

Article

# A New Method of Image Classification Based on Domain Adaptation

Fangwen Zhao <sup>1</sup>, Weifeng Liu <sup>1</sup> and Chenglin Wen <sup>2,\*</sup>

<sup>1</sup> School of Electrical and Control Engineering, Shaanxi University of Science and Technology, Xi'an 710021, China; 200611017@sust.edu.cn (F.Z.); liuwf@sust.edu.cn (W.L.)

<sup>2</sup> School of Automation, Guangdong University of Petrochemical Technology, Maoming 525000, China

\* Correspondence: wenc1@gdupt.edu.cn

**Abstract:** Deep neural networks can learn powerful representations from massive amounts of labeled data; however, their performance is unsatisfactory in the case of large samples and small labels. Transfer learning can bridge between a source domain with rich sample data and a target domain with only a few or zero labeled samples and, thus, complete the transfer of knowledge by aligning the distribution between domains through methods, such as domain adaptation. Previous domain adaptation methods mostly align the features in the feature space of all categories on a global scale. Recently, the method of locally aligning the sub-categories by introducing label information achieved better results. Based on this, we present a deep fuzzy domain adaptation (DFDA) that assigns different weights to samples of the same category in the source and target domains, which enhances the domain adaptive capabilities. Our experiments demonstrate that DFDA can achieve remarkable results on standard domain adaptation datasets.

**Keywords:** domain adaptation; unsupervised learning; maximum mean discrepancy



**Citation:** Zhao, F.; Liu, W.; Wen, C. A New Method of Image Classification Based on Domain Adaptation. *Sensors* **2022**, *22*, 1315. <https://doi.org/10.3390/s22041315>

Academic Editor: Gregorij Kurillo

Received: 28 December 2021

Accepted: 29 January 2022

Published: 9 February 2022

**Publisher's Note:** MDPI stays neutral with regard to jurisdictional claims in published maps and institutional affiliations.



**Copyright:** © 2022 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<https://creativecommons.org/licenses/by/4.0/>).

## 1. Introduction

In recent years, deep learning has achieved great success in computer vision [1] and natural language processing [2] tasks. Unfortunately, satisfactory performance gains come only when large amounts of labeled data are available for supervised training. In practice, it is often time-consuming and expensive to collect enough labeled data or even impossible in some cases, which limits the specific applications of deep neural networks. For a target domain with large amounts of unlabeled data, one natural idea is to transfer the neural network from a domain with richly labeled data to a domain with a shortage of labeled samples. For example, in the fields of medical image diagnosis [3,4] and fault diagnosis [5] where data is scarce, transfer learning is a powerful tool to solve such problems.

Traditional machine-learning algorithms assume that the training data and test data obey the same distribution; however, due to dataset bias, the data distribution on different domains often differs. The generalization ability of the algorithm will be weakened if the statistical distribution of samples is changed. The main idea of domain adaptation [6] is to use a large number of labeled samples from existing domains to facilitate the learning of new tasks by reducing the dataset bias on the target domain.

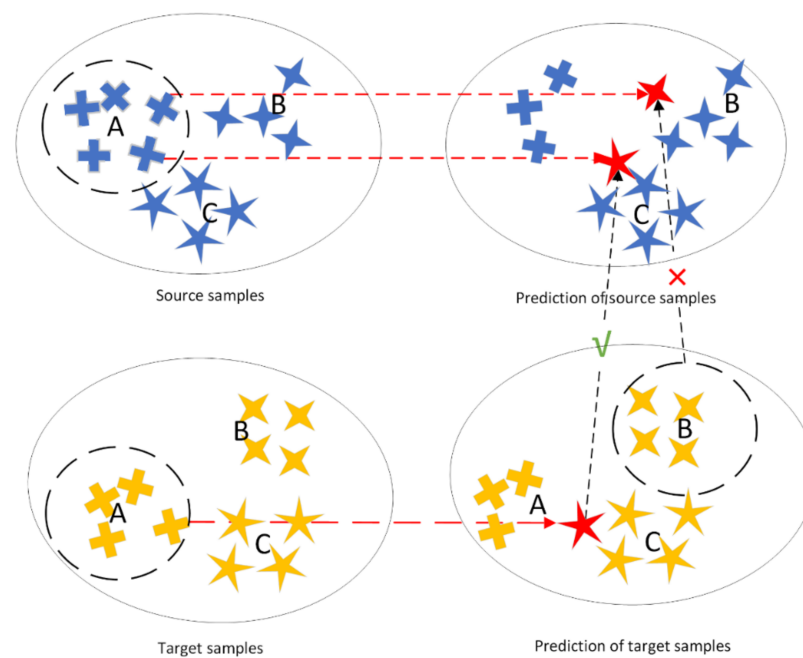
There are three common approaches to domain adaptation [6,7]: feature-based domain adaptation, instance-based domain adaptation, and classifier-based domain adaptation. The feature-based approach is a typical strategy for domain adaptation [8–11] and aims to learn a common feature representation by minimizing the distribution discrepancy between domains. The instance weighting method is another simple yet effective domain adaptive approach [12–14].

Some instances are selected from the source domain so that the probability distribution of the obtained sample subset is similar to the target sample and are then trained using traditional machine-learning methods. Apart from this, the classifier-based adaptation

makes the source domain model adapt to the target by regularizing the difference between the source domain and target domain model parameters [15–17]. Since the label of the target domain is required, this usually limits its scope of use.

Previous feature-based domain adaptation methods have mainly used explicit distances to align feature distributions between source and target domains [18–20] or minimized the distribution difference between domains by adversarial learning [21–23]. These methods align the global source and target domain distribution without considering the category information of the domain samples, which causes some discriminable local information and structures to be confused. Recently, researchers are increasingly investigating subdomain adaptation [24–26], which performs local alignment of the source and target domains in the feature space by introducing label information to the neural network. These subdomain-based alignment methods have gained a considerable degree of performance improvement due to capturing the fine-grained information of each category.

