

Article

Information Separation Network for Domain Adaptation Learning

Zeqing Zhang ^{1,2,*}, Zuodong Gao ^{1,†}, Xiaofan Li ^{1,†}, Cuihua Lee ¹ and Weiwei Lin ³ 

¹ School of Informatics, Xiamen University, Xiamen 361000, China; 23020191153236@xmu.edu.cn (Z.G.); lxf1293763074@foxmail.com (X.L.); chli@xmu.edu.cn (C.L.)

² School of Geosciences and Engineering, West Yunnan University of Applied Sciences, Dali 671006, China

³ School of Big Data and Artificial Intelligence, Fujian Polytechnic Normal University, Fuqing 350300, China; linww_cn@hotmail.com

* Correspondence: zhang_ze_qing@163.com; Tel.: +86-1396-071-1435

† These authors contributed equally to this work.

Abstract: The Bai People have left behind a wealth of ancient texts that record their splendid civilization, unfortunately fewer and fewer people can read these texts in the present time. Therefore, it is of great practical value to design a model that can automatically recognize the Bai ancient (offset) texts. However, due to the expert knowledge involved in the annotation of ancient (offset) texts, and its limited scale, we consider that using handwritten Bai texts to help identify ancient (offset) Bai texts for handwritten Bai texts can be easily obtained and annotated. Essentially, this is a problem of domain adaptation, and some of the domain adaptation methods were transplanted to handle ancient (offset) Bai text recognition. Unfortunately, none of them succeeded in obtaining a high performance due to the fact that they do not solve the problem of how to separate the style and content information of an image. To address this, an information separation network (ISN) that can effectively separate content and style information and eventually classify with content features only, is proposed. Specifically, our network first divides the visual features into a style feature and a content feature by a separator, and ensures that the style feature contains only style and the content feature contains only content by cross-domain cross-reconstruction; thus, achieving the separation of style and content, and finally using only the content feature for classification. This greatly reduces the impact brought by cross-domain. The proposed method achieves leading results on five public datasets and a private one.

Keywords: Bai character; domain adaptation; deep learning



Citation: Zhang, Z.; Gao, Z.; Li, X.; Lee, C.; Lin, W. Information Separation Network for Domain Adaptation Learning. *Electronics* **2022**, *11*, 1254. <https://doi.org/10.3390/electronics11081254>

Academic Editor: Arkaitz Zubiaga

Received: 31 March 2022

Accepted: 12 April 2022

Published: 15 April 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

The Bai People are the 15th largest national minority in China, mainly living in Yunnan, Guizhou and Hunan provinces, with the majority of the population gathered in the Dali Bai Autonomous Prefecture of Yunnan. The Bai text, as a unique Bai script, on one hand, has carried the splendid culture of the Bai People for thousands of years, most of which have been passed down through ancient books or offset printing. On the other hand, fewer and fewer modern people know them, let alone recognizing them. Hence, in the perspective of practicality, it is of great value to design a model that can recognize ancient (offset) Bai texts.

With the renaissance of deep neural networks [1,2], more and more computer vision tasks have achieved remarkable breakthroughs, yet the data-starved nature of deep models prevents them from obtaining a good result when less training data is available. The current situation is that the annotation of ancient (offset) Bai text requires expert knowledge and the difficulty to collect them is self-evident, and a superior performing recognition network cannot be trained, using only this data as the training set. Fortunately, a large handwritten Bai text dataset has been collected by Zeqing et al. [3]. Therefore, this handwritten Bai text dataset and part of the unlabeled ancient book (offset) dataset are used to jointly train a deep learning model capable of recognizing ancient book (offset) Bai text.

Obviously, the handwritten Bai text and antique (offset) Bai text have the same content in different styles, and from a deep learning perspective, this task can be treated as a cross-domain adaptation problem. Now there are many mature models in the domain adaptation, and eight latest methods were reported to tackle the problem of cross-domain adaptation of Bai text, but the results are not so promising. The truth is that these methods do not have the ability to separate the content and style information of text images directly. For example, DANN [4] expects the visual features extracted to be style-independent, and MCD [5] expects different classifiers to obtain the same labels in the target domain. Therefore, the visual features obtained by these methods cannot completely remove the style information and, in return, will be affected by different domains.

To overcome this, an information separation network (ISN) that can effectively separate content and style information and eventually classify with content features only, is proposed, greatly reducing the influence brought by different domains. Based on the fact that the original visual features contain not only content information, a separator to separate content and style information in visual features into content and style features is designed, as well as a combiner to combine content and style features back into visual features. In addition, a discriminator that can be used to distinguish between different styles of visual features, ensures that the combined visual features contain the original content and style features used in the combination, thus, finally completing the separation of content and style information. Last but not least, the proposed method achieves the state-of-the-art (SOTA) results on a private dataset and five public ones.

The contribution is threefold:

- A Bai text cross-domain adaptive dataset using an existing Bai handwritten dataset and our own collection of ancient (offset) Bai text datasets is constructed.
- An information separation network (ISN) that can effectively separate content and style information in visual features is designed.
- The proposed method achieves the state-of-the-art (SOTA) results on a private dataset and five public ones.

2. Related Work

2.1. Bai Text Recognition

Zeqing et al. [3], as a pioneer in using deep learning to solve the Bai text recognition problem, collected a dataset with 400 Bai characters in different handwriting, and an average of roughly 2000 samples per Bai text, which is a huge dataset of roughly 800,000 images. In their work, Zeqing et al. considered the use of Chinese characters similar to Bai characters to improve the model's Bai character recognition capability through knowledge transfer [6], and achieved remarkable results. However, the application value of their research was greatly limited, for they focused on the recognition of handwritten characters only, but the real application scenario is more about the recognition of ancient (offset) Bai characters. While we take a more direct approach to the recognition of ancient (offset) Bai characters, which has more practical value.

2.2. Domain Adaptation

Domain adaptation aims to transfer the knowledge from a source domain to a target domain. DANN [4], a pioneer in solving the domain adaptation problem, provides a model that can constrain the network with a discriminator so that the features extracted from the source and target domains are as similar as possible, eliminating the effect of different domains. DWL [7] proposes a dynamic weighted learning method (DWL) for domain adaptation. By monitoring the degree of alignment and discriminability in real-time, their method dynamically adjusts the weight of alignment learning and discriminability learning, so as to avoid excessive alignment or excessive pursuit of discriminability. ADDA [8] first learns a discriminative representation using the labels in the source domain and then a separate encoding that maps the target data to the same space using an asymmetric mapping learned through a domain-adversarial loss.

MCD [5] models the domain adaptation problem [9] as a semi-supervised learning [10] problem by making multiple classifiers collaborate with each other and obtain more robust classification outputs by adopting some techniques commonly used in semi-supervised learning, such as ensuring the consistency of classifier outputs, thus greatly improving the performance of the model. SYM [11] designs symmetric object classifiers, which serve as domain discriminators as well. This idea, like MCD [5], comes from semi-supervised learning, and is still a technique that has not been utilized widely. CDAN [12] conditions the adversarial model on the discriminative information conveyed in the classifier predictions. This approach is equivalent to not constraining the features directly, but rather constraining the output of the classifier, giving a more powerful and effective constraint to the classifier and thus improving the accuracy of the model.

