Cross-domain Cross-task Knowledge Distillation Network for Unsupervised Domain Adaptation*

Tai Ma
Modern Intelligent Manufacturing Industry College West Yunnan University of Applied Sciences

Chongwen Wang
School of Surveying and Information Engineering West Yunnan University of Applied Sciences

Hai Zi
Modern Intelligent Manufacturing Industry College West Yunnan University of Applied Sciences

Zeqing Zhang (✉ 3464723072@qq.com)
School of Surveying and Information Engineering West Yunnan University of Applied Sciences

Article

Keywords: Distillation Network, Unsupervised Domain Adaptation, Cross-domain Cross-task

Posted Date: October 14th, 2023

DOI: https://doi.org/10.21203/rs.3.rs-3365367/v1

Additional Declarations: No competing interests reported.

Cross-domain Cross-task Knowledge Distillation Network for Unsupervised Domain Adaptation*

Tai Ma[1], Chongwen Wang[2], Hai Zi[3], Zeqing Zhang[4]*

1 Modern Intelligent Manufacturing Industry College, West Yunnan University of Applied Sciences, Dali 671000, Yunnan, P. R. China 735172265@qq.com

2 School of Surveying and Information Engineering, West Yunnan University of Applied Sciences, Dali 671000, Yunnan, P. R. China wchongwen@gmail.com

3 Modern Intelligent Manufacturing Industry College, West Yunnan University of Applied Sciences, Dali 671000, Yunnan, P. R. China 1538859874@qq.com

4 School of Surveying and Information Engineering, West Yunnan University of Applied Sciences, Dali 671000, Yunnan, P. R. China 3464723072@qq.com

Abstract

Unsupervised domain adaptation has attracted extensive attention in recent years because it can naturally solve the problem of new domains encountered in depth model testing. Generally, unsupervised domain adaptive methods combine classification loss and adversarial loss, so that the features extracted from the model can be classified and are domain independent, thus alleviating the problems caused by cross domain models. However, it is impossible for the features extracted from the model to not contain any domain information. Therefore, we propose a new solution to make the source domain features extracted from the model fully include the style of the target domain images. Specifically, a compression network is trained by the target domain images, and the source domain features extracted by the classification network are distilled with the target domain image features extracted by the compression network to make it have the style of the target domain images. Finally, our method achieves leading or comparable results on five public data sets.

Keywords: Distillation Network, Unsupervised Domain Adaptation, Cross-domain Cross-task

*Project supported by the Science and Technology Planning Project of Yunnan Provincial Science and Technology Department (Grant Nos. 2018FD058).

1. Introduction

A large number of computer vision tasks have made great breakthroughs with the cutting-edge technologies such as deep convolutional neural networks (CNN) [32][5]. Despite of success already achieved, even a subtle departure from training data may still cause existing shallow or deep models to make spurious predictions. Directly applying a deep CNN well trained on millions of images to a new domain encounters performance degradation, while collecting labeled samples for the new domain is expensive and time-consuming.

Unsupervised domain adaptation has the natural ability to solve the problem of model spanning a new domain, and has received extensive attention in recent years. Unsupervised domain adaptation takes labeled source domain data and unlabeled target domain data as training sets, which can train a depth model that can be tested in the target domain data set.

At present, the mainstream methods [33][20][9] to solve the unsupervised domain adaptation problem is to expect the feature extraction module in the depth model to extract the domain independent information in the picture, so as to mitigate the impact of cross domain. However, the domain information is usually difficult to completely erase, so the performance of these methods is still not ideal.

In order to solve this problem, we do not need to let the feature extraction module in the depth model extract domain independent information, but let the feature extraction module endow the extracted features of the source domain with the style of the images in the target domain, so that the final classifier can adapt to the images in the target domain in advance, and finally alleviate the cross domain problem caused by crossing from the source domain to the target domain.

According to this idea, we propose a Cross-domain Cross-task Knowledge Distillation Network (CCKDN), which uses cross-domain and cross-task knowledge distillation to make the source domain features extracted by the feature extraction module have the style of the target domain. First, a compression network and a reconstruction network are trained with the images of the target domain. In this way, the compression network learns how to compress a image of the target domain into features of different dimensions. Then, these features are used to distill the source domain features extracted by the feature extraction module, so that the source domain features have the style of the target domain images. Because this knowledge distillation is from compression network to classification network, and from target domain to source domain, we call this distillation cross-domain cross-task knowledge distillation.

In addition, in order to make the features of the source domain include the style of the image in the target domain, we also talk about the features of the source domain to reconstruct the image through the reconstruction network, and use a discriminator to constrain it to make it as close as possible to the image reconstructed from the target domain. In this way, with the training, the features of the source domain image and the target domain image can be reconstructed into the image with the style of the target domain image, which fully ensures that the features of the source domain image include the style of the target domain image. Finally, because the classifier has learned the source domain features with labels and the image style of the target domain, it has a strong adaptability to the image of the target domain, greatly alleviated the problems caused by cross domain, and achieved leading performance.

The work and contribution in this paper are twofold:

· We propose a Cross-domain Cross-task Knowledge Distillation Network (CCKDN), which can make the features of the source domain image have the style of the target domain image through distillation loss, so that the classifier can adapt to the target domain image in advance.

·Our method has achieved leading or comparable results on five public data sets.

2. Related Work

Unsupervised domain adaptation, first proposed by [30]. This year, with the rapid development of deep learning technology, it has attracted more and more attention [20][33][8][21][31][18].