Although these aforementioned methods achieved good performance, there is no sample selection process, i.e., alignment of all samples in the same category, which may affect the final results. A straightforward example is shown in Figure 1. During the training period, samples from the source domain category A may be misclassified into the categories B or C after being predicted by the network. If the target domain is aligned with the same category on the source domain, the following incongruities will occur.



**Figure 1.** Matching errors may occur when subcategory alignment is performed directly.

For samples with the true label A in the source domain that are misclassified as B or C, it may happen to be aligned with a sample in the target domain that is also misclassified as C, but is more likely to be mismatched with several samples with the true label B. This confusion degrades the ability of domain adaptation. For more accurate alignment, we propose a fuzzy maximum mean discrepancy (FMMD), which considers the predicted information of instances in the source domain to alleviate this problem. Based on this, we design a new network architecture Deep Fuzzy Domain Adaption (DFDA) to apply FMMD, and DFDA can be easily optimized by the standard gradient descent method. The experimental results show that our method outperforms state-of-the-art metric-based approaches on benchmark datasets.

## 2. Related Work

Before discussing the FMMD, in this section, we introduce three related aspects of work: feature-based domain adaptation, subdomain adaptation, and maximum mean discrepancy (MMD).

### 2.1. Feature-Based Domain Adaptation

Research [27] has shown that simple transfer learning methods, such as pretrain-finetune can reduce training time but only slightly improve the result. Feature-based domain adaptation embeds adaptive modules into neural networks to reduce the distribution differences between domains and significantly improve the performance of neural networks on the target domain. There are mainly two methods: adversarial learning and statistical moment matching. The domain adaptation method based on adversarial learning brings the idea of GAN (Generative adversarial networks) [28] to the problem of domain adaptation.

The discriminator learns to distinguish between the source domain and target domain features, and the generator learns feature representations with domain invariance to confuse the discriminator. Previous works [24,25,29] have achieved good results. The method based on statistical moment matching measures the difference in distance between domains in terms of the mean or higher order moments and minimizes the difference as much as possible.

MMD reduces the mean between domains in the Reproducing Kernel Hilbert Space (RKHS), CORAL reduces the difference between the covariance matrices the two domains, and CMD (Central moment discrepancy) [30] aligns the higher-order central moments of two domain distributions to perform domain adaptation. Many improved algorithms are subsequently proposed on this basis. Most state-of-the-art methods are domain adversarial net-based adaptation methods. However, domain adversarial networks are often difficult to converge during training, and our method based on the MMD metric also achieves remarkable results.

### 2.2. Subdomain Adaptation

Some recent approaches have improved the performance of domain adaptation by introducing category information into the network. CDAN (Conditional adversarial domain adaptation) [24] conditions the adversarial adaptation model based on the discriminative information in the classifier predictions. MADA (Multi-adversarial domain adaptation) [22] uses a category discriminator for each category, capturing the multimodal structure and fine-grained alignment of the data distribution between domains.

A joint adaptation network (JAN) [31] combines the joint probability distributions of features and labels, aligning both the marginal and conditional distributions of the two domains. A DTN (deep transfer network) [32] uses the output of the discriminator to calculate the conditional MMD and aligns the conditional probability distribution by minimizing the conditional MMD. Deep subdomain adaptation network (DSAN) [26] reduces the MMD distance between samples of the same category on the source and target domains based on the label.

Inspired by the concept of the instance-based transfer method and the different above-mentioned methods, we assign different weights to samples of the same category between domains by adding the predicted value of the samples and selecting some suitable samples for fine alignment. These methods perform domain feature alignment based on the hard pseudo-labeling of samples by the classifier, which will undoubtedly lead to improper alignment due to the presence of noise in these labels.

### 2.3. Maximum Mean Discrepancy

MMD calculates the mean distance of the distribution in the RKHS as the distribution discrepancy between the two domains, and its effectiveness has been proven [33]. There is considerable research around MMD, and the first method to apply MMD to transfer learning was Transfer Component Analysis (TCA) [34]. Deep Domain Confusion (DDC) [35] first

adds the MMD to the loss of the deep network feature layer, Deep Adaptation Network (DAN) [36] replaces the MMD with MK-MMD and adapts it in multiple network layers.

Weight MMD (WMMD) [37] alleviates the effect of category imbalance on domain adaptation through class prior distributions. A joint adaptation network (JAN) [31] uses MK-MMD to simultaneously align the marginal distribution and conditional distribution of the two domains. Dynamic Distribution Adaptation (DDA) [38] quantitatively calculates the variability of marginal distribution and conditional distribution between domains based on JAN, and the performance is greatly improved. MMD has become the most popular distance in transfer learning, and DFDA is also based on the MMD approach.

### 3. Methodology

In the unsupervised domain adaptive scenario, we are given a source domain  $D_s = \{x_i^s, y_i^s\}_{i=1}^{n_s}$  with  $n_s$  labeled samples whose labels  $y_i^s \in \mathcal{Y}_s$ , and target domain  $D_t = \{x_j^t\}_{j=1}^{n_t}$  has  $n_t$  unlabeled samples. Suppose the source and target domains have the same feature space, i.e.,  $\mathcal{X}_s = \mathcal{X}_t$ , and the same category space  $\mathcal{Y}_s = \mathcal{Y}_t$ .  $D_s$  and  $D_t$  are sampled from different data distributions  $\mathcal{P}$  and  $\mathcal{Q}$ , respectively, and  $\mathcal{P} \neq \mathcal{Q}$ . The goal of domain adaptation is how to use the source domain data to build a learner to predict the label of the target domain.

Deep neural networks can learn more transfer representations than traditional machine learning handcrafted features. Several popular deep transfer methods have emerged by adding adaptation layers to neural networks to align the distribution of features between domains [39,40]. These methods reduce the distribution discrepancy between domains by minimizing the distance between the source and target domain features after domain-invariant representations have been extracted by the neural network. However, these methods mainly learn a global domain shift without considering category information to align with subcategories.

