mathematics

MDPI

Article

Deep Transfer Learning Method Based on Automatic Domain Alignment and Moment Matching

Jingui Zhang [1,2,†], Chuangji Meng [1,†], Cunlu Xu [1,2,*], Jingyong Ma [3] and Wei Su [1,2]

[1] School of Information Science and Engineering, Lanzhou University, 222 Tianshui South Road, Lanzhou 730000, China; zhangjingui@lzu.edu.cn (J.Z.); 4120105144@stu.xjtu.edu.cn (C.M.); suwei@lzu.edu.cn (W.S.)
[2] Key Laboratory of Media Convergence Technology and Communication, Lanzhou 730030, China
[3] College of Pastoral Agriculture Science and Technology, Lanzhou University, Lanzhou 730020, China; majy@lzu.edu.cn
* Correspondence: clxu@lzu.edu.cn
† These authors contributed equally to this work.

Abstract: Domain discrepancy is a key research problem in the field of deep domain adaptation. Two main strategies are used to reduce the discrepancy: the parametric method and the nonparametric method. Both methods have achieved good results in practical applications. However, research on whether the combination of the two can further reduce domain discrepancy has not been conducted. Therefore, in this paper, a deep transfer learning method based on automatic domain alignment and moment matching (DA-MM) is proposed. First, an automatic domain alignment layer is embedded in the front of each domain-specific layer of a neural network structure to preliminarily align the source and target domains. Then, a moment matching measure (such as MMD distance) is added between every domain-specific layer to map the source and target domain features output by the alignment layer to a common reproduced Hilbert space. The results of an extensive experimental analysis over several public benchmarks show that DA-MM can reduce the distribution discrepancy between the two domains and improve the domain adaptation performance.

Keywords: deep transfer learning; domain adaptation; automatic domain alignment; maximum mean discrepancy

MSC: 68T07; 68U10

Citation: Zhang, J.; Meng, C.; Xu, C.; Ma, J.; Su, W. Deep Transfer Learning Method Based on Automatic Domain Alignment and Moment Matching. *Mathematics* **2022**, *10*, 2531. https://doi.org/10.3390/math10142531

Academic Editor: Jakub Nalepa

Received: 21 June 2022
Accepted: 17 July 2022
Published: 21 July 2022

1. Introduction

Deep neural networks have achieved substantial success in all aspects of machine learning applications [1–3]. However, the training and testing data for these networks may not have the same distribution. In addition, obtaining sufficient labeled target data is difficult [4]. Transfer learning [5,6] attempts to build effective classifiers that can be used in the target domain by using the labeled data of the source domain that obey different but related distributions. In such cases, the source and target data are obtained from similar but not identical domains and usually follow different distributions. Therefore, reducing the distribution differences between the source and target domains has become the main obstacle [7–11].

Recently, research based on deep learning has achieved remarkable results in different fields [12–15]. In most studies, the discrepancy between the source domain and target domain is reduced by learning the domain-invariant features, usually mainly using two main strategies. One strategy is based on minimizing the domain adaptation loss, which contains hyperparameters for a regular term, such as the minimization of the maximum mean discrepancy (MMD) [16–18] or the domain-confusion loss [19,20]. The common

purpose of this strategy is to obtain more domain-invariant representations. We call this method the parametric approach.

The other strategy uses a nonparametric method [21–24] with the goal of reducing the domain shift between the two domains by designing a specific normalized distribution layer, such as adaptive batch normalization (AdaBN) [25] and domain alignment layers (DA layers) [26]. This strategy endows these layers with the ability to automatically learn the degree of alignment of different layers of the network without introducing additional loss terms (e.g., MMD or domain confusion) into the optimization function and their related hyperparameters. However, most of these methods require additional optimization steps and hyperparameters in order to establish a connection between the training and testing domains. Such an additional computational burden can greatly complicate the training of a deep neural network. For example, the hyperparameters that control the trade-off between the supervised learning loss term of the source domain and the additional loss (e.g., MMD, JMMD, B-JMMD, domain confusion) term need to be fine-tuned, and even some well-designed additional loss items (B-JMMD, BDA) require hyperparameters with specific functions, for example, to balance the marginal and conditional distribution adaptations [27]. Again, the additional computational burden can complicate the training of a deep neural network, which is unappealing to most researchers.

The advantage of the nonparametric method is that it does not require prior knowledge (which layers need to be aligned), and there are no hyperparameters that need to be fine-tuned [25,26]. The distributions, such as the BN layer, can be aligned automatically [25]. In addition, AutoDial [26] can learn domain-invariant features without introducing additional loss terms (e.g., MMD or domain confusion) into the optimization function or any associated hyperparameters. Even in powerful deep learning models, the problem of domain shift can be alleviated but not eliminated, which raises the question: Does adding additional loss items such as MMD after the initial DA-layer alignment further improve the alignment?

The contribution of this work can be summarized from the following aspects: (a) The performance of the MMD parametric method can be significantly improved by embedding the automatic domain alignment layer (DA layer) between each domain-specific layer in a deep neural network. The MMD parameter can benefit from the preliminary alignment performed by the DA layer between the source domain and the target domain. (b) For the automatic alignment method, although AutoDial causes the model to learn domain-invariant features without introducing additional loss terms (e.g., MMD or domain confusion) into the optimization function and the associated hyper-parameters, there is still room for improvement. Incorporating MMD can further improve the degree of alignment and achieve better performance. (c) Compared with the automatic alignment method, our method needs to add DA layers to only a small number of domain-specific layers, which greatly reduces the number of network layers while improving the performance. (d) We conducted numerous comparative experiments to demonstrate the effectiveness of the proposed method.

2. Related Work

In this section, previous research on deep transfer learning and deep domain adaptation is discussed, and relevant differences between these approaches and our proposed method are identified. One of the main problems in deep transfer learning is reducing the discrepancy between the distributions of the source domain and target domain through two main strategies.

The first strategy is based on a parametric approach, $\mathrm{L}_s(X_s, Y_s) + lamda * DA_{loss}(X_s, Y_s, X_t)$, where X_s and Y_s represent the labeled data in source data. X_t represents the target data, $\mathrm{L}_s(X_s, Y_s)$ represents the source domain loss applied to the source samples, while $DA_{loss}(X_s, Y_s, X_t)$ is an entropy loss applied to the target samples. $lamda$ is the regularization coefficients of the target domain predictor.

(a) Feature alignment based on MMD: $\mathrm{L}_s + lamda * L_{MMD}$, where L_s represents the source domain loss applied to the source samples, L_{MMD} represents the MMD loss applied

to the target samples. The minimization of the MMD [16–18] can be described as: the source and target data are projected into a common subspace, and then the distributions of the representations of the source domain and target domain are optimized by minimizing the mean embedding distance between the two domains to make them as similar as possible. The deep domain confusion (DDC) [28] approach introduces one adaptation layer to AlexNet [29] that uses a linear kernel MMD and an additional domain-confusion loss, causing the model to learn domain-invariant representations. DDC used the classical MMD loss to regularize the representation in the final CNN layer. To improve the effectiveness of adaptation, DAN [16] matches the mean embeddings of marginal distributions by introducing the multi-kernel MMD in the corresponding domain-specific layers. Thus, DAN further extends the method to multi-kernel MMD and multi-layer adaptation.

