Learning Semantic Representations and Discriminative Features in Unsupervised Domain Adaptation

Rushendra Sidibomma
Department of Computer Science and Engineering
Indian Institute of Information Technology
Sri City, India
rushendra.s20@iiits.in

Rakesh Kumar Sanodiya
Department of Computer Science and Engineering
Indian Institute of Information Technology
Sri City, India
rakesh.pcs16@gmail.com

***Abstract*—In domain adaptation, the goal is to train a neural network on the source domain and obtain a good accuracy on the target domain. In such a scenario, it is important to transfer the knowledge from the labelled source domain to the unlabelled target domain due to the expensive cost of manual labelling. Following the trail of works in the recent time, feature level alignment seems to be the most promising direction in unsupervised domain adaptation. In most of the recent works using this feature alignment, the semantic information present in the labelled source domain has not been exploited. Among the works that have tried to learn this semantic representations, the discriminative features have not been taken into consideration which results in lower accuracy on target domain. In this paper, we present a novel approach, joint discriminative and semantic transfer network (JDSTN) that not only aligns the semantic representations of source and target domain, but also enhances the discriminative features and thereby improving the accuracy significantly. This is achieved by using pseudo-labels to align the feature centroids of source and target domains while introducing losses that promote the learning of discriminative features.**

I. Introduction

Traditional machine learning approaches assume that training and testing data are independent and identically distributed. Therefore, supervised machine learning algorithms perform well only when the test data is from the same distribution as the labelled data that has been used for training. This leads to significantly high errors when testing on data from different application domains. One tedious method would be to use labelled data from various application domains so that the model can learn the corresponding features from each domain. However, manual large scaled labelled data on target domain is too expensive or sometimes impossible to collect in practice. This motivates us to find an effective algorithm that can adapt the learned classifier from a source domain with a large number of labelled samples to a target domain with limited or no labelled samples. This is possible because the source and target domains have different, but related distributions. This can be achieved by learning source invariant features from the label-rich source domain and transferring them to the target domain. The single major obstacle with such a learning paradigm is the domain discrepancy. For example, a classifier model trained on manually labelled images, in most scenarios, do not perform well on testing with similar images captured under different lighting conditions or angle.

This dependency on labelled data renders the advancements in deep learning and especially the CNNs somewhat handicapped. This motivates us to find an effective algorithm for unsupervised domain adaptation. There have been numerous attempts to eliminate or at least reduce this domain shift. Traditional domain adaptation methods have aimed to do this before deep learning methods changed the game. Most of these methods use the features extracted from the raw images and the approaches can be classified into feature selection, distribution adaptation or subspace learning. Feature selection methods predominantly used speeded up robust features (SURF) as extracted features. Structural correspondence learning (SCL) [1] was a very popular representational model to find the common features across multiple domains. It selects the pivot features and uses them to find a common low dimensional feature space. MGSA [2] reduces the semantic gap between visual features and semantics using joint feature selection and feature adaptation. FFSL [3] integrated structure preservation and feature selection. Distribution adaptation methods commonly incorporated MMD [4], Reproducing Kernel Hilbert Space (RKHS) and Transfer Component Analysis (PCA) to align the distributions. Subspace learning methods have been discussed in later sections.

These early methods of domain adaptation focused solely on aligning the source and target domain distributions (marginal and conditional). However, despite this reduction in domain variation, classification accuracies did not match the expectations. This is because even with perfect alignment of source and target domain distributions, there is no guarantee that the samples from the same class will map nearby in the feature space. This led to a new paradigm namely adversarial domain adaptation methods. These are analogous to GANs [5]. Various measures have been used to the align these features. However, we focus on the semantic alignment of features, which has yielded excellent results as shown by [6], [7], [8] and [9].

