International Journal of Innovative
Computing, Information and Control
Volume **17**, Number **5**, October **2021**

ICIC International ©2021 ISSN 1349-4198
pp. 1501–1511

ADAPTATION FEATURE NORM METHOD BASED ON L2-NORMALIZATION AND SCALING PARAMETER

Liquan Zhao[1,*], Siying Zhou[1] and Yanfei Jia[2]

[1]Key Laboratory of Modern Power System Simulation and Control
and Renewable Energy Technology, Ministry of Education
Northeast Electric Power University
No. 169, Changchun Road, Jilin 132012, P. R. China
[*]Corresponding author: zhaoliquan@neepu.edu.cn; zhouruanruan1996@163.com

[2]College of Information and Electric Engineering
Beihua University
No. 3999, Beijing Road, Jilin 132013, P. R. China
jia_yanfei@163.com

Received March 2021; revised July 2021

Abstract. *The stepwise adaptation feature norm (SAFN) is one of the transfer learning methods. It proposes that only the samples with larger feature norms are easier to classify, and samples with larger feature norms have better domain transfer performance. It ignores the classification effect of samples with small feature norms. To improve accuracy, this paper proposes an improved method based on the stepwise adaptation feature norm. It is designed as an L2-normalization layer and adds it prior to the last fully connected layer. It constrains the feature norms of samples to a fixed hypersphere. This layer enables more attention to be before samples with small feature norms and mitigates the importance of samples with large feature norms. The layer also decreases the distance between the samples in the same category and increases the distances between different categories. This makes the decision boundaries clearer for classification. Besides, it also adds a scaling parameter after the L2-normalization layer to make the hyperspherical surface area larger. This makes features be better distributed over the hyperspherical surface. The proposed method is compared with other methods including ResNet-50, DAN, DANN, JAN, CDAN, SAFN on the datasets Office-31, ImageCLEF-DA, and Office-Home. The simulation results show that the proposed method has the highest accuracy in different datasets.*

Keywords: Transfer learning, Stepwise adaptation feature norm, L2-normalization, Scaling parameter

1. **Introduction.** Transfer learning is one of the artificial intelligence methods. Its goal is to transform learned knowledge in the source domain to the target domain. In traditional machine learning, it requires labeling an amount of training data for training a new model. This consumes a large amount of manpower and material. In some real applications, some data have no label, and training model is difficult. The transfer learning method can solve the problem by transforming the learned knowledge from the dataset that has an amount of training data with labels to the new model. Besides, most of the data or tasks are correlative, so the transfer learning method also can share the model parameters in the source domain to the new model to improve the efficiency of model training, to avoid training models from scratch. The transfer learning method has been applied in

DOI: 10.24507/ijicic.17.05.1501

word recognition [1], insulator identification [2], biomedical event trigger extraction [3], medical image processing [4], and classifying of animals [5].

Domain adaptation is one of the special transfer learning methods. It maps the data in the source domain and the target domain to feature space and adjusts the parameters to reduce the distance between features in different domains. Therefore, the trained objective function in the source domain can be transformed to the target domain, in order to improve the accuracy of the target domain. Domain adaptation has several applications in the field of vision, including image classification, semantic segmentation, target re-identification, etc. [6-8]. Domain adaptation methods can be divided into supervised domain adaptation and unsupervised domain adaptation, based on the availability of the target data label. In this paper, we concern with unsupervised domain adaptation. The unsupervised domain adaptation transforms the knowledge that is learned from the source domain that contains a large quantity of labeled data, to the target domain that only has the unlabeled data.

The research of most domain adaptive methods can be divided into two parts. One is to minimize the source domain error of the classification task. The other is to minimize some statistical discrepancies between the source domain and the target domain to obtain domain invariant features. They want to make the data distribution between the source domain and the target domain be "overlap". However, strict alignment of the source domain and the target domain is difficult to achieve. To reduce the data distance between the source domain and the target domain, Xu et al. [32] proposed the SAFN method by introducing the difference between the feature norms of the source domain and the target domain into loss function, to reduce the error of the domain adaptive method. It reduced the transfer effect of domain shift through feature norm adaptive. This method considered that the larger the feature norm is, the easier it is to classify. The effect of features with a small intra-category norm on the classification was ignored. The dataset contains many classes. The SAFN method only considered the distribution of the feature norm in the source and the target domains and ignored the differences in each class. This can lead to a reduction in classification accuracy. To improve accuracy, an improved method is proposed.

The main contributions of this paper are summarized as the following.

1) It designs an L2-normalization layer and adds it to the last fully connected layer in the SAFN method. It limits each feature norm onto a hypersphere of fixed radius, to make features from the same category be closer together in the source domain.
2) It designs a scale layer to increase the inter-class distance between different classes in the standardized space. If we only add the L2-normalization layer, the feature norms of all samples will be mixed on a smaller hypersphere. The scale layer can expand the hypersphere to make the decision boundaries between different classes more obvious.

In Section 1, we introduce transfer learning, domain adaptation, and our contributions. In Section 2, we review the developments related to domain adaptation methods. In Section 3, we describe our proposed method. In Section 4, we discuss and analyze the simulation results. In Section 5, conclusions are given.