RevGrad [20] proposed a gradient reversal layer to compensate for the domain-specific back-propagated gradients. Bousmalis et al. [22] devised a domain separation network that can extract better domain-invariant features by explicitly modeling the private and shared components of the domain representations in a network. Different from the previous deep transfer methods, JMMD approximates the shift of joint distributions after the network activations in the second-order tensor product Hilbert space [30]. However, it is unclear how to determine which components of the representations support the reasoning about the original joint distributions. B-JMMD adaptively utilizes the importance of marginal and conditional distributions behind multiple domain-specific layers across domains and realizes the adaptive effect of a balanced distribution of deep network architectures [27]. At the same time, however, a balance factor is introduced.

(b) Adversarial-based deep transfer learning: $L_s + lamda * L_{adv}$, where L_{adv} represents the adversarial loss applied to the target samples. Another strategy [19,20] relies on domain-confusion loss, which can predict if a sample comes from the source domain or the target domain by training an auxiliary classifier. Intuitively, the domain-invariant features can be obtained by maximizing this term, (i.e., poor performance is punished by using auxiliary classifiers). Researchers have attempted to minimize domain classification loss by making the feature distribution of the two domains as indistinguishable as possible [31]. This approach assumes that to transfer effectively, a good representation should not discriminate between the source domain and the target domain but should be discriminated for the main learning tasks.

(c) Embedding domain adaptation modules into network-based deep transfer learning: $L_s + lamda * L_{others}$ [16], where L_{others} represents the other loss applied to the target samples. This approach reduces the domain discrepancy by embedding domain adaptation modules into deep networks [28] and jointly learns adaptive classifiers and transferable features from labeled data in the source domain and from unlabeled data in the target domain. The model explicitly learns a residual function with reference to the target classier by inserting several additional layers into the deep network. Bousmalis et al. [22] learned domain adaptation and deep hash features simultaneously using a DNN.

A second strategy for unsupervised domain adaptation is the nonparametric approach.

Recently, researchers have begun to investigate alternative directions [21–24], such as reducing the possibility of the domain shift by introducing specific distribution normalization layers. Inspired by the popular batch normalization (BN) technique [23], a simple nonparametric approach called AdaBN was used to modify the Inception-BN network. AdaBN aligns the source and target representations of learning by using different mean/variance terms of the source and target domains when performing BN at the time of inference. Inspired by Li et al. [25], AutoDial introduces novel domain alignment layers (DA layers) embedded at different levels of a deep architecture. Different from Li et al. [23], all previous deep domain adaptation methods determine which layers should be adapted in advance, and AutoDial [26] endows the DA layers with the ability to automatically learn the alignment degree. This nonparametric approach causes the model to learn domain-invariant features without introducing additional loss terms (e.g., MMD or domain confusion) into the optimization function or associated hyperparameters.

3. Preliminary Information

3.1. Problem Definition

In this paper, we focus on unsupervised domain adaptation. Given a source domain D_s and a target domain D_t, the source domain D_s is composed of $\{(x_i^s, y_i^s)\}_{i=1}^{n_s}$ labeled examples, and the target domain D_t is composed of n_t unlabeled examples $\{(x_i^t)\}_{i=1}^{n_t}$. Generally, in machine learning problems, we assume the feature space $\chi_s = \chi_t$, the label space $y_s = y_t$, but in transfer learning, the marginal distribution $P_s(x_s) \neq P_t(x_t)$, the conditional distribution $P_s(y_s|x_s) \neq P_t(y_t|x_t)$. Transfer learning aims to obtain the label of the target domain through knowledge learning in the source domain, and domain adaptation solves the problem of transfer learning across domains by reducing the distribution difference between the two distributions.

3.2. Domain Alignment Layers (DA Layers)

The idea behind the AdaBN [25] algorithm is to align the source and target domain distributions independently to a standard normal distribution by using a certain method. In this process, the target samples do not affect the predictor network parameters. Due to insufficient target domain samples, the domain adaptability of the network structure still has some deficiencies. Compared with the AdaBN algorithm, the DA-layer approach considers the roles of the target domain samples and capitalizes on them. Specifically, DA layers [26] introduce a coupling parameter to mix the source and target domain samples and a cross-domain bias. The coupling parameter and the cross-domain bias jointly influence the model network parameters so that the DA layers of the network branches where the source and target domain predictors are located correspond with each other [26].

Generally, the inputs of DA layers in two predictors are represented by x_s and x_t, and the corresponding distributions are expressed as q_s and q_t. The coupling parameters are denoted by δ. The samples of the source and target domains are independent distributions in the first through sixth layers. In the seventh and eighth layers, the samples of the target domain become involved in generating predictors by introducing the coupling parameters δ, which causes the boundary between the source and target domains to become blurred, thereby reducing domain discrepancies [26]. The output of the DA layers of the source and target domain predictors are denoted by Formulas (1) and (2), respectively:

$$DA(x_s;\delta) = \frac{x_s - \mu_{st,\delta}}{\sqrt{\varepsilon + \sigma_{st,\delta}^2}}, \tag{1}$$

$$DA(x_t;\delta) = \frac{x_t - \mu_{ts,\delta}}{\sqrt{\varepsilon + \sigma_{ts,\delta}^2}}, \tag{2}$$

To avoid the problem that the denominator is zero in the case of zero variance, we introduce a decimal $\varepsilon > 0$. Here, $\mu_{st,\delta}$ and $\sigma_{st,\delta}^2$ represent the expectation and variance of $x \sim q_\delta^{\mathrm{st}}$: $\mu_{st,\delta} = E_{x\sim q_\delta^{st}}[x]$ and $\sigma_{st,\delta}^2 = Var_{x\sim q_\delta^{\mathrm{st}}}[x]$, respectively. Similarly, $\mu_{ts,\delta}$ and $\sigma_{ts,\delta}^2$ represent the expectation and variance of $x \sim q_\delta^{\mathrm{ts}}$: $\mu_{\mathrm{ts},\delta} = E_{x\sim q_\delta^{\mathrm{ts}}}[x]$ and $\sigma_{\mathrm{ts},\delta}^2 = Var_{x\sim q_\delta^{\mathrm{ts}}}[x]$, respectively. q_δ^{st} and q_δ^{ts} represent a cross-domain distribution between the source and target domains, respectively, and are denoted by Formulas (3) and (4) as follows:

$$q_\delta^{st} = \delta q^s + (1-\delta)q^t, \tag{3}$$

$$q_\delta^{\mathrm{ts}} = \delta q^t + (1-\delta)q^s, \tag{4}$$

where $\delta \in [0.5, 1]$. When the coupling coefficient $\delta = 0.5$, $q_\delta^{\mathrm{st}} = q_\delta^{ts}$; that is, the complete coupling is achieved, and therefore, no domain alignment is generated at this time; when $\delta = 1$, the independent alignment transformation is performed on the two domains, which is equivalent to AdaBN [25]. The DA layers may compute different functions for two predictors; that is, the source and target domains are completely aligned. The coupling

coefficient δ is configured to transform the independent alignment transformation of the two domains into a fully coupled state. It should be noted that λ is acquired during training and can automatically adjust the alignment degree between specific domains without requiring manual parameter adjustment.