DSAN introduces the category information of the source domain and the predicted value information of the target domain into the network and aligns each category separately, which achieves remarkable results on both object recognition tasks and digit classification tasks. However, its assumption that every sample of the same class belongs to class  $c$  with the same probability  $\omega^c$ , as shown in Figure 1, is unreasonable in some cases. We borrow the idea of an instance-based method to increase the feature weight that is beneficial to the target classification task, decrease the weights of features that are unhelpful to the target classification, and give different weights to each feature to distinguish the importance of the sample, which achieves better performance in the benchmark dataset.

#### 3.1. Fuzzy Maximum Mean Discrepancy

Formally, the MMD between distributions  $P$  and  $Q$  is defined as

$$\text{MMD}(H_k, P, Q) := \sup_{\|f\|_{H_k} \leq 1} \mathbb{E}_{X \sim P}[\varphi(x_i^s)] - \mathbb{E}_{Y \sim Q}[\varphi(x_j^t)] \quad (1)$$

where  $H_k$  is the reproducing kernel Hilbert space (RKHS) endowed with characteristic kernel  $k(\cdot, \cdot)$ .  $\mathbb{E}[\cdot]$  denotes the mean of the embedded samples, and  $\varphi(\cdot)$  denotes some feature map to map the original samples to RKHS. To calculate this difference, an unbiased estimate of MMD is obtained by calculating the squared distance between empirical kernel mean embedding on the sample  $X$  and  $Y$  instead of sample expectation.

$$\widehat{\text{MMD}}(H_k, P, Q) = \left\| \frac{1}{n_s} \sum_{x_i^s \in D_s} \varphi(x_i^s) - \frac{1}{n_t} \sum_{x_j^t \in D_t} \varphi(x_j^t) \right\|_H^2 \quad (2)$$



DSAN introduces label information to the network and proposes the Local Maximum Mean Discrepancy (LMMD) to align the distributions of the relevant subdomains within the same category in the source and target domains.

$$\widehat{\text{LMMD}}(H_k, P, Q) = \frac{1}{C} \sum_{c=1}^C \left\| \sum_{x_i^s \in D_s} \omega_i^{sc} \varnothing(x_i^s) - \sum_{x_j^t \in D_t} \omega_j^{tc} \varnothing(x_j^t) \right\|_H^2 \quad (3)$$

where  $\omega_i^{sc}$  and  $\omega_j^{tc}$  denote the weight of  $x_i^s$  and  $x_j^t$  belonging to class  $c$ , respectively. Assuming that the weight of each sample in the same category is equal:

$$\omega_i^c = \frac{1}{\sum_{(y_i \in C)} y_i} \quad (4)$$

However, there are gaps in quality between samples, and the alignment of outliers can easily become confusing. In particular, when a category is too sparse in a batch, the sample weight is prone to unreasonable assignments. To address this issue, we propose the FMMD as follows:

$$\widehat{\text{FMMD}}(H_k, P, Q) = \frac{1}{C} \sum_{c=1}^C \left\| \sum_{x_i^s \in D_s} \omega_i^{sc} \hat{y}_i \varnothing(x_i^s) - \sum_{x_j^t \in D_t} \omega_j^{tc} \varnothing(x_j^t) \right\|_H^2 \quad (5)$$

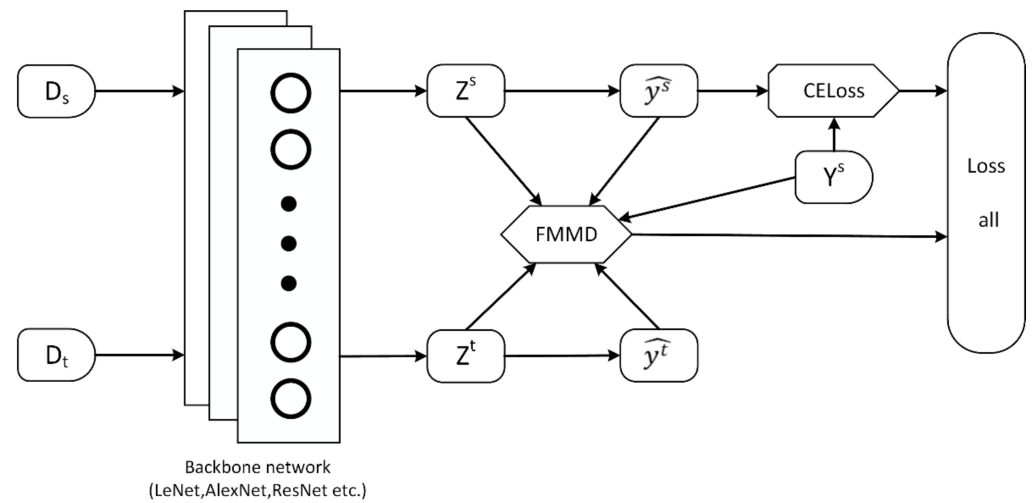
where  $\hat{y}_i$  is the output of the source domain samples after the neural network, which represents the confidence that  $x_i$  belongs to class  $c$ . It is used to evaluate the quality of the features here. If the predicted value of the sample is closer to the label value, it means that it is a good sample and increases the weight of the sample. On the contrary, if the predicted value deviates from the true value, the feature weight of this sample will decrease. As in DSAN, the weights on the target domain sample are calculated using the predicted values instead of the true label.

### 3.2. Deep Fuzzy Domain Adaption

We propose a new network architecture Deep Fuzzy Domain Adaption (DFDA) in order to embed FMMD into the network. Different from the previous method that only uses the label of the source domain, the predicted information of the source domain samples are also added to the network. As shown in Figure 2, DFDA can be trained end-to-end by the standard stochastic gradient descent method. The loss of the entire network is as follows:

$$\ell_{\text{loss}} = \ell_C(x^s, y^s) + \lambda \mathcal{D}(z^s, z^t, y^s, \hat{y}^s, \hat{y}^t) \quad (6)$$

where  $\ell_C(x^s, y^s)$  denotes the loss of source domain data in the neural network. As  $\ell_C(x^s, y^s)$  becomes smaller, the accuracy of the source domain is constantly improving.  $\mathcal{D}(z^s, z^t, y^s, \hat{y}^s, \hat{y}^t)$  denotes the FMMD metric function;  $z^s$  and  $z^t$  denote the feature vectors output by  $x^s$  and  $x^t$  via the neural network, respectively; and  $\hat{y}^s$  and  $\hat{y}^t$  denote the predicted results of the samples on the source and target domains, respectively. As the feature divergence between the target and source domains becomes smaller, the accuracy of target domain prediction increases with the increase of the source domain sample accuracy.