DANN [9] focuses on combining domain adaptation and deep feature learning in one training process. Its goal is to embed the domain adaptively into the process of learning representation, so that the final classification decision is based on the characteristics that are both discriminative and invariant to domain changes, that is, the same or very similar distribution in the source and target domains. In this way, the obtained feedforward network can be applied to the target domain without being affected by the displacement between the two domains. ADDA [33] puts forward a new and unified framework for adaptation of confrontation domain, which enables us to effectively check the differences of different factors among existing methods and clearly check their shared similarities. Our framework unifies design choices, such as weight sharing, basic model and adversarial loss, and includes previous work. At the same time, it also promotes the instantiation of existing instances of novel design improvements.

Generally, CDAN [21] is a principled framework, which restricts the adaptive model of adversary based on the identification information transmitted in classifier prediction. Symnets [39] designed the source and target task classifiers symmetrically, in which both domain classifiers produced classification output. SGF [25] and GFK [11][10] build a bridge between domains on the Glassman manifold, but they cannot be easily applied to deep networks. In the deep method, DSN [1], DISE [2] and DLOW [12] build a bridge between the source and the target by reconstructing the input image at the pixel level.

3. Method

3.1. Notations and Definitions

Unsupervised domain adaptation task requires the use of a labeled source domain dataset $S = \{(x^S, y) | x \in X^S, y \in Y\}$, where X^S and Y denote the set of images and labels in the source domain dataset, respectively, and an unlabeled target domain dataset $T = \{x^t | x^t \in X^t\}$, where X^t denotes the set of images in the target domain dataset. Although the images in the target domain dataset are not labeled, they are usually from the same classes as the images in the source domain. The essence of this task is to train a classifier $f : X^t \to Y$ with these two datasets that is capable of classifying images in the target domain.

3.2. Overview

The framework of our approach is shown in Figure 1, which is called Cross-domain Cross-task Knowledge Distillation Network (CCMDN). Our method consists of a Compression Network (*Co*), a Classification Network (*Cl*), three Feature Transform Modules (*FTM*), a Reconstruction Network (*Re*) and a Discriminator (*D*).

The Compression Network is composed of several Convolution Blocks (*CB*) and Down-sampling Layers (*DL*), whose main role is to continuously compress a source domain image to obtain its low, medium and high dimensional features. These features contain not only the low, medium and high frequency information, but also rich style and content information of the source domain image.

The Classification Network is composed of a Feature Extraction Module (*FEM*), a Global Pooling Layer (*GPL*) and a Classifier (*C*). The Feature Extraction Module has multiple Residual Blocks (*RB*) and Down-sampling Layers (*DL*), and the Classifier is a multi-layer perception. The main function of Classification Network is to classify a image.

The Feature Transform Module can transform the source domain image features extracted by the Classification Net work into the space where the target domain image features extracted by the Compression Network are located, and use distillation loss to make the source domain image features as similar as the target domain image features as possible. In this way, the Classification Network can have a strong adaptability to the images in the target domain.

The Reconstruction Network consists of multiple Convolution Blocks (*CB*) and Up-sampling Layers (*UL*). It can restore the image features of the target domain compressed by the Compression Network, and restore the image features of the source domain extracted by the Classification Network to the image style of the target domain as much as possible.

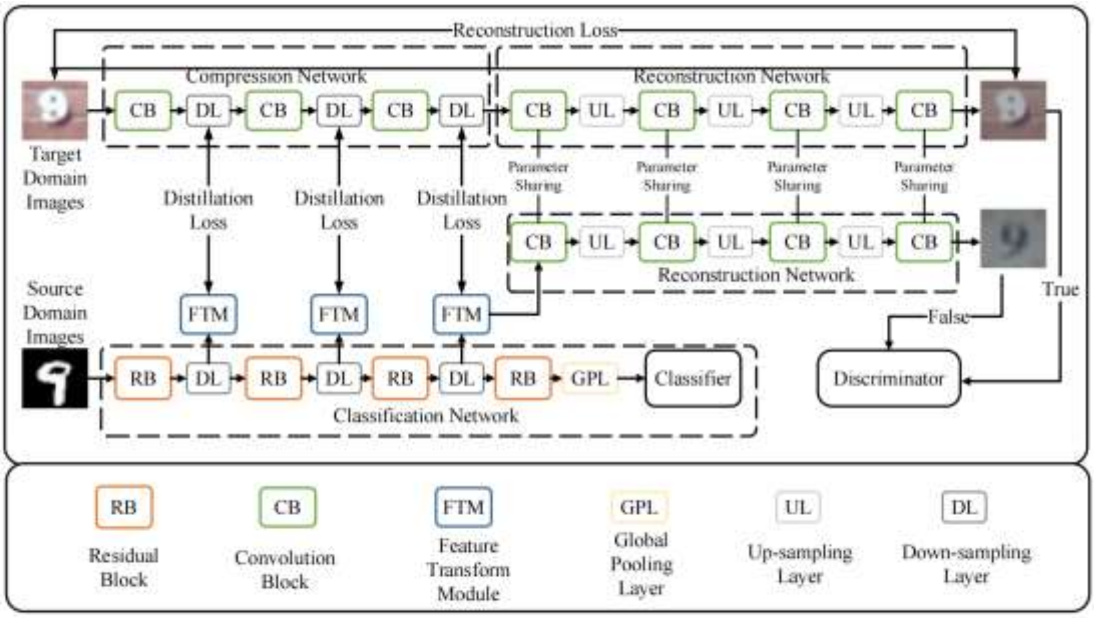


Figure 1. The overall framework of our proposed Cross-domain Cross-task Knowledge Distillation Network (CCMDN).