2. **Related Work.** Domain adaptation methods can be divided into supervised domain adaptation and unsupervised domain adaptation. Our proposed method belongs to unsupervised domain adaptation, so we mainly introduce the unsupervised domain adaptation methods in the following. There are two kinds of methods. One kind of method uses a domain discriminator to determine whether the feature comes from the source domain or the target domain. When the domain discriminator cannot distinguish the feature, the

domain-invariant feature is learned. Ganin and Lempitsky [9] added a domain discriminator layer and gradient inversion layer to a standard deep network to ensure invariance between the source domain and the target domain. This method increased the difference between the features generated in the two domains, and made the distinguisher unable to distinguish features. Ganin et al. proposed the domain adaptation neural network method (DANN) by adding an adversarial mechanism to train the neural network [10]. It also used the gradient inversion layer as a trick during training. The DANN only considered sample features for distribution matching and did not consider that each domain has its independent features. To solve the problem, domain separation network (DSN) was proposed as an extension of DANN [11]. The DSN method improved the ability of the model to extract domain-invariant features by constructing a new source domain and the target domain. The new domains were composed of two parts that are a private component and a shared component across domains. When the data distribution represents a complex multi-modal structure, DANN was unable to capture this multi-modal structure. Pei et al. [12] proposed a multi-adversarial domain adaptive method. The method could capture multi-modal structures and align fine granularity of different data distributions through multiple domain discriminators, reduce the phenomenon of domain alignment errors. To solve the problem that the DANN method ignores the complex data distribution, Long et al. [13] proposed the CDAN approach to deal with conditional adversarial problems. The trained generators may generate unambiguous features near class boundaries. To solve this problem, Saito et al. [14] proposed to maximize the discrepancy between the outputs of the two classifiers and minimize the target features generated by the feature generator to adjust the distribution of sources and the targets. Cao et al. [15] proposed partial adversarial domain adaptation (PADA) to mitigate negative transfer. It reduced the weight of outlier classifications during domain alignment and used the distribution of features in the matching shared label space to facilitate the positive transfer. The adversarial domain adaptation training is a game between generator and discriminator. The loss function does not represent the model training progress, and it requires a longer time to reach a certain equilibrium.

Another kind of unsupervised domain adaptation method is based on the maximum mean discrepancy (MMD) method [16]. The source and the target domain samples are firstly mapped to the feature space, respectively. Secondly, it minimizes the distance of feature distribution between the source domain and the target domain in the feature space. If a large number of samples generated from the source domain and the target domain have an equal mean discrepancy in the feature space, then it could be assumed that the source domain and the target domain are highly similar and belong to the same distribution. Pan et al. [17] proposed transfer component analysis (TCA) to learn some transfer components across domains by reproducing the maximum mean discrepancy in the kernel Hilbert space. Long et al. [18] proposed joint distribution adaptation (JDA) to reduce the differences between edge and conditional distributions. By adding an additional loss term to the JDA, some improved methods were proposed. The ARTL [19] added a structural risk minimization framework to JDA and used manifold regularization to learn the adaptive classifier. Hou et al. [20] added the selection of the target domain data to JDA. The method updated the distribution of class conditions. VDA [21] added the calculation of intra-class distance and class spacing to optimize the objective of JDA. JGSA [22] added intra-class distance, class spacing, label persistence to JDA. Joint geometric and statistical alignment was used to learn two coupled projections to solve the shift between domains.

The deep domain confusion (DDC) method was the first method that introduced the MMD methods to deep networks [23]. It measured and minimized the distance between

cross-domain data distributions to solve the adaptive problem of deep networks. However, DDC had only one adaptation layer in the network. With the domain discrepancy increasing, the transferability of the deep feature decreases significantly. DDC used a single core MMD, and a single fixed core was not the optimal one. To solve the problem, DAN was proposed by designing deep adaptive network architecture [24]. It used three adaptive layers to compute the distance between the source domain and the target domain and used an MK-MMD to match the cross-domain distribution by multiple task-specific CNN layers. Enhanced transferability of task-specific layers did not consider the relationship between two sub-domains in different domains of the same category for DAN.

