Neurocomputing 312 (2018) 135–153

Contents lists available at ScienceDirect

Neurocomputing

journal homepage: www.elsevier.com/locate/neucom

Deep visual domain adaptation: A survey

Mei Wang, Weihong Deng*

School of Information and Communication Engineering, Beijing University of Posts and Telecommunications, Beijing 100876, China

ARTICLE INFO

Article history:
Received 8 January 2018
Revised 22 May 2018
Accepted 24 May 2018
Available online 30 May 2018

Communicated by Dr. Ivor Tsang

Keywords:
Deep domain adaptation
Deep networks
Transfer learning
Computer vision applications

ABSTRACT

Deep domain adaptation has emerged as a new learning technique to address the lack of massive amounts of labeled data. Compared to conventional methods, which learn shared feature subspaces or reuse important source instances with shallow representations, deep domain adaptation methods leverage deep networks to learn more transferable representations by embedding domain adaptation in the pipeline of deep learning. There have been comprehensive surveys for shallow domain adaptation, but few timely reviews the emerging deep learning based methods. In this paper, we provide a comprehensive survey of deep domain adaptation methods for computer vision applications with four major contributions. First, we present a taxonomy of different deep domain adaptation scenarios according to the properties of data that define how two domains are diverged. Second, we summarize deep domain adaptation approaches into several categories based on training loss, and analyze and compare briefly the state-of-the-art methods under these categories. Third, we overview the computer vision applications that go beyond image classification, such as face recognition, semantic segmentation and object detection. Fourth, some potential deficiencies of current methods and several future directions are highlighted.

1. Introduction

Over the past few years, machine learning has achieved great success and has benefited real-world applications. However, collecting and annotating datasets for every new task and domain are extremely expensive and time-consuming processes, sufficient training data may not always be available. Fortunately, the big data era makes a large amount of data available for other domains and tasks. For instance, although large-scale labeled video databases that are publicly available only contain a small number of samples, statistically, the YouTube face dataset (YTF) consists of 3.4 K videos. The number of labeled still images is more than sufficient [1]. Hence, skillfully using the auxiliary data for the current task with scarce data will be helpful for real-world applications.

However, due to many factors (e.g., illumination, pose, and image quality), there is always a distribution change or domain shift between two domains that can degrade the performance, as shown in Fig. 1. Mimicking the human vision system, domain adaptation (DA) is a particular case of transfer learning (TL) that utilizes labeled data in one or more relevant source domains to execute new tasks in a target domain. Over the past decades, various shallow DA methods have been proposed to solve a domain shift between the source and target domains. The common algorithms for shallow DA can mainly be categorized into two classes: instance-based DA [2,3] and feature-based DA [4–7]. The first class reduces the discrepancy by reweighting the source samples, and it trains on the weighted source samples. For the second class, a common shared space is generally learned in which the distributions of the two datasets are matched.

Recently, neural-network-based deep learning approaches have achieved many inspiring results in visual categorization applications, such as image classification [8], face recognition [9], and object detection [10]. Simulating the perception of the human brain, deep networks can represent high-level abstractions by multiple layers of non-linear transformations. Existing deep network architectures [11] include convolutional neural networks (CNNs) [8,12–14], deep belief networks (DBNs) [15], and stacked autoencoders (SAEs) [16], among others. Although some studies have shown that deep networks can learn more transferable representations that disentangle the exploratory factors of variations underlying the data samples and group features hierarchically in accordance with their relatedness to invariant factors, Donahue et al. [17] showed that a domain shift still affects their performance. The deep features would eventually transition from general to specific, and the transferability of the representation sharply decreases in higher layers. Therefore, recent work has addressed this problem by deep DA, which combines deep learning and DA.

* Corresponding author.
E-mail address: whdeng@bupt.edu.cn (W. Deng).

https://doi.org/10.1016/j.neucom.2018.05.083
0925-2312/

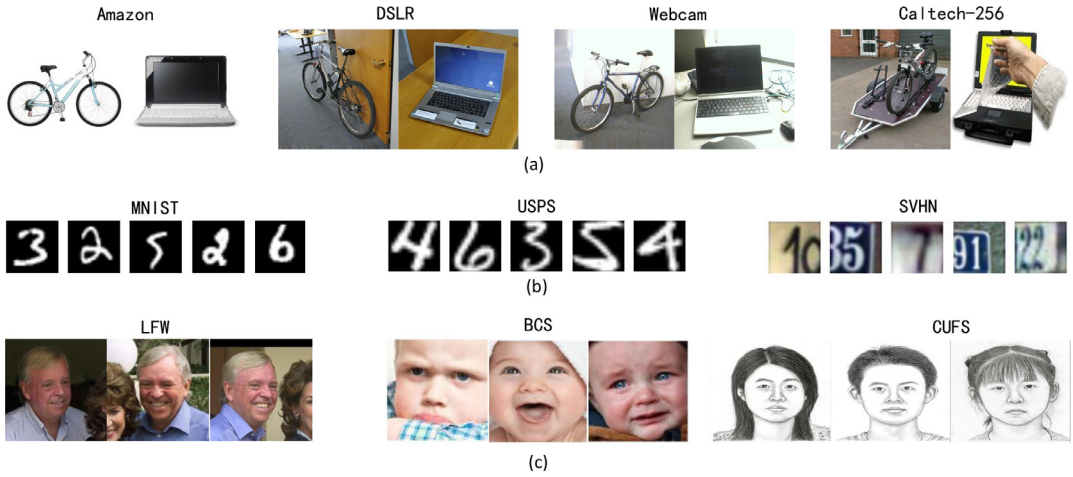


Fig. 1. (a) Some object images from the "Bike" and "Laptop" categories in Amazon, DSLR, Webcam, and Caltech-256 databases. (b) Some digit images from MNIST, USPS, and SVHN databases. (c) Some face images from LFW, BCS and CUFS databases. Realworld computer vision applications, such as face recognition, must learn to adapt to distributions specific to each domain.

There have been other surveys on TL and DA over the past few years [18–23]. Pan and Yang [18] categorized TL under three subsettings, including inductive TL, transductive TL, and unsupervised TL, but they only studied homogeneous feature spaces. Shao et al. [19] categorized TL techniques into feature-representation-level knowledge transfer and classifier-level knowledge transfer. The survey written by Patel et al. [21] only focused on DA, a subtopic of TL. Day and Khoshgoftaar [20] discussed 38 methods for heterogeneous TL that operate under various settings, requirements, and domains. Zhang et al. [22] were the first to summarize several transferring criteria in detail from the concept level. These five surveys mentioned above only cover the methodologies on shallow TL or DA. The work presented by Csurka [23] briefly analyzed the state-of-the-art shallow DA methods and categorized the deep DA methods into three subsettings based on training loss: classification loss, discrepancy loss and adversarial loss. However, Csurka's work mainly focused on shallow methods, and it only discussed deep DA in image classification applications.

In this paper, we focus on analyzing and discussing deep DA methods. Specifically, the key contributions of this survey are as follows: (1) we present a taxonomy of different deep DA scenarios according to the properties of data that define how two domains are diverged. (2) extending Csurka's work, we improve and detail the three subsettings (training with classification loss, discrepancy loss and adversarial loss) and summarize different approaches used in different DA scenes. (3) Considering the distance of the source and target domains, multi-step DA methods are studied and categorized into hand-crafted, feature-based and representation-based mechanisms. (4) We provide a survey of many computer vision applications, such as image classification, face recognition, style translation, object detection, semantic segmentation and person re-identification.

The remainder of this survey is structured as follows. In Section 2, we first define some notations, and then we categorize deep DA into different settings (given in Fig. 2). In the next three sections, different approaches are discussed for each setting, which are given in Table 1 and Table 2 in detail. Then, in Section 6, we introduce some successful computer vision applications of deep DA. Finally, the conclusion of this paper and discussion of future works are presented in Section 7.

2. Overview

2.1. Notations and definitions

In this section, we introduce some notations and definitions that are used in this survey. The notations and definitions match those from the survey papers by [18,23] to maintain consistency across surveys. A domain $\mathcal{D}$ consists of a feature space $\mathcal{X}$ and a marginal probability distribution $P(X)$, where $X = \{x_1, \ldots, x_n\} \in \mathcal{X}$. Given a specific domain $\mathcal{D} = \{\mathcal{X}, P(X)\}$, a task $\mathcal{T}$ consists of a feature space $\mathcal{Y}$ and an objective predictive function $f(\cdot)$, which can also be viewed as a conditional probability distribution $P(Y|X)$ from a probabilistic perspective. In general, we can learn $P(Y|X)$ in a supervised manner from the labeled data $\{x_i, y_i\}$, where $x_i \in \mathcal{X}$ and $y_i \in \mathcal{Y}$.

Assume that we have two domains: the training dataset with sufficient labeled data is the source domain $\mathcal{D}^s = \{\mathcal{X}^s, P(X)^s\}$, and the test dataset with a small amount of labeled data or no labeled data is the target domain $\mathcal{D}^t = \{\mathcal{X}^t, P(X)^t\}$. We see that the partially labeled part, $\mathcal{D}^{tl}$, and the unlabeled parts, $\mathcal{D}^{tu}$, form the entire target domain, that is, $\mathcal{D}^t = \mathcal{D}^{tl} \cup \mathcal{D}^{tu}$. Each domain is together with its task: the former is $\mathcal{T}^s = \{\mathcal{Y}^s, P(Y^s|X^s)\}$, and the latter is $\mathcal{T}^t = \{\mathcal{Y}^t, P(Y^t|X^t)\}$. Similarly, $P(Y^s|X^s)$ can be learned from the source labeled data $\{x_i^s, y_i^s\}$, while $P(Y^t|X^t)$ can be learned from labeled target data $\{x_i^{tl}, y_i^{tl}\}$ and unlabeled data $\{x_i^{tu}\}$.

2.2. Different settings of domain adaptation

The case of traditional machine learning is $\mathcal{D}^s = \mathcal{D}^t$ and $\mathcal{T}^s = \mathcal{T}^t$. For TL, Pan and Yang [18] summarized that the differences between different datasets can be caused by domain divergence $\mathcal{D}^s \neq \mathcal{D}^t$ (i.e., distribution shift or feature space difference) or task divergence $\mathcal{T}^s \neq \mathcal{T}^t$ (i.e., conditional distribution shift or label space difference), or both. Based on this summary, Pan et al. categorized TL into three main groups: inductive, transductive and unsupervised TL.

According to this classification, DA methods are transductive TL solutions with the assumption that the tasks are the same, i.e., $\mathcal{T}^s = \mathcal{T}^t$, and the differences are only caused by domain divergence, $\mathcal{D}^s \neq \mathcal{D}^t$. Therefore, DA can be split into two main categories based on different domain divergences (distribution shift or feature space difference): homogeneous and heterogeneous DA.

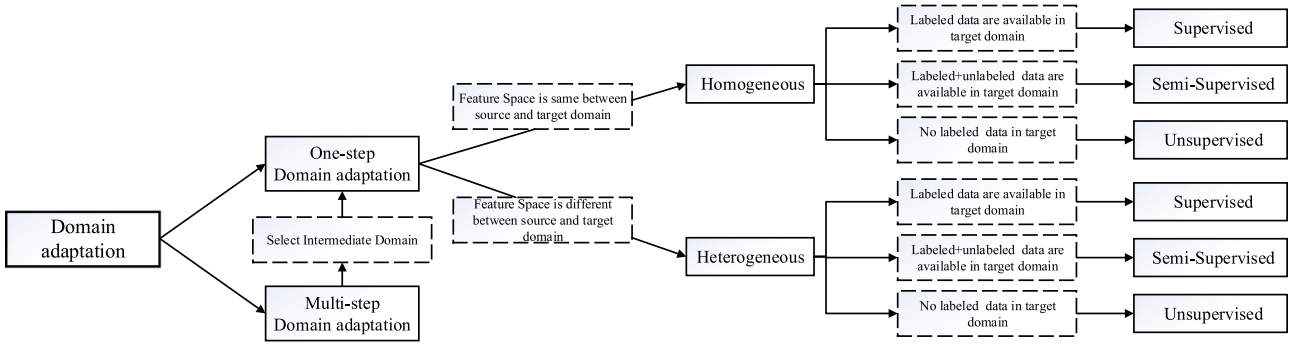


Fig. 2. An overview of different settings of domain adaptation.

Table 1
Different deep approaches to one-step DA.

One-step DA approaches	Brief description	subsettings
Discrepancy-based	fine-tuning the deep network with labeled or unlabeled target data to diminish the domain shift	class criterion [26,27] [28–32] [26,33–36] statistic criterion [32,34,37–39] [40–43] architecture criterion [44–48] [49] geometric criterion [50]
Adversarial-based	using domain discriminators to encourage domain confusion through an adversarial objective	generative models [51–53] non-generative models [26,54–57] [58]
Reconstruction-based	using the data reconstruction as an auxiliary task to ensure feature invariance	encoder–decoder reconstruction [43,59–61] adversarial reconstruction [62–64]

Table 2
Different deep approaches to multi-step DA.

Multi-step approaches	Brief description
Hand-crafted	users determine the intermediate domains based on experience [65]
Instance-based	selecting certain parts of data from the auxiliary datasets to compose the intermediate domains [25,50]
Representation-based	freeze weights of one network and use their intermediaterepresentations as input to the new network [66]

Then, we can further categorize DA into supervised, semi-supervised and unsupervised DA in consideration of labeled data of the target domain. The classification is given in Fig. 2.

- In the **homogeneous DA** setting, the feature spaces between the source and target domains are identical ($\mathcal{X}^s = \mathcal{X}^t$) with the same dimension ($d^s = d^t$). Hence, the source and target datasets are generally different in terms of data distributions ($P(X)^s \neq P(X)^t$).

In addition, we can further categorize the homogeneous DA setting into three cases:

1. In the supervised DA, a small amount of labeled target data, $\mathcal{D}^{tl}$, are present. However, the labeled data are commonly not sufficient for tasks.
2. In the semi-supervised DA, both limited labeled data, $\mathcal{D}^{tl}$, and redundant unlabeled data, $\mathcal{D}^{tu}$, in the target domain are available in the training stage, which allows the networks to learn the structure information of the target domain.
3. In the unsupervised DA, no labeled but sufficient unlabeled target domain data, $\mathcal{D}^{tu}$, are observable when training the network.

- In the **heterogeneous DA** setting, the feature spaces between the source and target domains are nonequivalent ($\mathcal{X}^s \neq \mathcal{X}^t$), and the dimensions may also generally differ ($d^s \neq d^t$).

Similar to the homogeneous setting, the heterogeneous DA setting can also be divided into supervised, semi-supervised and unsupervised DA.

All of the above DA settings assumed that the source and target domains are directly related; thus, transferring knowledge can be accomplished in one step. We call them one-step DA. In reality, however, this assumption is occasionally unavailable. There is little overlap between the two domains, and performing one-step DA will not be effective. Fortunately, there are some intermediate domains that are able to draw the source and target domains closer than their original distance. Thus, we use a series of intermediate bridges to connect two seemingly unrelated domains and then perform one-step DA via this bridge, named multi-step (or transitive) DA [24,25]. For example, face images and vehicle images are dissimilar between each other due to different shapes or other aspects, and thus, one-step DA would fail. However, some intermediate images, such as 'football helmet', can be introduced to be an intermediate domain and have a smooth knowledge transfer. Fig. 3 shows the differences between the learning processes of one-step and multi-step DA techniques.