3. Method

3.1. Notations and Definitions

Suppose that we have a source domain dataset (handwritten Bai character dataset) denoted as $\mathcal{S} = \{(x^s, y) | x \in \mathcal{X}^s, y \in \mathcal{Y}\}$, where \mathcal{X}^s and \mathcal{Y} denote the set of samples and labels in the source domain dataset, respectively. Similarly, we have a target domain dataset (ancient books or offset printed Bai character dataset) denoted as $\mathcal{T} = \{x^t | x^t \in \mathcal{X}^t\}$, where \mathcal{X}^t denotes the set of samples in the target domain dataset. The main goal of this paper is to obtain a classifier $f : \mathcal{X}^t \rightarrow \mathcal{Y}$ that can recognize the target domain images by using the labeled source domain dataset \mathcal{S} and the unlabeled target domain dataset \mathcal{T} as training sets.

3.2. Overview

The framework of the proposed approach is shown in Figure 1, which is called information separation network (ISN). The framework contains five modules: a backbone (B), a separator (Se), a combiner (Co), a discriminator (D) and a classifier (C).

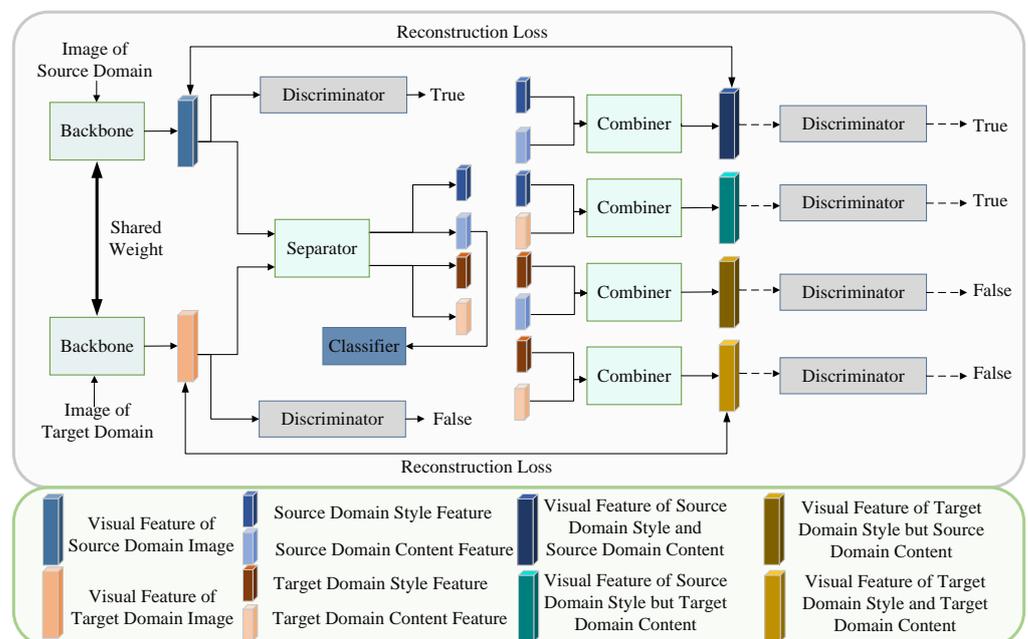


Figure 1. The overall framework of our proposed method.

The backbone (ResNet101) can be most of the convolutional neural networks, and its main role is to extract the image’s visual features, which usually contain both style and content information.

In view of this property of visual features, a novel separator for separating visual features into two features, style features and content features, is proposed. The sepa-

rated features are expected to contain only one of the style and content information in visual features.

The combiner can put the style and content features separated by the separator back together as a visual feature, and the visual feature is expected to contain both the style and the content of the feature, even if the style and content features are from different visual features.

The discriminator is mainly used to distinguish the visual features of the source domain and the target domain. Besides the only difference in style, the contents of the visual features of the source and target domains are the same. The key of the discriminator, therefore, is to distinguish which style a certain visual feature belongs to.

The classifier plays the part of classifying features that contain only content information, which can be used as a basis for classification. In a word, the classifier can ensure that the content features contain rich content information.

3.3. Pre-Training Stage

If the visual features extracted by the backbone do not contain enough information, the subsequent training separator and combiner would be meaningless. Therefore, in order to achieve a better visual feature extraction, according to the suggestions of ADDA [8], the backbone needs to be pre-trained with the source domain dataset first. The loss is as follows:

$$\mathcal{L}^{pre} = \mathbb{E}[L(C(c^s), y)], \quad (1)$$

where c^s is the content feature of the source domain visual features separated by the separator, and L is a classification loss, such as cross-entropy loss [13]. After pre-training, the backbone can ensure that the visual features extracted by itself can fully contain the information of the source domain data image, including both content and style information. The reason why we have to make sure that the visual features contain these information is that the content information is the key for the classifier to classify visual features into different categories. At the same time, if the visual features do not contain any style information of the image, there will be no sharp decline in the performance in cross domain testing. Therefore, visual features must also contain rich style information. The separation of content and style information is the key to realizing cross-domain adaptation, and also the main research focus of this paper.

3.4. Information Separation Stage

Similar to previous works [4,8], the adversarial learning [14] is applied in order to realize cross-domain adaptation. The difference lies in that we do not directly restrict the visual features to be independent of style, but by constraining the separator and combiner, the separator can separate the style and content information of visual features. To achieve this, a discriminator is first trained to distinguish visual features in the source and target domains with the following losses:

$$\begin{aligned} \mathcal{L}^D = & -\mathbb{E}[D(v^s)] + \mathbb{E}[D(v^t)] \\ & + \beta \mathbb{E}[(\|\nabla_{\hat{v}} D(\hat{v})\|_2 - 1)^2], \end{aligned} \quad (2)$$

where v^s is a visual feature of the source domain and v^t is a visual feature of the target domain. The last term is the Wasserstein loss [15] by enforcing the Lipschitz constraint [16], where $\hat{v} = \mu v^s + (1 - \mu)v^t$ with $\mu \sim U(0, 1)$. β is a hyperparameter, and as suggested in [15], $\beta = 10$ is fixed. Since the visual features contained in the source and target domains have the same content information but differ in their style information, so the discriminator distinguishes the visual features in the source and target domains mainly based on the style information contained in the visual features.