To enhance feature representation by aligning sub-domains, a deep sub-domain adaptation network [25] was proposed based on the DAN network. It captured fine-grained information for each category and used the local maximum which means discrepancy to estimate the difference in distribution between the two sub-domains. Compared with the global alignment method, sub-domain adaptation makes the source domain and target domain align more accurately within the same category. Long et al. [26] proposed the joint adaptation network. This network can align the joint distribution of multiple domain-specific layers in each domain to maximize the maximum mean discrepancy of the joint distribution. And it was easy to estimate from small batches of samples, which made the source domain and target domain distributions to be more distinguishable. Li et al. [27] replaced the discriminator in generative adversarial network (GAN) with the MMD distance. It matched all statistical orders between the dataset and the model samples by generating a simple sample. It solved the problem that the game process of GAN was not easy to converge and the model collapses. The above methods all assume that the source and the target domain conditional distributions are the same and that the marginal distributions are different. Xu et al. [28] assumed that the conditional distributions of the source domain and the target domain were different. The method used MMD to constrain the extracted features on the source and target domains so that the feature distributions were as similar as possible. The MMD was also used to constrain the softmax classification results on both domains so that the distributions of the two classification results were as equal as possible. It used the data, labels, and pseudo labels of the source and the target domains to make the marginal distribution and conditional distribution as close as possible. Considering the inconsistency between the prior distribution of the source domain label and the target domain, Yan et al. [29] proposed to introduce an auxiliary weight for each category in the source domain to re-weigh the source samples, so that the source domain and the target domain share the same category weight. This method used a weighted MMD model to test whether the target domain was consistent with the pseudo-label generated by the source domain. It reduced the classification impact of the wrong label, and the distribution achieves a better pulling effect. Sanodiya and Yao [30] used MMD to minimize both the marginal and conditional distributions and used parameters to measure the proportion of conditional distribution in the sample. So the samples in the source domain and the target domain that are at the edge of the class or far away from the center of the class are more aggregated. The method also combines relative distances, target domain variance, and Laplace terms to maintain distinguishing information and obtain better inter-class divisibility and intra-class compactness. Zhang et al. [31] proposed a discriminative relevance regularization term (DRR) by combining the MMD loss and DRR into a deep network to obtain a relevance regularization adaptation network.

Xu et al. [32] revealed the nature of model degradation through a large number of observations. It pointed out that the target domain feature norms were much smaller than the

source domain feature norms. The previous methods did not think over this factor. Therefore, the existing statistical disparities may not accurately describe the domain shift. The authors proposed the maximum mean feature norm discrepancy (MMFND) based on the MMD. It described the mean-feature-norm distance between the source domain and the target domain. It achieved a better transfer effect by feature norm adaptation.

3. **Method.** The source classification loss function of the SAFN method is

$$L_y\left(x_i^s, y_i^s, \theta_g, \theta_f, \theta_y\right) = -\sum_{k=1}^{|C_s|} I_{\left[k=y_i^s\right]} \log p_k \tag{1}$$

where x_i^s and y_i^s are the ith input image and corresponding label in the source domain, respectively. The θ_g, θ_f, θ_y are the parameters of the feature extract model, the last but one and last fully connected layers, respectively. C_s is the number of categories. $I[\cdot]$ is a function that is used to determine whether the $\log p_k$ equals y_i^s. $p = softmax\left(F\left(G\left(x_i^s\right)\right)\right) = \left[p_1, \ldots, p_{|C_s|}\right]$. G is the feature extract model. F is a task-specific classifier. The features optimized by the softmax loss function do not have higher intra-class similarities and lower inter-class similarities. Therefore, the distance between similar categories is larger, and the distance between the decision boundaries of classification is smaller. This makes the distance between different categories be smaller, and affects the accuracy.

To solve the problem, we propose to use the L2-norm and scale scaling method to constrict the softmax function, to improve the accuracy. Firstly, we use L2-norm to normalize the feature vector $x = [x_1, x_2, \ldots, x_{C_s}]$ to avoid the larger and smaller features affecting accuracy. Regardless of whether the sample features are large or small, they are fixed on a hypersphere by the L2-norm, such that all samples receive similar attention. That is, averaging over all samples reduces the relative importance of samples with large feature norms, which is equivalent to giving samples smaller feature norms more attention to improving accuracy.

It can be expressed as

$$y = \frac{x_i}{\sqrt{\sum_{i=1}^{D_s} x_i^2}} = \frac{x_i}{\|x_i\|_2} \tag{2}$$

The L2-norm makes all embedded features $\hat{x}_i$ only be mapped to a hypersphere surface with $R = 1$. The hypersphere is too small so that the embedded features can only be distributed on a limited surface area. All embedded features are mixed, and not easy to distinguish. This makes it impossible to gather the embedded features with the same category, and impossible together with the embedded features with different categories. To solve this problem, we introduce a scaling factor α to embedded features. The constriction is expressed as the following:

$$\|f(G(x_i))\|_2 = \alpha \tag{3}$$

The scaling factor can change the size of the hypersphere. Therefore, we use a larger scaling factor to increase the surface area of the hypersphere. The larger surface area can make the embedding features better distribute over the hyperspherical surface. It also has a larger surface area for the embedded features with the same category to coalesce together, and space between embedded features with different categories to keep each other away. Based on (3), we design the improved source classification loss function that is expressed as the following:

$$\begin{cases} \hat{L}_y\left(x_i^s, y_i^s, \theta_g, \theta_f, \theta_y\right) = -\dfrac{1}{n_s} \displaystyle\sum_{(x_i,y_i)\in D_s} \sum_{k=1}^{|C_s|} I_{[k=y_i^s]} \log \frac{\exp\left(W_{y_i}^T(f(G(x_i))) + b_i\right)}{\sum_{j=1}^{C_s} \exp\left(W_{y_j}^T(f(G(x_i))) + b_j\right)} \\ \text{subject to } \|f(G(x_i))\|_2 = \alpha,\ \forall i = 1, 2, \ldots, C_s \end{cases} \tag{4}$$

where the D_s denotes the source domain, and C_s indicates the number of categories. The j denotes the number of elements within the category, and W and b are the weights and biases that can be trained in the network, respectively.