With this constraint, the source domain image features extracted by the Classification Network can be fully close to the target domain image features. The main function of the Discriminator (*D*) is to cooperate with the Reconstruction Network to pull in the image reconstructed from the features of the target domain and the source domain.

3.3. Cross-domain Cross-task Knowledge Distillation

The Classification Network only uses labeled source domain data for training, so when the target domain image is input, the performance will inevitably decline due to cross domain reasons. Our solution is to distill the knowledge of the target domain extracted from the Compression Network to the Feature Extraction Module of the Classification Network through knowledge distillation, so that the source domain image features extracted by the Feature Extraction Module have been mixed with the knowledge of the target domain image, so the classifier in the Classification Network is equivalent to learning the labeled target domain source domain mixed features. In this way, even if the input image is in the target domain, the classifier is not completely unfamiliar with it, and finally alleviates the cross domain problem.

The knowledge of a image must be rich and diverse. These knowledge exist in low-frequency, medium frequency and high-frequency information, corresponding to the Compression Network, that is, in the low-dimensional, medium and high-dimensional features extracted by the Compression Network. Therefore, in order for the Feature Extraction Module to fully learn the knowledge of the target domain images obtained in the Compression Network, it is necessary to distill the features of different dimensions at the same time.

Knowledge Distillation of Low-dimensional Features.

The low-dimensional features of the target domain image extracted by the Compression Network basically maintain most of the details of the target domain image. Although these details are very important for reconstructing a image, only part of them are meaningful for classifying the image. The low-dimensional features of source domain images extracted by the feature extraction module need to learn the part that is different from itself but can assist in classification. Of course, this is difficult to do, because the model cannot distinguish which information can be used to classify which can not. Therefore, the common choice is to let the low dimensional features of the source domain images extracted by the Feature Extraction Module learn the low-dimensional features of the target domain images extracted by the Compression Network as much as possible. The only useful part is the information that can help the classifier to classify animage.

However, it is obvious that the size and dimension of the features extracted by the Compression Network and the features extracted by the Classification Network may be different. Therefore, the features extracted by the Classification Network need to be transformed into the space where the features extracted by the Compression Network are located through a Feature Transform Module before distillation learning.

Assuming that the low-dimensional target domain image feature extracted by the Compression Network is f_l^{t} and the low-dimensional source domain image feature extracted by the Classification Network is f_l^{s} , the distillation loss is as follows:

$$L^{DL\_low} = \left\| FTM_{low}(f_l^s) - f_l^t \right\|_2^2, \qquad (1)$$

where FTM_{low} is a Feature Transform Module used to transform low-dimensional features.

Knowledge Distillation of Medium-dimensional Features.

The medium-dimensional feature extracted from the Compression Network has removed a lot of detail information. At this time, the medium-dimensional feature contains more information such as the structure of the object in the image, which is very helpful to distinguish the class of the image. Therefore, the medium-dimensional features of the source domain extracted by the Classification Network can learn the medium-dimensional features of the target domain as much as possible, so that the source domain features can learn the structure and other information that will appear in the target

domain, so that the classifier can finally adapt to the images of the target domain.

Assuming that the medium-dimensional target domain image feature extracted by the Compression Network is $f_{\mathrm{m}}^{\mathrm{t}}$ and the medium-dimensional source domain image feature extracted by the Classification Network is $f_{\mathrm{m}}^{\mathrm{s}}$, the distillation loss is as follows:

$$L^{DL\_med} = \left\| FTM_{med}(f_m^s) - f_m^t \right\|_2^2, \qquad (2)$$

where FTM_{med} is a Feature Transform Module used to transform medium-dimensional features.

Knowledge Distillation of Medium-dimensional Features.

The high-dimensional features of the target domain extracted by the Compression Network have been very abstract, and there is basically no detailed information. Some of the structural information has also disappeared in the continuous down-sampling. Now most of the retained information is high-level semantic information, which is very important for classification. Therefore, the high-dimensional source do main features extracted by the Classification Network must fully learn these features, so that the classifier can adapt to the high-level semantic information of the target domain in advance, and finally improve its classification performance. Moreover, some of the high-level semantic information of the target domain and the source domain overlap, and the non overlapping part is the final reason for the decline of cross domain performance. Therefore, it is necessary to distill this part of knowledge.

Assuming that the high-dimensional target domain image feature extracted by the Compression Network is $f_{\mathrm{h}}^{\mathrm{t}}$ and the high-dimensional source domain image feature extracted by the Classification Network is $f_{\mathrm{h}}^{\mathrm{s}}$, the distillation loss is as follows:

$$L^{DL\_hig} = \left\| FTM_{hig}(f_h^s) - f_h^t \right\|_2^2, \qquad (3)$$

where FTM_{hig} is a Feature Transform Module used to transform high-dimensional features.

Total Knowledge Distillation Loss.

To sum up, the total distillation loss is as follows:

$$L^{DL} = L^{DL\_low} + L^{DL\_med} + L^{DL\_hig}, \qquad (4)$$

Since our distillation loss is from the target domain data set to the source domain data set, it is cross-domain. In addition, our distillation loss is from the Compression Network to the Classification Network, so it is cross-task. Therefore, we call our distillation loss as cross-domain cross-task distillation loss, which is the first one proposed by us.

3.4. Other Losses

Reconstruction Loss.

In order to ensure that the Compression Network can well compress the images of the target domain, it needs to conduct joint training with the Reconstruction Network. Its purpose is to learn how to compress and reconstruct a image.

The loss is as follows:

$$L^{\mathrm{Re}} = \left\| x^{t} - \mathrm{Re}(Co(x^{t})) \right\|_{2}^{2}, \tag{5}$$

where *Re* and *Co* represent Reconstruction Network and Compression Network, respectively.