978-1-6654-5572-5/23/$31.00 © 2023 IEEE

A. *Our Contributions*

The merits of this paper are as follows:

- A novel approach namely joint discriminative and semantic transfer network (JDSTN) for unsupervised domain adaptation that not only aligns the semantic representations of the features in source and target domains but also enforces the learning of discriminative features by making the class clusters in feature space more compact. The model also makes use of the pseudo labels in order to reduce the conditional entropy in target domain, leading to discriminative clusters in target domain.
- We compare our new approach with other state-of-the art approaches on a benchmark dataset using the same architecture to get unbiased results. An extensive analysis has also been done on our model using ablation studies and t-SNE plots.

II. Related Works

This notion of transfer learning and domain adaptation has been around for a while. The need for transferring knowledge from one domain to the other can be linked to dataset bias. Dataset bias can be described as some inherent bias in the data collected, which causes some features or structural characteristics to be over-represented or over-weighted. Due to this, the data distribution does not resemble the actual distribution found in the real world and causes skewed outcomes and low accuracy. This has been popularized by [10]. Over the course of the last decade, there have been relentless works in this domain covered by [11] and [12]. From these surveys, the common approaches to domain adaptation can be summarized in the following part.

Domain discrepancy based methods typically try to reduce or eliminate the distance between the distributions of source and target domains. Such approaches have resulted in increased accuracies as seen in [4], [13] and [14]. [4] uses Maximum Mean Discrepancy (MMD) whereas [13] uses a multi kernel MMD to align the distributions. [14] approaches this by matching the second order statistics of the distributions. The recent work of [15] can be used to align any arbitrary k^{th} order statistics. Despite having yielded a higher accuracy each time, these methods are considered ”shallow” because aligning the distributions do not lead to discriminative features in the subspace.

In the recent times, adversarial learning has been widely used for unsupervised domain adaptation [16], [17], [18] and [19]. These methods involve training a discriminator to classify the domain of the object and the feature extractor has to fool the discriminator. Such an adversarial training scenario ensures that the feature extractor extracts the desired information from the input. The reason that this approach became widely used is because it enforces alignment at the class-level instead of the domain-level. This allows us more control over what we wish to enforce upon the feature extractor. However, this does not solve the problem completely because the absence of labels in target domain makes it extremely difficult to align the features.

Many recent works have tried to make the best use the limited target labels available to transfer the semantic knowledge. Few-shot adversarial learning methods have been explored by [20], [21] and [22]. [17] uses the average output probability with source samples for each class and then for each target sample. [21] uses cross category similarity for semantic transfer. However, we consider a more challenging problem, where there are no labelled target samples. some of the works like [23] and [24] have toiled on this issue. However, we approach this issue by using pseudo labels and instead of assuming that they are correct, the model tries to align the shift brought by them as proposed in [6].

III. Methodology

This section describes the model, algorithm and the experimental setup in detail.

A. *Problem Definition*

We start by defining the notations used throughout the following sections. As a part of our dataset, let us consider a labelled source domain and an unlabelled target domain (the target labels are not provided to the model at any stage of the algorithm except while evaluating it). Let the labelled data from the source domain be denoted by $\{X_s, Y_s\}$ where $X_s \in \mathbb{R}^{D \times n_s}$ sampled from the distribution χ_s, $Y_s \in \mathbb{R}^{1 \times n_s}$ and the unlabelled data from target domain denoted by $X_t \in \mathbb{R}^{D \times n_t}$, which has been sampled from the distribution χ_t. Here, χ_s denotes the source distribution, χ_t denotes the target distribution, D denotes the dimensionality of a data point i.e. the number of features it has and n_s, n_t respectively denote the number of the source and target samples available. Further, χ_s and χ_t are unequal but related. We also assume that the domain of features and labels of the source samples are respectively equal to the domain of features and source of the target samples. The goal is to train the model using only the source domain and obtain a high accuracy on the target domain by ensuring the transfer of knowledge by learning domain invariant features.