3. Approaches of deep domain adaptation

In a broad sense, deep DA is a method that utilizes a deep network to enhance the performance of DA. Under this definition, shallow methods with deep features [17,67–70] can be considered as a deep DA approach. DA is adopted by shallow methods, whereas deep networks only extract vectorial features and are not helpful for transferring knowledge directly. For example, Lu et al. [71] extracted the convolutional activations from a CNN as the tensor representation, and then performed tensor-aligned

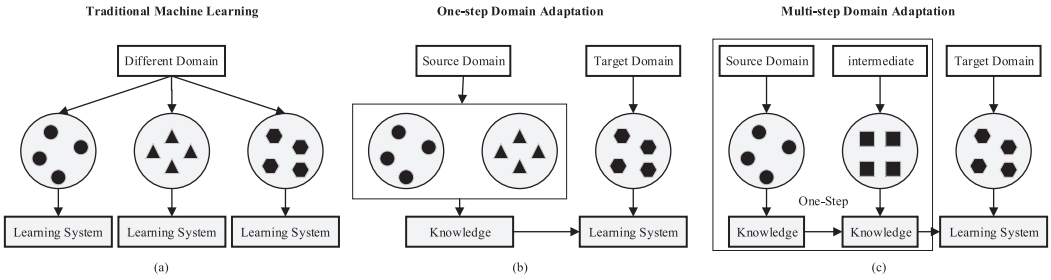


Fig. 3. Different learning processes between (a) traditional machine learning, (b) one-step domain adaptation and (c) multi-step domain adaptation [18].

invariant subspace learning to realize DA. This approach reliably outperforms current state-of-the-art approaches based on traditional hand-crafted features because sufficient representational and transferable features can be extracted through deep networks, which can work better on discrimination tasks [17].

In a narrow sense, deep DA is based on deep learning architectures designed for DA and can obtain a firsthand effect from deep networks via back-propagation. The intuitive idea is to embed DA into the process of learning representation and to learn a deep feature representation that is both semantically meaningful and domain invariant. With the "good" feature representations, the performance of the target task would improve significantly. In this paper, we focus on the narrow definition and discuss how to utilize deep networks to learn "good" feature representations with extra training criteria.

3.1. Categorization of one-step domain adaptation

In one-step DA, the deep approaches can be summarized into three cases, which refers to [23]. Table 1 shows these three cases and brief descriptions. The first case is the discrepancy-based deep DA approach, which assumes that fine-tuning the deep network model with labeled or unlabeled target data can diminish the shift between the two domains. Class criterion, statistic criterion, architecture criterion and geometric criterion are four major techniques for performing fine-tuning.

- **Class criterion:** uses the class label information as a guide for transferring knowledge between different domains. When the labeled samples from the target domain are available in supervised DA, soft label and metric learning are always effective [26–28,30,31]. When such samples are unavailable, some other techniques can be adopted to substitute for class labeled data, such as pseudo labels [29,32–34] and attribute representation [26,35].

- **Statistic criterion:** aligns the statistical distribution shift between the source and target domains using some mechanisms. The most commonly used methods for comparing and reducing distribution shift are maximum mean discrepancy (MMD) [32,34,37–40], correlation alignment (CORAL) [41,42], Kullback–Leibler (KL) divergence [43] and $\mathcal{H}$ divergence, among others.

- **Architecture criterion:** aims at improving the ability of learning more transferable features by adjusting the architectures of deep networks. The techniques that are proven to be cost effective include adaptive batch normalization (BN) [44–46], weak-related weight [47], domain-guided dropout [48], and so forth.

- **Geometric criterion:** bridges the source and target domains according to their geometrical properties. This criterion assumes that the relationship of geometric structures can reduce the domain shift [50].

The second case can be referred to as an adversarial-based deep DA approach [54]. In this case, a domain discriminator that classifies whether a data point is drawn from the source or target domain is used to encourage domain confusion through an adversarial objective to minimize the distance between the empirical source and target mapping distributions. Furthermore, the adversarial-based deep DA approach can be categorized into two cases based on whether there are generative models.

- **Generative models:** combine the discriminative model with a generative component in general based on generative adversarial networks (GANs). One of the typical cases is to use source images, noise vectors or both to generate simulated samples that are similar to the target samples and preserve the annotation information of the source domain [51–53].

- **Non-generative models:** rather than generating models with input image distributions, the feature extractor learns a discriminative representation using the labels in the source domain and maps the target data to the same space through a domain-confusion loss, thus resulting in the domain-invariant representations [26,54–56,58].

The third case can be referred to as a reconstruction-based DA approach, which assumes that the data reconstruction of the source or target samples can be helpful for improving the performance of DA. The reconstructor can ensure both specificity of intra-domain representations and indistinguishability of inter-domain representations.

- **Encoder-decoder reconstruction:** by using stacked autoencoders (SAEs), encoder–decoder reconstruction methods combine the encoder network for representation learning with a decoder network for data reconstruction [43,59–61].

- **Adversarial reconstruction:** the reconstruction error is measured as the difference between the reconstructed and original images within each image domain by a cyclic mapping obtained via a GAN discriminator, such as dual GAN [62], cycle GAN [63] and disco GAN [64].

3.2. Categorization of multi-step domain adaptation

In multi-step DA, we first determine the intermediate domains that are more related with the source and target domains than their direct connection. Second, the knowledge transfer process will be performed between the source, intermediate and target domains by one-step DA with less information loss. Thus, the key of multi-step DA is how to select and utilize intermediate domains; additionally, it can fall into three categories referring to [18]: hand-crafted, feature-based and representation-based selection mechanisms.

- **Hand-crafted:** users determine the intermediate domains based on experience [65].

Table 3
Different approaches used in different domain adaptation settings.

		Supervised DA	Unsupervised DA
Discrepancy-based	Class criterion	√	
	Statistic criterion		√
	Architecture criterion	√	√
	Geometric criterion	√	
Adversarial-based	Generative model		√
	Non-generative model		√
Reconstruction-based	encoder–decoder Model		√
	Adversarial Model		√

Table 4
Some common rules of thumb for deciding fine-tuned or frozen in the first n layers [73].

		The size of target dataset		
		Low	Medium	High
The distance between source and target	Low	Freeze	Try freeze or tune	Tune
	Medium	Try freeze or tune	Tune	Tune
	High	Try freeze or tune	Tune	Tune

- **Instance-based:** selecting certain parts of data from the auxiliary datasets to compose the intermediate domains to train the deep network [25,50].
- **Representation-based:** transfer is enabled via freezing the previously trained network and using their intermediate representations as input to the new one [66].

4. One-step domain adaptation

As mentioned in Section 2.1, the data in the target domain have three types regardless of homogeneous or heterogeneous DA: (1) supervised DA with labeled data, (2) semi-supervised DA with labeled and unlabeled data and (3) non-supervised DA with unlabeled data. The second setting is able to be accomplished by combining the methods of setting 1 and setting 3; thus, we only focus on the first and third settings in this paper. The cases where the different approaches are mainly used for each DA setting are shown in Table 3. As shown, more work is focused on unsupervised scenes because supervised DA has its limitations. When only few labeled data in the target domain are available, using the source and target labeled data to train parameters of models typically results in overfitting to the source distribution. In addition, the discrepancy-based approaches have been studied for years and produced more methods in many research works, whereas the adversarial-based and reconstruction-based approaches are a relatively new research topic but have recently been attracting more attention.

4.1. Homogeneousdomain adaptation

4.1.1. Discrepancy-based approaches

Yosinski et al. [72] proved that transferable features learned by deep networks have limitations due to fragile co-adaptation and representation specificity and that fine-tuning can enhance generalization performance (Fig. 4). Fine-tuning (can also be viewed as a discrepancy-based deep DA approach) is to train a base network with source data and then directly reuse the first n layers to conduct a target network. The remaining layers of the target network are randomly initialized and trained with loss based on discrepancy. During training, the first n layers of the target network can be fine-tuned or frozen depending on the size of the target dataset and its similarity to the source dataset [73]. Some common rules of thumb for navigating the 4 major scenarios are given in Table 4.

- **Class criterion**

The class criterion is the most basic training loss in deep DA. After pre-training the network with source data, the remaining layers of the target model use the class label information as a guide to train the network. Hence, a small number of labeled samples from the target dataset is assumed to be available.

Ideally, the class label information is given directly in supervised DA. Most work commonly uses the negative log-likelihood of the ground truth class with softmax as their training loss, $\mathcal{L} = -\sum_{i=0}^{N} y_i \log \hat{y}_i$ ($\hat{y}_i$ are the softmax predictions of the model, which represent class probabilities) [26,27,30,74]. To extend this, Hinton et al. [31] modified the softmax function to soft label loss:

$$q_i = \frac{\exp(z_i/T)}{\sum_j (\exp(z_j/T))} \tag{1}$$

where z^i is the logit output computed for each class. T is a temperature that is normally set to 1 in standard softmax, but it takes a higher value to produce a softer probability distribution over classes. By using it, much of the information about the learned function that resides in the ratios of very small probabilities can be obtained. For example, when recognizing digits, one version of 2 may obtain a probability of 10^6 of being a 3 and 10^9 of being a 7; in other words, this version of 2 looks more similar to 3 than 7. Inspired by Tzeng et al., [26] fine-tuned the network by simultaneously minimizing the domain confusion loss (belonging to adversarial-based approaches, which will be presented in Section 4.1.2) and soft label loss. Using soft labels rather than hard labels can preserve the relationships between classes across domains. Gebru et al. [35] modified existing adaptation algorithms based on [26] and utilized soft label loss at the fine-grained class level $\mathcal{L}_{csoft}$ and attribute level $\mathcal{L}_{asoft}$ (Fig. 5).

In addition to softmax loss, there are other methods that can be used as training loss to fine-tune the target model in supervised DA. Embedding metric learning in deep networks is another method that can make the distance of samples from different domains with the same labels be closer while those with different labels are far away. Based on this idea, [28] constructed the semantic alignment loss and the separation loss accordingly. Deep transfer metric learning is proposed by Hu et al. [30], which applies the marginal Fisher analysis criterion and MMD criterion (described in

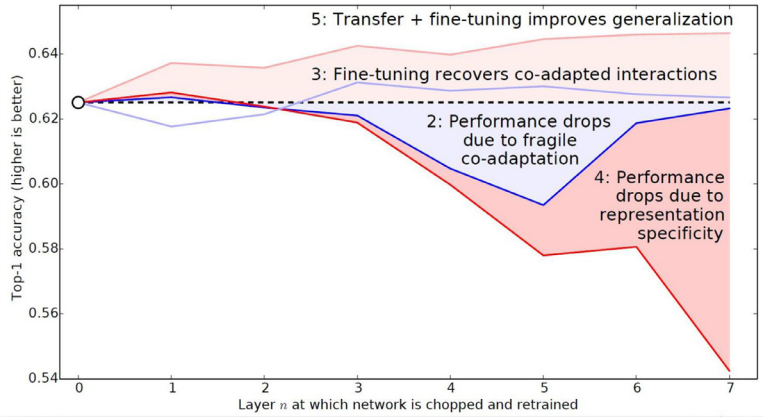


Fig. 4. The average accuracy over the validation set for a network trained with different strategies. Baseline B: the network is trained on dataset B. (2) BnB: the first n layers are reused from baseline B and frozen. The higher layers are trained on dataset B. (3) BnB+: the same as BnB but where all layers are fine-tuned. (4) AnB: the first n layers are reused from the network trained on dataset A and frozen. The higher layers are trained on dataset B. (5) AnB+: the same as AnB but where all layers are fine-tuned [72].

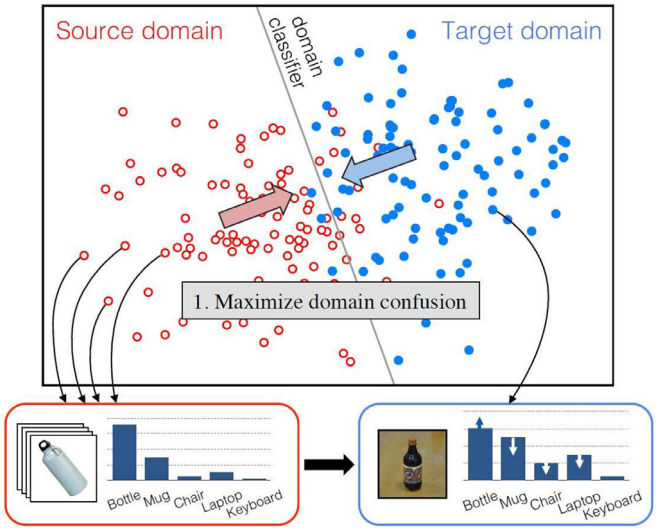


Fig. 5. Deep DA by combining domain confusion loss and soft label loss [26].

Statistic Criterion) to minimize their distribution difference:

$$\min \mathcal{J} = S_c^{(M)} - \alpha S_b^{(M)} + \beta D_{ts}^{(M)}\left(\mathcal{X}^s, \mathcal{X}^t\right) + \gamma \sum_{m=1}^{M} \left(\left\| W^{(m)} \right\|_F^2 + \left\| b^{(m)} \right\|_2^2 \right) \tag{2}$$

where α, β and γ are regularization parameters and $W^{(m)}$ and $b^{(m)}$ are the weights and biases of the *mth* layer of the network. $D_{ts}^{(M)}\left(\mathcal{X}^s, \mathcal{X}^t\right)$ is the MMD between representations of the source and target domains. S_c and S_b define the intra-class compactness and the interclass separability.

However, what can we do if there is no class label information in the target domain directly? As we all know, humans can identify unseen classes given only a high-level description. For instance, when provided the description "tall brown animals with long necks", we are able to recognize giraffes. To imitate the ability of humans, [75] introduced high-level semantic attributes per class. Assume that $a^c = (a_1^c, \ldots, a_m^c)$ is the attribute representation for class c, which has fixed-length binary values with m attributes in all the classes. The classifiers provide estimates of $p(a_m|x)$ for each attribute a_m. In the test stage, each target class y obtains its attribute vector a^y in a deterministic way, i.e., $p(a|y) = [\![a = a^y]\!]$. By applying Bayes rule, $p(y|a) = \frac{p(y)}{p(a^y)}[\![a = a^y]\!]$, the posterior of a test class can be calculated as follows:

$$p(y|x) = \sum_{a \in \{0,1\}^M} p(y|a)p(a|x) = \frac{p(y)}{p(a^y)} \prod_{m=1}^{M} p(a_m^y|x) \tag{3}$$

Gebru et al. [35] drew inspiration from these works and leveraged attributes to improve performance in the DA of fine-grained recognition. There are multiple independent softmax losses that simultaneously perform attribute and class level to fine-tune the target model. To prevent the independent classifiers from obtaining conflicting labels with attribute and class level, an attribute consistency loss is also implemented.

Occasionally, when fine-tuning the network in unsupervised DA, a label of target data, which is called a pseudo label, can preliminarily be obtained based on the maximum posterior probability. Yan et al. [34] initialized the target model using the source data and then defined the class posterior probability $p(y_j^t = c|x_j^t)$ by the output of the target model. With $p(y_j^t = c|x_j^t)$, they assigned pseudo-label $\widehat{y_j^t}$ to x_j^t by $\widehat{y_j^t} = \arg\max_c p(y_j^t = c|x_j^t)$. In [29], two different networks assign pseudo-labels to unlabeled samples, another network is trained by the samples to obtain target discriminative representations. The deep transfer network (DTN) [33] used some base classifiers, e.g., SVMs and MLPs, to obtain the pseudo labels for the target samples to estimate the conditional distribution of the target samples and match both the marginal and the conditional distributions with the MMD criterion. When casting the classifier adaptation into the residual learning framework, [32] used the pseudo label to build the conditional entropy $E(\mathcal{D}^t, f^t)$, which ensures that the target classifier f^t fits the target-specific structures well.