**Figure 2.** The architecture of DFDA, where CELoss is the cross-entropy loss function.

### 3.3. Theoretical Analysis

We analyzed the effectiveness of DFDA based on the domain adaptation theory [39,41].

**Theorem 1.** Let  $\mathcal{H}$  be a hypothesis space of VC dimension  $d$ . Given two domains  $S$  and  $T$ , then, for any  $\delta \in (0, 1)$  with probability at least  $1 - \delta$ , for every  $h \in \mathcal{H}$ :

$$\epsilon_T(h) \leq \epsilon_S(h) + \frac{1}{2}d_{\mathcal{H}\Delta\mathcal{K}}(S, T) + \lambda \quad (7)$$

$\epsilon_S(h)$  and  $\epsilon_T(h)$  are the empirical errors in the source and target domains, respectively, where  $\epsilon_S(h)$  can be easily minimized by the label information of the source domain sample.  $\lambda = \epsilon_S(h^*) + \epsilon_T(h^*)$  denotes the combined error of the ideal hypothesis, where  $h^* = \operatorname{argmin}_{h \in \mathcal{H}} \epsilon_S(h) + \epsilon_T(h)$  represents an ideal joint hypothesis that achieves the minimum combined errors on both the source and target domains. If combined error is large, there will not be a classifier that performs well in both the source and target domains; therefore, we usually assume that  $\lambda$  is a relatively small and negligible value.

In this case, the second term,  $d_{\mathcal{H}\Delta\mathcal{K}}(S, T)$ , which represents the distribution difference between the source domain and the target domain, is an important component to constrain the error bounds in the target domain. DFDA aligns relatively high-quality samples so that the target domain and the source domain's similar samples are more accurately matched. Compared with the previous method,  $d_{\mathcal{H}\Delta\mathcal{K}}(S, T)$  will become smaller. In summary, utilizing the prediction of the source samples in unsupervised domain adaptation is effective.

## 4. Experiment

This section mainly introduces the data sets and experimental environment used in the experiment and describes in detail the analysis. We evaluate our algorithm on three widely used benchmark datasets, including the Office31, Office Home, and large-scale digital recognition datasets, and compare it with several state-of-the-art distance metric-based transfer learning models: DDC, DAN, JAN, Deep CORAL, and DSAN, to assess the effectiveness of our approach.

### 4.1. Setup

Office31 [42] is one of the most widely used datasets for the domain adaptation, and it contains three domains: Amazon (A), Webcam (W), and DSLR (D), with 2817, 498, and 795 samples, respectively, with each including 31 object classes. The samples in Amazon

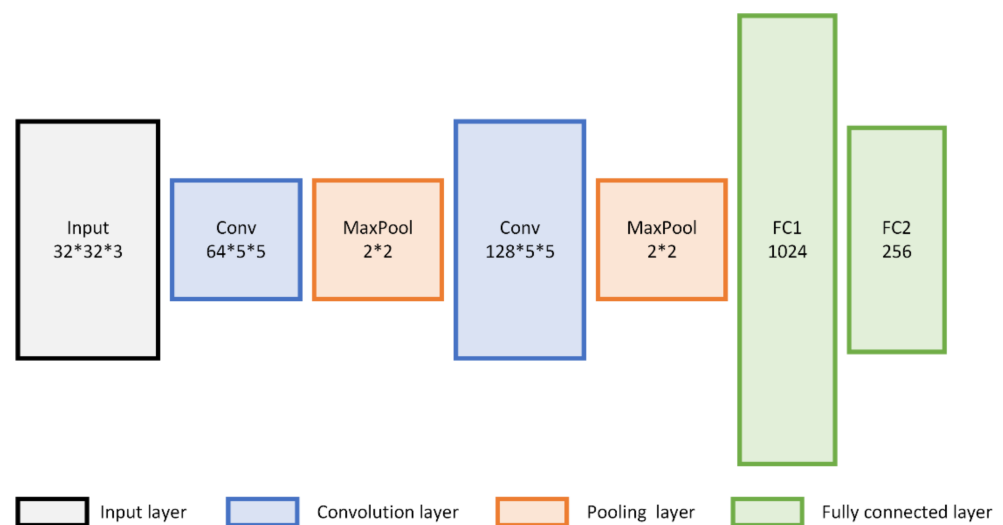
are downloaded from the Amazon website, the samples in Webcam are low-resolution images taken by surveillance equipment, and the samples in DSLR are high-resolution images taken by SLR cameras. We evaluate all methods across all six tasks  $A \rightarrow W$ ,  $W \rightarrow A$ ,  $W \rightarrow D$ ,  $D \rightarrow W$ ,  $A \rightarrow D$ , and  $D \rightarrow A$  as DSAN.

Office Home [43] is much larger than the Office31 data set. Each domain has 65 object categories and contains a total of 15,588 color images in office and home scenarios. These images come from four domains: artistic images (A), clip art (C), product images (P), and real-world images (R). Similarly, we use all domain combinations and construct 12 transfer tasks.

Digital recognition dataset contains three widely used benchmarks: MNIST [44], USPS [45] and SVHN [46], MNIST contains 60,000 training images and 10,000 test images, and USPS contains 7291 training images and 2007 test images, all the images in MNIST and USPS are  $28 \times 28$  grayscale images. Here, we follow the settings in DSAN and JDA, and randomly sample 2000 and 1800 images in MNIST and USPS, respectively, to form a new dataset. SVHN contains  $32 \times 32$  color images, but each image may contain multiple digits. We conduct experiments on three transfer tasks  $MNIST \rightarrow USPS$ ,  $USPS \rightarrow MNIST$ , and  $SVHN \rightarrow MNIST$ .

