IEEE Access
Multidisciplinary | Rapid Review | Open Access Journal

SPECIAL SECTION ON DEEP LEARNING: SECURITY AND FORENSICS RESEARCH ADVANCES AND CHALLENGES

Received July 4, 2019, accepted July 25, 2019, date of publication August 6, 2019, date of current version August 19, 2019.

*Digital Object Identifier 10.1109/ACCESS.2019.2933591*

# Domain Adaptation by Stacked Local Constraint Auto-Encoder Learning

**XISHUAI PENG[1], (Member, IEEE), YUANXIANG LI[1], (Member, IEEE), YI LU MURPHEY[2], (Fellow, IEEE), AND JIANHUA LUO[1]**

[1]School of Aeronautics and Astronautics, Shanghai Jiao Tong University, Shanghai 200240, China
[2]College of Engineering and Computer Science, University of Michigan–Dearborn, Dearborn, MI 48128, USA

Corresponding author: Yuanxiang Li (yuanxli@sjtu.edu.cn)

This work was supported in part by the National Natural Science Fund of China under Grant U1406404, and in part by the Civil Aviation Special Project of China under Grant MJZ-2016-S-44.

**ABSTRACT** Domain adaptation (DA), a particular case of transfer learning, is an effective technology for learning a discriminative model in scenarios where the data from the training (source) and the testing (target) domains share common class labels but follow different distributions. The differences between domains, called domain shifts, are caused by variations in the acquisition devices and environmental conditions, such as changing illuminations, pose, and collecting-device noises, that are related to a specific domain, denoted as domain-specific noises in this paper. The research on stacked denoising autoencoder (SDA) has demonstrated that noise-robust features could be learned through training a model to reduce the man-made (simulated) noises. However, little research has been conducted to learn the domain-invariant features through training SDA to reduce the domain-specific noises from the real word. In this paper, we propose a novel variant of SDA for DA, called the stacked local constraint auto-encoder (SLC–AE), which aims to learn domain-invariant features through iteratively optimizing the SDA and the low-dimensional manifold. The core idea behind the SLC–AE is that both the source and target samples are corrupted due to the domain-specific noises, and each corrupted sample could be de-noised by calculating the weighted sum of its neighbor samples defined on the intrinsic manifold. Because the neighbor samples on the intrinsic manifold are semantically similar, their weighted sum preserves the generic information and reduces the domain-specific noises. To properly evaluate the performance of the SLC–AE, we conducted extensive experiments using seven benchmark data sets, *i.e., MNIST, USPS, COIL20, SYN SIGNS, GTSRB, MSRC* and *VOC 2007*. Compared to twelve different state-of-the-art methods, the experimental results demonstrated that the proposed SLC–AE model made significant improvement over the performance of SDA and achieved the best average performance on the seven data sets.

**INDEX TERMS** Computer vision, machine learning, image recognition, feature extraction.

## I. INTRODUCTION

Supervised learning with deep architectures has made remarkable contributions to machine learning and computer vision, leading to the development of robust algorithms that are applicable to a broad range of application problems. However, properly learning the parameters of a deep architecture usually requires a large number of labeled data, which is quite expensive and time-consuming. Moreover, in real-world applications, the training and the testing data usually follow different distributions or underlying structures. In such scenarios, the performance of the conventional machine learning models is significantly decreased on the testing data, even though the labeled training data are large. For instance, in the case of handwritten digit recognition, the support vector machine (SVM) model, a conventional machine learning model, trained using the training data from the USPS data set, can achieve about 88.7% accuracy on the testing data from the same data set [1]. However, it can achieve only about 33.2% accuracy on the testing data from the MNIST data set [2], which shares the 10 same classes of digits but follows different feature distributions

The associate editor coordinating the review of this manuscript and approving it for publication was Kim-Kwang Raymond Choo.

from the USPS data set. This problem of the conventional machine learning models also has been demonstrated in other real-world applications such as *indoor WiFi localization* [3], *text categorization* [4] *and video event recognition* [5].

Domain adaptation (DA), a particular case of transfer learning, has proven to be an effective technology for learning a discriminative model in scenarios where the training (source) and the testing (target) domains share the same task, *i.e., class labels remain the same across domains* but follow different data distributions. Although some special types of DA problems have been studied under different names such as covariate shift [6], class imbalance [7], and sample selection bias [8], [9], it began by taking advantage of deep architectures to learn domain-invariant features and has gained significant interest very recently in computer vision. The approaches proposed in the literature to explore deep architectures for DA can be grouped into three main categories. The first group of methods considers the CNN models as feature extractors, and then the extracted deep features are used by the conventional shallow DA methods [10]–[13]. The second group of methods first trains a deep network on the source domain, and then fine-tunes or adjusts it using the target domain data [14]–[17]. The third group of methods, which aims to design new models for DA based on the traditional deep learning architectures, could be considered the most promising. This paper is most related to the third type of methods, with emphasis on designing a novel DA model based on a stacked denoising autoencoder (SDA).