According to the properties of the discriminator, as long as the visual features combined by the combiner can be identified by the discriminator as containing only the style information of the style feature and not the style information of the content feature in-

volved in the combination, it is guaranteed that the style features separated by the separator contain style information. According to this idea, we have the following losses:

$$\begin{aligned} \mathcal{L}^{SC} = & -\mathbb{E}[D(\text{Co}(s^s, c^s))] - \mathbb{E}[D(\text{Co}(s^s, c^t))] \\ & + \mathbb{E}[D(\text{Co}(s^t, c^t))] + \mathbb{E}[D(\text{Co}(s^t, c^s))], \end{aligned} \quad (3)$$

where $s^s, c^s = \text{Se}(v^s)$ are the style features and content features separated from the visual features of the source domain, and $s^t, c^t = \text{Se}(v^t)$ are the style features and content features separated from the visual features of the target domain. The loss expects that the visual features combined with the source domain style features will be identified as source domain style visual features, regardless of whether the content features come from the source or target domain, and vice versa. In order to do this, the separator must ensure that the separated content features do not contain any style information of the visual features, which of course does not mean that the style features do not contain any content information. In order to make the content information be included in the content features as much as possible, we cannot ignore the classifier in the process of training the separator and the combiner, and the classifier [17] only participates in the classification with the help of the content features; therefore, it needs the content features to contain rich content information. That is, we only need to continue to involve the loss used in pre-training in the subsequent training as well, which is as follows:

$$\mathcal{L}^{BSC} = \mathbb{E}[L(C(c^s), y)]. \quad (4)$$

In addition, if we want to further improve the quality of the separated content and style features, and make sure they contain as much information as possible of the original visual features, the following reconstruction losses need to be introduced:

$$\begin{aligned} \mathcal{L}^{Re} = & \mathbb{E}[|\text{Co}(\text{Se}(v^s)) - v^s|] \\ & + \mathbb{E}[|\text{Co}(\text{Se}(v^t)) - v^t|]. \end{aligned} \quad (5)$$

To sum up, in the information separation stage, the overall optimization objective \mathcal{L} of the separator, combiner, classifier and backbone is:

$$\mathcal{L} = \mathcal{L}^{SC} + \lambda_1 \mathcal{L}^{BSC} + \lambda_2 \mathcal{L}^{Re}, \quad (6)$$

where λ_1 and λ_2 are hyperparameters. In all our experiments, the λ_1 is fixed to be 1 and λ_2 is 0.01. The training process of the proposed model is summarized in Algorithm 1.

Algorithm 1 Proposed approach.

Require: The source dataset \mathcal{S} and the target dataset \mathcal{T} .

Ensure: Random initialization of the parameters of $B, C, D, \text{Co}, \text{Se}$;

- 1: **while** B does not converge **do**
 - 2: **for** samples in \mathcal{S} **do**
 - 3: The samples are used to optimize B by Equation (1);
 - 4: **end for**
 - 5: **end while**
 - 6: **while** the model does not converge **do**
 - 7: **for** source and target samples in $\text{zip}\{\mathcal{S}, \mathcal{T}\}$ **do**
 - 8: The samples are used to optimize D by Equation (2);
 - 9: The samples are used to optimize B, C, Co and Se by Equation (6);
 - 10: **end for**
 - 11: **end while**
-

3.5. Testing Stage

The test is conducted in the target domain test set, and the only modules involved in the final test are the backbone, separator and classifier. The visual features are first extracted from the target domain images with the backbone, then the content features are separated with the separator, and finally, the content features are classified with the classifier.

4. Experimental Results

4.1. Experimental Setup

Datasets. The proposed method is evaluated on the following private dataset: Bai Character Cross-domain dataset, and five public ones: Digits [8], VisDA-2017 [18], Office-31 [19], Office-Home [20] and ImageCLEF-DA [21].

Specifically, the Bai Character Cross-domain (BCC) dataset consists of images from three different domains: Handwritten (H), Ancient books (A) and Offset (O), and each domain contains 400 categories. The main function of this dataset is to train the proposed model to perform cross-domain tasks from handwritten texts to antique (H→A) or offset (H→O) texts. Digits consists of three datasets with different domains: MNIST [22], USPS [23] and SVHN [24] digits datasets, and we take the adaptations in three directions into consideration: MNIST→USPS (M→U), USPS→MNIST (U→M), and SVHN→MNIST (S→M). The Visual Domain Adaptation Challenge 2017 (VisDA-2017) is oriented to the task of vision domain adaptation, which includes the tasks of target classification and target segmentation. This paper addresses the task of target classification, and the dataset has a total of 12 classes. Office-31 contains images of 31 categories drawn from three domains: Amazon (A), Webcam (W) and DSLR (D), and the proposed method was evaluated on the one-source to one-target domain adaptation. OfficeHome is a more challenging recent dataset that consists of images from 4 different domains: Art (Ar), Clip Art (Cl), Product (Pr) and Real-World (Rw) [25]. Each domain contains 65 object categories found typically in office and home environments. ImageCLEF-DA aims to provide an evaluation forum for the cross-language annotation and retrieval of images.

Implementation Details. The backbone changes as the training dataset changes, but usually it is the ResNet [2] pre-trained on ImageNet. The separator, discriminator and combiner are all three-layer multilayer perceptron (MLP) containing 2048-dimensional hidden layers, which are activated by ReLU. The output of the separator and the combiner is also connected to a ReLU layer, while the output of the discriminator is not connected to any activation function. The classifier is a fully connected network plus a softmax layer. The dimensionality of both content and style features was 512 dimensions. All modules were optimized with Adam [26], and the learning rate was 0.0001 from the beginning to end and with $\beta_1 = 0.9$ and $\beta_2 = 0.999$.

4.2. Comparison with SOTA Methods on BCC

Results on BCC are reported in Table 1. Eight latest domain adaptive methods are transplanted to the BCC dataset, but the best performing ETD only had an average accuracy of 72.3%. This is because none of these methods consider separating the content and style of visual features, so the features used in their classification are still impacted by the style information. In contrast, the proposed method outperforms all other methods on the BCC dataset due to the design and application of content and style separation, surpassing the second place by 1.5% on the H→A task, 3.2% on the H→A task and 2.9% on average, which demonstrates its leading position and fully illustrates its great effectiveness.

Table 1. Accuracy (%) on BCC (ResNet-101).

Method	H→A	H→O	Average
DAN [27]	55.6	59.2	57.4
DANN [4]	58.8	63.7	61.3
ADDN [8]	60.7	64.5	62.6
CoGAN [28]	64.1	67.9	66.0
CDAN [12]	66.3	70.8	68.6
CAT [29]	65.6	69.7	67.7
ETD [30]	70.1	74.5	72.3
LWC [31]	69.0	74.6	71.8
ISN (ours)	72.6	77.7	75.2

4.3. Comparison with SOTA Methods on Other Datasets

Results on Digits [8] are reported in Table 2. The proposed model achieves 96.5%, 96.7% on tasks of MNIST→USPS and USPS→MNIST, respectively, which outperforms the state-of-the-art (SOTA) methods. Although in the task of SVHN→MNIST, it is outperformed by the best method CAT [29] by 0.3%, the average performance of the method in all tasks still is the best among all methods. The reason why our method is unable to significantly outperform other methods is that the individual tasks on the Digits dataset are relatively simple, and even the most unsophisticated method can achieve high accuracy, while many methods, including ours, have already achieved accuracy of 95%+, which is thought to the problem making it difficult to improve performance on that dataset again.

Table 2. Accuracy (%) on Digits dataset.