#### 4.2. Implementation Detail

For the digital recognition dataset, all the images are resized to  $32 \times 32$  as the input of the network. We use the modified LeNet [42] as the feature extraction network, which mainly contains two convolutional layers with a convolution kernel size of  $5 \times 5$ , followed by two fully connected layers fc1 and fc2 with 1024 and 256 units, respectively, and uses the output of fc2 as inputs of FMMD. The modified LeNet model is shown in Figure 3.



**Figure 3.** The structure of the modified LeNet model.

For the other two datasets, we follow the settings in DSAN and employ ResNet50 as the feature extraction network, and a bottleneck layer fcb with 256 units is added after the last average pooling layer to reduce the dimensionality. Finally, we use the output of fcb as the FMMD input. We use the pre-trained model on ImageNet to fine-tune all convolutional and pooling layers, and we train the classifier layer via backpropagation.

We follow the settings of DSAN. For each task, we use a mini-batch stochastic gradient descent (SGD) with a momentum of 0.9, and the weight attenuation coefficient is  $5 \times 10^{-4}$ . The learning rate is adjusted during SGD using the following formula:  $\eta_p = \eta_0 / (1 + 10p)^{0.75}$ , where  $p$  is the training progress linearly varying from 0 to 1,

$\eta_0 = 0.01$ , which is optimized to promote convergence and low error on the source domain. Instead of fixing the adaptation factor  $\lambda_p$ , we dynamically adjust it via Equation (8).

$$\lambda_p = \frac{2}{1 + \exp(-10p)} - 1 \quad (8)$$

This progressive strategy can effectively suppress noisy activations at the early stages of training.

For a fair comparison of the above methods, the same network architecture is used on the same dataset. We implement these with publicly available code (<https://github.com/jindongwang/transferlearning/> accessed on 27 September 2021) and report the average classification accuracy and standard error for three random trials. For all MMD-based approaches, we adopt a Gaussian kernel, and the bandwidth is set to the median pairwise squared distance on the training data.

### 4.3. Results

Tables 1–3 show the results of the different methods on the digital recognition dataset, Office-31, and Office Home, respectively. DFDA outperforms the compared methods on most tasks. There is a 1.7% improvement on the digital recognition dataset and an average accuracy improvement of more than 0.8% on the Office-31 dataset. These facts show that our approach is indeed effective and enhances the domain adaptive capability compared to DSAN.

**Table 1.** Accuracy (%) on digit recognition tasks for unsupervised domain adaptation.

Method	MNIST→USPS	USPS→MNIST	SVHN→MNIST	Avg
baseline	94.5	76.0	71.2	80.5
DDC	96.0	95.3	77.5	89.6
Deep CORAL	96.0	87.7	72.4	85.4
DSAN	97.2	<b>97.1</b>	87.7	94.0
DFDA	<b>97.5</b>	<b>97.1</b>	<b>92.5</b>	<b>95.7</b>

**Table 2.** Accuracy (%) on Office-31 for unsupervised domain adaptation (resnet50).

Method	A→W	D→W	W→D	A→D	D→A	W→A	Avg
ResNet50	68.4 ± 0.5	96.7 ± 0.5	99.3 ± 0.1	68.9 ± 0.2	62.5 ± 0.3	60.7 ± 0.3	76.1
DDC	75.8 ± 0.2	95.0 ± 0.2	98.2 ± 0.1	77.5 ± 0.3	67.4 ± 0.4	64.0 ± 0.5	79.7
Deep CORAL	77.7 ± 0.2	97.6 ± 0.2	99.7 ± 0.1	81.1 ± 0.4	64.6 ± 0.4	64.0 ± 0.3	80.8
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
DSAN	<b>93.6 ± 0.2</b>	98.3 ± 0.1	<b>100 ± 0.0</b>	90.2 ± 0.7	73.5 ± 0.5	74.8 ± 0.4	88.4
DFDA	93.5 ± 0.3	<b>98.7 ± 0.1</b>	<b>100 ± 0.0</b>	<b>90.3 ± 0.5</b>	<b>75.9 ± 0.4</b>	<b>76.8 ± 0.2</b>	<b>89.2</b>

**Table 3.** Accuracy (%) on Office-Home for unsupervised domain adaptation (resnet50).

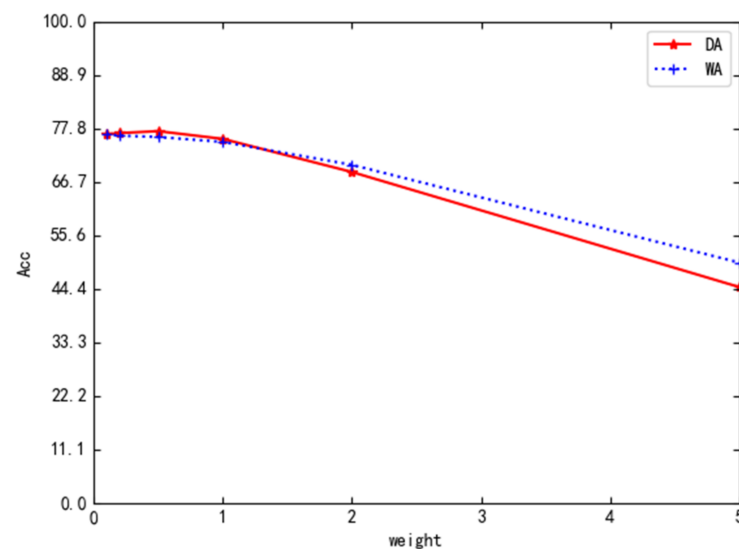
Method	A→C	A→P	A→R	C→A	C→P	C→R	P→A	P→C	P→R	R→A	R→C	R→P	Avg
ResNet	34.9	50.0	58.0	37.4	41.9	46.2	38.5	31.2	60.4	53.9	41.2	59.9	46.1
DAN	43.6	57	67.9	45.8	56.5	60.4	44.0	43.6	67.7	63.1	51.5	74.3	56.3
DANN	45.6	59.3	70.1	47.0	58.5	60.9	46.1	43.7	68.5	63.2	51.8	76.8	57.6
JAN	45.9	61.2	68.9	50.4	59.7	61.0	45.8	43.4	70.3	63.9	52.4	76.8	58.3
DSAN	54.4	70.8	75.4	60.4	67.8	68.0	<b>62.6</b>	55.9	78.5	73.8	60.6	83.1	67.6
DFDA	<b>54.8</b>	<b>71.4</b>	<b>75.5</b>	<b>61.8</b>	<b>69.7</b>	<b>68.6</b>	62.5	<b>56.6</b>	<b>79.0</b>	<b>74.2</b>	<b>61.2</b>	<b>83.7</b>	<b>68.2</b>