3.3. Loss Function

The DA layers [26] fully consider the influence of the target and can make full use of the unique components of each domain during alignment. The network parameters are restricted by both the source and target domain functions. During training, under the constraints of the two functions and through continuous learning, the network ultimately learns the most suitable parameter to construct the optimal source and target domain predictors. This approach effectively utilizes the unlabeled samples of the target domain to better separate the samples that represent different categories. The predictor for the source domain network branch is measured by the SoftMax loss function and expressed by $\mathrm{L}^s(\phi)$ according to Formula (5):

$$L^s(\phi) = -\frac{1}{n}\sum_{i=1}^{n} \log f_s^{\phi}(y_i^s; x_i^s), \tag{5}$$

where $f_s^{\phi}(y_i^s; x_i^s)$ is the probability that sample point x_i^s takes label y_i^s according to the source predictor. The predictor for the target domain network branch is measured by the cross-entropy loss function, which is expressed by $L^t(\phi)$, as shown in Formula (6):

$$L^t(\phi) = -\frac{1}{m}\sum_{i=1}^{m}\sum_{y\in\mathcal{Y}} f_t^{\phi}(y; x_i^t)\log f_t^{\phi}(y; x_i^t), \tag{6}$$

where $f_s^{\phi}(y_i^s; x_i^s)$ represents the probability that sample point x_i^t takes label y according to the target predictor. The loss function of the complete network structure is measured using the weightings of Formulas (5) and (6).

3.4. MMD Metric

The MMD [16], which measures the distance between two distributions in a regenerative Hilbert space, is a kernel learning method. If P and Q are used as the inputs to the MMD distance metric [16], then the distance can be estimated according to Formula (7) as follows:

$$\mathrm{MMD}(P, Q) = \hat{D}_{\mathcal{H}}(P, Q) = \sum_{\ell\in L} \hat{D}_{\mathcal{H}}^{\ell}(P, Q)\,, \tag{7}$$

where $\ell \in L$ is the domain-specific layer, and $D_{\mathcal{H}}^{\ell}(P, Q)$ is used to measure the cross-domain joint distribution of the middle layer L of the deep neural network, which is estimated by Formula (8):

$$\begin{aligned} \hat{D}_{\mathcal{H}}^{\ell}(P, Q) \quad &= \tfrac{1}{n_s^2}\sum_{i=1}^{n_s}\sum_{j=1}^{n_s} k(x_i^{s\ell}, x_j^{s\ell}) \\ &+ \tfrac{1}{n_t^2}\sum_{i=1}^{n_t}\sum_{j=1}^{n_t} k(x_i^{t\ell}, x_j^{t\ell}) \\ &- \tfrac{1}{n_s n_t}\sum_{i=1}^{n_s}\sum_{j=1}^{n_t} \mathrm{k}(x_i^{s\ell}, x_j^{t\ell}) \end{aligned} \tag{8}$$

where $\mathrm{k}\left(x_i^*, x_j^*\right)$ is a Hilbert space mapping in the form of an inner product and used to map the original variables into a high-dimensional space, x_i^{sl} denotes the activation value generated by the source domain in layer L of the neural network, and x_i^{tl} denotes the activation value generated by the target domain in layer L of the neural network. As a domain-specific portion of the AlexNet [29] network structure, the last three layers of the L layer are domain-specific layers. In the GoogLeNet [32] network structure, an inner product layer acts as the domain-specific layer.

4. Algorithm Design

In this paper, we propose a deep transfer learning method based on automatic domain alignment and moment matching. First, DA layers are embedded in the front of each domain-specific layer of the neural network structure for preliminary alignment of the source and target domains. Then, MMD parameters are added between every two specific layers of each domain to map the source and target domain features output by the DA layers to a common space, RKHS, thereby further reducing the distribution discrepancy between the two domains. A deep transfer network architecture based on automatic domain alignment and moment matching is shown in Figure 1.

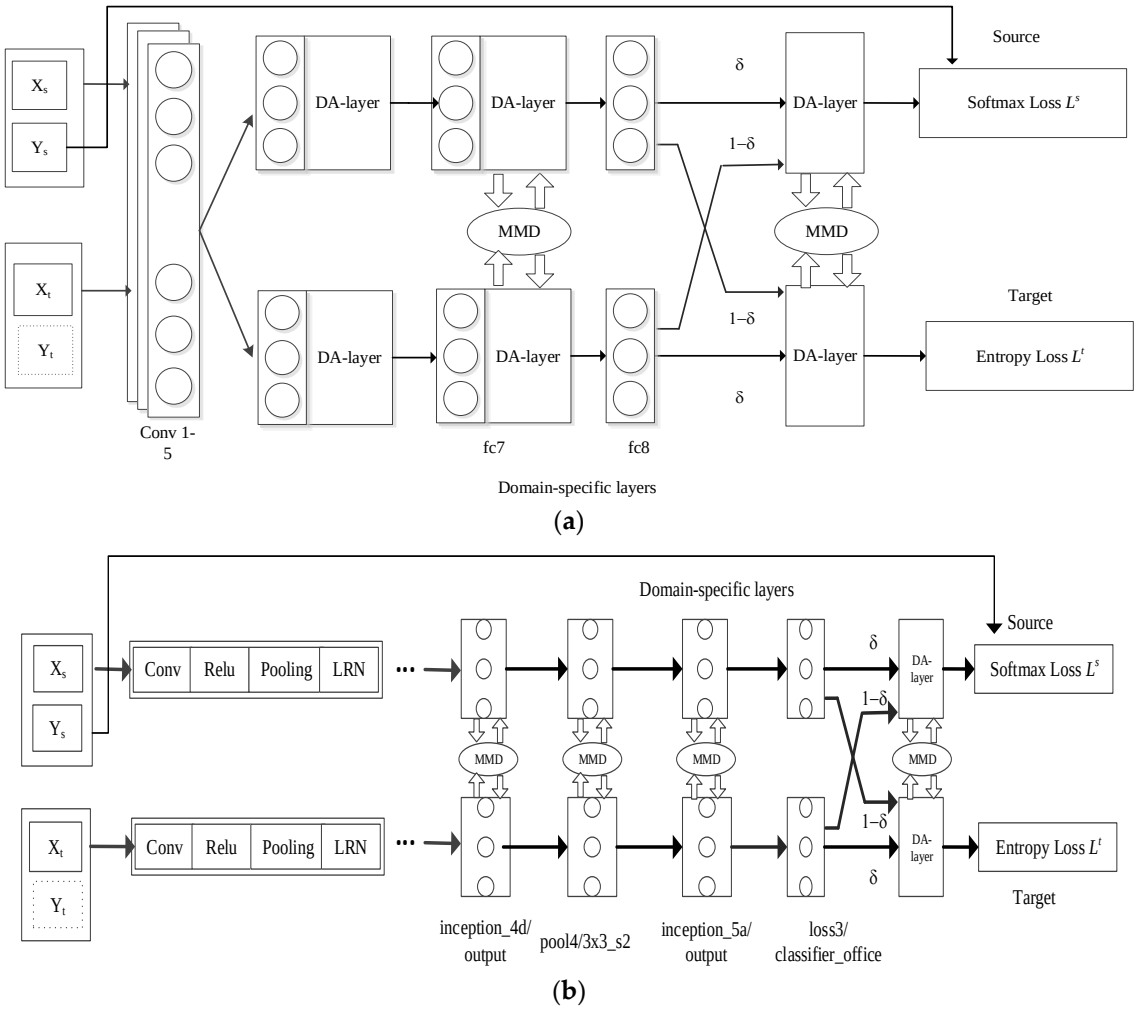


Figure 1. Network architecture of the proposed method: (**a**) architecture based on AlexNet for unsupervised domain adaptation; (**b**) architecture based on GoogLeNet for unsupervised domain adaptation.

Domain-based automatic alignment and moment matching can also be applied to deeper network structures, such as ResNet [33], VGGNet [34], or GoogLeNet [32]. The total loss function of the structure is determined according to Formula (9):

$$L(\phi) = L^s(\phi) + \lambda L^t(\phi) + \gamma MMD(P, Q|\phi), \tag{9}$$

where the term $L^s(\phi)$ is the standard log-loss applied to the source samples, while $L^s(\phi)$ is an entropy loss applied to the target samples. MMD is the first-order MMD, and P and Q are output by DA layers and input into the MMD distance layer. Here, λ and γ are the regularization coefficients of the target domain predictor and the MMD distance, respectively. The specific flow of the algorithm is shown in Algorithm 1.