B. *Model Description*

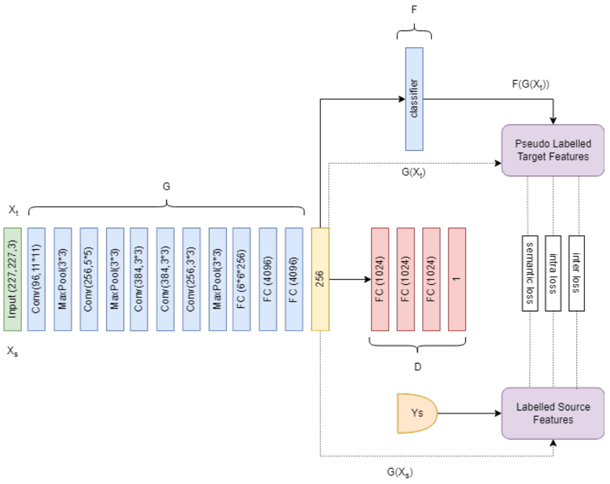
The model consists of the following networks:

1) Feature Extractor (G): A typical Alexnet architecture [25] has been used as the feature extractor. The output of this network are the deep features of the images and these features are then flattened and connected another pair of fully connected layers in order to safely transfer the features. These are the features that are further used for calculating the semantic loss, intra-class variance gap and the inter-class variance gap. All the losses propagate backwards through this network in order to promote the learning of source invariant and discriminiative features.

2) Classifier (f): The classifier takes as input the deep features extracted by G and outputs the class of the input image. The exact architecture has been shown in figure 1.

3) Discriminator (D): This network is trained to classify the domain of the input image based on the deep features extracted by the generator. This adversarial training of G and D trains the generator to extract domain invariant features which can fool the discriminator. This is one of the most promising approaches to enforce domain invariance in the recent line of works.

In this section, the working of the model has been explained in detail. In every iteration, there are two batches of data, one from the source domain and the other from the target domain. They are passed through the feature extractor, which outputs the features of these inputs in a high dimensional feature subspace. Since that target labels are not present, it will not be possible to enforce any type of alignment at this stage. Therefore, the model assigns pseudo-labels by passing the features of target images through the classifier. There is a possibility that a fraction of these pseudo-labels will be incorrect each time but their effect is reduced because they are suppressed by the correctly labelled ones. This is done by aligning the centroids of the source and target domain features as in [6]. On top of this, these newly pseudo-labelled features are used to increase the distance between feature points of different classes and reduce the distance between the feature points of same classes. The conditional target entropy further ensures dense class clusters.



(a)

Fig. 1: (**a**) Proposed model architecture: G is the feature extractor, F is the classifier and D is the discriminator. X_s and X_t are the source and target images respectively. Y_s represents the labels of the source domain.

C. Formulating the Loss Function

The final loss function consists of the following terms:

1) Classification Loss: The classification loss on the source domain is represented by

$$L_c = \frac{1}{n_s}\sum_{i=1}^{n_s} J(f_\theta(\mathbf{x}_i^s), \mathbf{y}_i^s)) \tag{1}$$

where L_c represents the classification loss in source domain, $J(.,.)$ represents the cross entropy loss, $\mathbf{x}_i^s$ and $\mathbf{y}_i^s$ represent an image in the source domain and the corresponding label.

2) Minimizing discrepancy between source and target domain: There are different possible measures to calculate the domain discrepancy which are covered by [4] and [15]. We chose to use to domain adversarial loss. In this method, an additional network is trained, called the discriminator (D) which is used to classify the domain based on the features extracted from the feature extractor (G). Both of these networks G and D are trained to fool each other and this adversarial training forces G into learning to extract domain invariant features from the images of source domain.

$$\begin{aligned} d(X_s, X_t) = &\mathbb{E}_{x\sim D_s}\left[log(1 - D \circ G(x))\right] \\ &\mathbb{E}_{x\sim D_t}\left[log(D \circ G(x))\right] \end{aligned} \tag{2}$$