Method	M→U	U→M	S→M	Average
DAN [27]	80.3	77.8	73.5	77.2
DRCN [32]	91.8	73.7	82.0	82.5
CoGAN [28]	91.2	89.1	-	-
ADDA [8]	89.4	90.1	76.0	85.2
CyCADA [33]	95.6	96.5	90.4	94.2
CDAN [12]	93.9	96.9	88.5	93.1
MCD [5]	94.2	94.1	96.2	94.8
CAT [29]	90.6	80.9	98.1	89.9
TPN [34]	92.1	94.1	93.0	93.1
LWC [31]	95.6	97.1	97.1	96.6
ETD [30]	96.4	96.3	97.9	96.9
ISN (ours)	96.5	96.7	97.8	97.0

Results on VisDA-2017 [18] are reported in Table 3. The proposed method did not achieve the highest performance in this dataset, but it was only 0.7% lower than the highest BSP [35]. However, it had more balanced performance than other methods and did not have very low precision in some categories. For example, in the category of trucks, our method achieved the highest precision of 44.5%, which was 6.1% higher than the second place BSP, while the precision of DANN [4] in this category was only 7.8%. This is a strong indication that the method is able to balance all categories better, which is the reason its overall performance is still comparable to that of BSP [35].

Table 3. Accuracy (%) on VisDA-2017.

Method	Plane	Bycycl	Bus	Car	Horse	Knife	Mcycl	Person	Plant	Sktbrd	Train	Truck	Mean
ResNet101 [2]	55.1	53.3	61.9	59.1	80.6	17.9	79.7	31.2	81.0	26.5	73.5	8.5	52.4
DAN [27]	87.1	63.0	76.5	42.0	90.3	42.9	85.9	53.1	49.7	36.3	85.8	20.7	61.1
DANN [4]	81.9	77.7	82.8	44.3	81.2	29.5	65.1	28.6	51.9	54.6	82.8	7.8	57.4
MCD [5]	87.0	60.9	83.7	64.0	88.9	79.6	84.7	76.9	88.6	40.3	83.0	25.8	71.9
BSP [35]	92.4	61.0	81.0	57.5	89.0	80.6	90.1	77.0	84.2	77.9	82.1	38.4	75.9
ISN (ours)	83.6	77.1	82.3	63.9	86.8	78.7	83.8	75.0	81.6	66.3	79.4	44.5	75.2

Results on Office31 [19] are reported in Table 4. The proposed method achieved the highest accuracy on three tasks of Webcam→D, Amazon→DSLR and DSLR→Amazon, respectively, and also achieved the second highest accuracy on the task DSLR→Webcam, just 0.4% lower compared to the first place. More importantly, its average performance exceeds all methods as the first method with an average accuracy over 89%, which fully illustrates the superiority of our method. Moreover, the same set of parameters were used for all tasks. As a matter of fact, if we do not use the same set of parameters, we can achieve an accuracy of 100.0% on the DSLR→Webcam task. While many methods adjust the hyperparameters according to different tasks, our method does have to, which further illustrates its stability [36].

Results on Office-Home [20] are reported in Table 5. The proposed method achieved the highest accuracy on four tasks of Ar→Pr, Ar→Rw, Pr→Ar and Rw→Pr, respectively, and the second highest accuracy on three tasks of Ar→Cl, Rw→Ar and Rw→Cl, respectively. Our method also achieved the second highest average performance on this dataset, just 1.0% lower than GVB [37] and on par with BNM [38], while greatly outperforming other methods. It is worth noting that, in so many tasks, the method also uses only the same set of hyperparameters, eliminating the complicated tuning process, which fully demonstrates its strong generalization ability.

Results on ImageCLEF-DA [21] are reported in Table 6. The proposed method achieved the highest accuracy on four tasks of I→P, P→I, C→P and P→C, respectively, and the second highest accuracy on two tasks of I→C and C→I, respectively. On this dataset, our method achieved the highest performance on average due to either the highest or second highest performance in all tasks, which is a good indication that it can be adapted to all tasks with good generalizability and excellent performance.

Table 4. Accuracy (%) on Office31.

Method	A→W	D→W	W→D	A→D	D→A	W→A	Average
ResNet50 [2]	68.4	96.7	99.3	68.9	62.5	60.7	76.1
DAN [27]	80.5	97.1	99.6	78.6	63.6	62.8	80.4
DANN [4]	82.6	96.9	99.3	81.5	68.4	67.5	82.7
ADDA [8]	86.2	96.2	98.4	77.8	69.5	68.9	82.9
CAT [29]	91.1	98.6	99.6	90.6	70.4	66.5	86.1
ETD [30]	92.1	100.0	100.0	88.0	71.0	67.8	86.2
DWL [7]	89.2	99.2	100.0	91.2	73.1	69.8	87.1
CDAN [12]	94.1	98.6	100.0	92.9	71.0	69.3	87.7
TAT [39]	92.5	99.3	100.0	93.2	73.1	72.1	88.4
TADA [40]	94.3	98.7	99.8	91.6	72.9	73.0	88.4
SYM [11]	90.8	98.8	100.0	93.9	74.6	74.6	88.4
BNM [38]	92.8	98.8	100.0	92.9	73.5	73.8	88.6
ALDA [41]	95.6	97.7	100.0	94.0	72.2	72.5	88.7
MDD [42]	94.5	98.4	100.0	93.5	74.6	72.2	88.9
ISN (ours)	92.7	99.6	100.0	94.1	74.6	73.5	89.1

Table 5. Accuracy (%) on Office-Home.

Method	Ar→Cl	Ar→Pr	Ar→Rw	Cl→Ar	Cl→Pr	Cl→Rw	Pr→Ar	Pr→Cl	Pr→Rw	Rw→Ar	Rw→Cl	Rw→Pr	Avg
ResNet50 [2]	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
MCD [5]	48.9	68.3	74.6	61.3	67.6	68.8	57.0	47.1	75.1	69.1	52.2	79.6	64.1
TAT [39]	51.6	69.5	75.4	59.4	69.5	68.6	59.5	50.5	76.8	70.9	56.6	81.6	65.8
ALDA [41]	53.7	70.1	76.4	60.2	72.6	71.5	56.8	51.9	77.1	70.2	56.3	82.1	66.6
SYM [11]	47.7	72.9	78.5	64.2	71.3	74.2	63.6	47.6	79.4	73.8	50.8	82.6	67.2
TADA [40]	53.1	72.3	77.2	59.1	71.2	72.1	59.7	53.1	78.4	72.4	60.0	82.9	67.6
MDD [42]	54.9	73.7	77.8	60.0	71.4	71.8	61.2	53.6	78.1	72.5	60.2	82.3	68.1
BNM [38]	56.2	73.7	79.0	63.1	73.6	74.0	62.4	54.8	80.7	72.4	58.9	83.5	69.4
DANN [4]	45.8	63.4	71.9	53.6	61.9	62.6	49.1	39.7	73.0	64.6	47.8	77.8	59.2
CDAN [12]	50.7	70.6	76.0	57.6	70.0	70.0	57.4	50.9	77.3	70.9	56.7	81.6	65.8
GVB [37]	57.0	74.7	79.8	64.6	74.1	74.6	65.2	55.1	81.0	74.6	59.7	84.3	70.4
ISN (ours)	56.4	74.8	80.0	60.1	73.5	73.6	66.0	50.4	80.2	72.9	60.1	85.3	69.4

Table 6. Accuracy (%) on ImageCLEF-DA.