Classification Loss.

The Classification Network also needs to learn the source domain images with labels. Of course, these features extracted from the source domain images have fused a large amount of information of the target domain images. Therefore, the classification loss can make the classifier well adapt to the target domain images. In the test environment, it will not cause a sudden drop in performance due to cross domain. The loss is as follows:

$$L^{Cl} = Cross\_Entropy(Cl(x^{s}), y), \tag{6}$$

where $Cross\_Entropy$ and Cl represent cross entropy loss function and Classification Network, respectively.

Adversarial Loss.

Table 1. Accuracy (%) on Digts.

Method	M→U	U→M	S→M	Average
DAN [20]	80.3	77.8	73.5	77.2
DRCN [10]	91.8	73.7	82.0	82.5
CoGAN [19]	91.2	89.1	-	-
ADDA [33]	89.4	90.1	76.0	85.2
CyCADA [14]	95.6	96.5	90.4	94.2
CDAN [21]	93.9	96.9	88.5	93.1
MCD [31]	94.2	94.1	96.2	94.8
CAT [8]	90.6	80.9	**98.1**	89.9
TPN [27]	92.1	94.1	93.0	93.1
LWC [37]	95.6	**97.1**	97.1	96.6
ETD [17]	96.4	96.3	97.9	96.9
CCMDN(ours)	**96.7**	96.9	**98.1**	**97.2**

Table 2. Accuracy (%) on VisDA-2017.

Method	plane	bcycl	bus	car	horse	knife	mcycl	person	plant	sktbrd	train	truck	mean
ResNet101 [13]	55.1	53.3	61.9	59.1	80.6	17.9	79.7	31.2	81	26.5	73.5	8.5	52.4
DAN [20]	87.1	63	76.5	42	90.3	42.9	85.9	53.1	49.7	36.3	85.8	20.7	61.1
DANN [9]	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 [31]	87	60.9	83.7	64	88.9	79.6	84.7	76.9	88.6	40.3	83	25.8	71.9
BSP [4]	92.4	61	81.0	57.5	89.0	80.6	90.1	77.0	84.2	77.9	82.1	38.4	75.9
CCMDN(ours)	89.7	77.5	83.6	61	90.2	80.4	91.0	77.1	83.8	76.4	86	40.1	**78.1**

In order to make the source domain image features extracted by the Classification Network fully contain the in formation of the target domain image features, we also add a Adversarial loss. This loss hopes that the source domain image features extracted by the Classification Network and the target domain image features extracted by the Compression Network can be reconstructed into the target domain style images by the Reconstruction Network. The loss is as follows:

$$L^{\mathrm{Ad}} = -D(\mathrm{Re}(FEM(x^{s}))) + D(\mathrm{Re}(Co(x^{t}))), \qquad (7)$$

where *D* and *FEM* represent the Discriminator and the Feature Extraction Module of the Classification Network, respectively.

3.5. Overall Optimization Objective

To sum up, in the Cross-domain Cross-task Knowledge Distillation Network, the overall optimization objective L is:

$$L = \min_{Cl,Co,\mathrm{Re}} \max_{D} L^{DL} + \lambda_1 L^{\mathrm{Re}} + \lambda_1 L^{Ad} + \lambda_1 L^{Cl}, \qquad (8)$$

where λ_1, λ_2 and λ_3 are hyperparameters.

4. Experimental Results

4.1. Experimental Setup

Datasets. The proposed method is evaluated on the following private dataset five public dataset: Digts [33], VisDA-2017 [29], Office-31 [30], Office-Home [34] and ImageCLEF-DA [23].

Specifically, Digts consists of three datasets with different domains: MNIST [16] (http://yann.lecun.com/exdb/mnist/), USPS [28] (https://www.csie.ntu.edu.tw/~cjlin/libsvmtools/datasets/multiclass.html#usps) and SVHN [24] (http://ufldl.stanford.edu/housenumbers/) 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 (http://data-bdd.berkeley.edu) 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. Office Home (https://hemanthdv.github.io/officehome-dataset/) 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). Each domain contains 65 object categories found typically in office and home environments. ImageCLEF-DA (https://paperswithcode.com/dataset/imageclef-da) aims to provide an evaluation forumfor the cross–language annotation and retrieval of images.

Table 3. Accuracy (%) on Office-31.

Method	A→W	D→W	W→D	A→D	D→A	W→A	Average
ResNet50 [13]	68.4	96.7	99.3	68.9	62.5	60.7	76.1
DAN [20]	80.5	97.1	99.6	78.6	63.6	62.8	80.4
DANN [9]	82.6	96.9	99.3	81.5	68.4	67.5	82.7
ADDA [33]	86.2	96.2	98.4	77.8	69.5	68.9	82.9
CAT [8]	91.1	98.6	99.6	90.6	70.4	66.5	86.1
ETD [17]	92.1	**100**	**100**	88	71	67.8	86.2
DWL [26]	89.2	99.2	**100**	91.2	73.1	69.8	87.1
CDAN [21]	94.1	98.6	**100**	92.9	71	69.3	87.7
TAT [18]	92.5	99.3	**100**	93.2	73.1	72.1	88.4
TADA [35]	94.3	98.7	99.8	91.6	72.9	73	88.4
SYM [39]	90.8	98.8	**100**	93.9	74.6	**74.6**	88.4
BNM [6]	92.8	98.8	**100**	92.9	73.5	73.8	88.6
ALDA [3]	**95.6**	97.7	**100**	**94**	72.2	72.5	88.7
MDD [38]	94.5	98.4	**100**	93.5	74.6	72.2	88.9
CCMDN(ours)	94.2	99.2	**100**	93.9	**74.7**	74	**89.3**