SDA, a traditional deep model, aims to learn noise-robust features through denoising the data corrupted by man-made noises, *such as Gaussian noise or binary masking noise* [18], [19]. From the first SDA-based DA method [20] to the very recent variants, *such as discriminative SDA* [21], *low-rank-weight SDA* [22], and *adversarial collaborative auto-encoder* [23], the collected data are all assumed to be 'clean'. From the perspective of DA, the differences between various domains, *such as the variations in illuminations, pose, and collecting-device noises*, are caused by the different data acquisition devices and environmental conditions in different application domains, which could be considered as the domain-specific noises. In other words, the collected data in the source and target domains are not 'clean', since they have been corrupted by the domain-specific noises. The domain-specific noises are from the real world and are much more complicated than the man-made or adversarial noises; thus, training a SDA to reduce the real-world noises instead of the simulated noises makes the SDA more effective in solving real-world problems [4], [24]. However, the supervised training procedure of SDA requires data pairs, each data pair contains the 'clean' sample and its corresponding 'corrupted' version. With the assumption that the originally collected data are 'clean', the 'corrupted' sample can be generated simply by adding simulated noises to the 'clean' sample, whereas under the assumption that the originally collected data are 'corrupted', how to generate the 'clean' data from the real-world 'corrupted' version is much more challenging and still an opening problem [4], [24].

The classical noise reduction methods remove the real-world noises by imposing a smooth constraint on the local data structure, *i.e. the changes among neighbor samples are continuous*; thus, a corrupted sample may be purified by calculating the average of its neighbor samples [25], [26]. Compared to the problem of finding the neighbor samples in spatial or temporal space, finding neighbor samples in domain space is a non-trivial problem. It is well-known that an object could have significantly different appearances in the images of different domains due to the domain-specific noises, *e.g. illumination, occlusion, etc*. Manifold learning (ML) is an effective technology for learning the intrinsic structure of data, which is invariant in different feature spaces [24], [27]. In the scenarios of DA, recent research has demonstrated that dynamically optimizing the manifold learning procedure in the low-dimensional space has superior capability for learning meaningful local structure, where the neighbor samples are semantically similar [28].

In this paper, we take advantage of recent DA research on ML and SDA to propose a novel DA architecture, named the stacked local constraint auto-encoder (SLC–AE), which aims to optimize SDA and low-dimensional manifolds using an iterative procedure. The core idea behind the SLC–AE is that a 'clean' sample can be estimated from the 'corrupted' input by calculating the weighted sum of the neighbor samples defined on the dynamic manifold. Because the neighbor samples on the low-dimensional manifold are semantically similar, their weighted sum contains the generic information and is invariant to domain-specific noises. The contributions of this paper can be summarized as follows:

1) A novel deep DA framework, SLC–AE, is proposed to learn the domain-invariant features by iteratively optimizing the SDA and the low-dimensional manifold.
2) A novel training scheme is proposed for SDA with the assumption that the domain data have been corrupted by the domain-specific noises. It trains a SDA to be less sensitive to the domain-specific noises from the real-world, which improves the model's robustness and transfer-ability.
3) A novel low-dimensional manifold learning method is proposed that employs discrimination and locality constraints, aiming to minimize the distance of semantically similar samples.

The remainder of this paper is organized as follows. We introduce the related works in Section II, address the problem formulation and the detailed description of the proposed approach in Section III, report the experimental evaluations in Section IV and conclude the paper with more discussions in Section V.

## II. RELATED WORKS

In this section, we briefly review the research on deep DA architectures, SDA and ML, according to the three main

contributions of this paper, and emphasize the differences between the previous research and the proposed framework, SLC–AE.

### A. DEEP DOMAIN ADAPTATION ARCHITECTURES

Based on the tasks in the source and the target domains, supervised transfer learning may be categorized into two subsettings, *i.e., inductive transfer learning and transductive transfer learning* [29]. In the inductive transfer learning setting, the target task is different from the source task, no matter whether the source and target domains are the same or not. In the transductive transfer learning setting, the source and target tasks are the same, while the source and target domain are different. As a specific case of transductive transfer learning, DA assumes that the feature spaces between domains are the same but the marginal probability distributions of data are different.