Based on (4), the whole loss function of our proposed method is expressed as the following:

$$\begin{cases} L\left(\theta_g, \theta_f, \theta_F\right) = \hat{L}_y\left(x_i^s, y_i^s, \theta_g, \theta_f, \theta_y\right) + \dfrac{\lambda}{n_s + n_t} \displaystyle\sum_{x_i \in D_s \cup D_t} L_d(h(x_i; \theta_1) + \Delta r, h(x_i; \theta_2)) \\ \text{subject to } \|f(G(x_i))\|_2 = \alpha,\ \forall i = 1, 2, \ldots, C_s \end{cases} \tag{5}$$

where D_t denotes the target domain, n_t denotes an unlabeled sample in the target domain, and $h(x)$ represents a mapping function that is constructed by L2-norm and depth characterization module. It can map the input from the original space to the regenerated Hilbert kernel space. L_d is the 2-norm distance. θ_1 and θ_2 are the model parameters at the last iteration and the current iteration, respectively. Δr denotes the positive residual scalar. It is used to control the feature norm amplification. λ is the hyper-parameter weight.

The network structure of the proposed method is shown in Figure 1. The X_s and X_t are source domain images and the target domain images, respectively. The feature extraction module G is composed of the ResNet network and is used to extract the source domain and the target domain. The F model is a task-specific classifier with fully connected layers. Each layer consists of a fully connected layer, batch normalization layer, ReLU active function, and dropout layer. The proposed L2-normalization layer and scale layer are added between the $(l-1)$th layer and lth layer in F model. L_t is the distance between the two domains after the current iteration update and L_{t+1} is the distance between the domains after the next iteration update. Shared L is the distance between the source domain and the target domain.

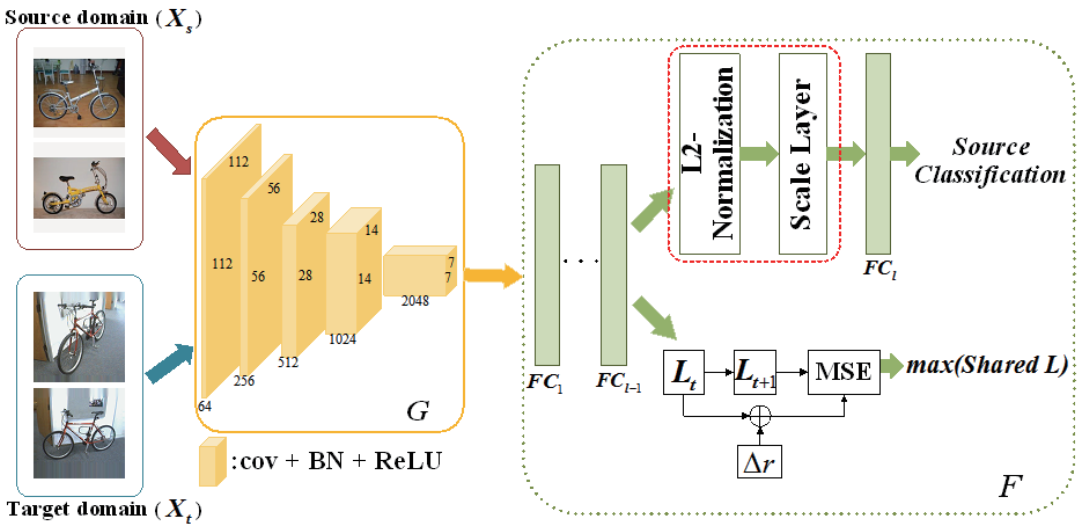

FIGURE 1. Proposed network structure

In our proposed method, the scaling parameter α plays a crucial role in determining the L2-soft maximum loss performance. There are two methods to determine the α. One is to set a fixed scaling parameter valued according to experience, and the other is to obtain the scaling parameter value by network learning. If we use the network learning method to obtain the scaling parameter value, the scaling parameter of network learning will be larger, which leads to the relaxation of L2 constraints. Therefore, we determine the scaling parameter value according to the experiments on three data sets. To test the accuracy with different scaling parameters, we randomly select the task Amazon (A)→DSLR (D) and A→Webcam (W) from the Office-31 dataset, the task P→I from the ImageCLEF-DA dataset, and task Artistic images (Ar)→Real-word images (Rw) from the Office-Home dataset. The accuracies with different scaling parameters are shown in Figure 2. For the task Ar→Rw, when the value of the scaling parameter is 8, the accuracy is the largest. For the task A→D and A→W, when the value of the scaling parameter is 10, the accuracy is the largest. For the task Pascal VOC 2012 (P)→ILSVRC 2012 (I), when the value of the scaling parameter is 16, the accuracy is the largest. The value of the scaling parameter being smaller or larger will reduce the accuracy. When the value of the scaling parameter is larger than 16, the accuracies start to reduce for all tasks. With a scaling parameter of 16, better transfer accuracy was obtained for all tasks. Therefore, we select 16 as the value of the scaling parameter α.