We also observed only about a 0.6% performance improvement in the Office Home dataset. The main reason is that, in the batch size of 64 samples, compared to the digital recognition dataset with only 10 categories, the probability of each class of sample present in the Office Home dataset with 65 categories is much smaller. That is, the former expects

about 6.4 occurrences per category, while the latter is less than 1. This means that there are more samples available for quality selection in the digital recognition dataset, and thus the improvement is the largest. In conclusion, our method performed better in larger batch sizes compared with smaller batches.

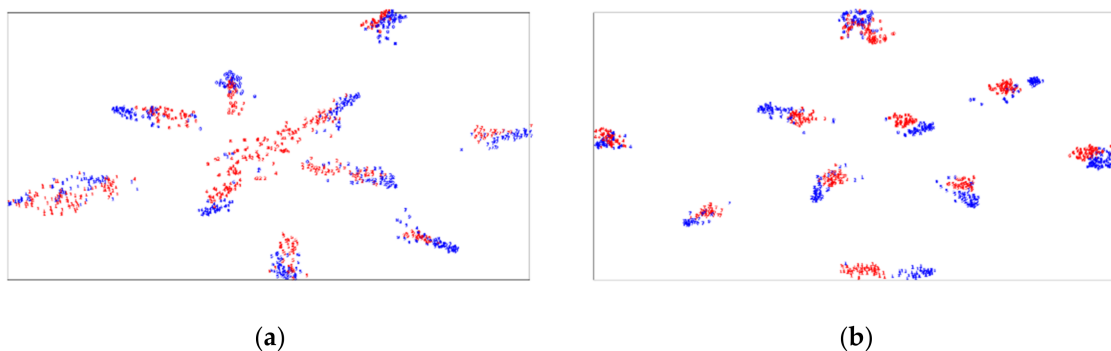
#### 4.4. Parameter Sensitivity Analysis

Although the weight coefficient  $\lambda_p$  of the transfer loss is changed in the experiment, we also studied the effects of different  $\lambda_p$ . Figure 3 demonstrates the variation of average accuracy of DFDA on tasks  $D \rightarrow A$  and  $W \rightarrow A$  for  $\lambda_p \in \{0.1, 0.2, 0.5, 1, 2, 5\}$ . As shown in Figure 4, the accuracy improves slightly with increasing  $\lambda_p$  and then decreases; however, good results can be achieved in the range of less than 1.



**Figure 4.** Performance of DFDA on task DA and WA with different  $\lambda_p$ .

To demonstrate the effect of transfer learning, we utilize t-SNE (<https://lvdmaaten.github.io/tsne/> accessed on 12 October 2021) to visualize in Figure 5a,b the network activations of task  $SVHN \rightarrow MNIST$  learned by DSAN and DFDA. Blue points are source samples, and red are target samples. Figure 5a shows the representations learned by DSAN, and it can be seen that although some categories can be well matched, there are still some that are confused. In contrast, the same categories on the source and target domains are properly aligned in Figure 5b. It is clear from the figure that our results are better than DSAN. We can find that the source and target domains are not well aligned using DSAN and some points are hard to classify. The main reason is that DSAN did not conduct quality assessment of the samples and align all samples, which caused confusion.



**Figure 5.** (a,b) The feature visualization results of DSAN and DFDA for the  $SVHN \rightarrow MNIST$  task, respectively.



#### 4.5. Discussion on the Advantage of DFDA

To give an overview of the results, we compared DFDA with several other MMD-based metrics in terms of the execution time and computational cost. The performance test was performed on a computer with a NVIDIA RTX3090 GPU, and the results are shown in Table 4.

**Table 4.** The performance of different methods on the Office-31 dataset. (Batch size = 64).

	ResNet50	DDC	DSAN	DFDA
FLOPs	$4.2 \times 10^9$	12,845,086	13,635,232	13,639,200
Epoch time (s)	21.9	32.1	32.3	32.5
Accuracy (%)	76.1	79.7	88.4	89.2

Our approach does increase some matrix operations over the previous approach; however, these computational costs are negligible compared to the computational effort brought by ResNet50, and considerable performance gains can be obtained. The average increase in computing time per epoch is only 0.3 s, which we believe is worth the cost.

#### 5. Conclusions

The previous subdomain adaptive approach aligns all samples within the relevant subdomain without considering the quality differences of individual features. In this paper, we proposed a new method of DFDA to measure the distance between domains, which evaluates the quality of individual features in the source domain by adding predicted information of the source domain to the network. Particularly in the setting of a large batch size, this can select more suitable samples among multiple same-class samples on the source domain to match with the target domain, which effectively improved the performance of the transfer model. Compared with previous methods, our proposed DFDA achieved 1.7% and 0.8% performance improvements on the digital recognition and Office-31 datasets, respectively, and these results support the effectiveness of our proposed method.

**Author Contributions:** The article was completed by three people, all writing and experimental simulations were completed by F.Z., C.W. was mainly responsible for the theoretical analysis of the algorithm, and W.L. provided technical support for the problems that appeared in the experiment. All authors have read and agreed to the published version of the manuscript.