With the recent progress in computer vision due to deep architectures, research endeavoring to take advantage of deep models for DA started gaining interest. On the one hand, recent research has demonstrated that the deep features contain the feature hierarchies, *i.e., features from higher levels of the hierarchy are formed by the composition of lower level features*. Analogous to the functional modules that can be reused in different systems in computer programming, the low-level deep features can be easily re-purposed to novel tasks in computer vision, even when the new tasks differ significantly from the task originally used to train the model [10], [30]. On the other hand, DA provides deep architectures with a solution to relieve the quantity and quality requirements of the labeled target data for model training, *i.e. borrowing the knowledge from the related domains*. The complementary relationship between deep architectures and DA makes deep DA one of the most promising research topics.

Continuing along the DA research on shallow models, the most fruitful deep DA solution is to minimize the data distributions between domains through maximum mean discrepancy (MMD). For instance, deep domain confusion (DDC) [31] was proposed to extract the source and the target features using two separate deep models, one layer of each model was selected, and their discrepancy was minimized on MMD. Instead of using a single layer and linear MMD, the deep adaptation network (DAN) [32] was proposed to minimize the domain discrepancy represented by the sum of multi-kernel MMD on several layers of deep models. These works considered only the discrepancy in marginal distributions between domains; joint adaptation networks (JAN) [33] improved these works by jointly minimizing MMD on both marginal and conditional distributions between domains. Inspired by the research on the generative adversarial network (GAN) [34], domain-adversarial neural networks (DANN) [35] was proposed to use an adversarial objective with respect to a domain discriminator instead of MMD as the distance measurement of domains. These methods learn the domain-invariant features through minimizing the distance of the features that are extracted from one or multiple activation layers of deep models on a specific distance metric. In contrast, the SLC–AE is a data reconstruction-based method that minimizes the difference between the source and the target features through the data reconstruction of the semantically similar samples using a layer-wise training scheme.

### B. STACKED DENOISING AUTOENCODER

The research most related to the SLC–AE is the deep DA architectures based on SDA, which is one of the most successful variants of the stacked auto-encoder (SAE) [4], [24], [36]. In SAE, the compressed (dimension reduced) features are learned for the input through minimizing the reconstruction error. Different from SAE, in SDA, the inputs of SAE are first corrupted with man-made noises, and then the model is used to reconstruct the 'clean' data from the 'corrupted' data for learning noise-robust features. The shared model weights and explicit reconstruction loss function make SDA an effective deep model for DA to capture the common subspace between domains.

The first SDA-based DA method was proposed to adapt sentiment classification between reviews of different products [20]. The authors in [18] investigated the performance of SDA for DA in computer vision applications. First, the SDA was trained to reconstruct the inputs on the union of the source and target data, and then a classifier, i.e., a linear SVM, was trained on the resulting features. The experimental results demonstrated that using the features extracted by SDA achieved significantly better performance than directly using the raw data. The noise formulations used in these SDA-based methods, such as Gaussian noise or binary masking noise, are simple. Although the number of training iterations could be increased to provide more complex noises, it is computationally-intensive and time-consuming. The work presented in [19] reduced the computation cost and improved the performance of SDA by providing a close-form solution for equivalently training SDA on a more complicated noise, *i.e., an infinitely large number of binary noises*. Taking advantage of the research on adversarial training, the authors in [23] proposed a SDA variant based on adversarial noises, which were dynamically learned during the training procedure to improve the models' denoising capability. These studies all assumed that the originally collected training data were 'clean' and demonstrated that the robustness of the system relies on the formulations of noises, *i.e., the richer the noisy patterns are, the better the performance will be* [18], [19].

In this paper, the originally collected data in different domains are assumed to be 'corrupted', due to the domain-specific noises. Compared to the previous research, the SLC–AE is a model robust to real-world noises, *i.e. domain-specific noises*. Instead of simulating the real-world noises to corrupt the 'clean' inputs, the core problem in SLC–AE is how to estimate the 'clean' data from the 'corrupted' inputs for the supervised training of the SDA.

Low-rank constraint has been employed in SDA for noise reduction in the recent literature. The work in [22] proposed regularizing the encoding and decoding weights of a SDA using a low-rank penalty in the form of nuclear norm. The work in [4] proposed a deep low-rank coding (DLRC), which aims to learn discriminative low-rank coding in the guidance of an iterative supervised structure term. The authors in [24] proposed a deep robust encoder (DRE) to learn a low-rank dictionary for 'clean' data generation. The work in [27] applied the DRE to extract features from noisy EGG data. Different from these works, in the SLC–AE, the 'clean' samples are estimated from the corresponding 'corrupted' input through calculating the weight sum of neighbor samples, which are defined on a dynamically learned manifold.