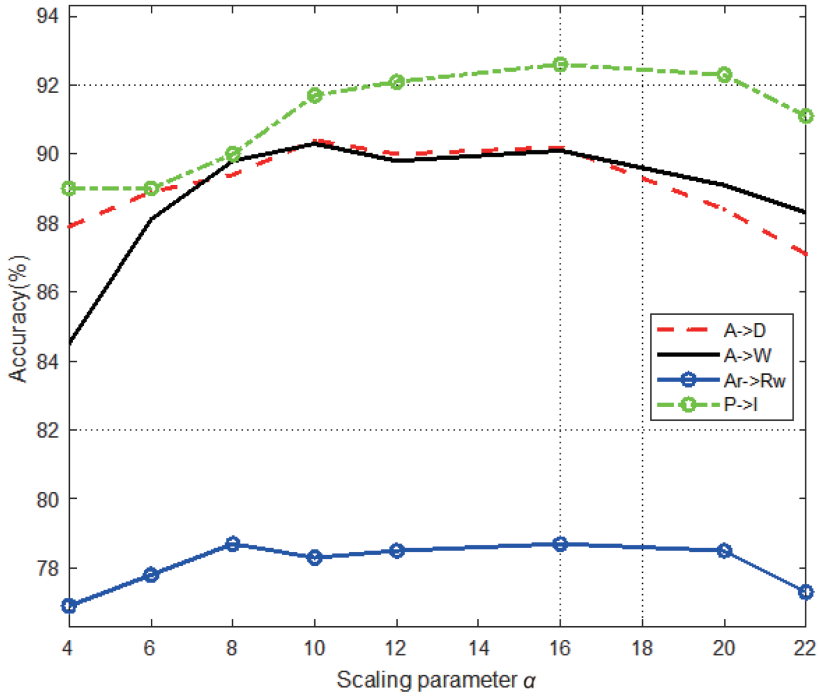

FIGURE 2. Relationship between scaling parameter and accuracy

4. **Simulation and Discussion.** We use three datasets to compare our proposed method with other transfer learning methods that are ResNet-50 [33], DAN [24], DANN [10], JAN [26], CDAN [13], and SAFN [32]. The three datasets are Office-31, ImageCLEF-DA, Office-Home. The Office-31 dataset contains 4652 images with 31 classes. It is a widely-used benchmark for visual domain adaptation. The images are collected from online websites, digital SLR cameras, and webcams. Therefore, the Office-31 dataset is divided into three domains: Amazon (A), Webcam (W), and DSLR (D). All transfer

learning methods are tested in six transfer tasks. A→W, D→W, W→D, A→D, D→A, and W→A. The ImageCLEF-DA dataset contains three public datasets that are Caltech-256 (C), ILSVRC 2012 (I), and Pascal VOC 2012 (P). We compare our proposed method with other methods for the following tasks: I→P, P→I, I→C, C→I, C→P, and P→C. The Office-Home dataset is more complex than the Office-31 and ImageCLEF-DA datasets. This dataset is composed of four domains and 65 classes with 15500 images. The four domains are Artistic images (Ar), Clip Art (CI), Product images (Pr), and Real-word images (Rw). These four domains are less similar to one another than those in the other two datasets. We test our proposed method alongside the other methods with this dataset for 12 tasks: Ar→CI, Ar→Pr, Ar→Rw, CI→Ar, CI→Pr, CI→Rw, Pr→Ar, Pr→CI, Pr→Rw, Rw→Ar, Rw→CI, and Rw→Pr.

We follow a standard protocol where the source domain samples are labeled and the target domain samples are unlabeled. We use a uniform set of hyperparameters in the Office-31, ImageCLEF-DA, and Office-Home datasets. For a fairer comparison, the same set of hyperparameters is chosen for our method as for the competitor's method. So, all of the parameters are set the same as SAFN method, where $\lambda = 0.05$, $\Delta r = 1.0$, batch size = 32, weight decay = 0.0005, momentum = 0.9, and learning rate = 0.001. The operating system is Ubuntu 18.04. The CPU is Intel Xeon E5-2678 v3 with 2.5 GHZ main frequencies. The GPU is NVIDIA GeForce GTX 1080Ti. The basic network used in the experiments is ResNet-50, and the deep learning framework is PyTorch.

Table 1 shows the accuracy results of our proposed method and those of other domain adaptation methods on the Office-31 dataset. The accuracies of the proposed method are 1.4% in task A→W, 0.7% in task D→W, 2.8% in task A→D, 2.9% in task D→A, and 2.3% in task W→A, in each case higher than that of SAFN. In the task W→D, the two methods have the same accuracy. The accuracies of the proposed method are 0.9% in task D→W, 0.7% in task A→D, 2.6% in task D→A, and 4.0% in task W→A, in each case higher than that of CDAN. In tasks A→W and W→D, the accuracies of the proposed method are lower than those of CDAN. In all tasks, the accuracies of the proposed method are higher than that of other methods except for SAFN and CDAN. Although the accuracies of the proposed method are lower than that of the CDAN method in tasks A→W and W→D, the accuracies of the proposed method are higher than those of the CDAN method in the other four tasks. The proposed method has the highest average accuracy (87.3%), followed by CDAN (86.6%), then SAFN (85.7%).