Algorithm 1: Deep transfer Learning Based on Automatic Domain Alignment and Moment Matching.

Input: Source data with the label X_s, Y_s and target data X_t. λ is the regularization coefficient of the target predictor; and γ is the regularization coefficient of the MMD distance.
Output: Test accuracy; Test loss;

(1) Set i = 0. Train a baseline neural network with X_s, Y_s and test with X_t;
(2) **for** iteration i **do**
(3) Learn the coupling coefficient δ via Formulas (3) and (4) and calculate the output of the DA layer embedded in the front of the 7th and 8th layers using Formulas (1) and (2).
(4) Learn the source domain and target domain predictors from Formulas (5) and (6) and fine-tune the parameters λ to achieve the best classification results.
(5) Fine-tune the parameters γ of the MMD between specific layers in the field to achieve the best alignment effect.
(6) **return** accuracy-test, loss-test

5. Experiments and Discussion

In this section, we evaluated the performance of the proposed DA-MM algorithm by conducting experiments on two popular datasets, Office-31 [35] and Office-Caltech, and using both AlexNet [29] and GoogLeNet [32] models. All our methods are based on the Caffe [36] framework.

5.1. Datasets

(1) Office-31: The Office-31 datasets contain a collection of image data from three different fields: Amazon (A), Webcam (W), and DSLR (D). Among them, Amazon (A) consists of images downloaded from Amazon.com (accessed on 1 March 2019), with a total of 31 categories and 2817 images. Webcam (W) consists of images captured by a web camera. There are 31 categories and 795 photos. The DSLR (D) images were mainly captured by digital SLR cameras and included a total of 498 photos in the same 31 categories. We use all the combinations of the domains of the datasets and obtain six transfer learning tasks: A $\rightarrow$ W, D $\rightarrow$ W, W $\rightarrow$ D, A $\rightarrow$ D, D $\rightarrow$ A, and W $\rightarrow$ A.

(2) Office-Caltech: The Office-Caltech dataset is another standard benchmark used in the domain adaptation field. It consists of Office 10 and Caltech 10 datasets and contains 10 categories that overlap with the Office-31 [35] and Caltech-256 [37] datasets. Each category is considered to be an independent domain. The Office-Caltech dataset provides an additional 12 transfer learning tasks. In the experiment, to observe the deviation of the dataset more fairly, only the six combinations that included category C were considered as the source and target domains.

5.2. Implementation Details

This paper validates the previous methods based on the AlexNet and GoogLeNet network structures, verifies the proposed method through experiments, and adjusts the two network structure models under the Caffe framework. The accuracy and speed of DDC [28], DAN [16], JAN [30], and AutoDial [26] increased in the field of target domain classification. We use mini-batch stochastic gradient descent with momentum to train our networks and use the following meta-parameters: momentum: 0.9, weight decay 0.0005, initial learning rate 0.003. For AlexNet, the batch_size of the source domain and target domain is 64 in the training phase, the batch_size of the target domain is 1 in the test phase; the training epoch is 795 (webcam) $\times$ 64 (batch_size), where amazon is 2817, webcam is 795, dslr is 498 for the target domain, max_iteration is 10,000, training time is about 60 min (Geforce GTX TITAN 6G). For GoogLeNet, the batch_size of the source domain and target domain is 16 in the training phase, the batch_size of the target domain is 1 in the test phase; the training epoch is 795 (webcam) $\times$ 16 (batch_size), max_iteration is 100,000, training time is about 300 min (Geforce GTX TITAN 6G). Other hyperparameters in this paper are consistent with those in [26], except for hyperparameters λ and γ. For the Office-31

dataset, we set λ to 0.1 and γ to 1, and for the Office-Caltech dataset, we set λ to 0.2 and γ to 1. Based on the stability of the DA-MM results and the form of the AutoDial results, we retained the average value of the results of each method.

5.3. Results

The results of unsupervised domain adaptation on the Office-31 dataset based on AlexNet and GoogLeNet are presented in Tables 1 and 2, respectively. Table 3 shows the result of unsupervised domain adaptation on the Office-Caltech dataset based on GoogLeNet network structure. To fairly compare DDC [28], DAN [16], RevGrad [20], DRCN [38], RTN [17], JAN [30], and AutoDial [26] in the same evaluation scenario, the results of unsupervised domain adaptation and the classification accuracies of these methods were taken directly from the literature.

Table 1. Classification accuracy (%) of the Office-31 dataset for unsupervised domain adaptation (AlexNet).

Method	A→W	D→W	W→D	A→D	D→A	W→A	Avg
AlexNet[29]	61.6	95.4	99.0	63.8	51.1	49.8	70.1
DDC[28]	61.8	95.0	98.5	64.4	52.1	52.2	70.6
DAN[16]	68.5	96.0	99.0	67.0	54.0	53.1	72.9
RevGrad[20]	72.6	96.4	99.2	67.1	54.5	52.7	72.7
DRCN[38]	68.7	96.4	99.0	66.8	56.0	54.9	73.6
RTN[17]	73.3	96.8	99.6	71.0	50.5	51.0	73.7
JAN[30]	74.9	96.6	99.5	71.8	58.3	55.0	76.0
AutoDIAL[26]	75.5	96.6	99.5	73.6	58.1	**59.4**	77.1
Ours	**77.2**	**98.7**	**100.0**	**76.1**	**61.1**	**59.4**	**78.7**

Table 2. Classification accuracy (%) of the Office-31 dataset for unsupervised domain adaptation (GoogLeNet).

Method	A→W	D→W	W→D	A→D	D→A	W→A	Avg
GoogleNet[32]	70.3.	94.3	100.0	70.5	60.1	57.9	75.5
DDC[28]	72.5	95.5	98.1	73.2	61.6	61.6	77.1
DAN[16]	76.0	95.9	98.6	74.4	61.5	60.3	77.8
JAN[30]	78.1	96.4	99.3	77.5	68.4	65.0	80.8
AutoDIAL[26]	84.2	97.9	99.9	**82.3**	64.6	64.2	82.2
Ours	**87.1**	**98.0**	**99.4**	**82.3**	**70.8**	**69.3**	**84.5**

Table 3. Classification accuracy (%) of the Office-Caltech dataset for unsupervised domain adaptation (GoogLeNet).

Method	A→C	D→C	W→C	C→D	C→A	C→W	Avg
GoogleNet[32]	90.8.	87.5	87.7	89.0	95.3	89.8	90.0
AutoDIAL[26]	91.8	90.9	90.2	89.5	95.6	92.2	91.2
Ours	**93.4**	**92.5**	**92.4**	**90.1**	**95.7**	**94.9**	**93.2**

Based on the results, we can make the following observations. The proposed method is superior to all the comparison methods in most transfer tasks (11 out of 12 tasks). Specifically, the classification accuracy of the proposed AlexNet-based method on the Office-31 dataset exceeds that of the comparison method DAN (MMD only) by 5.8% and that of the method AutoDial (DA layer only) by 1.6%, while the GoogLeNet result exceeds that of DAN (MMD only) by 6.7% and that of AutoDial (DA layer only) by 2.3%. On the Office-Caltech dataset, the average classification accuracy of the GoogLeNet model is 93.2%, which is an improvement of 2.0% on average compared with the best comparison method, AutoDial. These results imply that (1) the performance of an MMD parametric method can

be significantly improved by embedding the automatic domain alignment layer (DA layer) in front of the MMD parameter between each domain-specific layer in the deep neural network, and the MMD parameter benefits from this 'preliminary alignment' by the DA layer between the source domain and the target domain. (2) Although AutoDial leads to learning domain-invariant features without requiring additional loss terms (e.g., MMD, domain confusion) in the optimization function or associated hyperparameters, there is still room for improvement. Adding the MMD further improves the degree of alignment and achieves better performance.