Table 4. 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 [13]	34.9	50	58	37.4	41.9	46.2	38.5	31.2	60.4	53.9	41.2	59.9	46.1
MCD [31]	48.9	68.3	74.6	61.3	67.6	68.8	57	47.1	75.1	69.1	52.2	79.6	64.1
TAT [18]	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 [3]	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 [39]	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 [35]	53.1	72.3	77.2	59.1	71.2	72.1	59.7	53.1	78.4	72.4	60	82.9	67.6
MDD [38]	54.9	73.7	77.8	60	71.4	71.8	61.2	53.6	78.1	72.5	**60.2**	82.3	68.1
BNM [6]	56.2	73.7	79	63.1	73.6	74	62.4	54.8	80.7	72.4	58.9	83.5	69.4
DANN [9]	45.8	63.4	71.9	53.6	61.9	62.6	49.1	39.7	73	64.6	47.8	77.8	59.2
CDAN [21]	50.7	70.6	76	57.6	70	70	57.4	50.9	77.3	70.9	56.7	81.6	65.8
GVB [7]	**57**	74.7	79.8	**64.6**	**74.1**	**74.6**	65.2	**55.1**	**81**	**74.6**	59.7	84.3	**70.4**
CCMDN(ours)	56.5	**74.9**	**80.1**	60.2	73.6	73.7	**66.1**	50.5	80.3	73	**60.2**	**85.4**	69.5

Implementation Details. The Residual Blocks are composed of a 3 by 3 convolution layer, a 1 by 1 convolution layer and a 3 by 3 convolution, and have jump connections. Convolution Blocks each contain two convolution layers, which are activated using LeakyReLU layers. The classifier is a three-layer multi-layer perceptron. The Discriminator consists of 6 convolution layers and 3 down-sampling layers. All modules were optimized with Adam [15] with $\beta_1 = 0.9$ and $\beta_2 = 0.999$, and the learning rate was 0.01 from the beginning to end. In all data sets, the hyperparameter λ_1 is 0.1, λ_2 is 1, and λ_3 is 10. Because we use the same set of hyperparameters on all data sets, we can still achieve ideal results, which fully shows that our method has stronggen eralization ability, and does not need to adjust parameters according to different data sets and tasks, saving a lot of time.

4.2. Comparison with SOTA Methods

Results on Digits [33] are reported in Table 1. Our CCMDN achieves the highest average performance of 97.2% on this dataset, which is 0.3% higher than the second highest ETD. Although it does not appear to be much higher, our method is the first to exceed the average performance of 97%, and this performance has basically saturated on this dataset, so the improvement is only insignificant. Second,

our method is also the highest on task M→U and task S→M, and only 0.2% below the highest on task U→M. Our method is also the first to exceed 96% and approach 97% on task M→U, which shows that our method can adapt to unused settings and perform well on all tasks in this dataset.

Results on VisDA-2017 [29] are reported in Table 2. On this dataset, our method achieved an average performance improvement of 2.2%, which is a very significant progress. This is because our method can achieve good performance in all classes, while other methods have very low performance in some

Table 5. Accuracy (%) on ImageCLEF-DA.

Method	I→P	P→I	I→C	C→I	C→P	P→C	Average
ResNet50 [13]	74.8	83.9	91.5	78	65.5	91.2	80.7
DAN [20]	74.5	82.2	92.8	86.3	69.2	89.8	82.5
DANN [9]	75	86	96.2	87	74.3	91.5	85
JAN [22]	76.8	88	94.7	89.5	74.2	91.7	85.8
HAFN [36]	76.9	89	94.4	89.6	74.9	92.9	86.3
CAT [8]	76.7	89	94.5	89.8	74	93.7	86.3
ETD [17]	81	91.7	97.9	**93.3**	79.5	**95**	89.7
CCMDN(ours)	**82.3**	**92.4**	**98.2**	93.2	**80.1**	94.7	**90.1**

Table 6. Ablation Study.

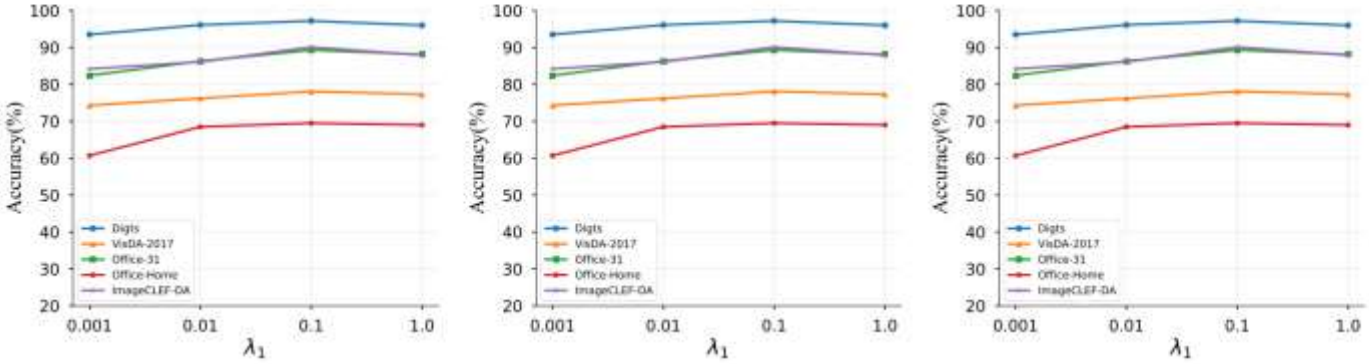
	L^{DL_low}	L^{DL_med}	L^{DL_hig}	Digts	VisDA-2017	Office-31	Office-Home	ImageCLEF-DA
1				88.4	70.5	80.3	62.7	82.6
2	✓			91.5	72.4	83	63.4	84.2
3		✓		92.1	72.8	83.3	64.4	84.7
4			✓	92.3	73	83.7	64.5	85.1
5		✓	✓	96.3	77.3	87.2	68.7	88
6	✓		✓	96.2	77.1	86.6	68.4	87.8
7	✓	✓		95.7	76.8	86.6	68	87.5
8	✓	✓	✓	**97.2**	**78.1**	**89.3**	**69.5**	**90.1**