The CDAN requires to compute the entropies and uses entropies as weights. The weights are not learned by the network itself. It requires artificially adjusting the weights before using it. If the adjusted weights are not suitable, the accuracy is lower. The

Table 1. Accuracy (%) on Office-31 for unsupervised domain adaptation (ResNet-50)

Method	**A→W**	**D→W**	**W→D**	**A→D**	**D→A**	**W→A**	**Avg**
ResNet-50	68.4±0.2	96.7±0.1	99.3±0.1	68.9±0.2	62.5±0.3	60.7±0.3	76.1
DAN	80.5±0.4	97.1±0.2	99.6±0.1	78.6±0.2	63.6±0.3	62.8±0.2	80.4
DANN	82.0±0.2	96.9±0.2	99.1±0.1	79.7±0.4	68.2±0.4	67.4±0.5	82.2
JAN	85.4±0.3	97.4±0.2	99.8±0.2	84.7±0.3	68.6±0.3	70.0±0.4	84.3
CDAN	93.1±0.5	98.2±0.2	100±0.0	89.8±0.3	70.1±0.4	68.0±0.4	86.6
SAFN	88.8±0.4	98.4±0.0	99.8±0.0	87.7±1.3	69.8±0.5	69.7±0.2	85.7
Proposed method	**90.2±0.7**	**99.1±0.0**	**99.8±0.0**	**90.5±0.4**	**72.7±0.3**	**72.0±0.3**	**87.3**

SAFN and our proposed method are parameter-free and more suitable for real applications. Besides, the CDAN method uses the generative adversarial network. It makes the network structure more complex. The network of SAFN and our proposed method are more lightweight than CDAN.

Table 2 shows the accuracy of the proposed method and that of other methods for the ImageCLEF-DA dataset. It can be seen that the proposed method exhibits the best performance in all tasks. The accuracies of the proposed method are 1.1% higher than that of SAFN for task I→P, 0.9% higher for task P→I, 1.4% higher for task I→C, 1.2% higher for task C→I, and 1.3% higher for task C→P, 1.1% higher for task P→C. For the ImageCLEF-DA dataset, the average accuracy of our proposed method is 89.3%, which is higher than that of SAFN (88.1%). The proposed method has the highest average accuracy, followed by SAFN, then CDAN. With the ImageCLEF-DA dataset, the average accuracy of our proposed method is 89.3%, which is 1.2% higher than that of SAFN. These results show that the proposed method has better transfer capability on the ImageCLEF-DA dataset.

TABLE 2. Accuracy (%) on ImageCLEF-DA for unsupervised domain adaptation (ResNet-50)

Method	**I→P**	**P→I**	**I→C**	**C→I**	**C→P**	**P→C**	**Avg**
ResNet-50	74.8±0.3	83.9±0.1	91.5±0.3	78.0±0.2	65.5±0.3	91.2±0.3	80.7
DAN	74.5±0.4	82.2±0.2	92.8±0.2	86.3±0.4	69.2±0.4	89.8±0.4	82.5
DANN	75.0±0.6	86.0±0.3	96.2±0.4	87.0±0.5	74.3±0.5	91.5±0.6	85.0
JAN	76.8±0.4	88.0±0.2	94.7±0.2	89.5±0.3	74.2±0.3	91.7±0.3	85.8
CDAN	76.7±0.3	90.6±0.3	97.0±0.4	90.5±0.4	74.5±0.3	93.5±0.4	87.1
SAFN	78.0±0.4	91.7±0.5	96.2±0.1	91.1±0.3	77.0±0.5	94.7±0.3	88.1
Proposed method	**79.1±0.3**	**92.6±0.4**	**97.6±0.1**	**92.3±0.6**	**78.3±0.4**	**95.8±0.1**	**89.3**

Table 3 shows the accuracy results for the proposed method and the ResNet-50, DAN, DANN, JAN, CDAN, and SAFN methods on the Office-Home dataset. For this dataset, the proposed method also has the highest accuracy results, followed by SAFN. The accuracies of the proposed method are 1.8% higher than SAFN for task Ar→CI, 2.4% higher for task Ar→Pr, 2.4% higher for task Ar→Rw, 2.1% higher for task CI→Ar, 1.2% higher for task CI→Pr, 0.7% higher for task CI→Rw, 1.9% higher for task Pr→Ar, 1.8% higher for task Pr→CI, 2.8 higher for task Pr→Rw, 1.8% higher for task Rw→Ar, 4.7% higher for task Rw→CI and 1.4% higher for task Rw→Pr. With the Office-Home dataset, the average accuracy of our proposed method is 69.2%, which is 1.9% higher than that of SAFN.

Based on the analyses of Table 1, Table 2, and Table 3, the proposed method has the highest average accuracy than ResNet-50, DAN, DANN, JAN, CDAN, and SAFN methods on the Office-Home dataset, ImageCLEF-DA dataset, and Office-31 dataset. It means that the proposed method has a better transfer capability than other methods.