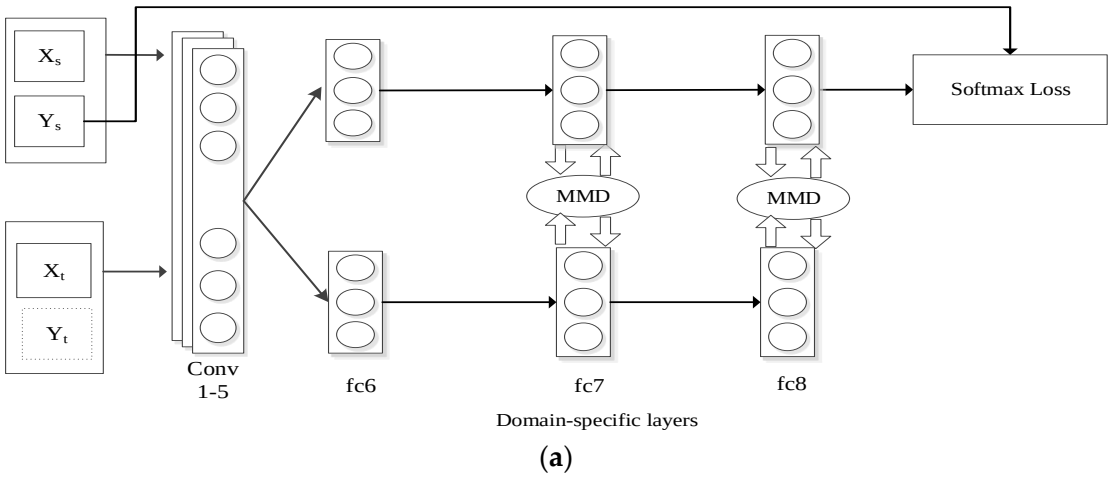
5.4. Ablation Study and Discussion

(1) Ablation study

As shown in Table 4, the results of the proposed method are significantly better than those of the method with no DA layers (exceeding 7.1% for the AlexNet-based network architecture and 6.7% for the GoogLeNet-based network architecture). The proposed method also outperformed the method with no MMD layers (by more than 1.6% for the AlexNet-based network architecture and 2.3% for the GoogLeNet-based network architecture). Corresponding network structure diagrams of these two methods are shown in Figures 1 and 2.

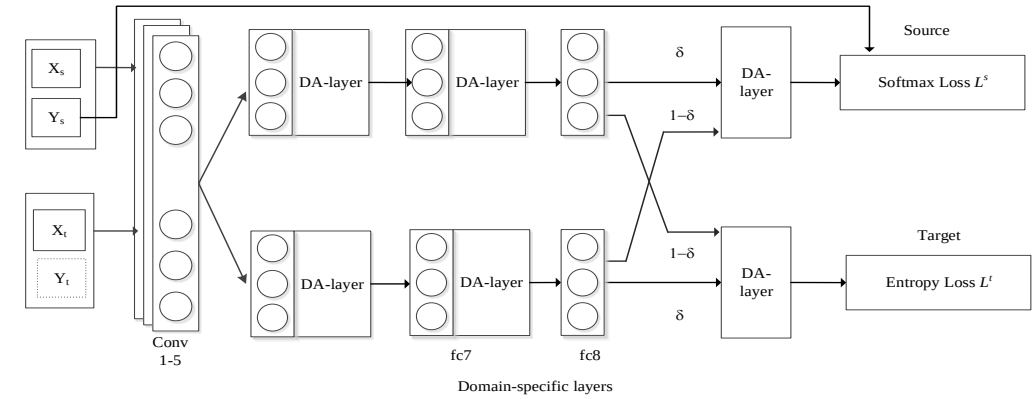
Table 4. Ablation study: classification accuracy (%) of the Office-31 dataset for unsupervised domain adaptation (AlexNet and GoogLeNet).

Method	Figure	A→W	D→W	W→D	A→D	D→A	W→A	Avg
AlexNet (source only)	-	61.6	95.3	99.0	63.8	51.1	49.8	70.1
Ours (AlexNet + MMD)	Figure 2a	61.8	95.0	98.5	64.4	52.1	52.2	70.6
Ours (AlexNet + DA layer)	Figure 2b	75.5	96.6	99.5	73.6	58.1	59.4	77.1
Ours (AlexNet + MMD + DA layer)	Figure 1a	77.2	98.7	100.0	76.1	61.1	59.4	78.7
GoogleNet (source only)	-	70.3	94.3	100.0	70.5	60.1	57.9	75.5
Ours (GoogleNet + MMD)	Figure 2c	76.0	95.9	98.6	74.4	61.5	60.3	77.8
Ours (GoogleNet + DA layer)	Figure 2d	84.2	97.9	99.9	82.35	64.6	64.2	82.2
Ours (GoogleNet + MMD + DA layer)	Figure 1b	87.1	98.0	99.4	82.3	70.8	69.3	84.5

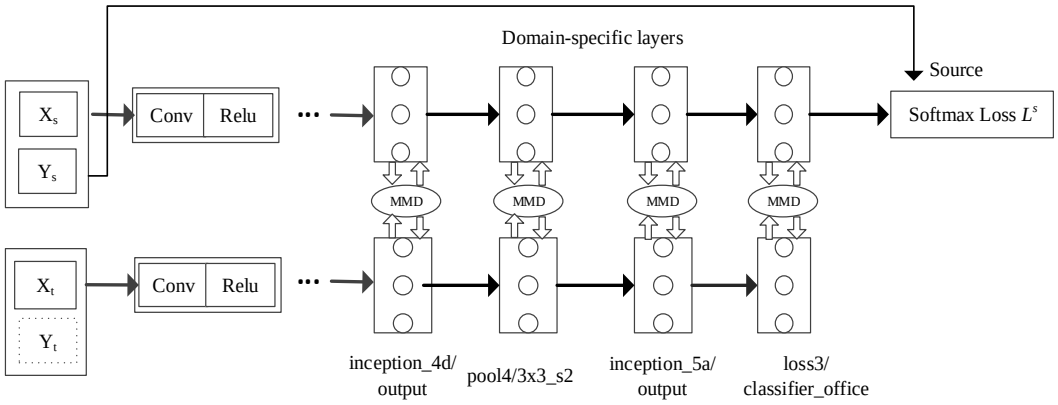


(**a**)