### C. MANIFOLD LEARNING

Based on the assumption that the high-dimensional data are embedded in the low-dimensional manifold, ML methods aim to learn the intrinsic structure that is invariant in different feature spaces. Instead of restricting the distribution discrepancy between domains to be minimized in the low-dimensional space, ML-based DA methods aim to minimize the distance of the data manifolds between domains. For instance, the statistically invariant sample selection method [37] uses the Hellinger distance on the statistical manifold instead of MMD as the distance measurement. The authors in [38] used the geometric structure to assist a model to select the sub-spaces, so the shared features across domains could be discovered. The linear transformation used in these works may fall short of capturing the non-linear structure in the real-world data. Therefore, the authors in [39] proposed a deep non-linear architecture, denoted as bi-shift auto-encoder (BAE), which was used to minimize the distance between the source and target manifolds through reconstruction. The authors in [40] proposed a non-linear kernel based approach, denoted as geodesic flow kernel (GFK), to characterize the domain shift by integrating an infinite number of sub-spaces.

However, the similarity matrix used in these methods is fixed and defined on the corrupted high-dimensional data, which may not be valid in DA scenarios, *i.e. the geodesic nearest neighbor on a manifold may not be the Euclidian nearest neighbor in the high-dimensional space*. For example, two images from different domains that are similar in the low-dimensional manifold (geodesic nearest neighbor on a manifold), may not necessarily be similar in the high-dimensional space (Euclidian nearest neighbor in the high-dimensional space). This is because the types of domain-specific noises could be significantly different between the source and target domains, and some of them, *e.g., illumination, occlusion etc.*, might change the pixel intensities of the entire image. In such scenarios, the similarity matrix defined in the corrupted high-dimensional space might mislead the learning of the low-dimensional manifold. In order to address this problem, ML methods have been developed to iteratively learn the similarity matrix in the low-dimensional space and verify the matrix through reconstruction [28], called generalized auto-encoder (GAE). Experiments have demonstrated that the GAE is capable of achieving a meaningful manifold and promising performance for DA. Manifold learning in the case of DA is the final purpose of the GAE, whereas in the SLC–AE, it is an auxiliary or intermediate task to estimate the 'clean' data for the supervised training of SDA. To the best of the authors' knowledge, little research has been conducted on exploring a dynamic low-dimensional manifold to generate the 'clean' data, and using them to train the SDA model to learn robust and domain-invariant features.

**TABLE 1. Notations and their descriptions used in this paper.**

| | |
|---|---|
| $\mathcal{D}_s$ ,$\mathcal{D}_t$ | source/target domain |
| $P(x_s)$, $P(x_t)$ | marginal distribution of the source/target data |
| $P(y_s\|x_s)$, $P(y_s\|x_t)$ | conditional distribution of the source/target data |
| $\mathcal{X}_s$,$\mathcal{X}_t$ | the high-dimensional source/target data spaces |
| $\mathbf{x}_s$,$\mathbf{x}_t$ | the high-dimensional source/target samples |
| $\mathcal{Y}_s$, $\mathcal{Y}_t$ | the source/target label sets |
| $\mathcal{Z}_s$,$\mathcal{Z}_t$ | the source/target low-dimensional spaces |
| $\mathbf{z}_s$,$\mathbf{z}_t$ | the source/target low-dimensional features |
| $n_s$,$n_t$ | the number of source/target samples |

## III. STACKED LOCAL CONSTRAINT AUTO-ENCODER

In this section, we present the detailed architecture and the training procedure of the proposed SLC–AE model.

### A. PROBLEM DEFINITION

We begin with the definitions of terminologies. For clarity, the frequently used notations are summarized in Tab. 1.

*Definition 1 (Domain):* A domain $\mathcal{D}$ is composed of a feature space $\mathcal{X}$ and a marginal probability distribution $P(\mathbf{x})$, *i.e.,* $\mathcal{D} = \{\mathcal{X}, P(\mathbf{x})\}$, *where* $\mathbf{x} \in \mathcal{X}$.

*Definition 2 (Task):* Given $\mathcal{D}$, a task $\mathcal{T}$ is composed of a $C$-cardinality label set $\mathcal{Y}$ and a classifier $f(\mathbf{x})$, *i.e.* $\mathcal{T} = \{\mathcal{Y}, f(\mathbf{x})\}$, where $y \in \mathcal{Y}$, and $f(\mathbf{x}) = P(y|x)$ from the probabilistic viewpoint.