classes. For example, the performance of DANN [9] on the class truck is less than 10%, and that on the class person is less than 30%. Especially on the class truck, all methods do not exceed 40% precision, and our method is the first method that exceeds 40% precision. In addition, in some classes where the accuracy of all methods is very high, such as class horse and class mcycl, our method is still leading or comparable with other methods. It is precisely because our method has high accuracy in simple classes and not low accuracy in difficult classes that our method is finally ahead of other methods.

Results on Office-31 [30] are reported in Table 3. On the Office-31 dataset, the average performance of our method is still the highest, leading the second place by 0.4%. Although the improvement is not obvious, our method is the first method close to 90% precision on this dataset. It is very difficult to improve every 0.1% on this dataset, so it can fully illustrate the advantages of our method. Although we only achieved the highest performance on task W→D and task D→A, our method is still comparable to the method with the highest performance on other tasks. For example, on task A→D, our method is only 0.1% lower than BNM [6]. Because our method has achieved leading or comparable results in all tasks on this dataset, it finally leads to leading all other methods in average performance.

Results on Office-Home [34] are reported in Table 4. On this dataset, the average performance of our method is only 0.9% lower than that of the highest method GVB [7], and we also achieved the second highest performance. However, our method still achieves the highest performance in tasks Ar→Pr, Ar→Rw, Pr→Ar, Rw→Cl and Rw→Pr, and does not lag far behind in other tasks. The reason why we did not achieve the highest accuracy in this dataset is that we used the same set of super parameters for all tasks, and did not readjust the super parameters because of the change of tasks. Many other methods will change the super parameters of their own models according to different tasks. This fully shows that our method has strong universality and generalization.

Results on ImageCLEF-DA [23] are reported in Table 5. On this dataset, we still achieved the highest performance of 90.1%, which is the first of all methods to exceed 90%, leading the second place by 0.4%. This is a very big improve ment on this dataset. In addition to tasks C→I and P→C, we have achieved the highest performance on all other tasks, far ahead of other methods. In task C→I, it is only 0.1% lower than ETD [17], and in task P→C, it is only 0.3% lower than ETD [17], which fully demonstrates the superiority of our method and can achieve leading or comparable results in almost all tasks.



(a) λ_1 (b) λ_2 (c) λ_3

Figure 2. The effect of hyperparameters λ_1, λ_2 and λ_3.

4.3. Ablation Study

In the ablation study, we mainly discussed the effects of three distillation losses on the model, as shown in Table 6.

Knowledge Distillation of Low-dimensional Features.

Compare the first row with the second row. With this loss, the performance of our method has improved by 3.1%, 1.9%, 2.7%, 0.7% and 1.2% on five data sets. Comparing the fifth row and the eighth row, after removing the loss, our method decreased by 0.9%, 0.8%, 2.4%, 0.8% and 0.9% on the five data sets. This shows that this loss can greatly improve the learning of the source domain features extracted by the feature extraction module on the feature details of the target domain, but many of these details are redundant and useless, so it has no greater impact on the model than the other two distillation losses.

Knowledge Distillation of Medium-dimensional Features.

Compare the first row with the second row. With this loss, the performance of our method has improved by 3.7%, 2.3%, 3.0%, 1.7% and 2.1% on five data sets. Comparing the fifth row and the eighth row, after removing the loss, our method decreased by 1.0%, 1.0%, 2.7%, 1.1% and 1.3% on the five data sets. This shows that this loss can greatly improve the learning of the source domain features extracted by the feature extraction module on the feature structure of the target domain. These structure information will certainly play a very key role in the classifier to judge the category of a picture, but the structure information still belongs to low-level semantic information. Therefore, this loss does not increase the model as much as the loss of high-dimensional feature distillation, but greater than the loss of low-dimensional feature distillation.

Knowledge Distillation of High-dimensional Features.

Compare the first row with the second row. With this loss, the performance of our method has improved by 3.9%, 2.5%, 3.4%, 1.8% and 2.5% on five data sets. Comparing the fifth row and the eighth row, after removing the loss, our method decreased by 1.5%, 1.3%, 2.7%, 1.5% and 1.6% on the five data sets. This shows that this loss can greatly improve the learning of the source domain features extracted by the feature extraction module on the high-level semantic information of the target domain features. This information is very important, and many of it is used to distinguish categories and the source domain from the target domain. Therefore, the impact of this loss on the model performance is far greater than the other two distillation losses.

4.4. Hyperparameter Analysis

We mainly analyze the three hyperparameters that need to be adjusted. See Figure 2.

Hyperparameter λ_1. This hyperparameter is mainly used to control the training of Compression Network and Reconstructed Network. If the Compression Network is not well trained, it will be difficult to extract good features from the images in the target domain, which will inevitably affect

the distillation loss, and the knowledge of the images in the target domain cannot be effectively distilled into the source domain features. Overall, in all data sets, taking 0.1 as the hyperparameter can obtain a good result.