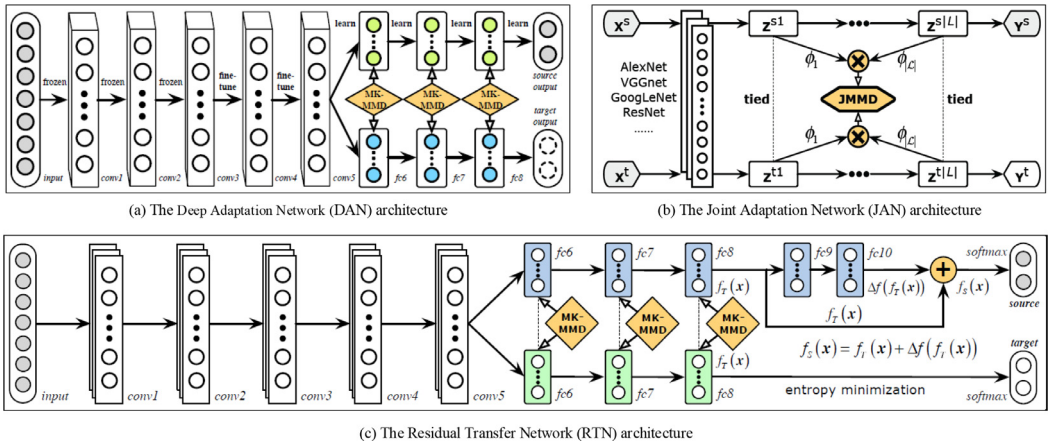
- **Statistic criterion**

Although some discrepancy-based approaches search for pseudo labels, attribute labels or other substitutes to labeled target data, more work focuses on learning domain-invariant representations via minimizing the domain distribution discrepancy in unsupervised DA.

MMD is an effective metric for comparing the distributions between two datasets by a kernel two-sample test [76]. Given two distributions s and t, the MMD is defined as follows:

$$MMD^2(s,t) = \sup_{\|\phi\|_{\mathcal{H}} \le 1} \left\| E_{\mathrm{x}^s \sim s}[\phi(\mathrm{x}^s)] - E_{\mathrm{x}^t \sim s}[\phi(\mathrm{x}^t)] \right\|_{\mathcal{H}}^2 \tag{4}$$

where ϕ represents the kernel function that maps the original data to a reproducing kernel Hilbert space (RKHS) and $\|\phi\|_{\mathcal{H}} \le 1$ defines a set of functions in the unit ball of RKHS $\mathcal{H}$.



(a) The Deep Adaptation Network (DAN) architecture

(b) The Joint Adaptation Network (JAN) architecture

(c) The Residual Transfer Network (RTN) architecture

Fig. 6. Different approaches with the MMD metric. (a) The deep adaptation network (DAN) architecture [38], (b) the joint adaptation network (JAN) architecture [37] and (c) the residual transfer network (RTN) architecture [32].

Based on the above, Ghifary et al. [40] proposed a model that introduced the MMD metric in feedforward neural networks with a single hidden layer. The MMD metric is computed between representations of each domain to reduce the distribution mismatch in the latent space. The empirical estimate of MMD is as follows:

$$MMD^2(D_s, D_t) = \left\| \frac{1}{M}\sum_{i=1}^{M}\phi(\mathrm{x}_i^s) - \frac{1}{N}\sum_{j=1}^{N}\phi(\mathrm{x}_j^t) \right\|_H^2 \tag{5}$$

Subsequently, Tzeng et al. [39] and Long et al. [38] extended MMD to a deep CNN model and achieved great success. The deep domain confusion network (DDC) by Tzeng et al. [39] used two CNNs for the source and target domains with shared weights. The network is optimized for classification loss in the source domain, while domain difference is measured by an adaptation layer with the MMD metric.

$$\mathcal{L} = \mathcal{L}_C(X^L, y) + \lambda MMD^2(X^s X^t) \tag{6}$$

where the hyperparameter λ is a penalty parameter. $\mathcal{L}_C(X^L, y)$ denotes classification loss on the available labeled data, X^L, and the ground-truth labels, y. $MMD^2(X^s X^t)$ denotes the distance between the source and target data. DDC only adapts one layer of the network, resulting in a reduction in the transferability of multiple layers. Rather than using a single layer and linear MMD, Long et al. [38] proposed the deep adaptation network (DAN) that matches the shift in marginal distributions across domains by adding multiple adaptation layers and exploring multiple kernels, assuming that the conditional distributions remain unchanged (Fig. 6). However, this assumption is rather strong in practical applications; in other words, the source classifier cannot be directly used in the target domain. To make it more generalized, a joint adaptation network (JAN) [37] aligns the shift in the joint distributions of input features and output labels in multiple domain-specific layers based on a joint maximum mean discrepancy (JMMD) criterion (Fig. 6). Zhang et al. [33] proposed DTN, where both the marginal and the conditional distributions are matched based on MMD (Fig. 6). The shared feature extraction layer learns a subspace to match the marginal distributions of the source and the target samples, and the discrimination layer matches the conditional distributions by classifier transduction. In addition to adapting features using MMD, residual transfer networks (RTNs) [32] added a gated residual layer for classifier adaptation. More recently, Yan et al. [34] proposed a weighted MMD model that introduces an auxiliary weight for each class in the source domain when the class weights in the target domain are not the same as those in the source domain.

If ϕ is a characteristic kernel (i.e., Gaussian kernel or Laplace kernel), MMD will compare all the orders of statistic moments. In contrast to MMD, CORAL [77] learned a linear transformation that aligns the second-order statistics between domains. Sun and Saenko [41] extended CORAL to deep neural networks (deep CORAL) with a nonlinear transformation.

$$\mathcal{L}_{CORAL} = \frac{1}{4d^2}\|C_S - C_T\|_F^2 \tag{7}$$

where $\|\cdot\|_F^2$ denotes the squared matrix Frobenius norm. C_S and C_T denote the covariance matrices of the source and target data, respectively.

By the Taylor expansion of the Gaussian kernel, MMD can be viewed as minimizing the distance between the weighted sums of all raw moments [78]. The interpretation of MMD as moment matching procedures motivated Zellinger et al. [79] to match the higher-order moments of the domain distributions, which we call central moment discrepancy (CMD). An empirical estimate of the CMD metric for the domain discrepancy in the activation space $[a, b]^N$ is given by

$$\begin{aligned} CMD_K(X^s, X^t) = &\frac{1}{(b-a)}\left\|E(X^s) - E(X^t)\right\|_2 \\ &+ \sum_{k=2}^{K}\frac{1}{|b-a|^k}\left\|C_k(X^s) - C_k(X^t)\right\|_2 \end{aligned} \tag{8}$$

where $C_k(X) = E((x - E(X))^k$ is the vector of all kth-order sample central moments and $E(X) = \frac{1}{|X|}\sum_{x\in X} x$ is the empirical expectation.

The association loss $\mathcal{L}_{assoc}$ proposed by Haeusser [80] is an alternative discrepancy measure, it enforces statistical associations between source and target data by making the two-step round-trip probabilities P_{ij}^{aba} be similar to the uniform distribution over the class labels.

- **Architecture criterion**

Some other methods optimize the architecture of the network to minimize the distribution discrepancy. This adaptation behavior can be achieved in most deep DA models, such as supervised and unsupervised settings.

Rozantsev et al. [47] considered that the weights in corresponding layers are not shared but related by a weight regularizer $r_w(\cdot)$ to account for the differences between the two domains (Fig. 7). The weight regularizer $r_w(\cdot)$ can be expressed as the exponential loss function:

$$r_w(\theta_j^s, \theta_j^t) = \exp\left(\left\|\theta_j^s - \theta_j^t\right\|^2\right) - 1 \tag{9}$$

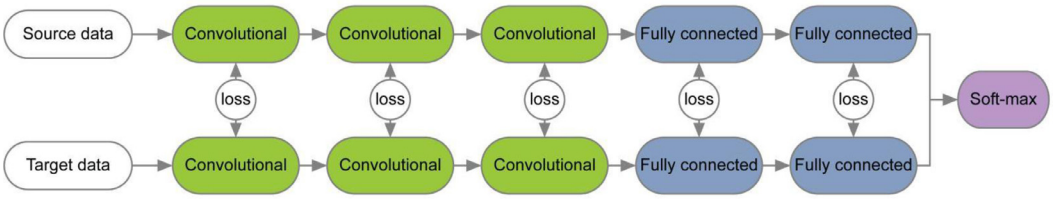


Fig. 7. The two-stream architecture with related weight [47].

where θ_j^s and θ_j^t denote the parameters of the *jth* layer of the source and target models, respectively. To further relax this restriction, they allow the weights in one stream to undergo a linear transformation:

$$r_w(\theta_j^s, \theta_j^t) = \exp\left(\left\|a_j\theta_j^s + b_j - \theta_j^t\right\|^2\right) - 1 \tag{10}$$

where a_j and b_j are scalar parameters that encode the linear transformation. The work of Shu et al. [81] is similar to [47] using weakly parameter-shared layers. The penalty term Ω controls the relatedness of parameters.

$$\Omega = \sum_{i=1}^{L}\left(\left\|W_S^{(l)} - W_T^{(l)}\right\|_F^2 + \left\|b_S^{(l)} - b_T^{(l)}\right\|_F^2\right) \tag{11}$$

where $\{W_S^{(l)}, b_S^{(l)}\}_{l=1}^L$ and $\{W_T^{(l)}, b_T^{(l)}\}_{l=1}^L$ are the parameters of the *lth* layer in the source and target domains, respectively.

Li et al. [44] hypothesized that the class-related knowledge is stored in the weight matrix, whereas domain-related knowledge is represented by the statistics of the batch normalization (BN) layer [82]. BN normalizes the mean and standard deviation for each individual feature channel such that each layer receives data from a similar distribution, irrespective of whether it comes from the source or the target domain. Therefore, Li et al. used BN to align the distribution for recomputing the mean and standard deviation in the target domain.

$$BN(X^t) = \lambda\left(\frac{x - \mu(X^t)}{\sigma(X^t)}\right) + \beta \tag{12}$$

where λ and β are parameters learned from the target data and $\mu(x)$ and $\sigma(x)$ are the mean and standard deviation computed independently for each feature channel. Based on [44], [83] endowed BN layers with a set of alignment parameters which can be learned automatically and can decide the degree of feature alignment required at different levels of the deep network. Furthermore, Ulyanov et al. [84] found that when replacing BN layers with instance normalization (IN) layers, where $\mu(x)$ and $\sigma(x)$ are computed independently for each channel and each sample, the performance of DA can be further improved.

Occasionally, neurons are not effective for all domains because of the presence of domain biases. For example, when recognizing people, the target domain typically contains one person centered with minimal background clutter, whereas the source dataset contains many people with more clutter. Thus, the neurons that capture the features of other people and clutter are useless. Domain-guided dropout was proposed by Xiao et al. [48] to solve the problem of multi-DA, and it mutes non-related neurons for each domain. Rather than assigning dropout with a specific dropout rate, it depends on the gain of the loss function of each neuron on the domain sample when the neuron is removed.

$$s_i = \mathcal{L}(g(x)_{\setminus i}) - \mathcal{L}(g(x)) \tag{13}$$

where $\mathcal{L}$ is the softmax loss function and $g(x)_{\setminus i}$ is the feature vector after setting the response of the *ith* neuron to zero. In [85], each source domain is assigned with different parameters, $\Theta^{(i)} = \Theta^{(0)} + \Delta^{(i)}$, where $\Theta^{(0)}$ is a domain general model, and $\Delta^{(i)}$ is a domain specific bias term. After the low rank parameterized CNNs are trained, $\Theta^{(0)}$ can serve as the classifier for target domain.

- **Geometric criterion**

The geometric criterion mitigates the domain shift by integrating intermediate subspaces on a geodesic path from the source to the target domains. A geodesic flow curve is constructed to connect the source and target domains on the Grassmannian. The source and target subspaces are points on a Grassmann manifold. By sampling a fixed [86] or infinite [87] number of subspaces along the geodesic, we can form the intermediate subspaces to help to find the correlations between domains. Then, both source and target data are projected to the obtained intermediate subspaces to align the distribution.

Inspired by the intermediate representations on the geodesic path, Chopra et al. [50] proposed a model called deep learning for DA by interpolating between domains (DLID). DLID generates intermediate datasets, starting with all the source data samples and gradually replacing source data with target data. Each dataset is a single point on an interpolating path between the source and target domains. Once intermediate datasets are generated, a deep nonlinear feature extractor using the predictive sparse decomposition is trained in an unsupervised manner.

4.1.2. Adversarial-based approaches

Recently, great success has been achieved by the GAN method [88], which estimates generative models via an adversarial process. GAN consists of two models: a generative model G that extracts the data distribution and a discriminative model D that distinguishes whether a sample is from G or training datasets by predicting a binary label. The networks are trained on the label prediction loss in a mini-max fashion: simultaneously optimizing G to minimize the loss while also training D to maximize the probability of assigning the correct label:

$$\begin{aligned}\min_G \max_D V(D,G) = {} & \mathbb{E}_{x\sim p_{data}(x)}[\log D(x)] \\ & + \mathbb{E}_{z\sim p_z(z)}[\log(1 - D(G(z)))]\end{aligned} \tag{14}$$

In DA, this principle has been employed to ensure that the network cannot distinguish between the source and target domains. Tzeng et al. [58] proposed a unified framework for adversarial-based approaches and summarized the existing approaches according to whether to use a generator, which loss function to employ, or whether to share weights across domains (Fig. 8). In this paper, we only categorize the adversarial-based approaches into two subsettings: generative models and non-generative models.

- **Generative models**

Synthetic target data with ground-truth annotations are an appealing alternative to address the problem of a lack of training data. First, with the help of source data, generators render unlimited quantities of synthetic target data, which are paired with synthetic source data to share labels or appear as if they were sampled from the target domain while maintaining labels, or something else. Then, synthetic data with labels are used to train

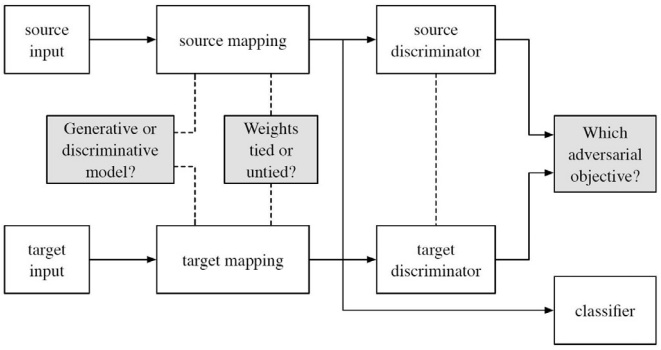


Fig. 8. Generalized architecture for adversarial domain adaptation. Existing adversarial adaptation methods can be viewed as instantiations of a framework with different choices regarding their properties [58].

the target model as if no DA were required. Adversarial-based approaches with generative models are able to learn such a transformation in an unsupervised manner based on GAN.

The core idea of CoGAN [51] is to generate synthetic target data that are paired with synthetic source ones (Fig. 9). It consists of a pair of GANs: GAN_1 for generating source data and GAN_2 for generating target data. The weights of the first few layers in the generative models and the last few layers in the discriminative models are tied. This weight-sharing constraint allows CoGAN to achieve a domain-invariant feature space without correspondence supervision. A trained CoGAN can adapt the input noise vector to paired images that are from the two distributions and share the labels. Therefore, the shared labels of synthetic target samples can be used to train the target model.