3) Minimizing semantic loss: Learning semantic representations ensures that the features of same class in different domains are mapped nearby. Minimizing this loss is crucial despite minimizing the domain discrepancy loss because aligning domains does not guarantee that domain invariant features are learnt. For this, the centroid alignment loss proposed by [6] has been used. This is done by aligning the deep-feature centroids of the source and target domain. Since this is a scenario of unsupervised domain adaptation, where the ground truth for the target domain are absent, we rely on the pseudo labels. These are the predictions of the model as of that iteration. The centroids of each class for each domain are calculated. Thereafter, the semantic loss is defined as,

$$L_{sem} = \sum_{k=1}^{K} \Phi(\mathbf{C}_s^k, \mathbf{C}_t^k) \tag{3}$$

where,

$$\mathbf{C}_s^k = \frac{1}{n_s}\sum_{x_i \in X_s} G(x_i) \tag{4}$$

and

$$\mathbf{C}_t^k = \frac{1}{n_t}\sum_{x_i \in X_t} G(x_i) \tag{5}$$

$\mathbf{C}_s^k$ is the centroid of class k in the sourse domain and $\mathbf{C}_t^k$ is the centroid of class k in target domain and Φ is the distance measure. JDSTN uses Euclidean distance as the similarity measure.

4) Maximizing the inter-class variance gap: This term promotes discriminative feature learning by distancing the samples that belong to different classes.

$$d_{inter}(\mathbf{h}_i^s, \mathbf{h}_j^s) = max(0, m_2 - ||\mathbf{h}_i^s - \mathbf{h}_j^s||_2)^2 \tag{6}$$

where d_{inter} represents the inter-class variance gap loss, $\mathbf{h}_i^s$, $\mathbf{h}_j^s$ respectively represent the deep features of the i^{th} sample x_i and the j^{th} sample x_j of the source domain such that x_i and x_j belong to different classes. When d_{inter} for each pair of samples from the source domain is summed, the inter-class variance loss is computed as,

$$L_{inter}(\mathbf{h}_i^s, \mathbf{h}_j^s) = \sum_{i,j=1,y_i \neq y_j}^{n_s} max(0, m_2 - ||\mathbf{h}_i^s - \mathbf{h}_j^s||_2)^2 \quad (7)$$

5) Minimizing the intra-class variance gap: This term promotes discriminative feature learning by bringing close the samples that belong to the same class.

$$d_{intra}(\mathbf{h}_i^s, \mathbf{h}_j^s) = max(0, ||\mathbf{h}_i^s - \mathbf{h}_j^s||_2 - m_1)^2 \quad (8)$$

where d_{intra} represents the inter-class variance gap loss, $\mathbf{h}_i^s$ and $\mathbf{h}_j^s$ respectively represent the deep features of the i^{th} sample x_i and the j^{th} sample x_j of the source domain such that x_i and y_i belong to same class. When d_{intra} for each pair of samples from the source domain is summed, the intra-class variance loss is computed as,

$$L_{intra}(\mathbf{h}_i^s, \mathbf{h}_j^s) = \sum_{i,j=1,y_i = y_j}^{n_s} max(0, ||\mathbf{h}_i^s - \mathbf{h}_j^s||_2 - m_1)^2 \quad (9)$$

6) Minimizing the target conditional entropy: Despite the discriminative features enforced by L_{inter} and L_{intra}, the decision boundary might be influenced by some data points lying in the low density regions (i.e. the samples which are not classified with high probability). To prevent such a scenario, it is necessary to introduce a conditional entropy loss on the target domain. Minimizing this loss ensures that the decision boundary does not cross high density data regions.

$$L_{ent} = \frac{1}{n_t} \sum_{i=1}^{n_t} \sum_{j=1}^{K} -p_j log(p_j) \quad (10)$$

where K is the number of classes, n_t is the number of target samples, p_j is the softmax-output of the j^{th} output node.

D. Final Loss Function

In order to form the objective function, the terms obtained from (1), (2), (3), (7), (9), (10) are combined as-

$$L = L_c + L_d + L_{sem} + L_{inter} + L_{intra} + L_{ent} \quad (11)$$

(11) sums up all the required loss terms, This function is invoked after every iteration in order to get a polished accuracy.