Method	I→P	P→I	I→C	C→I	C→P	P→C	Average
ResNet50 [2]	74.8	83.9	91.5	78.0	65.5	91.2	80.7
DAN [27]	74.5	82.2	92.8	86.3	69.2	89.8	82.5
DANN [4]	75.0	86.0	96.2	87.0	74.3	91.5	85.0
JAN [43]	76.8	88.0	94.7	89.5	74.2	91.7	85.8
HAFN [44]	76.9	89.0	94.4	89.6	74.9	92.9	86.3
CAT [29]	76.7	89.0	94.5	89.8	74.0	93.7	86.3
ETD [30]	81.0	91.7	97.9	93.3	79.5	95.0	89.7
ISN (ours)	81.1	92.2	96.6	92.9	80.0	96.1	89.8

4.4. Visualization Experiments

Figure 2 depicts the t-SNE [45] visualizations of content features learned in the proposed method on the task of MNIST→USPS. With the increase of training epochs from (a) to (d), it can be observed that the aggregation of similar features in the source and target domains was getting better. At the 100th epoch, the content features generated by the target domain were almost completely wrapped in the content features generated by the source domain. This is because the content features in the visual features of the source and target domains should be exactly the same, except for the different style features. Therefore, when the content features are separated, it is only logical that they should be in the same distribution. The visualization of the content features also shows why the method is able to obtain such high accuracy on the task of MNIST→USPS.

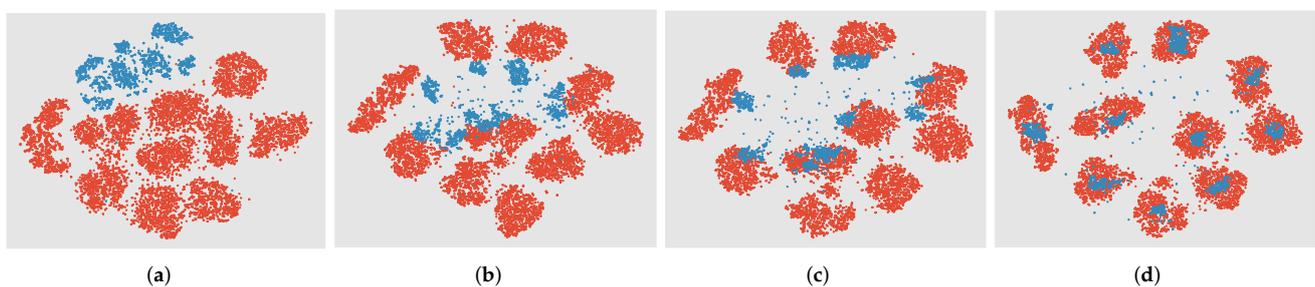


Figure 2. The t-SNE visualizations of content features generated by the proposed method with the increase of the epoch on MNIST→USPS. Red and blue points indicate the source and target samples, respectively. (a) Epoch:0, (b) Epoch:25, (c) Epoch:50, (d) Epoch:100.

4.5. Hyperparameter Analysis

The two hyperparameters for adjustment, λ_1 and λ_2 , are mainly discussed here.

Hyperparameter λ_1 . The effect of λ_1 is evaluated and shown in Figure 3. It can be observed that the best results were achieved on the BCC and Digits datasets when λ_1 was set to 1. This is because λ_1 controls the adversarial loss and determines how well the

content and style features are separated. When it is too small, the content features and style features cannot be completely separated, leading to a decrease in accuracy. When this hyperparameter is too large, the weight of classification loss is relatively small, and although content and style features can be well separated, the model cannot guarantee that the separated content information is useful for the classification process, which also leads to the decrease of model accuracy. As a result, it is reasonable to set λ_1 to 1.

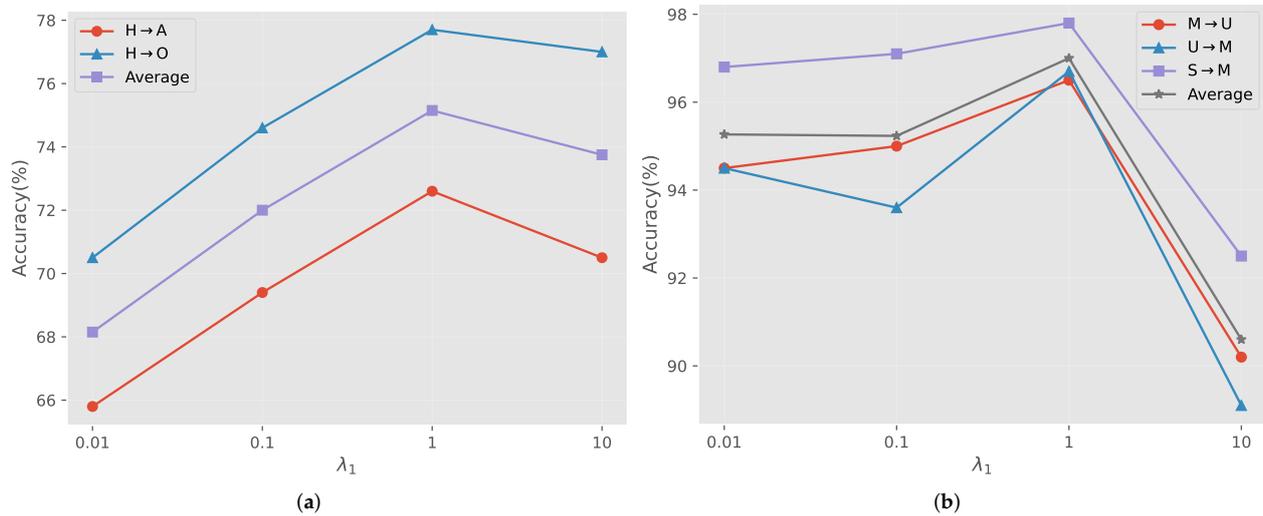


Figure 3. The effect of hyperparameter λ_1 on BCC and Digits datasets. (a) BCC, (b) Digits.

Hyperparameter λ_2 . The evaluation of the effect of λ_2 is shown in Figure 4. λ_2 controls the reconstruction loss, which can ensure that the separated content features and style features contain all the information of the original visual features as much as possible, greatly enhancing the quality of the content features. If it is too small, it is difficult to ensure that the content features can contain all the content information of the original features, which leads to the decrease of accuracy. Be that as it may, it is still a classification task in nature, and the reconstruction loss, as a regularization loss, should not be too large, otherwise it is bound to bring down the performance of the classification task. In summary, the hyperparameter λ_2 set to be 0.01 is most effective for all datasets.