More work focuses on generating synthetic data that are similar to the target data while maintaining annotations. Yoo et al. [89] transferred knowledge from the source domain to pixel-level target images with GANs. A domain discriminator ensures the invariance of content to the source domain, and a real/fake discriminator supervises the generator to produce similar images to the target domain. Shrivastava et al. [90] developed a method for simulated+unsupervised (S+U) learning that uses a combined objective of minimizing an adversarial loss and a self-regularization loss, where the goal is to improve the realism of synthetic images using unlabeled real data. In contrast to other works in which the generator is conditioned only on a noise vector or source images, Bousmalis et al. [52] proposed a model that exploits GANs conditioned on both (Fig. 10). The classifier T is trained to predict class labels of both source and synthetic images, while the discriminator is trained to predict the domain labels of target and synthetic images. In addition, to expect synthetic images with similar foregrounds and different backgrounds from the same source images, a content similarity is used that penalizes large differences between source and synthetic images for foreground pixels only by a masked pairwise mean squared error [91]. The goal of the network is to learn G, D and T by solving the optimization problem:

$$\min_{G,T} \max_{D} V(D,G) = \alpha \mathcal{L}_d(D,G) + \beta \mathcal{L}_t(T,G) + \gamma \mathcal{L}_c(G) \tag{15}$$

where α, β, and γ are parameters that control the trade-off between the losses. $\mathcal{L}_d$, $\mathcal{L}_t$ and $\mathcal{L}_c$ are the adversarial loss, softmax loss and content-similarity loss, respectively.

- **Non-generative models**

The key of deep DA is learning domain-invariant representations from source and target samples. With these representations,

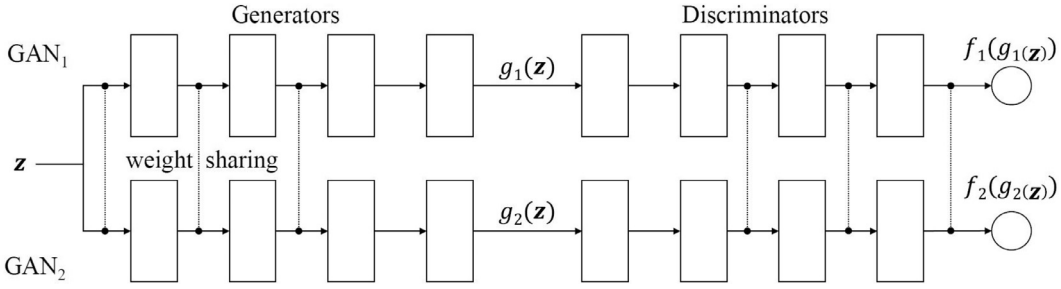


Fig. 9. The CoGAN architecture [51].

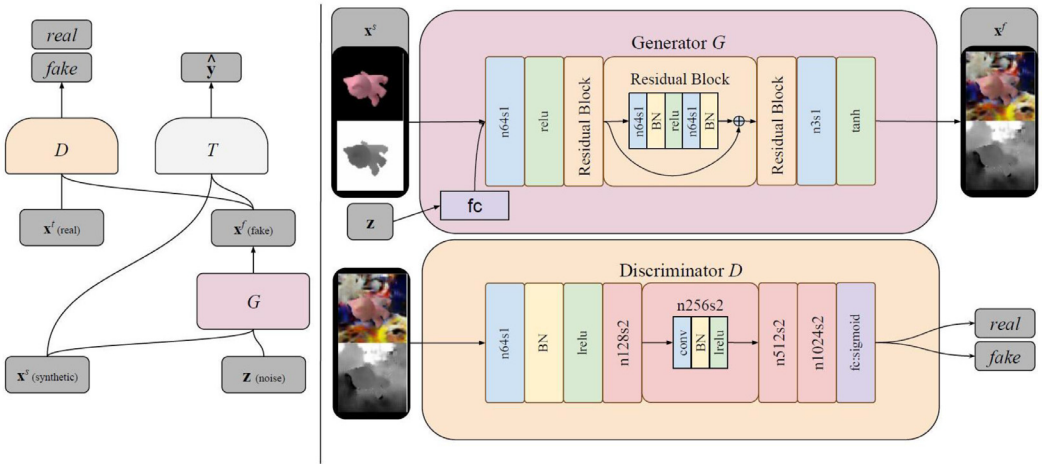


Fig. 10. The model that exploits GANs conditioned on noise vector and source images [52].

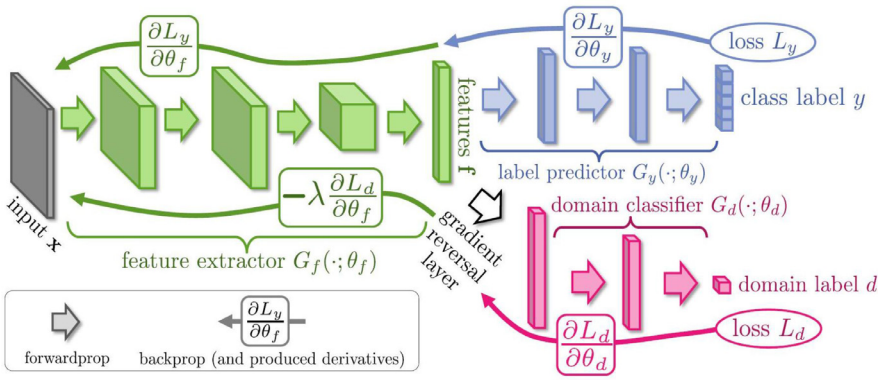


Fig. 11. The domain-adversarial neural network (DANN) architecture [55].

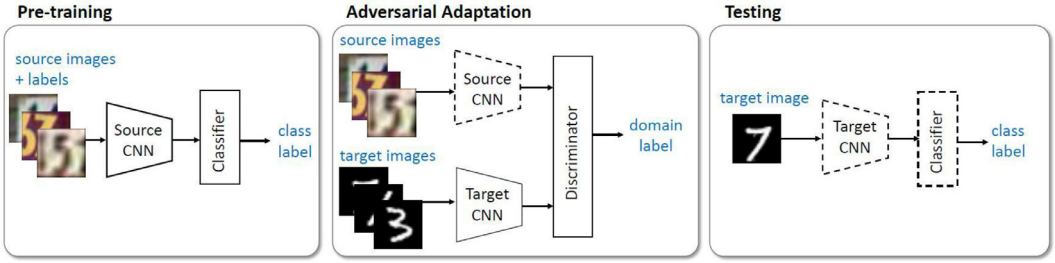


Fig. 12. The adversarial discriminativedomain adaptation (ADDA) architecture [58].

the distribution of both domains can be similar enough such that the classifier is fooled and can be directly used in the target domain even if it is trained on source samples. Therefore, whether the representations are domain-confused or not is crucial to transferring knowledge. Inspired by GAN, domain confusion loss, which is produced by the discriminator, is introduced to improve the performance of deep DA without generators.

The domain-adversarial neural network (DANN) [55] integrates a gradient reversal layer (GRL) into the standard architecture to ensure that the feature distributions over the two domains are made similar (Fig. 11). The network consists of shared feature extraction layers and two classifiers. DANN minimizes the domain confusion loss (for all samples) and label prediction loss (for source samples) while maximizing domain confusion loss via the use of the GRL. In contrast to the above methods, the adversarial discriminative domain adaptation (ADDA) [58] considers independent source and target mappings by untying the weights, and the parameters of the target model are initialized by the pre-trained source one (Fig. 12). This is more flexible because of allowing more domain-specific feature extractions to be learned. ADDA minimizes the source and target representation distances through iteratively minimizing these following functions, which is most similar to the original GAN:

$$\min_{M^s,C}\mathcal{L}_{cls}(X^s,Y^s) = -\mathbb{E}_{(x^s,y^s)\sim(X^s,Y^s)}\sum_{k=1}^{K}\mathbb{1}_{[k=y^s]}\log C(M^s(x^s))$$

$$\begin{aligned}\min_D \mathcal{L}_{advD}(X^s,X^t,M^s,M^t) = &-\mathbb{E}_{(x^s)\sim(X^s)}[\log D(M^s(x^s))]\\ &-\mathbb{E}_{(x^t)\sim(X^t)}[\log(1-D(M^t(x^t)))]\end{aligned}$$

$$\min_{M^s,M^t}\mathcal{L}_{advM}(M^s,M^t) = -\mathbb{E}_{(x^t)\sim(X^t)}[\log D(M^t(x^t))] \quad (16)$$

where the mappings M^s and M^t are learned from the source and target data, X^s and X^t. C represents a classifier working on the source domain. The first classification loss function $\mathcal{L}_{cls}$ is optimized by training the source model using the labeled source data. The second function $\mathcal{L}_{advD}$ is minimized to train the discriminator, while the third function $\mathcal{L}_{advM}$ is learning a representation that is domain invariant.

Tzeng et al. [26] proposed adding an additional domain classification layer that performs binary domain classification and designed a domain confusion loss to encourage its prediction to be as close as possible to a uniform distribution over binary labels. Unlike previous methods that match the entire source and target domains, Cao et al. introduced a selective adversarial network (SAN) [92] to address partial transfer learning from large domains to small domains, which assumes that the target label space is a subspace of the source label space. It simultaneously avoids negative transfer by filtering out outlier source classes, and it promotes positive transfer by matching the data distributions in the shared label space via splitting the domain discriminator into many class-wise domain discriminators. Motiian et al. [93] encoded domain labels and class labels to produce four groups of pairs, and replaced the typical binary adversarial discriminator by a four-class discriminator. Volpi et al. [94] trained a feature generator (S) to perform data augmentation in the source feature space and obtained a domain invariant feature through playing a minimax game against features from S.

Rather than using discriminator to classify domain label, some papers make some other explorations. Inspired by Wasserstein GAN [95], Shen et al. [96] utilized discriminator to estimate empirical Wasserstein distance between the source and target samples and optimized the feature extractor network to minimize the distance in an adversarial manner. In [97], two classifiers are treated as discriminators and are trained to maximize the discrepancy to detect target samples outside the support of the source, while a feature extractor is trained to minimize the discrepancy by generating target features near the support.

4.1.3. Reconstruction-based approaches

In DA, the data reconstruction of source or target samples is an auxiliary task that simultaneously focuses on creating a shared representation between the two domains and keeping the individual characteristics of each domain.

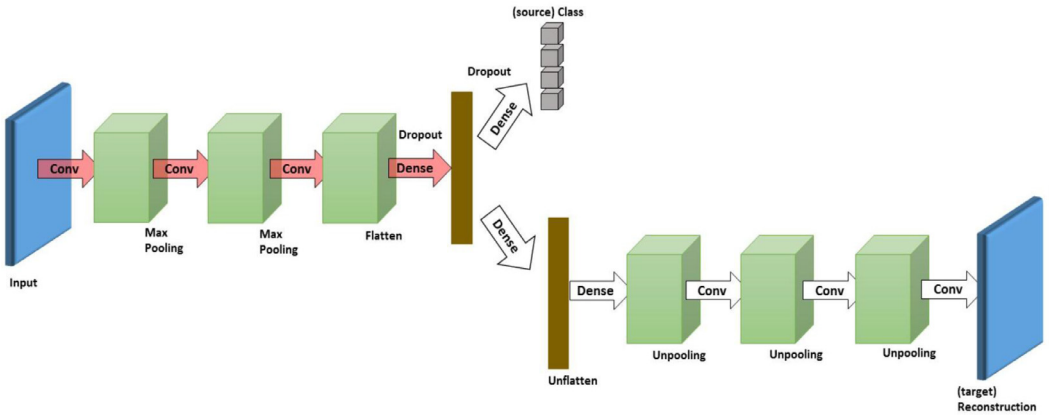


Fig. 13. The deep reconstruction classification network (DRCN) architecture [60].

- **Encoder–decoder reconstruction**

The basic autoencoder framework [98] is a feedforward neural network that includes the encoding and decoding processes. The autoencoder first encodes an input to some hidden representation, and then it decodes this hidden representation back to a reconstructed version. The DA approaches based on encoder–decoder reconstruction typically learn the domain-invariant representation by a shared encoder and maintain the domain-special representation by a reconstruction loss in the source and target domains.

Glorot et al. [99] proposed extracting a high-level representation based on stacked denoising autoencoders (SDA) [16]. By reconstructing the union of data from various domains with the same network, the high-level representations can represent both the source and target domain data. Thus, a linear classifier that is trained on the labeled data of the source domain can make predictions on the target domain data with these representations. Despite their remarkable results, SDAs are limited by their high computational cost and lack of scalability to high-dimensional features. To address these crucial limitations, Tsai and Chien [100] proposed the marginalized SDA (mSDA), which marginalizes noise with linear denoisers; thus, parameters can be computed in closed-form and do not require stochastic gradient descent.

The deep reconstruction classification network (DRCN) proposed in [60] learns a shared encoding representation that provides useful information for cross-domain object recognition (Fig. 13). DRCN is a CNN architecture that combines two pipelines with a shared encoder. After a representation is provided by the encoder, the first pipeline, which is a CNN, works for supervised classification with source labels, whereas the second pipeline, which is a deconvolutional network, optimizes for unsupervised reconstruction with target data.

$$\min \lambda \mathcal{L}_c(\{\theta_{enc}, \theta_{lab}\}) + (1-\lambda)\mathcal{L}_r(\{\theta_{enc}, \theta_{dec}\}) \tag{17}$$

where λ is a hyper-parameter that controls the trade-off between classification and reconstruction. θ_{enc}, θ_{dec} and θ_{lab} denote the parameters of the encoder, decoder and source classifier, respectively. $\mathcal{L}_c$ is cross-entropy loss for classification, and $\mathcal{L}_r$ is squared loss $\| x - f_r(x) \|_2^2$ for reconstruction in which $f_r(x)$ is the reconstruction of x.

Domain separation networks (DSNs) [59] explicitly and jointly model both private and shared components of the domain representations. A shared-weight encoder learns to capture shared representations, while a private encoder is used for domain-specific components in each domain. Additionally, a shared decoder learns to reconstruct the input samples by both the private and shared representations. Then, a classifier is trained on the shared representation. By partitioning the space in such a manner, the shared representations will not be influenced by domain-specific representations such that a better transfer ability can be obtained. Finding that the separation loss is simple and that the private features are only used for reconstruction in DSNs, [101] reinforced them by incorporating a hybrid adversarial learning in a separation network and an adaptation network.

Zhuang et al. [43] proposed transfer learning with deep autoencoders (TLDA), which consists of two encoding layers. The distance in distributions between domains is minimized with KL divergence in the embedding encoding layer, and label information of the source domain is encoded using a softmax loss in the label encoding layer. Ghifary et al. [61] extended the autoencoder into a model that jointly learns two types of data-reconstruction tasks taken from related domains: one is self-domain reconstruction, and the other is between-domain reconstruction.

- **Adversarial reconstruction**

Dual learning was first proposed by He et al. [102] to reduce the requirement of labeled data in natural language processing. Dual learning trains two "opposite" language translators, e.g., A to B and B to A. The two translators represent a primal-dual pair that evaluates how likely the translated sentences belong to the targeted language, and the closed loop measures the disparity between the reconstructed and the original ones. Inspired by dual learning, adversarial reconstruction is adopted in deep DA with the help of dual GANs.