E. Algorithm

The pseudo code of the algorithm is given in Algorithm 1.

IV. Experiment

In this section, we evaluate the performance of JDSTN on a benchmark dataset along with other primitive and related approaches for unsupervised domain adaptation.

Algorithm 1: Pseudocode for JDSTN

```
Input: Labelled source data S, unlabelled target data
  T, batch size N, feature extractor G, classifier f,
  global centroids $C_S^k$ and $C_T^k$, K is the number of
  classes and $\theta$ is the moving average coefficient.
Output:
while $Iterations > 0$ do
    $X_S$, $Y_S$ = RandomSample(S, N)
    $X_T$ = RandomSample(T, N)
    $Y_T$ = PseudoLabelling(G, f, $X_T$)
    $\mathbf{h}^S$ = G($X_S$)
    $\mathbf{h}^T$ = G($X_T$)
    $L_{sem}$, $L_{inter}$, $L_{intra}$, $L_{ent}$ = 0
    for k=1 to k=K do
        $\mathbf{C}_S^k \leftarrow \theta \mathbf{C}_S^k + (1-\theta)\ \mathbf{C}_S^k$
        $\mathbf{C}_T^k \leftarrow \theta \mathbf{C}_T^k + (1-\theta)\ \mathbf{C}_T^k$
        $L_{sem} \leftarrow L_{sem} + \phi(C_S^k, C_T^k)$
    $L_{inter} = \sum_{i,j=1,y_i=y_j}^{n_S} max(0, m_2 - ||\mathbf{h}_i^S - \mathbf{h}_j^S||_2)^2$
    $L_{intra} = \sum_{i,j=1,y_i=y_j}^{n_S} max(0, ||\mathbf{h}_i^S - \mathbf{h}_j^S||_2 - m_1)^2$
    $L_{ent} = \frac{1}{n_T} \sum_{i=1}^{n_T} \sum_{j=1}^{c} -p_j log(p_j)$
```

A. Dataset

Office 31 has been used to test our approach and also compare it with other models.

Office 31: The dataset contains 31 classes of objects across three domains: Amazon, Webcam and DSLR. The classes in the dataset comprise of objects that are commonly confronted in the general office setting such as backpacks, desk chairs, trashcan, staplers and so on. The Amazon domain consists of product images listed on the e-commerce website and therefore they are captured against a white background without any noise. It has a total of 2817 images and an average of 90 images per category, making it the biggest of all three domains. The Webcam domain consists of images captured using a low resolution camera (640*480 pixels). This, along with the fact that these images were captured under normal lighting conditions in the presence of other objects in the background leads to a significant amount of noise in the images. There are an average of 25 images per category in this domain. Finally, the DSLR domain consists of images captured using a high resolution camera (4288*480 pixels) which exhibit minimal noise. This domains consists of 498 images with an average of 15 images per category. Despite the imbalance of data available across the domains, Office 31 has been a benchmark in domain adaptation.

B. Experiment Setup

An overview of the model and architecture has been explained in section III-B and figure 1. For our experiment on Office-31 dataset, the input images have been scaled to 227*227 pixels and the features extracted from the generator are flattened and passed through the fully connected layers of size 4096 each. The model beings with a pretrained Alexnet

TABLE I: Classification accuracies (in %) on Office 31 datasets. The column headings are in the form source → target