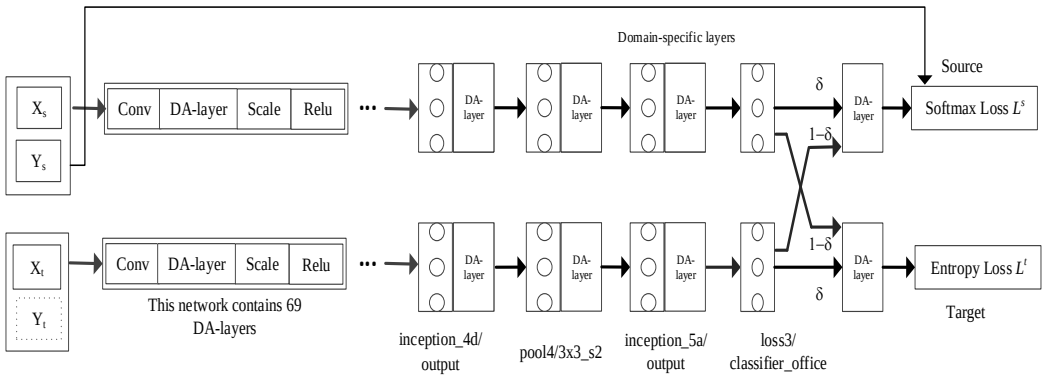
Figure 2. *Cont.*



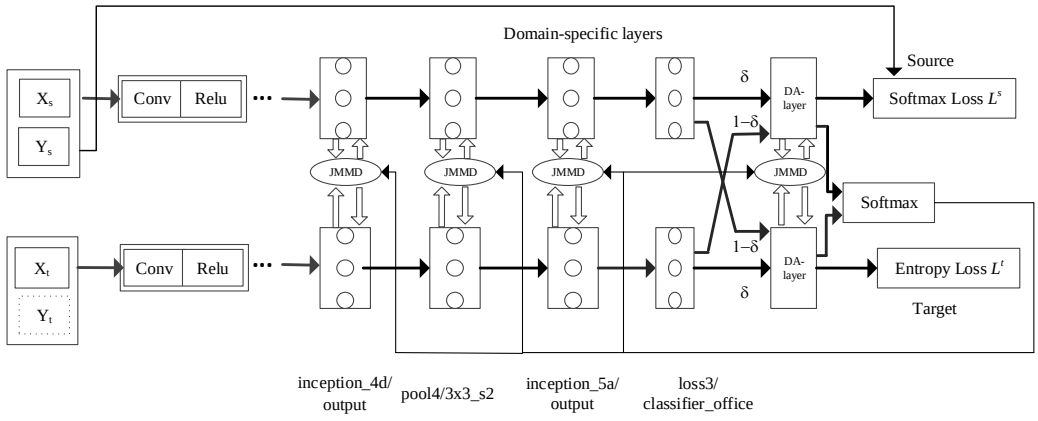
(**b**)



(**c**)

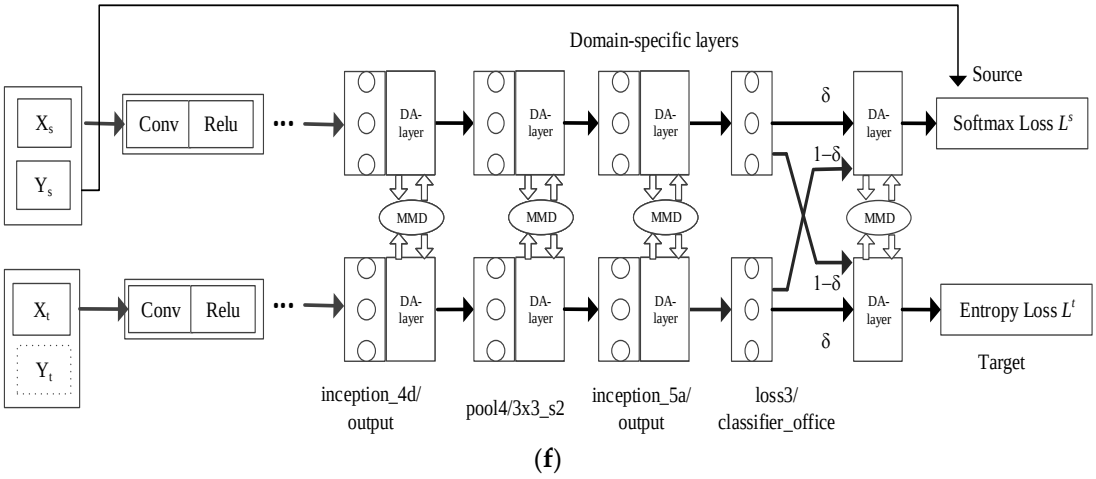


(**d**)



(**e**)

Figure 2. *Cont*.



(**f**)

Figure 2. Network architecture: (**a**) DAN (AlexNet): AlexNet + 2MMD; (**b**) AutoDial (AlexNet): AlexNet + 3DA layer; (**c**) DAN (GoogLeNet): GoogLeNet + 4MMD; (**d**) AutoDial (GoogLeNet): GoogLeNet + 69DA layer; (**e**) JAN (GoogLeNet): GoogLeNet + 4JMMD + 1DA layer; (**f**) Ours (GoogLeNet V1): GoogLeNet + 4MMD + 4DA layer.

Explanation of networks: (a) DAN (based on AlexNet) contains two MMD layers. (b) AutoDial (based on AlexNet) contains three DA layers. (c) DAN (based on GoogLeNet) contains four MMD layers based on the GoogLeNet structure. (d) AutoDial (based on GoogLeNet) contains 69 DA layers based on GoogLeNet structure. (e) JAN (GoogLeNet) contains four jmmd layers and a DA layer. (f) Ours (based on GoogLeNet V1) includes four MMD layers and four DA layers.

(2) Discussion

Comparisons with MMD-based method: Table 5 shows the classification accuracy of each method (%) for unsupervised domain adaptation. We adopt the results on task A→W as an example. The results of the proposed method outperform those of the MMD-based method (DAN) and the DA-layer-based method (AutoDial). Their network architectures are shown in Figure 2a (DAN based on AlexNet), Figure 2c (DAN based on GoogLeNet), Figure 2b (AutoDial based on AlexNet), and Figure 2d (AutoDial based on GoogLeNet). These results indicate that, to a certain extent, preliminary alignment by adding automatic domain alignment layers can help unleash the potential of the MMD parameter. (see Figure 2e (JAN based on GoogLeNet)).

Table 5. Ablation study: a description of the network architecture and the classification accuracy achieved by each method (%) for unsupervised domain adaptation (the results on task A→W were used as an example).

Method	Description of Network Architecture	Figure	A→W
AlexNet	AlexNet (source only)	-	61.6
DAN (AlexNet)	AlexNet + 2MMD	Figure 2a	68.5
AutoDial (AlexNet)	AlexNet + 3DA layer	Figure 2b	75.5
Ours (AlexNet)	AlexNet + 4MMD + 1DA layer	Figure 1a	77.2
GoogleNet	GoogleNet (source only)	-	70.3
DAN (GoogleNet)	GoogleNet + 4MMD	Figure 2c	76.0
JAN (GoogleNet)	GoogleNet + 4JMMD + 1DA layer	Figure 2e	81.5
AutoDial (GoogleNet)	GoogleNet + 69DA layer	Figure 2d	84.2
Ours (GoogleNet_v1)	GoogleNet + 4MMD + 4DA layer	Figure 2f	80.4
Ours (GoogleNet)	GoogleNet + 4MMD + 1DA layer	Figure 1b	87.1

Comparisons with AutoDial: Figure 3 shows the result of the A→D domain adaptations to compare the accuracy and loss of the proposed method and those of AutoDial

during the testing phase. As shown in Figure 3, the accuracy curve of DA-MM is substantially higher than that of AutoDial, while the loss curve of DA-MM is lower than that of AutoDial. Combined with the conclusions drawn in Figure 3, the proposed method partially alleviates the difficulty of domain adaptation compared with AutoDial. These results show that although AutoDial can automatically align across domains, the degree of alignment can be further improved. The proposed method uses MMD to improve the domain alignment degree based on automatic domain alignment, which further reduces the distribution discrepancy between domains. In addition, comparing the network architecture of AutoDial (Figure 2d) and the proposed method (Figure 1b), which has $69 * 2 = 138$ DA layers, the proposed method requires only one DA layer. Therefore, although the network structure of the proposed method is simpler, it achieves better results.