*Problem:* Given a labeled source domain $\mathcal{D}_s = \{(x_s^1, y_s^1), \ldots, (x_s^{n_s}, y_s^{n_s})\}$ and an unlabeled target domain $\mathcal{D}_t = \{x_t^1, \ldots, x_t^{n_t}\}$ under $\mathcal{X}_s = \mathcal{X}_t, \mathcal{Y}_s = \mathcal{Y}_t, P_s(x_s) \neq P_t(x_t)$ and $P_s(y_s|x_s) \neq P_t(y_t|x_t)$. In this paper, we aim to learn the domain-invariant features through training a SDA to reduce the domain-specific noises. Specifically, the encoder learns to map the 'corrupted' input from both the source and the target domains into a low-dimensional feature space $\mathcal{Z}$, where the low-dimensional features $z = \{z_s \cup z_t\}$ can be mapped by the decoder into the original data space for reconstructing the 'clean' version of the input, *i.e., the input removed the domain-specific noises*.

### B. SLC–AE ARCHITECTURE

The SAE architecture consists of two components, *namely, the encoder and the decoder*. The encoder maps the input onto the feature space, and then the decoder recovers the input using the encoded features. It may seem only an

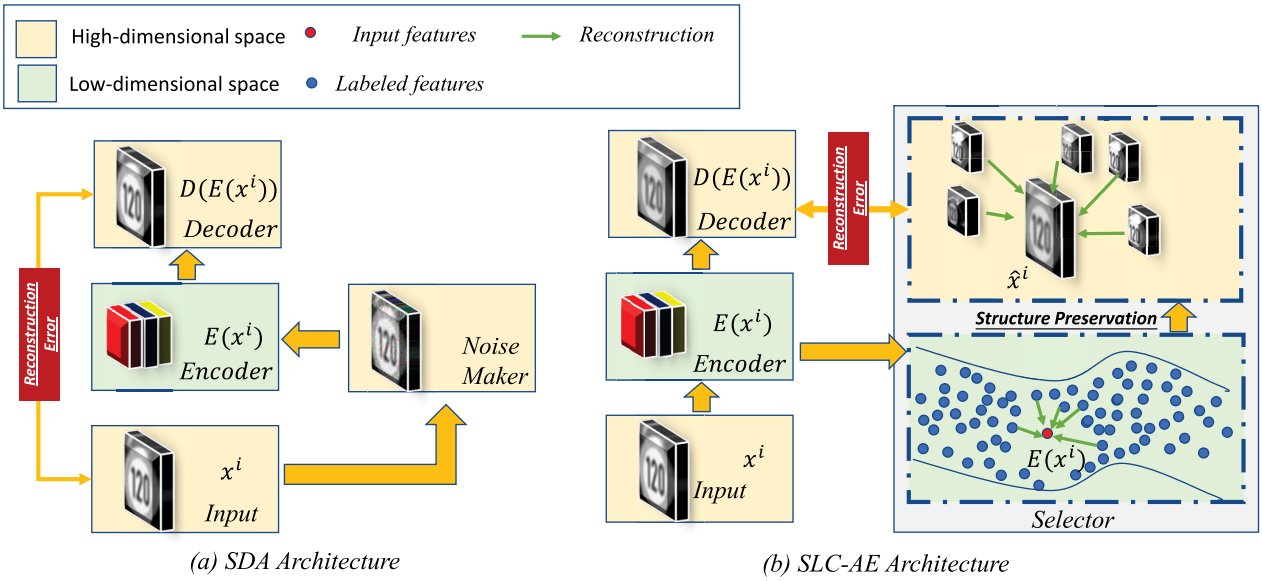


**FIGURE 1.** **(a) SDA consists of three components, *i.e. the encoder, decoder and noise-maker.* In SDA, the input data are first intentionally corrupted by the noise-maker. Then the encoder maps the corrupted data into the low-dimensional feature space. Last, the decoder reconstructs the input data by minimizing the reconstruction error. (b) SLC–AE consists of three components, *i.e. the encoder, decoder and selector.* Since the input data are assumed to be corrupted, they are first directly mapped into low-dimensional feature space by the encoder. Then the selector is used to collect semantically similar neighbor samples to construct the clean data. Last, the decoder is used to reconstruct the clean data.**

approximation of an identity operator, but it can be used to learn interesting features of data by constraining the number of hidden nodes.