The L2-normalization can reduce the differences between samples. The mode length difference is larger between samples. Although the differences between samples were reduced when we use the L2-normalization to deal with the features of the same dimension, it does not change the relative size of the feature norm. Samples with larger feature norms are easy to classify. It makes samples with smaller feature norms get more attention and are more easily classified. The efficiency of classification learning is improved for samples in the target domain with the small feature norm.

TABLE 3. Accuracy (%) on Office-Home for unsupervised domain adaptation (ResNet-50)

Method	Ar→ Cl	Ar→ Pr	Ar→ Rw	Cl→ Ar	Cl→ Pr	Cl→ Rw	Pr→ Ar	Pr→ Cl	Pr→ Rw	Rw→ Ar	Rw→ Cl	Rw→ Pr	Avg
ResNet-50	38.6	60.9	58.0	75.2	39.9	48.1	52.9	31.0	70.8	65.4	41.8	70.4	53.7
DAN	43.6	57.0	67.9	45.8	56.5	60.4	44.0	43.6	67.7	63.1	51.5	74.3	56.3
DANN	45.6	59.3	70.1	47.0	58.5	60.9	46.1	43.7	68.5	63.2	51.8	76.8	57.6
JAN	45.9	61.2	68.9	50.4	59.7	61.0	45.8	43.4	70.3	63.9	52.4	76.8	58.3
CDAN	49.0	69.3	74.5	54.4	66.0	68.4	55.6	48.3	75.9	68.4	55.4	80.5	63.8
SAFN	52.0	71.7	76.3	64.2	69.9	71.9	63.7	51.4	77.1	70.9	57.1	81.5	67.3
Proposed method	**53.8**	**74.1**	**78.7**	**66.3**	**71.1**	**72.6**	**65.6**	**53.2**	**79.9**	**72.7**	**61.8**	**82.9**	**69.2**

5. **Conclusions.** In this paper, an improved method is proposed that is based on the SAFN method. It adds the L2-normalization layer and the scaling layer before the last fully connected layer of the classifier. It makes the smaller feature norms get more attention. It also reduces the distance between the samples in the same category and increases the distances between different categories. On the Office-31 dataset, ImageCLEF-DA dataset, and Office-Home dataset, the proposed method still has the highest average accuracy than ResNet-50, DAN, DANN, JAN, CDAN, and SAFN methods.

This method only considers inter-class and intra-class discrepancies between samples and ignores the inter-domain norm discrepancies. It also affects the domain shift performance. In the future, we will introduce the inter-domain norm to close the distance between the source domain and target domain. It can make the boundaries of classification decisions in the target domain be clearer.

Acknowledgment. This research has been funded by the National Natural Science Foundation of China (61271115), Scientific and Technological Developing Scheme of Jilin Province (YDZJ202101ZYTS172), Research Foundation of Education Bureau of Jilin Province (JJKH20210042KJ, JJKH20210095KJ) and Doctoral Scientific Research Foundation of Beihua University (20171424).

REFERENCES

[1] R. Pramanik and S. Bag, Handwritten Bangla city name word recognition using CNN-based transfer learning and FCN, *Neural Computing and Applications*, vol.22, pp.1-13, 2021.

[2] Y. J. Wang, P. P. Cao, X. S. Wang and X. Y. Yang, Research on insulator self explosion detection method based on deep learning, *Journal of Northeast Electric Power University*, vol.40, no.3, pp.33-40, 2020.

[3] Y. Chen, A transfer learning model with multi-source domains for biomedical event trigger extraction, *BMC Genomics*, vol.22, no.1, pp.1-18, 2021.

[4] A. R. Badanidiyoor and G. K. Naravi, $\theta(1)$ time complexity parallel local binary pattern feature extractor on a graphical processing unit, *ICIC Express Letters*, vol.13, no.9, pp.867-874, 2019.

[5] M. A. Tabak, M. S. Norouzzadeh, D. W. Wolfson et al., Improving the accessibility and transferability of machine learning algorithms for identification of animals in camera trap images: MLWIC2, *Ecology and Evolution*, vol.10, no.19, pp.10374-10383, 2020.

[6] S. J. Pan and Q. Yang, A survey on transfer learning, *IEEE Transactions on Knowledge and Data Engineering*, vol.22, no.10, pp.1345-1359, 2010.

[7] L. Shao, F. Zhu and X. Li, Transfer learning for visual categorization: A survey, *IEEE Transactions on Neural Networks and Learning Systems*, vol.26, no.5, pp.1019-1034, 2015.

[8] M. Wang and W. Deng, Deep visual domain adaptation: A survey, *Neurocomputing*, vol.312, pp.135-153, 2018.

[9] Y. Ganin and V. Lempitsky, Unsupervised domain adaptation by backpropagation, *International Conference on Machine Learning*, vol.2015, pp.1180-1189, 2015.