Zhu et al. [63] proposed a cycle GAN that can translate the characteristics of one image domain into the other in the absence of any paired training examples (Fig. 14). Compared to dual learning, cycle GAN uses two generators rather than translators, which learn a mapping G: $X \rightarrow Y$ and an inverse mapping F: $Y \rightarrow X$. Two discriminators, D_X and D_Y, measure how realistic the generated image is ($G(X) \approx Y$ or $G(Y) \approx X$) by an adversarial loss and how well the original input is reconstructed after a sequence of two generations ($F(G(X)) \approx X$ or $G(F(Y)) \approx Y$) by a cycle consistency loss (reconstruction loss). Thus, the distribution of images from $G(X)$ (or $F(Y)$) is indistinguishable from the distribution Y (or X).

$$\begin{aligned}\mathcal{L}_{GAN}(G, D_Y, X, Y) &= \mathbb{E}_{y \sim p_{data}(y)}[\log D_Y(y)] \\ &\quad + \mathbb{E}_{x \sim p_{data}(x)}[\log(1 - D_Y(G(x)))] \\ \mathcal{L}_{cyc}(G, F) &= \mathbb{E}_{x \sim data(x)}[\|F(G(x)) - x\|_1] \\ &\quad + \mathbb{E}_{y \sim data(y)}[\|G(F(y)) - y\|_1]\end{aligned} \tag{18}$$

where $\mathcal{L}_{GAN}$ is the adversarial loss produced by discriminator D_Y with mapping function G: $X \rightarrow Y$. $\mathcal{L}_{cyc}$ is the reconstruction loss using L1 norm.

The dual GAN [62] and the disco GAN [64] were proposed at the same time, where the core idea is similar to cycle GAN. In dual

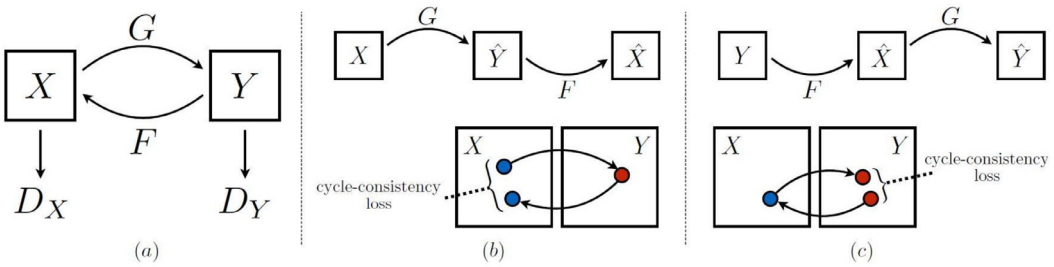


Fig. 14. The cycle GAN architecture [63].

GAN, the generator is configured with skip connections between mirrored downsampling and upsampling layers [53,103], making it a U-shaped net to share low-level information (e.g., object shapes, textures, clutter, and so forth). For discriminators, the Markovian patch-GAN [104] architecture is employed to capture local high-frequency information. In disco GAN, various forms of distance functions, such as mean-square error (MSE), cosine distance, and hinge loss, can be used as the reconstruction loss, and the network is applied to translate images, changing specified attributes including hair color, gender and orientation while maintaining all other components.

4.1.4. Hybrid approaches

To obtain better performance, some of the aforementioned methods have been used simultaneously. Tzeng et al. [26] combined a domain confusion loss and a soft label loss, while [32] used both statistic (MMD) and architecture criteria (adapt classifier by residual function) for unsupervised DA. Yan et al. [34] introduced class-specific auxiliary weights assigned by the pseudo-labels into the original MMD. In DSNs [59], encoder–decoder reconstruction approaches separate representations into private and shared representations, while the MMD criterion or domain confusion loss is helpful to make the shared representations similar and soft subspace orthogonality constraints ensure dissimilarity between the private and shared representations. Rozantsev et al. [47] used the MMD between the learned source and target representations and also allowed the weights of the corresponding layers to differ. Zhuang et al. [43] learned domain-invariant representations by encoder–decoder reconstruction approaches and the KL divergence.

4.2. Heterogeneous domain adaptation

In heterogeneous DA, the feature spaces of the source and target domains are not the same, $Xs \neq Xt$, and the dimensions of the feature spaces may also differ. According to the divergence of feature spaces, heterogeneous DA can be further divided into two scenarios. In one scenario, the source and target domain both contain images, and the divergence of feature spaces is mainly caused by different sensory devices (e.g., visual light (VIS) vs. near-infrared (NIR) or RGB vs. depth) and different styles of images (e.g., sketches vs. photos). In the other scenario, there are different types of media in source and target domain (e.g., text vs. image and language vs. image). Obviously, the cross-domain gap of the second scenario is much larger.

Most heterogeneous DA with shallow methods fall into two categories: symmetric transformation and asymmetric transformation. The symmetric transformation learns feature transformations to project the source and target features onto a common subspace. Heterogeneous feature augmentation (HFA) [105] first transformed the source and target data into a common subspace using projection matrices P and Q respectively, then proposed two new feature mapping functions, $\varphi_s(x^s) = \left[Px^s, x^s, 0_{d_t}\right]^T$ and $\varphi_t\left(x^t\right) = \left[Qx^t, 0_{d_s}, x^t\right]^T$, to augment the transformed data with their original features and zeros. These projection matrices are found using standard SVM with hinge loss in both the linear and nonlinear cases and an alternating optimization algorithm is proposed to simultaneously solve the dual SVM and to find the optimal transformations. Wang and Mahadevan [106] treated each input domain as a manifold which is represented by a Laplacian matrix, and used labels rather than correspondences to align the manifolds. The asymmetric transformation transforms one of source and target features to align with the other. Zhou et al. [107] proposed a sparse and class-invariant feature transformation matrix to map the weight vector of classifiers learned from the source domain to the target domain. The asymmetric regularized cross-domain transfer (ARC-t) [108] used asymmetric, non-linear transformations learned in Gaussian RBF kernel space to map the target data to the source domain. Extended from [109], ARC-t performed asymmetric transformation based on metric learning, and transfer knowledge between domains with different dimensions through changes of the regularizer. Since we focus on deep DA, we refer the interested readers to [20], which summarizes shallow approaches of heterogeneous DA.

However, as for deep methods, there is not much work focused on heterogeneous DA so far. The special and effective methods of heterogeneous deep DA have not been proposed, and heterogeneous deep DA is still performed similar to some approaches of homogeneous DA.

4.2.1. Discrepancy-based approach

In discrepancy-based approaches, the network generally shares or reuses the first n layers between the source and target domains, which limits the feature spaces of the input to the same dimension. However, in heterogeneous DA, the dimensions of the feature spaces of source domain may differ from those of target domain.

In first scenario of heterogeneous DA, the images in different domains can be directly resized into the same dimensions, so the Class Criterion and Statistic Criterion are still effective and are mainly used. For example, given an RGB image and its paired depth image, Gupta et al. [110] used the mid-level representation learned by CNNs as a supervisory signal to re-train a CNN on depth images. To transform an RGB object detector into a RGB-D detector without needing complete RGB-D data, Hoffman et al. [111] first trained an RGB network using labeled RGB data from all categories and finetuned the network with labeled depth data from partial categories, then combined mid-level RGB and depth representations at fc6 to incorporate both modalities into the final object class prediction. Mittal et al. [112] first trained the network using large face database of photos and then finetuned it using small database of composite sketches; Liu et al. [113] transferred the VIS deep networks to the NIR domain in the same way.

In second scenario, the features of different media can not be directly resized into the same dimensions. Therefore, discrepancy-based methods fail to work without extra process. Shu et al. [81] proposed weakly shared DTNs to transfer labeled information across heterogeneous domains, particularly from the text domain

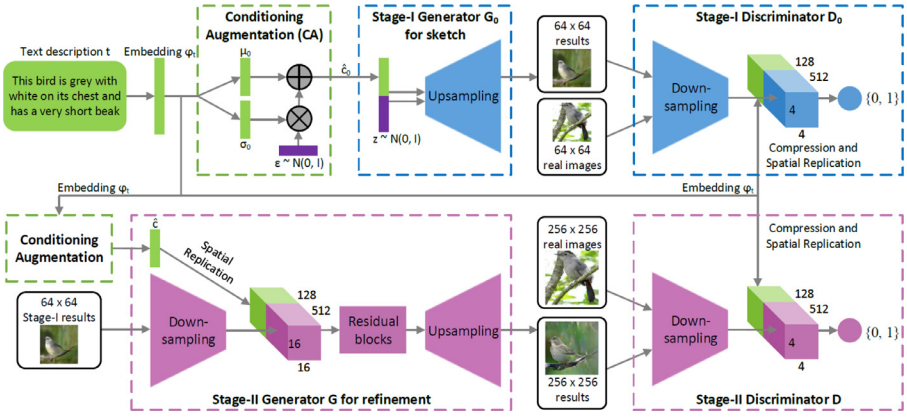


Fig. 15. The StackGAN architecture [119].

to the image domain. DTNs take paired data, such as text and image, as input to two SAEs, followed by weakly parameter-shared network layers at the top. Chen et al. [114] proposed transfer neural trees (TNTs), which consist of two stream networks to learn a domain-invariant feature representation for each modality. Then, a transfer neural decision forest (Transfer-NDF) [115,116] is used with stochastic pruning for adapting representative neurons in the prediction layer.

4.2.2. Adversarial-based approach

Using Generative Models can generate the heterogeneous target data while transferring some information of source domain to them. Taigman et al. [117] employed a compound loss function that consists of a multiclass GAN loss, a regularizing component and an f-constancy component to transfer unlabeled face photos to emoji images. To generate images for birds and flowers based on text, Reed et al. [118] trained a GAN conditioned on text features encoded by a hybrid character-level convolutional-recurrent neural network. Zhang et al. [119] proposed stacked generative adversarial networks (StackGAN) with conditioning augmentation for synthesizing photo-realistic images from text (Fig. 15). It decomposes the synthesis problem into several sketch-refinement processes. Stage-I GAN sketches the primitive shape and basic colors of the object to yield low-resolution image, and Stage-II GAN completes details of the object to produce a high-resolution photo-realistic image.

4.2.3. Reconstruction-based approach

The Adversarial Reconstruction can be used in heterogeneous DA as well. For example, the cycle GAN [63], dual GAN [62] and disco GAN [64] used two generators, G_A and G_B, to generate sketches from photos and photos from sketches, respectively. Based on cycle GAN [63], Wang et al. [120] proposed a multi-adversarial network to avoid artifacts of facial photo-sketch synthesis by leveraging the implicit presence of feature maps of different resolutions in the generator subnetwork.

5. Multi-step domain adaptation

For multi-step DA, the selection of the intermediate domain is problem specific, and different problems may have different strategies.

5.1. Hand-crafted approaches

Occasionally, the intermediate domain can be selected by experience, that is, it is decided in advance. For example, when the source domain is image data and the target domain is composed of text data, some annotated images will clearly be crawled as intermediate domain data.

With the common sense that nighttime light intensities can be used as a proxy for economic activity, Xie et al. [65] transferred knowledge from daytime satellite imagery to poverty prediction with the help of some nighttime light intensity information as an intermediate domain.

5.2. Instance-based approaches

In other problems where there are many candidate intermediate domains, some automatic selection criterion should be considered. Similar to the instance-transfer approaches proposed by Pan and Yang [18], because the samples of the source domain cannot be used directly, the mixture of certain parts of the source and target data can be useful for constructing the intermediate domain.

Tan et al. [25] proposed distant domain transfer learning (DDTL), where long-distance domains fail to transfer knowledge by only one intermediate domain but can be related via multiple intermediate domains. DDTL gradually selects unlabeled data from the intermediate domains by minimizing reconstruction errors on the selected instances in the source and intermediate domains and all the instances in the target domain simultaneously. With removal of the unrelated source data, the selected intermediate domains gradually become closer to the target domain from the source domain:

$$\begin{aligned}\mathcal{J}_1(f_e, f_d, v_S, v_T) = &\frac{1}{n_S}\sum_{i=1}^{n_S} v_S^i \left\|\hat{x}_S^i - x_S^i\right\|_2^2 \\ &+\frac{1}{n_I}\sum_{i=1}^{n_I} v_I^i \left\|\hat{x}_I^i - x_I^i\right\|_2^2 \\ &+\frac{1}{n_T}\sum_{i=1}^{n_T} \left\|\hat{x}_T^i - x_T^i\right\|_2^2 + R(v_S, v_T)\end{aligned} \tag{19}$$

where $\hat{x}_S^i$, $\hat{x}_T^i$ and $\hat{x}_I^i$ are reconstructions of source data S^i, target data T^i and intermediate data I^i based on the autoencoder, respectively, and f_e and f_d are the parameters of the encoder and decoder, respectively. $v_S = (v_S^1, \ldots, v_S^{n_S})^\top$ and $v_I = (v_I^1, \ldots, v_I^{n_I})^\top$, v_S^i, $v_I^i \in 0, 1$ are selection indicators for the *ith* source and intermediate instance, respectively. $R(v_S, v_T)$ is a regularization term that avoids all values of v_S and v_I being zero.

The DLID model [50] mentioned in Section 4.1.1 (Geometric Criterion) constructs the intermediate domains with a subset of the source and target domains, where source samples are gradually replaced by target samples.

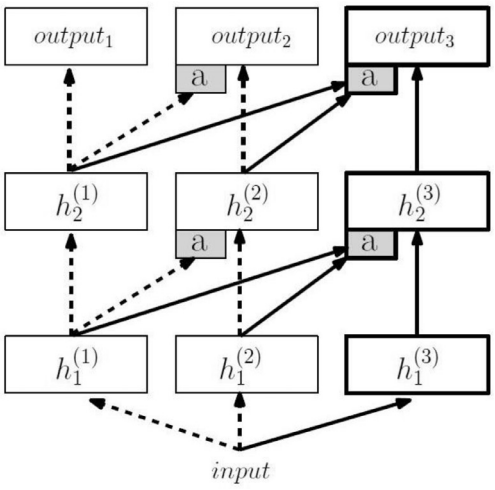


Fig. 16. The progressive network architecture [66].

5.3. Representation-based approaches

Representation-based approaches freeze the previously trained network and use their intermediate representations as input to the new network. Rusu et al. [66] introduced progressive networks that have the ability to accumulate and transfer knowledge to new domains over a sequence of experiences (Fig. 16). To avoid the target model losing its ability to solve the source domain, they constructed a new neural network for each domain, while transfer is enabled via lateral connections to features of previously learned networks. In the process, the parameters in the latest network are frozen to remember knowledge of intermediate domains.

6. Application of deep domain adaptation

Deep DA techniques have recently been successfully applied in many real-world applications, including image classification, object recognition, face recognition, object detection, style translation, and so forth. In this section, we present different application examples using various visual deep DA methods. Because the information of commonly used datasets for evaluating the performance is provided in [22] in detail, we do not introduce it in this paper.

6.1. Image classification

Because image classification is a basic task of computer vision applications, most of the algorithms mentioned above were originally proposed to solve such problems. Therefore, we do not discuss this application repeatedly, but we show how much benefit deep DA methods for image classification can bring. Because different papers often use different parameters, experimental protocols and tuning strategies in the preprocessing steps, it is quite difficult to perform a fair comparison among all the methods directly. Thus, similar to the work of Pan and Yang [18], we show the comparison results between the proposed deep DA methods and non-adaptation methods using only deep networks. A list of simple experiments taken from some published deep DA papers are presented in Table 5.