**Funding:** This research was funded by the National Natural Science Foundation of China under Grants 61933013, Shaanxi province key research and development program (2021GY-087).

**Institutional Review Board Statement:** Not applicable.

**Informed Consent Statement:** Not applicable.

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

#### References

1. He, K.; Zhang, X.; Ren, S.; Sun, J. Deep residual learning for image recognition. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Las Vegas, NV, USA, 27–30 June 2016; pp. 770–778.
2. Brown, T.B.; Mann, B.; Ryder, N.; Subbiah, M.; Kaplan, J.; Dhariwal, P.; Neelakantan, A.; Shyam, P.; Sastry, G.; Askell, A. Language models are few-shot learners. *arXiv* **2020**, arXiv:2005.14165.
3. Bruno, A.; Ardizzone, E.; Vitabile, S.; Midiri, M. A novel solution based on scale invariant feature transform descriptors and deep learning for the detection of suspicious regions in mammogram images. *J. Med. Signals Sens.* **2020**, *10*, 158. [[PubMed](#)]
4. Ghafoorian, M.; Mehrtash, A.; Kapur, T.; Karssemeijer, N.; Marchiori, E.; Pesteie, M.; Guttmann, C.R.; de Leeuw, F.-E.; Tempny, C.M.; Van Ginneken, B. Transfer learning for domain adaptation in mri: Application in brain lesion segmentation. In *Medical Image Computing and Computer Assisted Intervention—MICCAI 2017, Proceedings of the 20th International Conference, Quebec City, QC, Canada, 11–13 September 2017*; Springer: Berlin/Heidelberg, Germany, 2017; pp. 516–524.
5. Ye, L.; Ma, X.; Wen, C. Rotating machinery fault diagnosis method by combining time-frequency domain features and CNN knowledge transfer. *Sensors* **2021**, *21*, 8168. [[CrossRef](#)] [[PubMed](#)]
6. Pan, S.J.; Yang, Q. A survey on transfer learning. *IEEE Trans. Knowl. Data Eng.* **2009**, *22*, 1345–1359. [[CrossRef](#)]
7. Csurka, G. Domain adaptation for visual applications: A comprehensive survey. *arXiv* **2017**, arXiv:1702.05374.

8. Liu, J.; Shah, M.; Kuipers, B.; Savarese, S. Cross-view action recognition via view knowledge transfer. In Proceedings of the CVPR 2011, Colorado Springs, CO, USA, 20–25 June 2011; pp. 3209–3216.
9. Hu, D.H.; Yang, Q. Transfer learning for activity recognition via sensor mapping. In Proceedings of the Twenty-Second International Joint Conference on Artificial Intelligence, Barcelona, Spain, 16–22 July 2011.
10. Long, M.; Wang, J.; Ding, G.; Sun, J.; Yu, P.S. Transfer joint matching for unsupervised domain adaptation. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Columbus, OH, USA, 23–28 June 2014; pp. 1410–1417.
11. Duan, L.; Tsang, I.W.; Xu, D. Domain transfer multiple kernel learning. *IEEE Trans. Pattern Anal. Mach. Intell.* **2012**, *34*, 465–479. [[CrossRef](#)]
12. Chu, W.-S.; De la Torre, F.; Cohn, J.F. Selective transfer machine for personalized facial action unit detection. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Portland, OR, USA, 23–28 June 2013; pp. 3515–3522.
13. Tsai, Y.-H.H.; Yeh, Y.-R.; Wang, Y.-C.F. Learning cross-domain landmarks for heterogeneous domain adaptation. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Las Vegas, NV, USA, 27–30 June 2016; pp. 5081–5090.
14. Dai, W.; Yang, Q.; Xue, G.-R.; Yu, Y. Boosting for transfer learning. In Proceedings of the 24th International Conference on Machine Learning, Association for Computing Machinery, Corvallis, OR, USA, 20–24 June 2007; pp. 193–200.
15. Rozantsev, A.; Salzmann, M.; Fua, P. Beyond sharing weights for deep domain adaptation. *IEEE Trans. Pattern Anal. Mach. Intell.* **2018**, *41*, 801–814. [[CrossRef](#)]
16. Rozantsev, A.; Salzmann, M.; Fua, P. Residual parameter transfer for deep domain adaptation. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Salt Lake City, UT, USA, 18–23 June 2018; pp. 4339–4348.
17. Chen, C.; Jiang, B.; Jin, X. Parameter transfer extreme learning machine based on projective model. In Proceedings of the 2018 International Joint Conference on Neural Networks (IJCNN), Rio de Janeiro, Brazil, 8–13 July 2018; pp. 1–8.
18. Sun, B.; Saenko, K. Deep coral: Correlation alignment for deep domain adaptation. In Proceedings of the European Conference on Computer Vision, Amsterdam, The Netherlands, 11–14 October 2016; Springer: Berlin/Heidelberg, Germany, 2016; pp. 443–450.
19. Long, M.; Wang, J.; Cao, Y.; Sun, J.; Philip, S.Y. Deep learning of transferable representation for scalable domain adaptation. *IEEE Trans. Knowl. Data Eng.* **2016**, *28*, 2027–2040. [[CrossRef](#)]
20. Sener, O.; Song, H.O.; Saxena, A.; Savarese, S. Learning transferrable representations for unsupervised domain adaptation. In Proceedings of the Advances in Neural Information Processing Systems, Barcelona, Spain, 5–10 December 2016; pp. 2110–2118.
21. Ganin, Y.; Lempitsky, V. Unsupervised domain adaptation by backpropagation. In Proceedings of the 32nd International Conference on Machine Learning, Lille, France, 6–11 July 2015; pp. 1180–1189.
22. Ganin, Y.; Ustinova, E.; Ajakan, H.; Germain, P.; Larochelle, H.; Laviolette, F.; Marchand, M.; Lempitsky, V. Domain-adversarial training of neural networks. *J. Mach. Learn. Res.* **2016**, *17*, 1–35.
23. Yu, C.; Wang, J.; Chen, Y.; Huang, M. Transfer learning with dynamic adversarial adaptation network. In Proceedings of the 2019 IEEE International Conference on Data Mining (ICDM), Beijing, China, 8–11 November 2019; pp. 778–786.
24. Long, M.; Cao, Z.; Wang, J.; Jordan, M.I. Conditional adversarial domain adaptation. *arXiv* **2017**, arXiv:1705.10667.
25. Pei, Z.; Cao, Z.; Long, M.; Wang, J. Multi-adversarial domain adaptation. In Proceedings of the Thirty-Second AAAI Conference on Artificial Intelligence, New Orleans, LA, USA, 2–7 February 2018.
26. Zhu, Y.; Zhuang, F.; Wang, J.; Ke, G.; Chen, J.; Bian, J.; Xiong, H.; He, Q. Deep subdomain adaptation network for image classification. *IEEE Trans. Neural Netw. Learn. Syst.* **2020**, *32*, 1713–1722. [[CrossRef](#)] [[PubMed](#)]
27. He, K.; Girshick, R.; Dollár, P. Rethinking imagenet pre-training. In Proceedings of the IEEE/CVF International Conference on Computer Vision, Seoul, Korea, 27–28 October 2019; pp. 4918–4927.
28. Goodfellow, I.; Pouget-Abadie, J.; Mirza, M.; Xu, B.; Warde-Farley, D.; Ozair, S.; Courville, A.; Bengio, Y. Generative adversarial nets. *Adv. Neural Inf. Processing Syst.* **2014**, *27*, 2672–2680.
29. Tzeng, E.; Hoffman, J.; Saenko, K.; Darrell, T. Adversarial discriminative domain adaptation. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Honolulu, HI, USA, 21–26 July 2017; pp. 7167–7176.
30. Zellinger, W.; Grubinger, T.; Lughofer, E.; Natschläger, T.; Saminger-Platz, S. Central moment discrepancy (cmd) for domain-invariant representation learning. *arXiv* **2017**, arXiv:1702.08811.
31. Long, M.; Zhu, H.; Wang, J.; Jordan, M.I. Deep transfer learning with joint adaptation networks. In Proceedings of the 34th International Conference on Machine Learning, Sydney, Australia, 6–11 August 2017; pp. 2208–2217.
32. Zhang, X.; Yu, F.X.; Chang, S.-F.; Wang, S. Deep transfer network: Unsupervised domain adaptation. *arXiv* **2015**, arXiv:1503.00591.
33. Gretton, A.; Borgwardt, K.M.; Rasch, M.J.; Schölkopf, B.; Smola, A. A kernel two-sample test. *J. Mach. Learn. Res.* **2012**, *13*, 723–773.
34. Pan, S.J.; Tsang, I.W.; Kwok, J.T.; Yang, Q. Domain adaptation via transfer component analysis. *IEEE Trans. Neural Netw.* **2010**, *22*, 199–210. [[CrossRef](#)]
35. Tzeng, E.; Hoffman, J.; Zhang, N.; Saenko, K.; Darrell, T. Deep domain confusion: Maximizing for domain invariance. *arXiv* **2014**, arXiv:1412.3474.
36. Long, M.; Cao, Y.; Wang, J.; Jordan, M. Learning transferable features with deep adaptation networks. In Proceedings of the International Conference on Machine Learning, Lille, France, 6–11 July 2015; pp. 97–105.
37. Yan, H.; Ding, Y.; Li, P.; Wang, Q.; Xu, Y.; Zuo, W. Mind the class weight bias: Weighted maximum mean discrepancy for unsupervised domain adaptation. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Honolulu, HI, USA, 21–26 July 2017; pp. 2272–2281.