[10] Y. Ganin, E. Ustinova, H. Ajakan et al., Domain-adversarial training of neural networks, *Journal of Machine Learning Research*, vol.17, no.1, pp.1-35, 2016.
[11] K. Bousmalis, G. Trigeorgis, N. Silberman et al., Domain separation networks, *Proc. of the 30th International Conference on Neural Information Processing Systems*, New York, USA, pp.343-351, 2016.
[12] Z. Pei, Z. Cao, M. Long and J. Wang, Multi-adversarial domain adaptation, *Proc. of the AAAI Conference on Artificial Intelligence*, LA, USA, vol.32, no.1, pp.3934-3941, 2018.
[13] M. Long, J. Cao and M. Jordan, Conditional adversarial domain adaptation, *Proc. of the 32nd International Conference on Neural Information Processing Systems*, New York, USA, pp.1647-1657, 2018.
[14] K. Saito, K. Watanabe, Y. Ushiku et al., Maximum classifier discrepancy for unsupervised domain adaptation, *Proc. of the IEEE Conference on Computer Vision and Pattern Recognition*, Salt Lake City, USA, pp.3723-3732, 2018.
[15] Z. Cao, L. Ma, M. Long et al., Partial adversarial domain adaptation, *Proc. of the European Conference on Computer Vision (ECCV)*, Munich, Germany, pp.135-150, 2018.
[16] A. Gretton, K. M. Borgwardt, M. Rasch et al., A kernel two-sample test, *Journal of Machine Learning Research*, vol.13, no.1, pp.723-773, 2012.
[17] S. J. Pan, I. W. Tsang et al., Domain adaptation via transfer component analysis, *IEEE Transactions on Neural Networks*, vol.22, no.2, pp.199-210, 2011.
[18] M. Long, J. Wang, G. Ding et al., Transfer feature learning with joint distribution adaptation, *Proc. of the IEEE International Conference on Computer Vision*, Sydney, NSW, Australia, pp.2200-2207, 2013.
[19] M. Long, J. Wang, G. Ding et al., Adaptation regularization: A general framework for transfer learning, *IEEE Transactions on Knowledge and Data Engineering*, vol.26, no.5, pp.1076-1089, 2014.
[20] C. Hou, A. Yeh and Y. Wang, An unsupervised domain adaptation approach for cross-domain visual classification, *Proc. of the 12th IEEE International Conference on Advanced Video and Signal Based Surveillance*, Karlsruhe, Germany, pp.1-6, 2015.
[21] J. Tahmoresnezhad and S. Hashemi, Visual domain adaptation via transfer feature learning, *Knowledge & Information Systems*, vol.50, no.2, pp.585-605, 2017.
[22] J. Zhang, W. Li and P. Ogunbona, Joint geometrical and statistical alignment for visual domain adaptation, *Proc. of the IEEE Conference on Computer Vision and Pattern Recognition*, HI, USA, pp.1859-1867, 2017.
[23] E. Tzeng, J. Hoffman, N. Zhang et al., Deep domain confusion: Maximizing for domain invariance, *arXiv Preprint*, arXiv: 1412.3474, 2014.
[24] M. Long, Y. Cao, J. Wang et al., Learning transferable features with deep adaptation networks, *Proc. of the 32nd International Conference on Machine Learning, JMLR*, Lille, France, vol.37, pp.97-105, 2015.
[25] Y. Zhu, F. Zhuang, J. Wang et al., Deep subdomain adaptation network for image classification, *IEEE Transactions on Neural Networks and Learning Systems*, vol.32, no.4, pp.1713-1722, 2021.
[26] M. Long, H. Zhu, J. Wang et al., Deep transfer learning with joint adaptation networks, *Proc. of the 34th International Conference on Machine Learning, JMLR*, Sydney, Australia, vol.70, pp.2208-2217, 2017.
[27] Y. Li, K. Swersky and R. Zemel, Generative moment matching networks, *Proc. of the 32nd International Conference on Machine Learning, JMLR*, Lille, France, vol.37, pp.1718-1727, 2015.
[28] Z. Xu, F. X. Yu, S. F. Chang et al., Deep transfer network: Unsupervised domain adaptation, *arXiv Preprint*, arXiv: 1503.00591, 2015.
[29] H. Yan, Y. Ding, P. Li et al., Mind the class weight bias: Weighted maximum mean discrepancy for unsupervised domain adaptation, *Proc. of the IEEE Conference on Computer Vision and Pattern Recognition*, HI, USA, pp.2272-2281, 2017.
[30] R. K. Sanodiya and L. Yao, Unsupervised transfer learning via relative distance comparisons, *IEEE Access*, vol.8, pp.110290-110305, 2020.
[31] W. Zhang, X. Zhang, L. Lan et al., Enhancing unsupervised domain adaptation by discriminative relevance regularization, *Knowledge & Information Systems*, vol.62, no.9, pp.3641-3664, 2020.
[32] R. J. Xu, G. B. Li, J. H. Yang et al., Larger norm more transferable: An adaptive feature norm approach for unsupervised domain adaptation, *Proc. of the 12th International Conference on Computer Vision Systems*, Thessaloniki, Greece, pp.1426-1435, 2019.
[33] K. He, X. Zhang, S. Ren et al., Deep residual learning for image recognition, *Proc. of the IEEE Conference on Computer Vision and Pattern Recognition*, Las Vegas, NV, USA, pp.770-778, 2016.