In [37], [79], and [26], the authors used the Office-31 dataset[1] as one of the evaluation data sets, as shown in Fig. 1(a). The Office dataset is a computer vision classification data set with images from three distinct domains: Amazon (A), DSLR (D), and Webcam (W). The largest domain, Amazon, has 2817 labeled images and its corresponding 31 classes, which consist of objects commonly encountered in office settings. By using this dataset, previous works can show the performance of methods across all six possible DA tasks. Long et al. [37] showed comparison experiments among the standard AlexNet [8], the DANN method [55], and the MMD algorithm and its variations, such as DDC [39], DAN [38], JAN [37] and RTN [32]. Zellinger et al. [79] evaluated their proposed CMD algorithm in comparison to other discrepancy-based methods (DDC, deep CROAL [41], DLID [50], AdaBN [44]) and the adversarial-based method DANN. Tzeng et al. [26] proposed an algorithm combining soft label loss and domain confusion loss, and they also compared them with DANN and DLID under a supervised DA setting.

In [58], MNIST[2](M), USPS[3](U), and SVHN[4] (S) digit datasets (shown in Fig. 1(b)) are used for a cross-domain hand-written digit recognition task, and the experiment showed the comparison results on some adversarial-based methods, such as DANN, CoGAN [51] and ADDA [58], where the baseline is VGG-16 [12].

6.2. Face recognition

The performance of face recognition significantly degrades when there are variations in the test images that are not present in the training images. The dataset shift can be caused by poses, resolution, illuminations, expressions, and modality. Kan et al. [121] proposed a bi-shifting auto-encoder network (BAE) for face recognition across view angle, ethnicity, and imaging sensor. In BAE, source domain samples are shifted to the target domain, and sparse reconstruction is used with several local neighbors from the target domain to ensure its correction, and vice versa. Single sample per person domain adaptation network (SSPP-DAN) in [122] generates synthetic images with varying poses to increase the number of samples in the source domain and bridges the gap between the synthetic and source domains by adversarial training with a GRL in real-world face recognition (Fig. 17). Sohn et al. [1] improved the performance of video face recognition by using an adversarial-based approach with large-scale unlabeled videos, labeled still images and synthesized images. Considering that age variations are difficult problems for smile detection and that networks trained on the current benchmarks do not perform well on young children, Xia et al. [123] applied DAN [38] and JAN [37] (mentioned in Section 4.1.1) to two baseline deep models, i.e., AlexNet and ResNet, to transfer the knowledge from adults to infants.

6.3. Object detection

Recent advances in object detection are driven by region-based convolutional neural networks (R-CNNs [10], fast R-CNNs [124] and faster R-CNNs [125]). They are composed of a window selection mechanism and classifiers that are pre-trained labeled bounding boxes by using the features extracted from CNNs. At test time, the classifier decides whether a region obtained by sliding windows contains the object. Although the R-CNN algorithm is effective, a large amount of bounding box labeled data is required to train each detection category. To solve the problem of lacking labeled data, considering the window selection mechanism as being domain independent, deep DA methods can be used in classifiers to adapt to the target domain.

[1] https://cs.stanford.edu/~jhoffman/domainadapt/.

[2] http://yann.lecun.com/exdb/mnist/.

[3] http://statweb.stanford.edu/~tibs/ElemStatLearn/data.html.

[4] http://ufldl.stanford.edu/housenumbers/.

Table 5
Comparison between transfer learning and non-adaptation learning methods.

Data set (reference)	Source vs. Target	Baselines	Deep domain adaptation methods				
		AlexNet	**DDC**	**DAN**	**RTN**	**JAN**	**DANN**
	A vs. W	61.6 ± 0.5	61.8 ± 0.4	68.5	73.3 ± 0.3	75.2 ± 0.4	73.0 ± 0.5
	D vs. W	95.4 ± 0.3	95.0 ± 0.5	96.0 ± 0.3	96.8 ± 0.2	96.6 ± 0.2	96.4 ± 0.3
Office-31 Dataset	W vs. D	99.0 ± 0.2	98.5 ± 0.4	99.0 ± 0.3	99.6 ± 0.1	99.6 ± 0.1	99.2 ± 0.3
ACC (unit:%) [37]	A vs. D	63.8 ± 0.5	64.4 ± 0.3	67.0 ± 0.4	71.0 ± 0.2	72.8 ± 0.3	72.3 ± 0.3
	D vs. A	51.1 ± 0.6	52.1 ± 0.6	54.0 ± 0.5	50.5 ± 0.3	57.5 ± 0.2	53.4 ± 0.4
	W vs. A	49.8 ± 0.4	52.2 ± 0.4	53.1 ± 0.5	51.0 ± 0.1	56.3 ± 0.2	51.2 ± 0.5
	Avg	70.1	70.6	72.9	73.7	76.3	74.3
		AlexNet	**Deep CORAL**	**CMD**	**DLID**	**AdaBN**	**DANN**
	A vs. W	61.6	66.4	77.0 ± 0.6	51.9	74.2	73
	D vs. W	95.4	95.7	96.3 ± 0.4	78.2	95.7	96.4
Office-31 Dataset	W vs. D	99.0	99.2	99.2 ± 0.2	89.9	99.8	99.2
ACC (unit:%) [79]	A vs. D	63.8	66.8	79.6 ± 0.6	–	73.1	–
	D vs. A	51.1	52.8	63.8 ± 0.7	–	59.8	–
	W vs. A	49.8	51.5	63.3 ± 0.6	–	57.4	–
	Avg	70.1	72.1	79.9	–	76.7	–
		AlexNet	**DLID**	**DANN**	**Soft Labels**	Domain Confusion	Confusion + Soft
	A vs. W	56.5 ± 0.3	51.9	53.6 ± 0.2	82.7 ± 0.7	82.8 ± 0.9	82.7 ± 0.8
	D vs. W	92.4 ± 0.3	78.2	71.2 ± 0.0	95.9 ± 0.6	95.6 ± 0.4	95.7 ± 0.5
Office-31 Dataset	W vs. D	93.6 ± 0.2	89.9	83.5 ± 0.0	98.3 ± 0.3	97.5 ± 0.2	97.6 ± 0.2
ACC (unit:%) [26]	A vs. D	64.6 ± 0.4	–	–	84.9 ± 1.2	85.9 ± 1.1	86.1 ± 1.2
	D vs. A	47.6 ± 0.1	–	–	66.0 ± 0.5	66.2 ± 0.4	66.2 ± 0.3
	W vs. A	42.7 ± 0.1	–	–	65.2 ± 0.6	64.9 ± 0.5	65.0 ± 0.5
	Avg	66.2	–	–	82.17	82.13	82.22
MNIST, USPS,		**VGG-16**	**DANN**	**CoGAN**	**ADDA**		
and SVHN	M vs. U	75.2 ± 1.6	77.1 ± 1.8	91.2 ± 0.8	89.4 ± 0.2		
digits datasets	U vs. M	57.1 ± 1.7	73.0 ± 2.0	89.1 ± 0.8	90.1 ± 0.8		
ACC (unit:%) [58]	S vs. M	60.1 ± 1.1	73.9	–	76.0 ± 1.8		

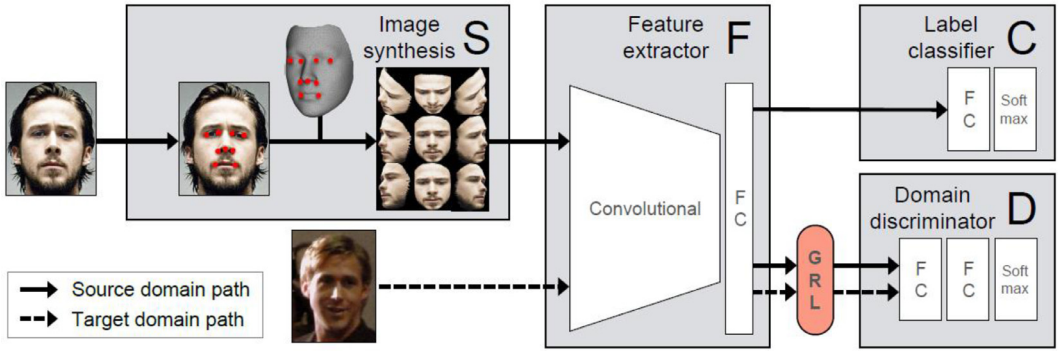


Fig. 17. The single sample per person domain adaptation network (SSPP-DAN) architecture [122].

Because R-CNNs train classifiers on regions just like classification, weak labeled data (such as image-level class labels) are directly useful for the detector. Most works learn the detector with limited bounding box labeled data and massive weak labeled data. The large-scale detection through adaptation (LSDA) [126] trains a classification layer for the target domain and then uses a pre-trained source model along with output layer adaptation techniques to update the target classification parameters directly. Rochan and Wang [127] used word vectors to establish the semantic relatedness between weak labeled source objects and target objects and then transferred the bounding box labeled information from source objects to target objects based on their relatedness. Extending [126] and [127], Tang et al. [128] transferred visual (based on the LSDA model) and semantic similarity (based on work vectors) for training an object detector on weak labeled category. Chen et al. [129] incorporated both an image-level and an instance-level adaptation component into faster R-CNN and minimized the domain discrepancy based on adversarial training. By using bounding box labeled data in a source domain and weak labeled data in a target domain, [130] progressively fine-tuned the pre-trained model with domain-transfer samples and pseudo-labeling samples.

6.4. Semantic segmentation

Fully convolutional network models (FCNs) for dense prediction have proven to be successful for evaluating semantic segmentation, but their performance will also degrade under domain shifts. Therefore, some work has also explored using weak labels to improve the performance of semantic segmentation. Hong et al. [131] used a novel encoder–decoder architecture with attention model by transferring weak class labeled knowledge in the source domain, while [132,133] transferred weak object location knowledge.

Much attention has also been paid to deep unsupervised DA in semantic segmentation. Hoffman et al. [134] first introduced it, in which global domain alignment is performed using FCNs with adversarial-based training, while transferring spatial layout is achieved by leveraging class-aware constrained multiple instance loss (Fig. 18). Zhang et al. [135] enhanced the segmentation performance on real images with the help of virtual ones. It uses the global label distribution loss of the images and local label distribution loss of the landmark superpixels in the target domain to effectively regularize the fine-tuning of the semantic segmenta-

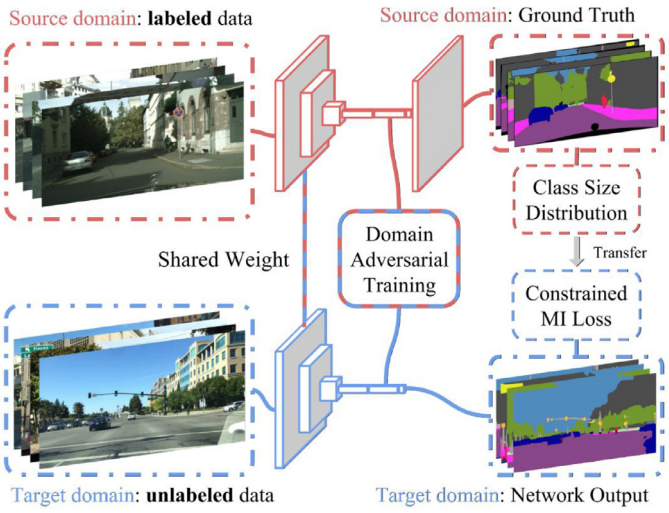


Fig. 18. The architecture of pixel-level adversarial and constraint-based adaptation [134].

tion network. Chen et al. [136] proposed a framework for cross-city semantic segmentation. The framework assigns pseudo labels to pixels/grids in the target domain and jointly utilizes global and class-wise alignment by domain adversarial learning to minimize domain shift. In [137], a target guided distillation module adapts the style from the real images by imitating the pre-trained source network, and a spatial-aware adaptation module leverages the intrinsic spatial structure to reduce the domain divergence. Rather than operating a simple adversarial objective on the feature space, [138] used a GAN to address domain shift in which a generator projects the features to the image space and a discriminator operates on this projected image space.

6.5. Image-to-image translation

Image-to-image translation has recently achieved great success with deep DA, and it has been applied to various tasks, such as style transferring. Specially, when the feature spaces of source and target images are not the same, image-to-image translation should be performed by heterogeneous DA.

More approaches of image-to-image translation use a dataset of paired images and incorporate a DA algorithm into generative networks. Isola et al. [53] proposed the pix2pix framework, which uses a conditional GAN to learn a mapping from source to target images. Tzeng et al. [56] utilized domain confusion loss and pair-wise loss to adapt from simulation to real-world data in a PR2 robot. However, several other methods also address the unpaired setting, such as CoGAN [51], cycle GAN [63], dual GAN [62] and disco GAN [64].

Matching the statistical distribution by fine-tuning a deep network is another way to achieve image-to-image translation. Gatys et al. [139] fine-tuned the CNN to achieve DA by the total loss, which is a linear combination between the content and the style loss, such that the target image is rendered in the style of the source image maintaining the content. The content loss minimizes the mean squared difference of the feature representation between the original image and generated image in higher layers, while the style loss minimizes the element-wise mean squared difference between the Gram matrix of them on each layer. [46] demonstrated that matching the Gram matrices of feature maps is equivalent to minimizing the MMD. Rather than MMD, Li et al. [42] proposed a deep generative correlation alignment network (DGCAN) that bridges the domain discrepancy between CAD synthetic and real images by applying the content and CORAL losses to different layers.

6.6. Person re-identification

In the community, person re-identification (re-ID) has become increasingly popular. When given video sequences of a person, person re-ID recognizes whether this person has been in another camera to compensate for the limitations of fixed devices. Recently, deep DA methods have been used in re-ID when models trained on one dataset are directly used on another. Xiao et al. [48] proposed the domain-guided dropout algorithm to discard useless neurons for re-identifying persons on multiple datasets simultaneously. Inspired by cycle GAN and Siamese network, the similarity preserving generative adversarial network (SPGAN) [140] translated the labeled source image to the target domain, preserving self similarity and domain-dissimilarity in an unsupervised manner, and then it trains re-ID models with the translated images using supervised feature learning methods.

6.7. Image captioning

Recently, image captioning, which automatically describes an image with a natural sentence, has been an emerging challenge in computer vision and natural language processing. Due to lacking of paired image-sentence training data, DA leverages different types of data in other source domains to tackle this challenge. Chen et al. [141] proposed a novel adversarial training procedure (captioner v.s. critics) for cross-domain image captioning using paired source data and unpaired target data. One captioner adapts the sentence style from source to target domain, whereas two critics, namely domain critic and multi-modal critic, aim at distinguishing them. Zhao et al. [142] fine-tuned the pre-trained source model on limited data in the target domain via a dual learning mechanism.

7. Conclusion

In a broad sense, deep DA is utilizing deep networks to enhance the performance of DA, such as shallow DA methods with features extracted by deep networks. In a narrow sense, deep DA is based on deep learning architectures designed for DA and optimized by back propagation. In this survey paper, we focus on this narrow definition, and we have reviewed deep DA techniques on visual categorization tasks.

Deep DA is classified as homogeneous DA and heterogeneous DA, and it can be further divided into supervised, semi-supervised and unsupervised settings. The first setting is the simplest but is generally limited due to the need for labeled data; thus, most previous works focused on unsupervised cases. Semi-supervised deep DA is a hybrid method that combines the methods of the supervised and unsupervised settings.

Furthermore, the approaches of deep DA can be classified into one-step DA and multi-step DA considering the distance of the source and target domains. When the distance is small, one-step DA can be used based on training loss. It consists of the discrepancy-based approach, the adversarial-based approach, and the reconstruction-based approach. When the source and target domains are not directly related, multi-step (or transitive) DA can be used. The key of multi-step DA is to select and utilize intermediate domains, thus falling into three categories, including hand-crafted, feature-based and representation-based selection mechanisms.

Although deep DA has achieved success recently, many issues still remain to be addressed. First, most existing algorithms focus on homogeneous deep DA, which assumes that the feature spaces between the source and target domains are the same. However,

this assumption may not be true in many applications. We expect to transfer knowledge without this severe limitation and take advantage of existing datasets to help with more tasks. Heterogeneous deep DA may attract increasingly more attention in the future.