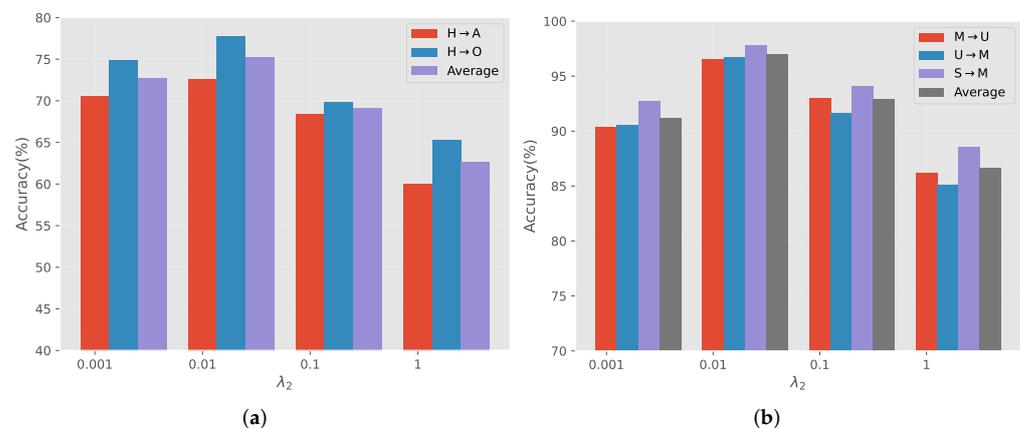


Figure 4. The effect of hyperparameter λ_2 on BCC and Digits datasets. (a) BCC, (b) Digits.

4.6. Ablation Experiments

The results of the ablation experiments are shown in Table 7. Comparing the first and second rows of Table 7, the addition of the adversarial loss gives the proposed model the ability to separate content information from style information in visual features, greatly

improving its ability to adapt across domains. With performance improvements of 33.4%, 29.6%, 29.7%, 9.2%, 19.5% and 6.2% in the six datasets, respectively, the information separation network's superior information separation capability is fully illustrated.

Table 7. The results of the ablation experiment using the average accuracy as a reference.

\mathcal{L}^{BSC}	\mathcal{L}^{Re}	BCC	Digits	VisDA-201	Office31	Office-Home	ImageCLEF-D
		40.5	65.5	53.6	77.3	48.5	82.6
✓		73.9	95.1	73.3	86.5	68.0	88.8
	✓	60.8	77.7	62.4	80.9	53.1	84.4
✓	✓	75.2	97.0	75.2	89.1	69.4	89.8

Comparing the second and fourth rows of Table 7, by adding the reconstruction loss as a regularization, greatly increased the quality of the content and style features, indirectly improving the performance of the model. Essentially speaking, this loss can be seen as a self-supervised loss [46–48], and this type of loss can help the model to automatically mine the potential knowledge in the data. After adding this loss, the accuracy performance of the model on six datasets improved by 1.3%, 1.9%, 1.9%, 2.6%, 1.4% and 1.0%, respectively.

4.7. Limitations Discussion

The backbone of our introduced method is usually pre-trained on top of ImageNet, even on the BCC dataset, with a huge handwritten Bai text dataset for pre-training. Therefore, if in some application scenarios it is not possible to collect a large dataset for pre-training, or if its classification target is far from ImageNet, then the performance of the network will be greatly affected. This is because the visual features drawn from the backbone contain little information at this point and the style and content features are more likely to be confused together, thus limiting our approach.

5. Conclusions and Future Works

A domain-adaptive Bai text dataset is constructed with the help of the existing handwritten Bai text dataset and the collected ancient (offset) Bai text dataset, and various domain adaptation methods are reported for this dataset. Meanwhile, an information separation network, which can effectively separate content and style information in visual features is designed, so that images with different domains are eventually sorted out with the same content features, eliminating the influence brought by different domains. What is more, the proposed cross-reconstruction method provides a strong guarantee for the success of information separation. Finally, experiments on multiple datasets and rich ablation, visualization experiments and hyperparameter analysis show that the proposed method can be equal or even superior to existing methods.

Currently, although our network aims to separate style information from content information in visual features, the accuracy improvement is limited on many datasets, so the style information in visual features is not completely stripped out, or in other words, the content features still contain some style information. In future work, we will take this as the goal and continue to investigate how to better separate style and content information in visual features.

Author Contributions: Conceptualization, Z.Z. and Z.G.; methodology, Z.Z.; software, Z.Z.; validation, Z.G., X.L. and W.L.; formal analysis, Z.Z.; investigation, Z.Z.; resources, Z.Z.; data curation, Z.Z.; writing—original draft preparation, Z.Z.; writing—review and editing, Z.Z.; visualization, Z.Z.; supervision, C.L.; project administration, Z.Z.; funding acquisition, Z.Z. All authors have read and agreed to the published version of the manuscript.

Funding: Project supported by the Science and Technology Planning Project of Yunnan Provincial Science and Technology Department (Grant Nos.2019J0313). <https://search.crossref.org/funding> (accessed on 26 March 2022). errors may affect your future funding.

Institutional Review Board Statement: This paper does not involve animal or human research.

Informed Consent Statement: This paper does not involve animal or human research.

Acknowledgments: Here I would like to thank my younger martial brother Gao Zuodong for his efforts and teacher Li Cuihua for his guidance.