Model	$A \rightarrow W$	$A \rightarrow D$	$W \rightarrow A$	$W \rightarrow D$	$D \rightarrow A$	$D \rightarrow W$	Avg.
Alexnet [25]	61.6	63.8	49.8	99.0	51.1	95.4	70.1
DDC [4]	61.8	64.4	52.2	98.5	52.1	95.0	70.6
DRCN [23]	68.7	66.8	54.9	99.0	56.0	96.4	73.6
RevGrad [16]	73.0	72.3	51.2	99.2	53.4	96.4	74.3
RTN [26]	73.3	71.0	51.0	**99.6**	50.5	96.8	73.7
JAN [27]	74.9	71.8	55.0	99.5	58.3	96.6	76.0
AutoDIAL [28]	75.5	73.6	59.4	99.5	58.1	96.6	77.1
MSTN [6]	75.8	72.7	55.3	99.2	61.2	96.6	76.7
JDSTN (ours)	**78.2**	**76.7**	**57.8**	99.2	**62.1**	**97.3**	**78.5**

and its layers are finetuned as the experiment progresses. The discriminator consists of two fully connected layers made up of 1024 neurons each. The value of θ, which is the moving average coefficient has been set at 0.7 as proposed in MSTN [6]. The adaptation factor, which is the weight balance of semantic loss changes after iteration as $\lambda = \frac{2}{1+e^{-10.x}}$ where x is the progress of the training from 0 to 1. Stochastic Gradient Descent with 0.9 momentum is used along with a learning rate annealed as $\mu = \frac{\mu_0}{(1+10.x)^{0.75}}$ with μ_0 = 0.01 as proposed in [16]. The batch size is set at 100.

C. Comparision

We compared the performance of JDSTN with several other primitive and related state-of-the-art domain adaptation approaches such as AlexNet [25], DDC [4], RevGrad [16], JAN [27], AutoDIAL [28] and MSTN [6].

DDC [4] starts with a typical trained Alexnet [25] architecture and later introduces an adaptation layer between the existing layers of the network where the Maximum Mean Discrepancy (MMD) is the least, which regularizes the classifier and promotes the learning of source invariant features. [16] proposes an architecture with a classifier and a domain classifier which is connected to the feature extractor via a gradient reversal layer that scales the gradient during backward propagation in order to promote domain invariant feature distributions. [27] proposes a different metric called the Joint Maximum Mean Discrepancy (JMMD) which uses the Hilbert space embedding of the distributions to measure the discrepancy of joint domain alignment. [28] also relies on additional domain adaptation layer to reduce the domain discrepancy however, the combination of entropy loss and this new layer aligns the source and domain distributions in to the extent required at each layer. [6] uses semantic alignment of source and target features using the concept of centroid alignment. Our semantic model is built on top of [6], with the inclusion of losses which promote the learning of discriminative features along with the formation of discriminative clusters in the target domain.

All the models were trained until convergence before noting down the accuracies. The results have been tabulated in I.

D. Results

In this section, the performance of our algorithm is compared with other Unsupervised Domain Adaptation approaches mentioned above. The training and testing methodology is similar to what has been proposed by the respective models.

Table I shows the accuracy results of JDSTN with respect to other approaches for all tasks of the Office 31. As seen in the table, our proposed algorithm performs better across all the tasks and the average accuracy of the tasks is the highest, compared to the other algorithms that focus solely on the domain alignment. It also surpasses the classification accuracy of MSTN which aligns the semantic representations of features but fails to promote discriminative features.

E. Analysis

In this section, we perform some quantitative analysis on our model in order to further consolidate the experimental results.

1) Ablation Study: The accuracies of the model has been noted, when each of the loss term is removed one by one. This is done to observe the possibility of the algorithm having redundant loss functions. This ablation study has been performed on the task A → W. First the target entropy (L_{ent}) is removed and this drops the accuracy to 75% because without it, the target features do not form discriminative clusters. Next, the inter class variance loss is removed and the accuaracy comes down to 77.5% because the features of each class are not discriminative enough. Thereafter, the intra class variance loss is removed and this lowers the accuracy to 75.1%. This is is because the class clusters formed are not compact enough and therefore the classifier does not have a clear decision boundary. On removing both intra and inter class variance loss, the accuracy drops down to 74%. Finally, The final accuracy with all the loss terms intact, comes down to 78.3% which is significantly high compared to other accuracies used in the ablation study. The summary of this study is shown in II.