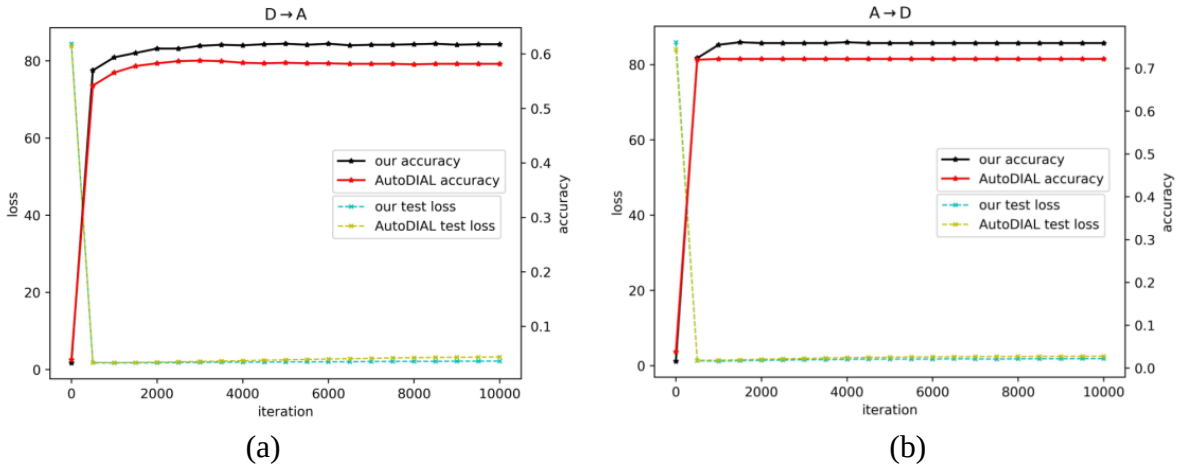


Figure 3. Comparison with the AutoDial method: (**a**) accuracy and test loss for task D→A; (**b**) accuracy and test loss for task A→D.

5.5. Parameter Sensitivity

We also conducted sensitivity tests to investigate the effects of the parameters λ and γ. Figure 4 illustrates the changes in the transfer classification performance as $\lambda \in \{0, 0.2, \ldots, 1\}$ and $\gamma \in \{0.1, 0.5, 1, 2, 3, \ldots, 10\}$ on the A→W task. The accuracy of the proposed method first increases and then decreases as λ and γ vary, forming a bell-shaped curve. This result confirms a good trade-off between the standard log-loss applied to the source samples and the entropy loss applied to the target samples, and the MMD distance can enhance feature transferability.

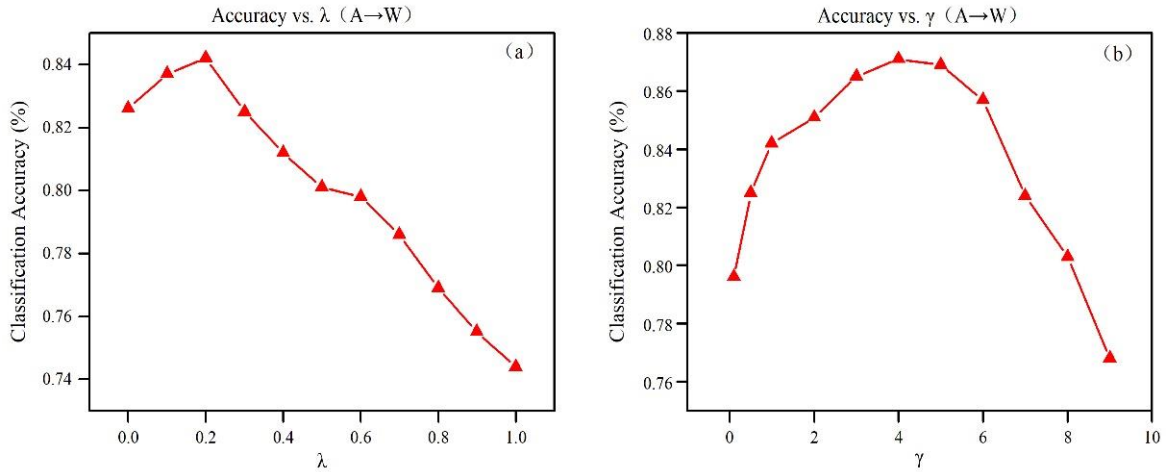


Figure 4. Sensitivity of parameters λ and γ: (**a**) accuracy vs. λ; (**b**) accuracy vs. γ.

6. Conclusions

In this paper, we propose an automatic domain alignment and moment matching (DA-MM) approach for deep domain adaptation. DA-MM aims to reduce the discrepancy between the source and target domain and improve the domain adaptation performance. Our results show that the combination of the moment matching method and the automatic domain alignment method can further reduce domain discrepancy and can significantly outperform several state-of-the-art domain adaptation methods. Although we combine the advantages of the two methods in this article, we introduce new hyperparameters that need to be manually adjusted while improving the performance. This problem requires us to continue to explore and optimize in future work. In addition, the method we proposed in this paper is a combination strategy, which can be applied more widely, not limited to the moment matching method mentioned in the article is only represented by MMD; the automatic domain alignment method is represented by AutoDial, and these two modules can both be replaced by any similar method to achieve similar effects and performance, which also requires further verification in the next work.

Author Contributions: Formal analysis, C.M. and J.M.; Funding acquisition, C.X. and W.S.; Methodology, J.Z. and C.M.; Writing—original draft, J.Z.; Writing—review & editing, J.Z. and J.M. All authors have read and agreed to the published version of the manuscript.

Funding: The CCF-Ant Research Fund (CCF-AFSG RF20200014), the Talent Innovation and Entrepreneurship Fund of Lanzhou (2020-RC-13), the Science and Technology Project of Gansu (21YF5GA102, 21YF5GA006, 21ZD8RA008), the Science and Technology Planning Project of Gansu Province (Technology Innovation Guidance Program, grant number 20CX9NA105).

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: Not applicable.

Acknowledgments: We would like to thank the editors and anonymous reviewers for their constructive and detailed comments on earlier versions of the manuscript.