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

References

1. Targ, S.; Almeida, D.; Lyman, K. Resnet in Resnet: Generalizing Residual Architectures. *arXiv* **2016**, arXiv:1603.08029.
2. 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.
3. Zhang, Z.; Lee, C.; Gao, Z.; Li, X. Basic research on ancient Bai characters recognition based on mobile APP. *Wirel. Commun. Mob. Comput.* **2021**, *2021*, 4059784. [[CrossRef](#)]
4. Ganin, Y.; Ustinova, E.; Ajakan, H.; Germain, P.; Larochelle, H.; Laviolette, F.; Marchand, M.; Lempitsky, V.S. Domain-Adversarial Training of Neural Networks. *J. Mach. Learn. Res.* **2016**, *17*, 59:1–59:35.
5. Saito, K.; Watanabe, K.; Ushiku, Y.; Harada, T. Maximum Classifier Discrepancy for Unsupervised Domain Adaptation. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Salt Lake City, UT, USA, 18–22 June 2018.
6. Yosinski, J.; Clune, J.; Bengio, Y.; Lipson, H. How transferable are features in deep neural networks? *Adv. Neural Inf. Process. Syst.* **2014**, *27*, 3320–3328.
7. Ni, X.; Lei, Z. Dynamic Weighted Learning for Unsupervised Domain Adaptation. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, Nashville, TN, USA, 20–25 June 2021; IEEE Computer Society: Washington, DC, USA, 2021. Available online: https://openaccess.thecvf.com/content/CVPR2021/html/Xiao_Dynamic_Weighted_Learning_for_Unsupervised_Domain_Adaptation_CVPR_2021_paper.html (accessed on 26 March 2022).
8. 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; IEEE Computer Society: Washington, DC, USA, 2017; pp. 2962–2971. Available online: https://openaccess.thecvf.com/content_cvpr_2017/html/Tzeng_Adversarial_Discriminative_Domain_CVPR_2017_paper.html (accessed on 26 March 2022).
9. Liang, Z.; Hongmei, C.; Yuan, H.; Keping, Y.; Shahid, M. A Collaborative V2X Data Correction Method for Road Safety. *IEEE Trans. Reliab.* **2022**, *4*, 1–12.
10. Blum, A.; Mitchell, T. *Combining Labeled and Unlabeled Data with Co-Training*; Morgan Kaufmann Publishers: San Mateo, CA, USA, 1998; pp. 92–100. Available online: <https://dl.acm.org/doi/pdf/10.1145/279943.279962> (accessed on 26 March 2022).
11. Zhang, Y.; Tang, H.; Jia, K.; Tan, M. Domain-Symmetric Networks for Adversarial Domain Adaptation. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, Long Beach, CA, USA, 16–20 June 2019.
12. Long, M.; Cao, Z.; Wang, J.; Jordan, M.I. Conditional Adversarial Domain Adaptation. In *Proceedings of the NeurIPS, Montreal, QC, Canada, 3–8 December 2018*; Bengio, S., Wallach, H.M., Larochelle, H., Grauman, K., Cesa-Bianchi, N., Garnett, R., Eds.; 2018; pp. 1647–1657. Available online: <https://proceedings.neurips.cc/paper/2018/hash/ab88b15733f543179858600245108dd8-Abstract.html> (accessed on 26 March 2022).
13. Martinez, M.; Stiefelwagen, R. Taming the Cross Entropy Loss. In *Proceedings of the GCPR, Stuttgart, Germany, 9–12 October 2018*; Brox, T., Bruhn, A., Fritz, M., Eds.; Lecture Notes in Computer Science; Springer: Berlin/Heidelberg, Germany, 2018; Volume 11269, pp. 628–637. Available online: https://link.springer.com/chapter/10.1007/978-3-030-12939-2_43 (accessed on 26 March 2022).
14. Goodfellow, I.J.; Pouget-Abadie, J.; Mirza, M.; Xu, B.; Warde-Farley, D.; Ozair, S.; Courville, A.C.; Bengio, Y. Generative Adversarial Nets. In *Proceedings of the NIPS, Montreal, QC, Canada, 8–13 December 2014*; Ghahramani, Z., Welling, M., Cortes, C., Lawrence, N.D., Weinberger, K.Q., Eds.; 2014; pp. 2672–2680. Available online: <https://proceedings.neurips.cc/paper/2014/hash/5ca3e9b122f61f8f06494c97b1afccf3-Abstract.html> (accessed on 26 March 2022).
15. Arjovsky, M.; Chintala, S.; Bottou, L. Wasserstein Generative Adversarial Networks. In *Proceedings of the ICML, Sydney, Australia, 6–11 August 2017*; Precup, D., Teh, Y.W., Eds.; Proceedings of Machine Learning Research; 2017; Volume 70, pp. 214–223. Available online: <https://proceedings.mlr.press/v70/arjovsky17a.html> (accessed on 26 March 2022).
16. Gulrajani, I.; Ahmed, F.; Arjovsky, M.; Dumoulin, V.; Courville, A. Improved Training of Wasserstein GANs. *arXiv* **2017**, arXiv:1704.00028v3.
17. Zhao, L.; Li, Z.; Al-Dubai, A.Y.; Min, G.; Li, J.; Hawbani, A.; Zomaya, A.Y. A Novel Prediction-Based Temporal Graph Routing Algorithm for Software-Defined Vehicular Networks. *IEEE Trans. Intell. Transp. Syst.* **2021**. [[CrossRef](#)]
18. Peng, X.; Usman, B.; Kaushik, N.; Hoffman, J.; Wang, D.; Saenko, K. VisDA: The Visual Domain Adaptation Challenge. *arXiv* **2017**, arXiv:1710.06924.
19. Saenko, K.; Kulis, B.; Fritz, M.; Darrell, T. Adapting Visual Category Models to New Domains. In *Proceedings of the ECCV (4), Crete, Greece, 5–11 September 2010*; Daniilidis, K., Maragos, P., Paragios, N., Eds.; Lecture Notes in Computer Science; Springer: Berlin/Heidelberg, Germany, 2010; Volume 6314, pp. 213–226.