TABLE II: Classification accuracies (in %) on Office 31 datasets without each of the loss terms

Model	$A \rightarrow W$
Without target entropy (L_{ent})	75%
Without inter-class loss (L_{inter})	77.5%
Without intra-class loss (L_{intra})	75.1%
Without inter and intra loss	74%
Including all terms	**78.2**%

2) t-SNE Plot: The success of JDSTN is mainly due to the discriminative features learnt by the classifier along with the semantic alignment. This leads to a better decision boundary for the classifier after learning domain invariant features. In order to show a visual representation of this effect, t-Distributed Stochastic Neighbor Embedding (t-SNE) has been used to visualise the high dimensional features in a plane.

From 2, it can be clearly observed that the features of each class are compactly mapped, but distant from those of other classes. This results in a higher accuracy than the other discussed domain adaptation techniques.

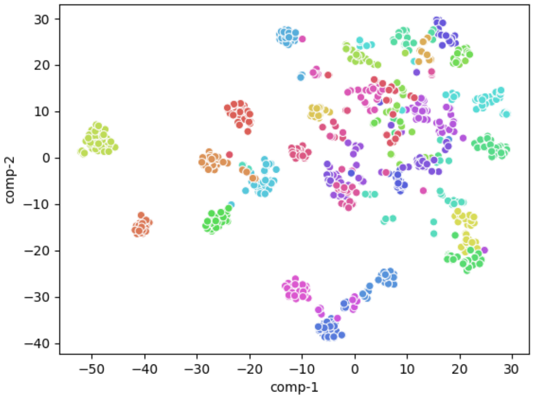


Fig. 2: (**a**) t-SNE plot of target features (after convergence)

V. Conclusion

In this paper, we have proposed a robust and improved approach for unsupervised domain adaptation. The algorithm learns to align the semantic representations of the domain invariant features. It also promotes the learning of discriminative features which ensure that the features of similar classes are closer to each other and the features of different classes are further away from each other. Such an approach has not been explored by any other domain adaptation algorithms (in this setting). Furthermore, the ablation study and the t-SNE plot show that each of the losses introduced are independent and have a strong positive effect in determining the final accuracy. Experimental results further consolidate that our proposed method is better over the primitive and other existing approaches.

References

[1] J. Blitzer, R. McDonald, and F. Pereira, "Domain adaptation with structural correspondence learning," in *Proceedings of the 2006 conference on empirical methods in natural language processing*, 2006, pp. 120–128.

[2] E. Rashedi, H. Nezamabadi-Pour, and S. Saryazdi, "A simultaneous feature adaptation and feature selection method for content-based image retrieval systems," *Knowledge-Based Systems*, vol. 39, pp. 85–94, 2013.

[3] J. Li, J. Zhao, and K. Lu, "Joint feature selection and structure preservation for domain adaptation." in *IjCAI*, 2016, pp. 1697–1703.

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

[5] I. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. Courville, and Y. Bengio, "Generative adversarial networks," *Communications of the ACM*, vol. 63, no. 11, pp. 139–144, 2020.

[6] S. Xie, Z. Zheng, L. Chen, and C. Chen, "Learning semantic representations for unsupervised domain adaptation," in *International conference on machine learning*. PMLR, 2018, pp. 5423–5432.

[7] S. Li, M. Xie, F. Lv, C. H. Liu, J. Liang, C. Qin, and W. Li, "Semantic concentration for domain adaptation," in *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2021, pp. 9102–9111.

[8] H. Li, Y. Chen, D. Tao, Z. Yu, and G. Qi, "Attribute-aligned domain-invariant feature learning for unsupervised domain adaptation person re-identification," *IEEE Transactions on Information Forensics and Security*, vol. 16, pp. 1480–1494, 2020.

[9] S. Zhao, B. Li, X. Yue, Y. Gu, P. Xu, R. Hu, H. Chai, and K. Keutzer, "Multi-source domain adaptation for semantic segmentation," *Advances in Neural Information Processing Systems*, vol. 32, 2019.