Hyperparameter λ_2. The hyperparameter mainly controls the adversarial loss, which can effectively ensure that the source domain features extracted by the Classification Network can be reconstructed into the target domain style pictures, so as to restrict the source domain features to contain as much information as possible about the target domain pictures, and finally lead to the classifier having good adaptability to the target domain pictures. According to the experimental results, the hyperparameter is taken as 1, and higher results can be obtained in all data sets.

Hyperparameter λ_3. This hyperparameter mainly controls the classification loss. Our model ultimately needs to train a powerful classifier, so the classification loss is the most important. On all data sets, we set the classification loss weight to 10 to enhance the proportion of classification loss. The experimental results also prove that this setting can achieve the highest results in all data sets.

5. Conclusion

This paper proposes a Cross-domain Cross-task Knowl edge Distillation Network, which can distill the knowledge of the target domain pictures obtained by the Compression Network to the source domain picture features obtained by the Classification Network, so that the source domain features contain a large number of target domain picture information, so that the classifier can adapt to the target domain pictures in advance, and finally alleviate the problems caused by cross domain. Finally, sufficient experiments show that our method can achieve leading results in multiple public data sets.

References

[1] Konstantinos Bousmalis, George Trigeorgis, Nathan Silberman, Dilip Krishnan, and Dumitru Erhan. Domain separation networks. *Advances in neural information processing systems*, 29, 2016. 2

[2] Wei-Lun Chang, Hui-Po Wang, Wen-Hsiao Peng, and Wei Chen Chiu. All about structure: Adapting structural information across domains for boosting semantic segmentation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 1900–1909, 2019. 2

[3] Minghao Chen, Shuai Zhao, Haifeng Liu, and Deng Cai. Adversarial-learned loss for domain adaptation. *CoRR*, abs/2001.01046, 2020. 6

[4] Xinyang Chen, Sinan Wang, Mingsheng Long, and Jianmin Wang. Transferability vs. discriminability: Batch spectral penalization for adversarial domain adaptation. In Kamalika Chaudhuri and Ruslan Salakhutdinov, editors, *ICML*, volume 97 of *Proceedings of Machine Learning Research*, pages 1081–1090. PMLR, 2019. 5

[5] Alexis Conneau, Holger Schwenk, Loïc Barrault, and Yann Lecun. Very deep convolutional networks for natural language processing, 2016. cite arxiv:1606.01781. 1

[6] Shuhao Cui, Shuhui Wang, Junbao Zhuo, Liang Li, Qingming Huang, and Qi Tian. Towards discriminability and diversity: Batch nuclear-norm maximization under label insufficient situations. In *CVPR*, pages 3940–3949. IEEE, 2020. 6, 7

[7] Shuhao Cui, Shuhui Wang, Junbao Zhuo, Chi Su, Qingming Huang, and Qi Tian. Gradually vanishing bridge for adversarial domain adaptation. In *CVPR*, pages 12452–12461. IEEE,
2020. 6, 7

[8] Zhijie Deng, Yucen Luo, and Jun Zhu. Cluster alignment with a teacher for unsupervised domain adaptation. In *ICCV*, pages 9943–9952. IEEE, 2019. 2, 5, 6, 7

[9] Yaroslav Ganin, Evgeniya Ustinova, Hana Ajakan, Pascal Germain, Hugo Larochelle, François Laviolette, Mario Marchand, and Victor S. Lempitsky. Domain-adversarial training of neural networks. *J. Mach. Learn. Res.*, 17:59:1–59:35, 2016. 1, 2, 5, 6, 7

[10] Muhammad Ghifary, W. Bastiaan Kleijn, Mengjie Zhang, David Balduzzi, and Wen Li. Deep reconstruction classification networks for unsupervised domain adaptation. *CoRR*, abs/1607.03516, 2016. 5

[11] Boqing Gong, Yuan Shi, Fei Sha, and Kristen Grauman. Geodesic flflow kernel for unsupervised domain adaptation. In *2012 IEEE conference on computer vision and pattern recognition*, pages 2066–2073. IEEE, 2012. 2

[12] Rui Gong, Wen Li, Yuhua Chen, and Luc Van Gool. Dlow: Domain flflow for adaptation and generalization. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 2477–2486, 2019. 2

[13] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition, 2015. citearxiv:1512.03385Comment: Tech report. 5, 6, 7

[14] Judy Hoffman, Eric Tzeng, Taesung Park, Jun-Yan Zhu, Phillip Isola, Kate Saenko, Alexei A. Efros, and Trevor Darrell. Cycada: Cycle-consistent adversarial domain adaptation. *CoRR*, abs/1711.03213, 2017. 5

[15] Diederik P. Kingma and Jimmy Ba. Adam: A method for stochastic optimization, 2014. cite arxiv:1412.6980Comment: Published as a conference paper at the 3rd International Conerence for Learning Representations, San Diego, 2015. 6

[16] Y. LeCun, L. Bottou, Y. Bengio, and P. Haffner. Gradient based learning applied to document recognition. *Proceedings of the IEEE*, 86(11):2278–2324, 1998. 5

[17] Mengxue Li, Yiming Zhai, You-Wei Luo, Pengfei Ge, and Chuan-Xian Ren. Enhanced transport distance for unsuper vised domain adaptation. In *CVPR*, pages 13933–13941. IEEE, 2020. 5, 6, 7

[18] Hong Liu, Mingsheng Long, Jianmin Wang, and Michael I. Jordan. Transferable adversarial training: A general approach to adapting deep classifiers. In Kamalika Chaudhuri and Rus lan Salakhutdinov, editors, *ICML*, volume 97 of *Proceedings of Machine Learning Research*, pages 4013–4022. PMLR, 2019. 2, 6