Compared to SAE, there is an additional component in SDA, the noise-maker, which is used to corrupt the inputs with various noises as illustrated in Fig 1(a). Since the SDA is trained to reduce the noises for reconstructing the 'clean' data in the training procedure, the usage of the noise-maker improves the robustness of SDA against various man-made noises. The loss function of SDA can be formulated as follows,

$$E_{sda}(W_e, b_e, W_d, b_d) = \sum_i ||D(E(G(\mathbf{x}^i))) - \mathbf{x}^i||_2 \quad (1)$$

where $W_e$ and $b_e$ are the parameters of the encoder $E$; $W_d$ and $b_d$ the parameters of the decoder $D$; $G$, the noise-maker, is used to add man-made noises to corrupt the inputs.

Using the noise-maker in SDA is based on the assumption that the originally collected data are 'clean', and thus the research on the conventional SDA aims to design an advanced noise-maker that is capable of simulating the real-world noises. Different from these works, in the SLC–AE, both the source and the target data are assumed to be 'corrupted', considering the domain-specific noises that are produced in the collection of realistic data. Therefore, rather than endeavoring to simulate the real-world noises, the core technique of the SLC–AE is how to rule out the domain-specific noises to generate the 'clean' data. As shown in Fig 1(b), there are three major components in the proposed SLC–AE architecture, *namely, the encoder, selector and decoder*, where the selector is the core component used to address the problem of 'clean' data generation. Since the selector is an additional component used only in the training procedure, once the model training procedure is completed, the SLC–AE requires no external time and memory space for testing. Moreover, analogous to SDA, the three major components of the SLC–AE may be simply stacked into a deep model.

The training procedure for the SLC–AE is described as follows. First, the inputs are directly mapped into a low-dimensional feature space by the encoder. Then the selector collects the neighbor samples for the inputs in the low-dimensional space and uses these neighbor samples to generate the 'clean' data in the high-dimensional space. Last, the decoder is used to reconstruct the high-dimensional 'clean' data by optimizing the following equation,

$$E(W_e, b_e, W'_d, b'_d) = \sum_i ||D(E(\mathbf{x}^i)) - \hat{\mathbf{x}}^i||_2 \quad (2)$$

where $\hat{\mathbf{x}}^i$, the 'clean' version of the 'corrupted' input $\mathbf{x}^i$, is generated by the selector, which will be introduced in the next subsection. We want to emphasize the differences in the objective functions between the SLC–AE and SDA, which clearly demonstrate the different assumptions held in the two approaches. First, the noise-maker $G$ in Eq. 1 is removed in Eq. 2, since the domain-specific noises from the real world are considered the major contamination sources in the SLC–AE. Second, the 'clean' sample used by the decoder for reconstruction in Eq. 2 is $\hat{\mathbf{x}}^i$, which is generated by the selector, whereas in Eq. 1 the 'clean' sample is the input itself $\mathbf{x}^i$, since the SDA assumes the inputs to be 'clean'.

### C. THE SELECTOR COMPONENT

As a core component of the SLC–AE, the selector is used to reduce the domain-specific noises by calculating the weighted sum of the neighbor samples, which are defined on a dynamically learned low-dimensional manifold. Specifically, the selector is first used to explore the manifold using the locality and discrimination constraints. Then in the high-dimensional space, it generates the 'clean' data using the weighted sum of the neighbor samples.

#### 1) LOCALITY CONSTRAINED AFFINITY MATRIX COMPUTATION

Local manifold learning is mainly characterized by constructing an affinity matrix, which indicates the affinities (or similarities) of vertex pairs in a graph. The matrix computation procedure is composed of two steps: neighbor selection and computation of affinity weights.

In the conventional ML methods, Euclidean distance between a pair of high-dimensional samples is commonly used for the affinity measurement. However, this will cause problems when there are domain shifts between the training and the testing data [41]. In this paper, the affinity between a pair of samples is measured by the reconstruction coefficients of their low-dimensional features. The affinity matrix can be obtained by optimizing the following equation,

$$\mathcal{L}_M = \min_M \sum_{i=1}^{n_s+n_t} [(\mathbf{z}^i - \hat{\mathbf{z}}^i)^2 + \mu \textstyle\sum_{j=1}^{n_s} |m_i^j| (\mathbf{z}^i - \mathbf{z}_s^j)^p] \quad (3)$$

where $\mathbf{z}^i$ is the output feature vector of the encoder using $\mathbf{x}^i$ as input; $\hat{\mathbf{z}}^i$ the reconstructed $\mathbf{z}^i$ using the labeled source data $\{\mathbf{z}_s^1, \ldots, \mathbf{z}_s^{n_s}\}$.

$$\hat{\mathbf{z}}^i = \sum_{j=1}^{n_s} m_i^j \mathbf{z}_s^j \quad (4)$$

where $m_i^j$, an element of the affinity matrix $M \in \mathbb{R}^{(n_s+n_t)\times n_s}$, denotes the coefficient of $z_s^j$ for reconstructing $z^i$. We set $m_i^j = 0$ if $i = j$ to avoid the trivial solution, *i.e. each feature vector is reconstructed by itself*.