In addition, deep DA techniques have been successfully applied in many real-world applications, including image classification, and style translation. We have also found that only a few papers address adaptation beyond classification and recognition, such as object detection, face recognition, semantic segmentation and person re-identification. How to achieve these tasks with no or a very limited amount of data is probably one of the main challenges that should be addressed by deep DA in the next few years.

Finally, since existing deep DA methods aim at aligning marginal distributions, they commonly assume shared label space across the source and target domains. However, in realistic scenario, the images of the source and target domain may be from the different set of categories or only a few categories of interest are shared. Recently, some papers [92,143,144] have begun to focus on this issue and we believe it is worthy of more attention.

Acknowledgments

This work was partially supported by the National Natural Science Foundation of China under Grant nos. 61573068, 61471048, and 61375031, and Beijing Nova Program under Grant no. Z161100004916088.

References

[1] K. Sohn, S. Liu, G. Zhong, X. Yu, M.-H. Yang, M. Chandraker, Unsupervised domain adaptation for face recognition in unlabeled videos, arxiv:1708.02191 (2018).
[2] L. Bruzzone, M. Marconcini, Domain adaptation problems: a DASVM classification technique and a circular validation strategy, IEEE Trans. Pattern Anal. Mach. Intell. 32 (5) (2010) 770–787.
[3] W.-S. Chu, F. De la Torre, J.F. Cohn, Selective transfer machine for personalized facial action unit detection, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2013, pp. 3515–3522.
[4] B. Gong, K. Grauman, F. Sha, Connecting the dots with landmarks: discriminatively learning domain-invariant features for unsupervised domain adaptation, in: Proceedings of the International Conference on Machine Learning, 2013, pp. 222–230.
[5] S.J. Pan, I.W. Tsang, J.T. Kwok, Q. Yang, Domain adaptation via transfer component analysis, IEEE Trans. Neural Netw. 22 (2) (2011) 199–210.
[6] M. Gheisari, M.S. Baghshah, Unsupervised domain adaptation via representation learning and adaptive classifier learning, Neurocomputing 165 (2015) 300–311.
[7] S. Pachori, A. Deshpande, S. Raman, Hashing in the zero shot framework with domain adaptation, Neurocomputing 275 (2018) 2137–2149.
[8] A. Krizhevsky, I. Sutskever, G.E. Hinton, Imagenet classification with deep convolutional neural networks, in: Proceedings of the Advances in Neural Information Processing Systems, 2012, pp. 1097–1105.
[9] Y. Taigman, M. Yang, M. Ranzato, L. Wolf, DeepFace: closing the gap to human-level performance in face verification, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2014, pp. 1701–1708.
[10] R. Girshick, J. Donahue, T. Darrell, J. Malik, Rich feature hierarchies for accurate object detection and semantic segmentation, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2014, pp. 580–587.
[11] W. Liu, Z. Wang, X. Liu, N. Zeng, Y. Liu, F.E. Alsaadi, A survey of deep neural network architectures and their applications, Neurocomputing 234 (2017) 11–26.
[12] K. Simonyan, A. Zisserman, Very deep convolutional networks for large-scale image recognition, arXiv:1409.1556 (2018).
[13] C. Szegedy, W. Liu, Y. Jia, P. Sermanet, S. Reed, D. Anguelov, D. Erhan, V. Vanhoucke, A. Rabinovich, Going deeper with convolutions, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2015, pp. 1–9.
[14] K. He, X. Zhang, S. Ren, J. Sun, Deep residual learning for image recognition, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2016, pp. 770–778.
[15] G.E. Hinton, S. Osindero, Y.-W. Teh, A fast learning algorithm for deep belief nets, Neural Comput. 18 (7) (2006) 1527–1554.
[16] P. Vincent, H. Larochelle, I. Lajoie, Y. Bengio, P.-A. Manzagol, Stacked denoising autoencoders: learning useful representations in a deep network with a local denoising criterion, J. Mach. Learn. Res. 11 (2010) 3371–3408.
[17] J. Donahue, Y. Jia, O. Vinyals, J. Hoffman, N. Zhang, E. Tzeng, T. Darrell, DeCAF: a deep convolutional activation feature for generic visual recognition, in: Proceedings of the International Conference on Machine Learning, 2014, pp. 647–655.
[18] S.J. Pan, Q. Yang, A survey on transfer learning, IEEE Trans. Knowl. Data Eng. 22 (10) (2010) 1345–1359.
[19] L. Shao, F. Zhu, X. Li, Transfer learning for visual categorization: a survey, IEEE Trans. Neural Netw. Learn. Syst. 26 (5) (2015) 1019–1034.
[20] O. Day, T.M. Khoshgoftaar, A survey on heterogeneous transfer learning, J. Big Data 4 (1) (2017) 29.
[21] V.M. Patel, R. Gopalan, R. Li, R. Chellappa, Visual domain adaptation: a survey of recent advances, IEEE Signal Process. Mag. 32 (3) (2015) 53–69.
[22] J. Zhang, W. Li, P. Ogunbona, Transfer Learning for Cross-dataset Recognition: A Survey, arXiv:1705.04396 (2017).
[23] G. Csurka, Domain adaptation for visual applications: a comprehensive survey, arXiv:1702.05374 (2018).
[24] B. Tan, Y. Song, E. Zhong, Q. Yang, Transitive transfer learning, in: Proceedings of the Twenty First ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, ACM, 2015, pp. 1155–1164.
[25] B. Tan, Y. Zhang, S.J. Pan, Q. Yang, Distant domain transfer learning, in: Proceedings of the AAAI, 2017, pp. 2604–2610.
[26] E. Tzeng, J. Hoffman, T. Darrell, K. Saenko, Simultaneous deep transfer across domains and tasks, in: Proceedings of the IEEE International Conference on Computer Vision, 2015, pp. 4068–4076.
[27] X. Peng, J. Hoffman, X.Y. Stella, K. Saenko, Fine-to-coarse knowledge transfer for low-res image classification, in: IEEE International Conference on Image Processing (ICIP), IEEE, 2016, pp. 3683–3687.
[28] S. Motiian, M. Piccirilli, D.A. Adjeroh, G. Doretto, Unified deep supervised domain adaptation and generalization, in: Proceedings of the IEEE International Conference on Computer Vision (ICCV), 2, 2017.
[29] K. Saito, Y. Ushiku, T. Harada, Asymmetric tri-training for unsupervised domain adaptation, arXiv:1702.08400 (2018).
[30] J. Hu, J. Lu, Y.-P. Tan, Deep transfer metric learning, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2015, pp. 325–333.
[31] G. Hinton, O. Vinyals, J. Dean, Distilling the knowledge in a neural network, arXiv:1503.02531 (2018).
[32] M. Long, H. Zhu, J. Wang, M.I. Jordan, Unsupervised domain adaptation with residual transfer networks, in: Proceedings of the Advances in Neural Information Processing Systems, 2016, pp. 136–144.
[33] X. Zhang, F.X. Yu, S.-F. Chang, S. Wang, Deep transfer network: unsupervised domain adaptation, arXiv:1503.00591 (2018).
[34] H. Yan, Y. Ding, P. Li, Q. Wang, Y. Xu, W. Zuo, Mind the class weight bias: weighted maximum mean discrepancy for unsupervised domain adaptation, arXiv:1705.00609 (2018).
[35] T. Gebru, J. Hoffman, L. Fei-Fei, Fine-grained recognition in the wild: A multi-task domain adaptation approach, in: 2017 IEEE International Conference on Computer Vision (ICCV), IEEE, 2017, pp. 1358–1367.
[36] W. Ge, Y. Yu., Borrowing treasures from the wealthy: Deep transfer learning through selective joint fine-tuning, in: IEEE Conference on Computer Vision and Pattern Recognition, 6, Honolulu, HI, 2017, pp. 10–19.
[37] M. Long, J. Wang, M.I. Jordan, Deep transfer learning with joint adaptation networks, arXiv:1605.06636 (2018).
[38] M. Long, Y. Cao, J. Wang, M. Jordan, Learning transferable features with deep adaptation networks, in: Proceedings of the International Conference on Machine Learning, 2015, pp. 97–105.
[39] E. Tzeng, J. Hoffman, N. Zhang, K. Saenko, T. Darrell, Deep domain confusion: maximizing for domain invariance, arXiv:1412.3474 (2018).
[40] M. Ghifary, W.B. Kleijn, M. Zhang, Domain adaptive neural networks for object recognition, in: Proceedings of the Pacific Rim International Conference on Artificial Intelligence, Springer, 2014, pp. 898–904.
[41] B. Sun, K. Saenko, Deep coral: correlation alignment for deep domain adaptation, in: Proceedings of the Workshops of Computer Vision–ECCV, Springer, 2016, pp. 443–450.
[42] X. Peng, K. Saenko, Synthetic to real adaptation with deep generative correlation alignment networks, arXiv:1701.05524 (2017).
[43] F. Zhuang, X. Cheng, P. Luo, S.J. Pan, Q. He, Supervised representation learning: transfer learning with deep autoencoders., in: Proceedings of the International Joint Conferences on Artificial Intelligence, IJCAI, 2015, pp. 4119–4125.
[44] Y. Li, N. Wang, J. Shi, J. Liu, X. Hou, Revisiting batch normalization for practical domain adaptation, arXiv:1603.04779 (2018).
[45] X. Huang, S. Belongie, Arbitrary Style Transfer in Real-Time with Adaptive Instance Normalization, in: 2017 IEEE International Conference on Computer Vision (ICCV), Venice, Italy, 2018, pp. 1510–1519.
[46] Y. Li, N. Wang, J. Liu, X. Hou, Demystifying neural style transfer, arXiv:1701.01036 (2018).
[47] A. Rozantsev, M. Salzmann and P. Fua. Beyond Sharing Weights for Deep Domain Adaptation. In IEEE Transactions on Pattern Analysis & Machine Intelligence. doi:10.1109/TPAMI.2018.2814042
[48] T. Xiao, H. Li, W. Ouyang, X. Wang, Learning deep feature representations with domain guided dropout for person re-identification, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2016, pp. 1249–1258.
[49] S.-A. Rebuffi, H. Bilen, A. Vedaldi, Learning multiple visual domains with residual adapters, in: Advances in Neural Information Processing Systems, 2017, pp. 506–516.

[50] S. Chopra, S. Balakrishnan, R. Gopalan, DLID: deep learning for domain adaptation by interpolating between domains, in: Proceedings of the ICML workshop on Challenges in Representation Learning, 2, 2013.

[51] M.-Y. Liu, O. Tuzel, Coupled generative adversarial networks, in: Proceedings of the Advances in Neural Information Processing Systems, 2016, pp. 469–477.

[52] K. Bousmalis, N. Silberman, D. Dohan, D. Erhan, D. Krishnan, Unsupervised pixel-level domain adaptation with generative adversarial networks, in: The IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 1, 2017, p. 7.

[53] P. Isola, J.-Y. Zhu, T. Zhou, A.A. Efros, Image-to-image translation with conditional adversarial networks, arXiv:1611.07004 (2018).

[54] Y. Ganin, E. Ustinova, H. Ajakan, P. Germain, H. Larochelle, F. Laviolette, M. Marchand, V. Lempitsky, Domain-adversarial training of neural networks, J. Mach. Learn. Res. 17 (59) (2016) 1–35.

[55] Y. Ganin, V. Lempitsky, Unsupervised domain adaptation by backpropagation, in: Proceedings of the International Conference on Machine Learning, 2015, pp. 1180–1189.

[56] E. Tzeng, C. Devin, J. Hoffman, C. Finn, P. Abbeel, S. Levine, K. Saenko, T. Darrell, Adapting deep visuomotor representations with weak pairwise constraints, CoRR, vol. abs/1511.07111 (2015).

[57] K.-C. Peng, Z. Wu, J. Ernst, Zero-shot deep domain adaptation, arXiv:1707.01922 (2017).

[58] E. Tzeng, J. Hoffman, K. Saenko, T. Darrell, Adversarial discriminative domain adaptation, in: IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 1, 2017, p. 4.

[59] K. Bousmalis, G. Trigeorgis, N. Silberman, D. Krishnan, D. Erhan, Domain separation networks, in: Proceedings of the Advances in Neural Information Processing Systems, 2016, pp. 343–351.

[60] M. Ghifary, W.B. Kleijn, M. Zhang, D. Balduzzi, W. Li, Deep reconstruction–classification networks for unsupervised domain adaptation, in: Proceedings of the European Conference on Computer Vision, Springer, 2016, pp. 597–613.

[61] M. Ghifary, W. Bastiaan Kleijn, M. Zhang, D. Balduzzi, Domain generalization for object recognition with multi-task autoencoders, in: Proceedings of the IEEE International Conference on Computer Vision, 2015, pp. 2551–2559.

[62] Z. Yi, H. Zhang, P.T. Gong, et al., DualGAN: unsupervised dual learning for image-to-image translation, arXiv:1704.02510 (2018).

[63] J.-Y. Zhu, T. Park, P. Isola, A.A. Efros, Unpaired image-to-image translation using cycle-consistent adversarial networks, arXiv:1703.10593 (2018).

[64] T. Kim, M. Cha, H. Kim, J. Lee, J. Kim, Learning to discover cross-domain relations with generative adversarial networks, arXiv:1703.05192 (2018).

[65] M. Xie, N. Jean, M. Burke, D. Lobell, S. Ermon, Transfer learning from deep features for remote sensing and poverty mapping, 1510.00098 (2015).

[66] A.A. Rusu, N.C. Rabinowitz, G. Desjardins, H. Soyer, J. Kirkpatrick, K. Kavukcuoglu, R. Pascanu, R. Hadsell, Progressive neural networks, arXiv:1606.04671 (2018).

[67] J. Hoffman, E. Tzeng, J. Donahue, Y. Jia, K. Saenko, T. Darrell, One-shot adaptation of supervised deep convolutional models, arXiv:1312.6204 (2018).

[68] A. Raj, V.P. Namboodiri, T. Tuytelaars, Subspace alignment based domain adaptation for rcnn detector, arXiv:1507.05578 (2018).

[69] H.V. Nguyen, H.T. Ho, V.M. Patel, R. Chellappa, DASH-N: joint hierarchical domain adaptation and feature learning, IEEE Trans. Image Process. 24 (12) (2015) 5479–5491.

[70] L. Zhang, Z. He, Y. Liu, Deep object recognition across domains based on adaptive extreme learning machine, Neurocomputing 239 (2017) 194–203.

[71] H. Lu, L. Zhang, Z. Cao, W. Wei, K. Xian, C. Shen, A. van den Hengel, When unsupervised domain adaptation meets tensor representations, in: Proceedings of the IEEE International Conference on Computer Vision, (ICCV), 2, 2017.

[72] J. Yosinski, J. Clune, Y. Bengio, H. Lipson, How transferable are features in deep neural networks? in: Proceedings of the Advances in Neural Information Processing Systems, 2014, pp. 3320–3328.

[73] B. Chu, V. Madhavan, O. Beijbom, J. Hoffman, T. Darrell, Best practices for fine-tuning visual classifiers to new domains, in: Proceedings of the Workshops of Computer Vision–ECCV, Springer, 2016, pp. 435–442.

[74] X. Wang, X. Duan, X. Bai, Deep sketch feature for cross-domain image retrieval, Neurocomputing 207 (2016) 387–397.

[75] C.H. Lampert, H. Nickisch, S. Harmeling, Learning to detect unseen object classes by between-class attribute transfer, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, CVPR, IEEE, 2009, pp. 951–958.