[19] Ming-Yu Liu and Oncel Tuzel. Coupled generative adversarial networks. In Daniel D. Lee, Masashi Sugiyama, Ulrike von Luxburg, Isabelle Guyon, and Roman Garnett, editors, *NIPS*, pages 469–477, 2016. 5

[20] Mingsheng Long, Yue Cao, Jianmin Wang, and Michael I. Jordan. Learning transferable features with deep adaptation networks. In Francis R. Bach and David M. Blei, editors, *ICML*, volume 37 of *JMLR Workshop and Conference Proceedings*, pages 97–105. JMLR.org, 2015. 1, 2, 5, 6, 7

[21] Mingsheng Long, Zhangjie Cao, Jianmin Wang, and Michael I. Jordan. Conditional adversarial domain adaptation. In Samy Bengio, Hanna M. Wallach, Hugo Larochelle, Kristen Grauman, Nicolò Cesa-Bianchi, and Roman Garnett, editors, *NeurIPS*, pages 1647–1657, 2018. 2, 5, 6

[22] Mingsheng Long, Han Zhu, Jianmin Wang, and Michael I. Jordan. Deep transfer learning with joint adaptation networks. In Doina Precup and Yee Whye Teh, editors, *ICML*, volume 70 of *Proceedings of Machine Learning Research*, pages 2208–2217. PMLR, 2017. 7

[23] Henning Müller, Paul Clough, Thomas Deselaers, and Barbara Caputo, editors. *ImageCLEF: Experimental Evaluation in Visual Information Retrieval*, volume 32 of *The Information Retrieval Series*. Springer, Berlin, 2010. 5, 7

[24] Yuval Netzer, Tao Wang, Adam Coates, Alessandro Bissacco, Bo Wu, and Andrew Y. Ng. Reading digits in natural images with unsupervised feature learning. 2011. 5

[25] Jie Ni, Qiang Qiu, and Rama Chellappa. Subspace interpolation via dictionary learning for unsupervised domain adaptation. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 692–699, 2013. 2

[26] Xiao Ni and Zhang Lei. Dynamic weighted learning for unsupervised domain adaptation. In *CVPR*. IEEE Computer Society, 2021. 6

[27] Yingwei Pan, Ting Yao, Yehao Li, Yu Wang, Chong-Wah Ngo, and Tao Mei. Transferrable prototypical networks for unsupervised domain adaptation. *CoRR*, abs/1904.11227, 2019. 5

[28] A. D. Parkins and A. K. Nandi. Genetic programming techniques for hand written digit recognition. *Signal Processing*, 84(12):2345–2365, December 2004. 5

[29] Xingchao Peng, Ben Usman, Neela Kaushik, Judy Hoffman, Dequan Wang, and Kate Saenko. Visda: The visual domain adaptation challenge. *CoRR*, abs/1710.06924, 2017. 5, 6

[30] Kate Saenko, Brian Kulis, Mario Fritz, and Trevor Darrell. Adapting visual category models to new domains. In Kostas Daniilidis, Petros Maragos, and Nikos Paragios, editors, *ECCV (4)*, volume 6314 of *Lecture Notes in Computer Science*, pages 213–226. Springer, 2010. 2, 5, 7

[31] Kuniaki Saito, Kohei Watanabe, Yoshitaka Ushiku, and Tatsuya Harada. Maximum classifier discrepancy for unsuper vised domain adaptation. *CoRR*, abs/1712.02560, 2017. 2, 5, 6

[32] Sasha Targ, Diogo Almeida, and Kevin Lyman. Resnet in resnet: Generalizing residual architectures. *CoRR*, abs/1603.08029, 2016. 1

[33] Eric Tzeng, Judy Hoffman, Kate Saenko, and Trevor Darrell. Adversarial discriminative domain adaptation. pages 2962– 2971. IEEE Computer Society, 2017. 1, 2, 5, 6

[34] Hemanth Venkateswara, Jose Eusebio, Shayok Chakraborty, and Sethuraman Panchanathan. Deep hashing network for unsupervised domain adaptation. In *CVPR*, pages 5385–5394. IEEE Computer Society, 2017. 5, 7

[35] Ximei Wang, Liang Li, Weirui Ye, Mingsheng Long, and Jianmin Wang. Transferable attention for domain adaptation. In *AAAI*, pages 5345–5352. AAAI Press, 2019. 6

[36] Ruijia Xu, Guanbin Li, Jihan Yang, and Liang Lin. Larger norm more transferable: An adaptive feature norm approach for unsupervised domain adaptation. In *ICCV*, pages 1426–1435. IEEE, 2019. 7

[37] Shaokai Ye, Kailu Wu, Mu Zhou, Yunfei Yang, Sia Huat Tan, Kaidi Xu, Jiebo Song, Chenglong Bao, and Kaisheng Ma. Light-weight calibrator: A separable component for unsupervised domain adaptation. In *CVPR*, pages 13733–13742. IEEE, 2020. 5

[38] Yuchen Zhang, Tianle Liu, Mingsheng Long, and Michael I. Jordan. Bridging theory and algorithm for domain adaptation. In Kamalika Chaudhuri and Ruslan Salakhutdinov, editors, *ICML*, volume 97 of *Proceedings of Machine Learning Research*, pages 7404–7413. PMLR, 2019. 6

[39] Yabin Zhang, Hui Tang, Kui Jia, and Mingkui Tan. Domain symmetric networks for adversarial domain adaptation. *CoRR*, abs/1904.04663, 2019. 2, 6