38. Wang, J.; Chen, Y.; Feng, W.; Yu, H.; Huang, M.; Yang, Q. Transfer learning with dynamic distribution adaptation. *ACM Trans. Intell. Syst. Technol.* **2020**, *11*, 1–25. [[CrossRef](#)]
39. Ben-David, S.; Blitzer, J.; Crammer, K.; Pereira, F. Analysis of representations for domain adaptation. *Adv. Neural Inf. Processing Syst.* **2007**, *19*, 137.
40. Sun, B.; Feng, J.; Saenko, K. Return of frustratingly easy domain adaptation. In Proceedings of the AAAI Conference on Artificial Intelligence, Phoenix, AZ, USA, 12–17 February 2016.
41. Ben-David, S.; Blitzer, J.; Crammer, K.; Kulesza, A.; Pereira, F.; Vaughan, J.W. A theory of learning from different domains. *Mach. Learn.* **2010**, *79*, 151–175. [[CrossRef](#)]
42. Saenko, K.; Kulis, B.; Fritz, M.; Darrell, T. Adapting visual category models to new domains. In *Computer Vision—ECCV 2010, Proceedings of the 11th European Conference on Computer Vision, Heraklion, Crete, Greece, 5–11 September 2010*; Springer: Berlin/Heidelberg, Germany, 2010; pp. 213–226.
43. Venkateswara, H.; Eusebio, J.; Chakraborty, S.; Panchanathan, S. Deep hashing network for unsupervised domain adaptation. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Honolulu, HI, USA, 21–26 July 2017; pp. 5018–5027.
44. LeCun, Y.; Bottou, L.; Bengio, Y.; Haffner, P. Gradient-based learning applied to document recognition. *Proc. IEEE* **1998**, *86*, 2278–2324. [[CrossRef](#)]
45. Hull, J.J. A database for handwritten text recognition research. *IEEE Trans. Pattern Anal. Mach. Intell.* **1994**, *16*, 550–554. [[CrossRef](#)]
46. Netzer, Y.; Wang, T.; Coates, A.; Bissacco, A.; Wu, B.; Ng, A.Y. Reading digits in natural images with unsupervised feature learning. In Proceedings of the NIPS Workshop on Deep Learning and Unsupervised Feature Learning, Granada, Spain, 12–17 December 2011; p. 5.