20. 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; IEEE Computer Society: Washington, DC, USA, 2017; pp. 5385–5394. Available online: https://openaccess.thecvf.com/content_cvpr_2017/html/Venkateswara_Deep_Hashing_Network_CVPR_2017_paper.html (accessed on 26 March 2022).
21. Müller, H.; Clough, P.; Deselaers, T.; Caputo, B. (Eds.) *ImageCLEF: Experimental Evaluation in Visual Information Retrieval*; The Information Retrieval Series; Springer: Berlin, Germany, 2010; Volume 32. [CrossRef]
22. LeCun, Y.; Bottou, L.; Bengio, Y.; Haffner, P. Gradient-based learning applied to document recognition. *Proc. IEEE* **1998**, *86*, 2278–2324. [CrossRef]
23. Parkins, A.D.; Nandi, A.K. Genetic programming techniques for hand written digit recognition. *Signal Process.* **2004**, *84*, 2345–2365. [CrossRef]
24. Netzer, Y.; Wang, T.; Coates, A.; Bissacco, A.; Wu, B.; Ng, A.Y. Reading Digits in Natural Images with Unsupervised Feature Learning. 2011. Available online: <https://research.google/pubs/pub37648/> (accessed on 26 March 2022).
25. Zhao, L.; Zheng, T.; Lin, M.; Hawbani, A.; Shang, J.; Fan, C. SPIDER: A Social Computing Inspired Predictive Routing Scheme for Software Document Vehicular Networks. *IEEE Trans. Intell. Transp. Syst.* **2021**. Available online: <https://ieeexplore.ieee.org/abstract/document/9594721> (accessed on 26 March 2022).
26. Kingma, D.P.; Ba, J. Adam: A Method for Stochastic Optimization. *arXiv* **2014**, arXiv:1412.6980
27. Long, M.; Cao, Y.; Wang, J.; Jordan, M.I. Learning Transferable Features with Deep Adaptation Networks. In *Proceedings of the ICML, Lille, France, 6–11 July 2015*; Bach, F.R., Blei, D.M., Eds.; JMLR Workshop and Conference Proceedings; 2015; Volume 37, pp. 97–105. Available online: <http://proceedings.mlr.press/v37/long15> (accessed on 26 March 2022).
28. Liu, M.Y.; Tuzel, O. Coupled Generative Adversarial Networks. In *Proceedings of the NIPS, Barcelona, Spain, 5–10 December 2016*; Lee, D.D., Sugiyama, M., von Luxburg, U., Guyon, I., Garnett, R., Eds.; 2016; pp. 469–477. Available online: <https://proceedings.neurips.cc/paper/2016/hash/502e4a16930e414107ee22b6198c578f-Abstract.html> (accessed on 26 March 2022).
29. Deng, Z.; Luo, Y.; Zhu, J. Cluster Alignment with a Teacher for Unsupervised Domain Adaptation. In Proceedings of the ICCV, Seoul, Korea, 27 October–2 November 2019; pp. 9943–9952.
30. Li, M.; Zhai, Y.; Luo, Y.W.; Ge, P.; Ren, C.X. Enhanced Transport Distance for Unsupervised Domain Adaptation. In Proceedings of the CVPR, New Orleans, LA, USA, 19–24 June 2020; pp. 13933–13941.
31. Ye, S.; Wu, K.; Zhou, M.; Yang, Y.; Tan, S.H.; Xu, K.; Song, J.; Bao, C.; Ma, K. Light-weight Calibrator: A Separable Component for Unsupervised Domain Adaptation. In Proceedings of the CVPR, New Orleans, LA, USA, 19–24 June 2020; pp. 13733–13742.
32. Ghifary, M.; Kleijn, W.B.; Zhang, M.; Balduzzi, D.; Li, W. Deep Reconstruction-Classification Networks for Unsupervised Domain Adaptation. In *European Conference on Computer Vision*; Springer: Cham, Switzerland 2016.
33. Hoffman, J.; Tzeng, E.; Park, T.; Zhu, J.Y.; Isola, P.; Saenko, K.; Efros, A.A.; Darrell, T. CyCADA: Cycle-Consistent Adversarial Domain Adaptation. In Proceedings of the International Conference on Machine Learning, Sydney, Australia, 6–11 August 2017.
34. Pan, Y.; Yao, T.; Li, Y.; Wang, Y.; Ngo, C.W.; Mei, T. Transferrable Prototypical Networks for Unsupervised Domain Adaptation. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, Long Beach, CA, USA, 16–20 June 2019.
35. Chen, X.; Wang, S.; Long, M.; Wang, J. Transferability vs. Discriminability: Batch Spectral Penalization for Adversarial Domain Adaptation. In *Proceedings of the ICML, Long Beach, CA, USA, 9–15 June 2019*; Chaudhuri, K., Salakhutdinov, R., Eds.; Proceedings of Machine Learning Research; 2019; Volume 97, pp. 1081–1090. Available online: <http://proceedings.mlr.press/v97/chen19i.html?ref=https://codemonkey.link> (accessed on 26 March 2022).
36. Zhao, L.; Li, J.; Al-Dubai, A.; Zomaya, A.Y.; Min, G.; Hawbani, A. Routing schemes in software-defined vehicular networks: Design, open issues and challenges. *IEEE Intell. Transp. Syst. Mag.* **2020**, *13*, 217–226. [CrossRef]
37. Cui, S.; Wang, S.; Zhuo, J.; Su, C.; Huang, Q.; Tian, Q. Gradually Vanishing Bridge for Adversarial Domain Adaptation. In Proceedings of the CVPR, New Orleans, LA, USA, 19–24 June 2020; pp. 12452–12461.
38. Cui, S.; Wang, S.; Zhuo, J.; Li, L.; Huang, Q.; Tian, Q. Towards Discriminability and Diversity: Batch Nuclear-Norm Maximization Under Label Insufficient Situations. In Proceedings of the CVPR, New Orleans, LA, USA, 19–24 June 2020; pp. 3940–3949.
39. Liu, H.; Long, M.; Wang, J.; Jordan, M.I. Transferable Adversarial Training: A General Approach to Adapting Deep Classifiers. In *Proceedings of the ICML, Long Beach, CA, USA, 9–15 June 2019*; Chaudhuri, K., Salakhutdinov, R., Eds.; Proceedings of Machine Learning Research; 2019; Volume 97, pp. 4013–4022. Available online: <https://proceedings.mlr.press/v97/liu19b.html> (accessed on 26 March 2022).
40. Wang, X.; Li, L.; Ye, W.; Long, M.; Wang, J. Transferable Attention for Domain Adaptation. In Proceedings of the AAAI, Atlanta, GA, USA, 8–12 October 2019; AAAI Press: Menlo Park, CA, USA, 2019; pp. 5345–5352. Available online: <https://ojs.aaai.org/index.php/AAAI/article/view/4472> (accessed on 26 March 2022).
41. Chen, M.; Zhao, S.; Liu, H.; Cai, D. Adversarial-Learned Loss for Domain Adaptation. In Proceedings of the AAAI Conference on Artificial Intelligence, New York, NY, USA, 7–12 February 2020.
42. Zhang, Y.; Liu, T.; Long, M.; Jordan, M.I. Bridging Theory and Algorithm for Domain Adaptation. In *Proceedings of the ICML, Long Beach, CA, USA, 9–15 June 2019*; Chaudhuri, K., Salakhutdinov, R., Eds.; Proceedings of Machine Learning Research; 2019; Volume 97, pp. 7404–7413. Available online: <http://proceedings.mlr.press/v97/zhang19i.html?ref=https://codemonkey.link> (accessed on 26 March 2022).

43. Long, M.; Zhu, H.; Wang, J.; Jordan, M.I. Deep Transfer Learning with Joint Adaptation Networks. In *Proceedings of the ICML, Sydney, Australia, 6–11 August 2017*; Precup, D., Teh, Y.W., Eds.; Proceedings of Machine Learning Research; 2017; Volume 70, pp. 2208–2217. Available online: <http://proceedings.mlr.press/v70/long17a.html> (accessed on 26 March 2022).
44. Xu, R.; Li, G.; Yang, J.; Lin, L. Larger Norm More Transferable: An Adaptive Feature Norm Approach for Unsupervised Domain Adaptation. In *Proceedings of the ICCV, Seoul, Korea, 27 October–2 November 2019*; pp. 1426–1435.
45. Van der Maaten, L.; Hinton, G. Visualizing data using t-SNE. *J. Mach. Learn. Res.* **2008**, *9*, 2579–2605.
46. Liu, X.; Zhang F.; Hou, Z.; Mian, L.; Wang, Z.; Zhang, J.; Tang, J.; Self-supervised Learning: Generative or Contrastive. *arXiv* **2020**. arXiv:2006.08218.
47. Hendrycks, D.; Mazeika, M.; Kadavath, S.; Song, D. Using Self-Supervised Learning Can Improve Model Robustness and Uncertainty. In *Proceedings of the Advances in Neural Information Processing Systems, Vancouver, BC, Canada, 8–14 December 2019*; pp. 15637–15648.
48. Welling, M.; Zemel, R.S.; Hinton, G.E. Self Supervised Boosting. In *Proceedings of the NIPS, Vancouver, BC, Canada, 9–14 December 2002*; Becker, S., Thrun, S., Obermayer, K., Eds.; MIT Press: Cambridge, MA, USA, 2002; pp. 665–672. Available online: <https://proceedings.neurips.cc/paper/2002/hash/cd0cbcc668fe4bc58e0af3cc7e0a653d-Abstract.html> (accessed on 26 March 2022).