It should be noted that there are two constraints in Eq. 3, a reconstruction error and a regularization term. The regularization term constrains the reconstruction coefficients to satisfy some geometric properties when minimizing the reconstruction error. For example, when $p = 0$, the regularization term constrains the reconstruction coefficients via $L1$ norm, which encourages the affinity matrix to be sparse [42]. The sparse constraint is used to encourage the number of the labeled samples used to reconstruct each $\mathbf{z}^i$ to be as small as possible. However, it is unable to ensure that the labeled samples used to reconstruct $\mathbf{z}^i$ have small distance to $\mathbf{z}^i$ as well. In this paper, the reconstruction coefficients are used to indicate the affinities of data pairs, thus the feature vectors used to reconstruct each other should be neighbors, *i.e., being close in distance*. Motivated by the work in [43], [44], we set $p = 2$, which encourages the optimization process to reconstruct each feature vectors using its neighbor samples. For instance, in order to minimize the total loss, a small reconstruction coefficient $m_i^j$ will be assigned for the feature vector of a labeled sample $\mathbf{z}_s^j$, if $(\mathbf{z}^i - \mathbf{z}_s^j)^2$ is large.

#### 2) DISCRIMINATION CONSTRAINED AFFINITY MATRIX REFINE

Once the affinity matrix is obtained, it will be refined by the selector using the additional discriminative information. The discriminative information used in the selector contains the ground-truth of the source data and the pseudo class labels of the target data. The ground-truth is manually annotated before the model training for the source data, and the pseudo class labels are generated by label propagation algorithms for the unlabeled target data during the procedure of model training. Label propagation algorithms are the techniques commonly used to assign a temporal class label for the unlabeled sample according to the relationship between the labeled and unlabeled samples. For instance, in the DA methods aiming to align the conditional (class-wise) distributions between domains, the conditional distribution of the unlabeled target data is generally estimated on the pseudo class labels assigned by a SVM trained on the source data [45]. In this paper, an unlabeled target sample could be assigned a pseudo class label through finding the nearest labeled source sample in the low-dimensional manifold,, *i.e., using the nearest neighbor classifier in the low-dimensional space*. This is because the low-dimensional manifold is assumed to be the intrinsic structure of data, where the neighbor samples are semantically similar.

In order to speed up the computation, for each sample, the 'reconstruction' set is constrained to have $k$ ($k << n_s$) nearest neighbor samples that are semantically similar, *i.e., sharing the same class label*. The detailed construction procedure of the 'reconstruction' set can be summarized as follows,

- For an unlabeled target sample, the pseudo label should first be assigned using the nearest neighbor classifier in the low-dimensional space. Then a set of the k-nearest neighbor samples having the same class label are selected as the 'reconstruction' set.
- For a labeled source sample, since the ground truth has been provided, a set of the k-nearest neighbor samples having the same class label are directly selected as the 'reconstruction' set.

Once the 'reconstruction' set is obtained, the affinity matrix can be refined as follows,

$$\widetilde{m}_i^j = \begin{cases} \dfrac{m_i^j}{\sum_{j\in\Omega_i} m_i^j}, & j \in \Omega_i \\ 0, & j \notin \Omega_i \end{cases} \quad (5)$$

where $\Omega_i$ is the 'reconstruction' set of a sample $\mathbf{x}^i$, and $\widetilde{m}_i^j$ is the refined affinity between the $i$th sample and the $j$th sample.