[76] K.M. Borgwardt, A. Gretton, M.J. Rasch, H.-P. Kriegel, B. Schölkopf, A.J. Smola, Integrating structured biological data by kernel maximum mean discrepancy, Bioinformatics 22 (14) (2006) e49–e57.

[77] B. Sun, J. Feng, K. Saenko, Return of frustratingly easy domain adaptation., in: Proceedings of the AAAI, 6, 2016, p. 8.

[78] Y. Li, K. Swersky, R. Zemel, Generative moment matching networks, in: Proceedings of the Thirty Second International Conference on Machine Learning (ICML), 2015, pp. 1718–1727.

[79] W. Zellinger, T. Grubinger, E. Lughofer, T. Natschläger, S. Saminger-Platz, Central moment discrepancy (CMD) for domain-invariant representation learning, 2017 arXiv:1702.08811.

[80] P. Haeusser, T. Frerix, A. Mordvintsev, D. Cremers, Associative domain adaptation, in: Proceedings of the International Conference on Computer Vision (ICCV), 2, 2017, p. 6.

[81] X. Shu, G.-J. Qi, J. Tang, J. Wang, Weakly-shared deep transfer networks for heterogeneous-domain knowledge propagation, in: Proceedings of the Twenty Third ACM International Conference on Multimedia, ACM, 2015, pp. 35–44.

[82] S. Ioffe, C. Szegedy, Batch normalization: accelerating deep network training by reducing internal covariate shift, in: Proceedings of the International Conference on Machine Learning, 2015, pp. 448–456.

[83] F.M. Carlucci, L. Porzi, B. Caputo, E. Ricci, S.R. Bulò, AutoDIAL: automatic domain alignment layers, in: Proceedings of the International Conference on Computer Vision, 2017.

[84] D. Ulyanov, A. Vedaldi, V. Lempitsky, Improved texture networks: Maximizing quality and diversity in feed-forward stylization and texture synthesis, IEEE Conference on Computer Vision and Pattern Recognition, 2017.

[85] D. Li, Y. Yang, Y.-Z. Song, T.M. Hospedales, Deeper, broader and artier domain generalization, in: Proceedings of the IEEE International Conference on Computer Vision (ICCV), IEEE, 2017, pp. 5543–5551.

[86] R. Gopalan, R. Li, R. Chellappa, Domain adaptation for object recognition: an unsupervised approach, in: Proceedings of the IEEE International Conference on Computer Vision (ICCV), IEEE, 2011, pp. 999–1006.

[87] B. Gong, Y. Shi, F. Sha, K. Grauman, Geodesic flow kernel for unsupervised domain adaptation, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), IEEE, 2012, pp. 2066–2073.

[88] I. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. Courville, Y. Bengio, Generative adversarial nets, in: Proceedings of the Advances in Neural Information Processing Systems, 2014, pp. 2672–2680.

[89] D. Yoo, N. Kim, S. Park, A.S. Paek, I.S. Kweon, Pixel-level domain transfer, in: Proceedings of the European Conference on Computer Vision, Springer, 2016, pp. 517–532.

[90] A. Shrivastava, T. Pfister, O. Tuzel, J. Susskind, W. Wang, R. Webb, Learning from simulated and unsupervised images through adversarial training, in: The IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 3, 2017, p. 6.

[91] D. Eigen, C. Puhrsch, R. Fergus, Depth map prediction from a single image using a multi-scale deep network, in: Proceedings of the Advances in Neural Information Processing Systems, 2014, pp. 2366–2374.

[92] Z. Cao, M. Long, J. Wang, M.I. Jordan, Partial transfer learning with selective adversarial networks, arXiv:1707.07901 (2017).

[93] S. Motiian, Q. Jones, S. Iranmanesh, G. Doretto, Few-shot adversarial domain adaptation, in: Proceedings of the Advances in Neural Information Processing Systems, 2017, pp. 6673–6683.

[94] R. Volpi, P. Morerio, S. Savarese, V. Murino, Adversarial feature augmentation for unsupervised domain adaptation, arXiv:1711.08561 (2018).

[95] M. Arjovsky, S. Chintala, L. Bottou, Wasserstein gan, arXiv:1701.07875 (2017).

[96] J. Shen, Y. Qu, W. Zhang, Y. Yu, Wasserstein distance guided representation learning for domain adaptation, arXiv:1707.01217 (2017).

[97] K. Saito, K. Watanabe, Y. Ushiku, T. Harada, Maximum classifier discrepancy for unsupervised domain adaptation, arXiv:1712.02560 (2018).

[98] Y. Bengio, Learning deep architectures for AI, Found. Trends Mach. Learn. 2 (1) (2009) 1–127.

[99] X. Glorot, A. Bordes, Y. Bengio, Domain adaptation for large-scale sentiment classification: a deep learning approach, in: Proceedings of the Twenty Eighth International Conference on Machine Learning (ICML), 2011, pp. 513–520.

[100] M. Chen, Z. Xu, K. Weinberger, F. Sha, Marginalized denoising autoencoders for domain adaptation, arXiv:1206.4683 (2018).

[101] J.-C. Tsai, J.-T. Chien, Adversarial domain separation and adaptation, in: Proceedings of the IEEE Twenty Seventh International Workshop on Machine Learning for Signal Processing (MLSP), IEEE, 2017, pp. 1–6.

[102] D. He, Y. Xia, T. Qin, L. Wang, N. Yu, T. Liu, W.-Y. Ma, Dual learning for machine translation, in: Proceedings of the Advances in Neural Information Processing Systems, 2016, pp. 820–828.

[103] O. Ronneberger, P. Fischer, T. Brox, U-Net: convolutional networks for biomedical image segmentation, in: Proceedings of the International Conference on Medical Image Computing and Computer-Assisted Intervention, Springer, 2015, pp. 234–241.

[104] C. Li, M. Wand, Precomputed real-time texture synthesis with Markovian generative adversarial networks, in: Proceedings of the European Conference on Computer Vision, Springer, 2016, pp. 702–716.

[105] L. Duan, D. Xu, I. Tsang, Learning with augmented features for heterogeneous domain adaptation, arXiv:1206.4660 (2018).

[106] C. Wang, S. Mahadevan, Heterogeneous domain adaptation using manifold alignment, in: Proceedings of the International Joint Conference on Artificial Intelligence, 22, 2011, p. 1541.

[107] J.T. Zhou, I.W. Tsang, S.J. Pan, M. Tan, Heterogeneous domain adaptation for multiple classes, in: Proceedings of the Artificial Intelligence and Statistics, 2014, pp. 1095–1103.

[108] B. Kulis, K. Saenko, T. Darrell, What you saw is not what you get: domain adaptation using asymmetric kernel transforms, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), IEEE, 2011, pp. 1785–1792.

[109] K. Saenko, B. Kulis, M. Fritz, T. Darrell, Adapting visual category models to new domains, in: Proceedings of the European Conference on Computer Vision, Springer, 2010, pp. 213–226.

[110] S. Gupta, J. Hoffman, J. Malik, Cross modal distillation for supervision transfer, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2016, pp. 2827–2836.

[111] J. Hoffman, S. Gupta, J. Leong, S. Guadarrama, T. Darrell, Cross-modal adaptation for RGB-D detection, in: Proceedings of the IEEE International Conference on Robotics and Automation, (ICRA), IEEE, 2016, pp. 5032–5039.

[112] P. Mittal, M. Vatsa, R. Singh, Composite sketch recognition via deep network-a transfer learning approach, in: Proceedings of the International Conference on Biometrics, (ICB), IEEE, 2015, pp. 251–256.
[113] X. Liu, L. Song, X. Wu, T. Tan, Transferring deep representation for NIR-VIS heterogeneous face recognition, in: Proceedings of the International Conference on Biometrics, (ICB), IEEE, 2016, pp. 1–8.
[114] W.-Y. Chen, T.-M. H. Hsu, Y.-H. H. Tsai, Y.-C. F. Wang, M.-S. Chen, Transfer neural trees for heterogeneous domain adaptation, in: Proceedings of the European Conference on Computer Vision, Springer, 2016, pp. 399–414.
[115] S. Rota Bulo, P. Kontschieder, Neural decision forests for semantic image labelling, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2014, pp. 81–88.
[116] P. Kontschieder, M. Fiterau, A. Criminisi, S. Rota Bulo, Deep neural decision forests, in: Proceedings of the IEEE International Conference on Computer Vision, 2015, pp. 1467–1475.
[117] Y. Taigman, A. Polyak, L. Wolf, Unsupervised cross-domain image generation, arXiv:1611.02200 (2018).
[118] S. Reed, Z. Akata, X. Yan, L. Logeswaran, B. Schiele, H. Lee, Generative adversarial text to image synthesis, arXiv:1605.05396 (2018).
[119] H. Zhang, T. Xu, H. Li, S. Zhang, X. Huang, X. Wang, D. Metaxas, StackGAN: text to photo-realistic image synthesis with stacked generative adversarial networks, in: Proceedings of the IEEE International Conference Computer Vision, (ICCV), 2017, pp. 5907–5915.
[120] L. Wang, V.A. Sindagi, V.M. Patel, High-quality facial photo-sketch synthesis using multi-adversarial networks, arXiv:1710.10182 (2018).
[121] M. Kan, S. Shan, X. Chen, Bi-shifting auto-encoder for unsupervised domain adaptation, in: Proceedings of the IEEE International Conference on Computer Vision, 2015, pp. 3846–3854.
[122] S. Hong, W. Im, J. Ryu, H.S. Yang, SSPP-DAN: deep domain adaptation network for face recognition with single sample per person 2018 arXiv:1702.04069.
[123] Y. Xia, D. Huang, Y. Wang, Detecting smiles of young children via deep transfer learning, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2017, pp. 1673–1681.
[124] R. Girshick, Fast R-CNN, in: Proceedings of the IEEE International Conference on Computer Vision, 2015, pp. 1440–1448.
[125] S. Ren, K. He, R. Girshick, J. Sun, Faster R-CNN: towards real-time object detection with region proposal networks, in: Proceedings of the Advances in Neural Information Processing Systems, 2015, pp. 91–99.
[126] J. Hoffman, S. Guadarrama, E.S. Tzeng, R. Hu, J. Donahue, R. Girshick, T. Darrell, K. Saenko, LSDA: large scale detection through adaptation, in: Proceedings of the Advances in Neural Information Processing Systems, 2014, pp. 3536–3544.
[127] M. Rochan, Y. Wang, Weakly supervised localization of novel objects using appearance transfer, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2015, pp. 4315–4324.
[128] Y. Tang, J. Wang, B. Gao, E. Dellandréa, R. Gaizauskas, L. Chen, Large scale semi-supervised object detection using visual and semantic knowledge transfer, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2016, pp. 2119–2128.
[129] Y. Chen, W. Li, C. Sakaridis, D. Dai, L. Van Gool, Domain adaptive faster R-CNN for object detection in the wild, arXiv:1803.03243 (2018).
[130] N. Inoue, R. Furuta, T. Yamasaki, K. Aizawa, Cross-domain weakly-supervised object detection through progressive domain adaptation, arXiv:1803.11365 (2018).
[131] S. Hong, J. Oh, H. Lee, B. Han, Learning transferrable knowledge for semantic segmentation with deep convolutional neural network, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2016, pp. 3204–3212.
[132] A. Kolesnikov, C.H. Lampert, Seed, expand and constrain: three principles for weakly-supervised image segmentation, in: Proceedings of the European Conference on Computer Vision, Springer, 2016, pp. 695–711.
[133] W. Shimoda, K. Yanai, Distinct class-specific saliency maps for weakly supervised semantic segmentation, in: Proceedings of the European Conference on Computer Vision, Springer, 2016, pp. 218–234.
[134] J. Hoffman, D. Wang, F. Yu, T. Darrell, FCNS in the wild: pixel-level adversarial and constraint-based adaptation, arXiv:1612.02649 (201).
[135] Y. Zhang, P. David, B. Gong, Curriculum domain adaptation for semantic segmentation of urban scenes, in: Proceedings of the IEEE International Conference on Computer Vision (ICCV), 2, 2017, p. 6.
[136] Y.-H. Chen, W.-Y. Chen, Y.-T. Chen, B.-C. Tsai, Y.-C.F. Wang, M. Sun, No more discrimination: Cross city adaptation of road scene segmenters, in: 2017 IEEE International Conference on Computer Vision (ICCV), IEEE, 2017, pp. 2011–2020.
[137] Y. Chen, W. Li, L. Van Gool, ROAD: reality oriented adaptation for semantic segmentation of urban scenes, arXiv:1711.11556 (2018b).
[138] S. Sankaranarayanan, Y. Balaji, A. Jain, S.N. Lim, R. Chellappa, Learning from synthetic data: addressing domain shift for semantic segmentation, arXiv:1711.06969 (2017).
[139] L.A. Gatys, A.S. Ecker, M. Bethge, Image style transfer using convolutional neural networks, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2016, pp. 2414–2423.
[140] W. Deng, L. Zheng, G. Kang, Y. Yang, Q. Ye, J. Jiao, Image-image domain adaptation with preserved self-similarity and domain-dissimilarity for person re-identification, arXiv:1711.07027 (2018).
[141] T.-H. Chen, Y.-H. Liao, C.-Y. Chuang, W.-T. Hsu, J. Fu, M. Sun, Show, adapt and tell: adversarial training of cross-domain image captioner, in: Proceedings of the IEEE International Conference on Computer Vision, (ICCV), 2, 2017.
[142] W. Zhao, W. Xu, M. Yang, J. Ye, Z. Zhao, Y. Feng, Y. Qiao, Dual learning for cross-domain image captioning, in: Proceedings of the Conference on Information and Knowledge Management, ACM, 2017, pp. 29–38.
[143] P.P. Busto, J. Gall, Open set domain adaptation, in: Proceedings of the IEEE International Conference on Computer Vision, (ICCV), 1, 2017, p. 3.
[144] J. Zhang, Z. Ding, W. Li, P. Ogunbona, Importance weighted adversarial nets for partial domain adaptation, arXiv:1803.09210 (2018).

Mei Wang received the B.E. degree in information and communication engineering from the Dalian University of Technology (DUT), Dalian, China, in 2013 and received M.E. degree in communication engineering from the Beijing University of Posts and Telecommunications (BUPT), Beijing, China, in 2016. From September 2018, she is a Ph.D. student in school of information and communication engineering of BUPT. Her research interests include pattern recognition and computer vision, with a particular emphasis in deep face recognition and transfer learning.

Weihong Deng received the B.E. degree in information engineering and the Ph.D. degree in signal and information processing from the Beijing University of Posts and Telecommunications (BUPT), Beijing, China, in 2004 and 2009, respectively. From Oct. 2007 to Dec. 2008, he was a postgraduate exchange student in the School of Information Technologies, University of Sydney, Australia. He is currently an professor in School of Information and Telecommunications Engineering, BUPT. His research interests include statistical pattern recognition and computer vision, with a particular emphasis in face recognition. He has published over 100 technical papers in international journals and conferences, such as IEEE TPAMI and CVPR. He serves as associate editor for IEEE Access, and guest editor for Image and Vision Computing Journal and the reviewer for dozens of international journals, such as IEEE TPAMI / TIP / TIFS / TNNLS / TMM / TSMC, IJCV, PR / PRL. His Dissertation titled “Highly accurate face recognition algorithms” was awarded the Outstanding Doctoral Dissertation Award by Beijing Municipal Commission of Education in 2011. He has been supported by the program for New Century Excellent Talents by the Ministry of Education of China in 2013 and Beijing Nova Program in 2016.