Syst.*, Barcelona, Spain, 2016, pp. 136–144.
- [11] E. Tzeng, J. Hoffman, N. Zhang, K. Saenko, and T. Darrell, "Deep domain confusion: Maximizing for domain invariance," 2014, *arXiv:1412.3474*. [Online]. Available: <http://arxiv.org/abs/1412.3474>
- [12] M. Long, Y. Cao, J. Wang, and M. I. Jordan, "Deep transfer learning with joint adaptation networks," in *Proc. Int. Conf. Mach. Learn.*, Sydney, NSW, Australia, vol. 70, 2017, pp. 2208–2217.
- [13] J. Shen, Y. R. Qu, W. N. Zhang, and Y. Yu, "Wasserstein Distance Guided Representation Learning for Domain Adaptation," in *Proc. 32nd AAAI Conf. Artif. Intell.*, New Orleans, LO, USA, 2018, pp. 4058–4065.
- [14] G.-Y. Zhou and J. X. Huang, "Modeling and mining domain shared knowledge for sentiment analysis," *ACM Trans. Inf. Syst.*, vol. 36, no. 2, pp. 1–36, Sep. 2017.
- [15] B. C. Sun and K. Saenko, "Deep coral: Correlation alignment for deep domain adaptation," *Proc. Eur. Conf. Comput. Vis.*, Amsterdam, The Netherlands, 2016, pp. 443–450.
- [16] X. Peng and K. Saenko, "Synthetic to real adaptation with generative correlation alignment networks," in *Proc. IEEE Winter Conf. Appl. Comput. Vis. (WACV)*, Lake Tahoe, NV, USA, Mar. 2018, pp. 1982–1991.
- [17] Y. Ganin, E. Ustinova, H. Ajakan, P. Germain, H. Larochelle, F. Laviolette, M. Marchand, and V. S. Lempitsky, "Domain-adversarial training of neural networks," *J. Mach. Learn. Res.*, vol. 17, no. 59, pp. 1–35, 2016.
- [18] Y. Ganin V. Lempitsky, "Unsupervised domain adaptation by backpropagation," in *Proc. 32nd Int. Conf. Mach. Learn.*, Lille, France, vol. 37, 2015, pp. 1180–1189.
- [19] W. Zhang, W. Ouyang, W. Li, and D. Xu, "Collaborative and adversarial network for unsupervised domain adaptation," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, Salt Lake City, Utah, USA, Jun. 2018, pp. 3801–3809.
- [20] Z. Y. Pei, Z. J. Cao, M. S. Long, and J. M. Wang, "Multi-adversarial domain adaptation," *Proc. 32nd AAAI Conf. Artif. Intell.*, New Orleans, LO, USA, 2018, pp. 3934–3941.
- [21] Y. C. Zhang, T. L. Liu, M. S. Long, and M. I. Jordan, "Bridging theory and algorithm for domain adaptation," in *Proc. Int. Conf. Mach. Learn.*, Los Angeles, CA, USA, 2019, pp. 7404–7413.
- [22] J. Zhang, Z. Ding, W. Li, and P. Ogunbona, "Importance weighted adversarial nets for partial domain adaptation," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, Lake City, Utah, USA, Jun. 2018, pp. 8156–8164.
- [23] M. Ghifary, W. B. Kleijn, M. Zhang, and D. Balduzzi, "Domain generalization for object recognition with multi-task autoencoders," in *Proc. IEEE Int. Conf. Comput. Vis. (ICCV)*, Santiago, Chile, Dec. 2015, pp. 2551–2559.
- [24] M. Ghifary, W. B. Kleijn, M. J. Zhang, D. Balduzzi, and W. Li, "Deep reconstruction-classification networks for unsupervised domain adaptation," in *Proc. Eur. Conf. Comput. Vis.*, Amsterdam, The Netherlands, 2016, pp. 597–613.
- [25] F. Z. Zhuang, X. H. Cheng, P. Luo, S. J. Pan, and Q. He, "Supervised representation learning: Transfer learning with deep autoencoders," in *Proc. 24th Int. Conf. Artif. Intell.*, Buenos Aires, Argentina, 2015, pp. 4119–4125.
- [26] J.-Y. Zhu, T. Park, P. Isola, and A. A. Efros, "Unpaired Image-to-Image translation using cycle-consistent adversarial networks," in *Proc. IEEE Int. Conf. Comput. Vis. (ICCV)*, Venice, Italy, Oct. 2017, pp. 2223–2232.
- [27] T. Kim, M. Cha, H. Kim, J. K. Lee, and J. Kim, "Learning to discover cross-domain relations with generative adversarial networks," in *Proc. 34th Int. Conf. Mach. Learn.*, Sydney, NSW, Australia, vol. 70, 2017, pp. 1857–1865.
- [28] X. Peng, H. Zhu, J. Feng, C. Shen, H. Zhang, and J. T. Zhou, "Deep clustering with sample-assignment invariance prior," *IEEE Trans. Neural Netw. Learn. Syst.*, early access, Dec. 31, 2019, doi: [10.1109/TNNLS.2019.2958324](https://doi.org/10.1109/TNNLS.2019.2958324).
- [29] X. Peng, J. Feng, S. Xiao, W.-Y. Yau, J. T. Zhou, and S. Yang, "Structured auto encoders for subspace clustering," *IEEE Trans. Image Process.*, vol. 27, no. 10, pp. 5076–5086, Oct. 2018.
- [30] K. Saenko, B. Kulis, M. Fritz, and T. Darrell, "Adapting visual category models to new domains," in *Proc. Eur. Conf. Comput. Vis.*, Crete, Greece, 2010, pp. 213–226.
- [31] B. Gong, Y. Shi, F. Sha, and K. Grauman, "Geodesic flow kernel for unsupervised domain adaptation," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Providence, RI, USA, Jun. 2012, pp. 2066–2073.
- [32] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Las Vegas, NV, USA, Jun. 2016, pp. 770–778.



**XU JIA** was born in Kaiyuan, Liaoning, China, in 1983. He received the B.S. degree in automation from Shenyang Aerospace University, Liaoning, in 2005, and the M.S. and Ph.D. degrees in pattern recognition and intelligent system from North-eastern University, Liaoning, in 2009 and 2012, respectively.

From 2013 to 2015, he was a Lecturer with the Liaoning University of Technology, where he has been an Assistant Professor with the School of Electronics and Information Engineering, since 2016. He is the author of more than 30 articles. His research interests include machine learning and image processing.



**FUMING SUN** was born in Dalian, Liaoning, China, in 1972. He received the Ph.D. degree from the University of Science and Technology of China, Hefei, China, in 2007.

From September 2012 to July 2013, he was a Visiting Scholar with the Department of Automation, Tsinghua University. He was a Professor with the School of Electronics and Information Engineering, Liaoning University of Technology, from 2004 to 2018. He is currently a Professor with the School of Information and Communication Engineering, Dalian Minzu University, Dalian, China. His current research interests include content-based image retrieval, image content analysis, and pattern recognition.

...