[10] A. Torralba and A. A. Efros, "Unbiased look at dataset bias," in *CVPR 2011*, 2011, pp. 1521–1528.

[11] 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.

[12] F. Zhuang, Z. Qi, K. Duan, D. Xi, Y. Zhu, H. Zhu, H. Xiong, and Q. He, "A comprehensive survey on transfer learning," *Proceedings of the IEEE*, vol. 109, no. 1, pp. 43–76, 2020.

[13] J. Long, E. Shelhamer, and T. Darrell, "Fully convolutional networks for semantic segmentation," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2015, pp. 3431–3440.

[14] B. Sun and K. Saenko, "Deep coral: Correlation alignment for deep domain adaptation," in *European conference on computer vision*. Springer, 2016, pp. 443–450.

[15] C. Chen, Z. Fu, Z. Chen, S. Jin, Z. Cheng, X. Jin, and X.-S. Hua, "Homm: Higher-order moment matching for unsupervised domain adaptation," in *Proceedings of the AAAI conference on artificial intelligence*, vol. 34, no. 04, 2020, pp. 3422–3429.

[16] Y. Ganin and V. Lempitsky, "Unsupervised domain adaptation by backpropagation," in *International conference on machine learning*. PMLR, 2015, pp. 1180–1189.

[17] E. Tzeng, J. Hoffman, T. Darrell, and K. Saenko, "Simultaneous deep transfer across domains and tasks," in *Proceedings of the IEEE international conference on computer vision*, 2015, pp. 4068–4076.

[18] J. Hoffman, E. Tzeng, T. Park, J.-Y. Zhu, P. Isola, K. Saenko, A. Efros, and T. Darrell, "Cycada: Cycle-consistent adversarial domain adaptation," in *International conference on machine learning*. Pmlr, 2018, pp. 1989–1998.

[19] S. Motiian, M. Piccirilli, D. A. Adjeroh, and G. Doretto, "Unified deep supervised domain adaptation and generalization," in *Proceedings of the IEEE international conference on computer vision*, 2017, pp. 5715–5725.

[20] S. Motiian, Q. Jones, S. Iranmanesh, and G. Doretto, "Few-shot adversarial domain adaptation," *Advances in neural information processing systems*, vol. 30, 2017.

[21] Z. Luo, Y. Zou, J. Hoffman, and L. F. Fei-Fei, "Label efficient learning of transferable representations acrosss domains and tasks," *Advances in neural information processing systems*, vol. 30, 2017.

[22] Y. Sun, E. Tzeng, T. Darrell, and A. A. Efros, "Unsupervised domain adaptation through self-supervision," *arXiv preprint arXiv:1909.11825*, 2019.

[23] M. Ghifary, W. B. Kleijn, M. Zhang, D. Balduzzi, and W. Li, "Deep reconstruction-classification networks for unsupervised domain adaptation," in *European conference on computer vision*. Springer, 2016, pp. 597–613.

[24] K. Bousmalis, G. Trigeorgis, N. Silberman, D. Krishnan, and D. Erhan, "Domain separation networks," *Advances in neural information processing systems*, vol. 29, 2016.

[25] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "Imagenet classification with deep convolutional neural networks," *Communications of the ACM*, vol. 60, no. 6, pp. 84–90, 2017.

[26] M. Long, H. Zhu, J. Wang, and M. I. Jordan, "Unsupervised domain adaptation with residual transfer networks," *Advances in neural information processing systems*, vol. 29, 2016.

[27] ——, "Deep transfer learning with joint adaptation networks," in *International conference on machine learning*. PMLR, 2017, pp. 2208–2217.

[28] F. Maria Carlucci, L. Porzi, B. Caputo, E. Ricci, and S. Rota Bulo, "Autodial: Automatic domain alignment layers," in *Proceedings of the IEEE international conference on computer vision*, 2017, pp. 5067–5075.