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

References

1. LeCun, Y.; Bengio, Y.; Hinton, G. Deep learning. *Nature* **2015**, *521*, 436–444. [CrossRef] [PubMed]
2. Chui, C.K.; Lin, S.-B.; Zhou, D.-X. Deep neural networks for rotation-invariance approximation and learning. *Anal. Appl.* **2019**, *17*, 737–772. [CrossRef]
3. Weibo, L.; Zidong, W.; Xiaohui, L.; Nianyin, Z.; Yurong, L.; Fuad, E.A. A survey of deep neural network architectures and their applications. *Neurocomputing* **2017**, *234*, 0925–2312.
4. Tahmoresnezhad, J.; Hashemi, S. Visual domain adaptation via transfer feature learning. *Knowl. Inf. Syst.* **2017**, *50*, 585–605. [CrossRef]
5. Jie, L.; Vahid, B.; Peng, H.; Hua, Z.; Shan, X.; Guangquan, Z. Transfer learning using computational intelligence: A survey. *Knowl-Based Syst.* **2015**, *80*, 14–23.
6. Pan, S.J.; Qiang, Y. A survey on transfer learning. *IEEE Trans. Knowl. Data Eng.* **2010**, *22*, 1345–1359. [CrossRef]
7. Mansour, Y.; Mohri, M.; Rostamizadeh, A. Domain Adaptation: Learning Bounds and Algorithms. *arXiv* **2009**, arXiv:0902.3430.
8. Donahue, J.; Jia, Y.; Vinyals, O.; Hoffman, J.; Zhang, N.; Tzeng, E.; Darrell, T. DeCAF: A Deep Convolutional Activation Feature for Generic Visual Recognition. *arXiv* **2014**, arXiv:1310.1531.
9. 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, 25–27 June 2013; pp. 3515–3522. [CrossRef]
10. Gong, B.; Grauman, K.; Sha, F. Connecting the Dots with Landmarks: Discriminatively Learning Domain-Invariant Features for Unsupervised Domain Adaptation. In Proceedings of the International Conference on Machine Learning, Atlanta, GA, USA, 17–19 June 2013.
11. Zeng, X.; Ouyang, W.; Wang, M.; Wang, X. Deep Learning of Scene-Specific Classifier for Pedestrian Detection. In Proceedings of the European Conference on Computer Vision, Zurich, Switzerland, 6–12 September 2014; pp. 472–487. [CrossRef]
12. Schwab, C.; Zech, J. Deep learning in high dimension: Neural network expression rates for generalized polynomial chaos expansions in UQ. *Anal. Appl.* **2018**, *17*, 19–55. [CrossRef]

13. Yilmaz, S.; Toklu, S. A deep learning analysis on question classification task using Word2vec representations. *Neural Comput. Appl.* **2020**, *32*, 2909–2928. [CrossRef]
14. Sun, Y.; Wang, X.; Tang, X. Deep Learning Face Representation from Predicting 10,000 Classes. In Proceedings of the 2014 IEEE Conference on Computer Vision and Pattern Recognition, Columbus, OH, USA, 24–27 June 2014; pp. 1891–1898.
15. Esteva, A.; Kuprel, B.; Novoa, R.A.; Ko, J.; Swetter, S.M.; Blau, H.M.; Thrun, S. Dermatologist-level classification of skin cancer with deep neural networks. *Nature* **2017**, *542*, 115–118. [CrossRef] [PubMed]
16. Long, M.; Cao, Y.; Wang, J.; Jordan, M.I. Learning Transferable Features with Deep Adaptation Networks. *arXiv* **2015**, arXiv:1502.02791.
17. Long, M.; Zhu, H.; Wang, J.; Jordan, M.I. Unsupervised Domain Adaptation with Residual Transfer Networks. In Proceedings of the Advances in Neural Information Processing Systems, Barcelona, Spain, 5–10 December; 2016.
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.
19. Tzeng, E.; Hoffman, J.; Darrell, T.; Saenko, K. Simultaneous Deep Transfer Across Domains and Tasks. In Proceedings of the IEEE International Conference on Computer Vision, Santiago, Chile, 7–13 December 2015; pp. 4068–4076. [CrossRef]
20. Ganin, Y.; Lempitsky, V.S. Unsupervised Domain Adaptation by Backpropagation. *arXiv* **2015**, arXiv:1409.7495.
21. Ghifary, M.; Kleijn, W.B.; Zhang, M.; Balduzzi, D.; Li, W. Deep Reconstruction-Classification Networks for Unsupervised Domain Adaptation. *arXiv* **2016**, arXiv:1607.03516.
22. Bousmalis, K.; Trigeorgis, G.; Silberman, N.; Krishnan, D.; Erhan, D. Domain Separation Networks. In Proceedings of the Advances in Neural Information Processing Systems, Barcelona, Spain, 5–10 December; 2016.
23. Ioffe, S.; Szegedy, C. Batch Normalization: Accelerating Deep Network Training by Reducing Internal Covariate Shift. In Proceedings of the International Conference on Machine Learning, Lille, France, 6–11 July 2015.
24. Carlucci, F.M.; Porzi, L.; Caputo, B.; Ricci, E.; Bulò, S.R. Just DIAL: DomaIn Alignment Layers for Unsupervised Domain Adaptation. In Proceedings of the International Conference on Image Analysis and Processing, Catania, Italy, 11–15 September 2017; pp. 357–369. [CrossRef]
25. Li, Y.; Wang, N.; Shi, J.; Hou, X.; Liu, J. Adaptive Batch Normalization for practical domain adaptation. *Pattern Recognit.* **2018**, *80*, 109–117. [CrossRef]
26. Carlucci, F.M.; Porzi, L.; Caputo, B.; Ricci, E.; Bulo, S.R. AutoDIAL: Automatic Domain Alignment Layers. In Proceedings of the IEEE International Conference on Computer Vision, Venice, Italy, 22–29 October 2017; pp. 5077–5085. [CrossRef]
27. Meng, C.; Xu, C.; Lei, Q.; Su, W.; Wu, J. Balanced joint maximum mean discrepancy for deep transfer learning. *Anal. Appl.* **2020**, *19*, 491–508. [CrossRef]
28. Tzeng, E.; Hoffman, J.; Zhang, N.; Saenko, K.; Darrell, T. Deep Domain Confusion: Maximizing for Domain Invariance. *arXiv* **2014**, arXiv:1412.3474.
29. Krizhevsky, A.; Sutskever, I.; Hinton, G.E. Imagenet classification with deep convolutional neural networks. *Adv. Neural Inf. Processing Syst.* **2012**, *60*, 84–90. [CrossRef]
30. Long, M.; Zhu, H.; Wang, J.; Jordan, M.I. Deep Transfer Learning with Joint Adaptation Networks. In Proceedings of the International Conference on Machine Learning, Sydney, Australia, 6–11 August 2017.
31. Tzeng, E.; Hoffman, J.; Saenko, K.; Darrell, T. Adversarial Discriminative Domain Adaptation. In Proceedings of the 2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Honolulu, HI, USA, 21–26 July 2017; pp. 2962–2971.
32. Szegedy, C.; Liu, W.; Jia, Y.; Sermanet, P.; Reed, S.; Anguelov, D.; Erhan, D.; Vanhoucke, V.; Rabinovich, A. Going Deeper with Convolutions. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Boston, MA, USA, 8–10 June 2015; pp. 1–9.
33. He, K.; Zhang, X.; Ren, S.; Sun, J. Deep Residual Learning for Image Recognition. In Proceedings of the 2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Las Vegas, NV, USA, 27–30 June 2016.
34. Simonyan, K.; Zisserman, A. Very Deep Convolutional Networks for Large-Scale Image Recognition. *arXiv* **2014**, arXiv:1409.1556.
35. Saenko, K.; Kulis, B.; Fritz, M.; Darrell, T. Adapting Visual Category Models to New Domains. In Proceedings of the European Conference on Computer Vision, Crete, Greece, 5–11 September 2010.
36. Jia, Y.; Shelhamer, E.; Donahue, J.; Karayev, S.; Long, J.; Girshick, R.B.; Guadarrama, S.; Darrell, T. Caffe: Convolutional Architecture for Fast Feature Embedding. In Proceedings of the 22nd ACM International Conference, Mountain View, CA, USA, 18–19 June 2014.
37. Griffin, G.; Holub, A.; Perona, P. *Caltech-256 Object Category Dataset*; California Institute of Technology: Pasadena, CA, USA, 2007.
38. Kim, S.; Kang, I.; Kwak, N. Semantic Sentence Matching with Densely-Connected Recurrent and Co-Attentive Information. *arXiv* **2019**, arXiv:1805.11360. [CrossRef]