#### 3) AFFINITY MATRIX BASED CLEAN DATA CONSTRUCTION

The affinity matrix explored in the low-dimensional space represents the intrinsic structure of data where the neighbor samples defined on it are semantically similar. We use the weighted sum of the neighbor samples as the 'clean' data with the assumption that the domain-specific noises can be reduced by averaging them in the neighbor samples.

$$\hat{\mathbf{x}}^i = \sum_{j \in \Omega_i} \widetilde{m}_i^j \mathbf{x}^j \tag{6}$$

where $\widetilde{m}_i^j$ is the coefficient of $\mathbf{x}^j$ used to reconstruct the clean data $\hat{\mathbf{x}}^i$ in the high-dimensional space. It should be noted that, $\widetilde{m}_i^j$ also denotes the coefficient of $\mathbf{z}^j$ used to reconstruct $\hat{\mathbf{z}}^i$ in the low-dimensional feature space. In this way, constructing the 'clean' data using Eq. 6 approximately preserves the low-dimensional manifold in the high-dimensional space.

### D. THE TRAINING PROCEDURE OF SLC–AE

As three components are required to be optimized to minimize Eq. 2, layer-wise pre-training technology [36] is used in the training procedure, which is described as Algorithm 1.

**Algorithm 1** Training Procedure of SLC–AE

**Input:**

training data $X = \{x_s^1, \ldots, x_s^{n_s}\} \cup \{x_t^1, \ldots, x_t^{n_t}\}$

the trade-off parameters for locality and reconstruction $\mu = 0.3$; the size of reconstruction set $k = 5$; the standard deviation of the Gaussian kernel $\sigma = 5$

**Output:**

$W_e, b_e$, the parameters of encoder

$W_d, b_d$, the parameters of decoder

//Initialize the parameters of auto-encoder:

Randomly initialize the parameters of the encoder and decoder $W_e, b_e, W_d, b_d$

//Initialize the prior affinity matrix:

Initialize the prior affinity matrix $M$ using Eq. 7

//Optimization:

**for** $epoch = 1 \rightarrow n_{epoch}$ **do**

    // Refine the affinity matrix using the selector

    Use discriminative information to construct the 'reconstruction set' as described in Section III-C.2

    Refine the affinity matrix $M$ through Eq. 5

    // Optimizing the encoder and decode

    **for** $subepoch = 1 \rightarrow n_{circle}$ **do**

        Update $W_e, b_e, W_d, b_d$ to minimize Eq. 2

    **end for**

    // Estimate the affinity matrix using the selector

    Update the affinity matrix $M$ via solving Eq. 3

**end for**

At first, since the parameters of the encoder are randomly initialized, the similarity of each data pair in the affinity matrix is initialized as follows,

$$m_{i,initial}^j = exp(-\frac{(x^i - x^j)^2}{2\sigma^2}) \tag{7}$$

where $\sigma$ is the standard deviation of the Gaussian kernel. The affinity matrix estimated by Eq. 7 was used by LE [46] as a regularization term to learn a low-dimensional manifold for traditional machine learning problems, whereas we use it as an initialization considering the domain shifts in scenarios of DA. Then, the selector is used to refine the affinity matrix using the discriminative information and Eq. 5. Last, the encoder and the decoder are updated to minimize the objective function in Eq. 2.

It should be noted that, as shown in Eq. 3, the selector requires the encoded features of all labeled data to estimate the affinity matrix. This may consume considerable time when the amount of labeled data is large. Thus, the affinity matrix is updated once the encoder and decoder are updated for several epochs. Our experiments demonstrated that this strategy results in a stable training procedure.

## IV. EXPERIMENTS

In this section, we evaluate the performance of the proposed SLC–AE on seven benchmark data sets, *i.e., COIL20* [47], *MNIST, USPS, SYN Signs* [48], *GTSRB* [49], *VOC 2007* [50] and *MSRC* [51]. Detailed descriptions of the seven data sets and experimental setting are presented in Section IV-A; the comparative results are reported in Section IV-B; and the in-depth analysis, including the parameter studies and ablation studies, is presented in Section IV-C and Section IV-D respectively.

### A. EXPERIMENTAL DATA SETS AND SETTING

Several examples of images from the seven benchmark data sets are illustrated in Fig. 2. The data in two panels of one column are used as the source and the target data, respectively, since they belong to the same classification problem but follow different distributions. For example, USPS and MNIST data sets contain the images of the same ten digit classes, but belong to different domains, due to the domain shifts caused by the domain-specific noises in the real world, *e.g. the writing styles, stroke thicknesses, shapes, orientations, etc.*

Tab. 2 illustrates the statistical information of the seven data sets and the four transferable sets used in the experiments, *i.e., COILA-B, USPS-MNIST, SYN-GTSRB, and MSRC-VOC*. In each transferable set, there are two domains, *i.e., the source and the target domains*. In the experiments, we used $DS_{\mathcal{D}_s - \mathcal{D}_t}$ to denote the dissimilarity between domains, which can be estimated using the following equation,

$$DS_{\mathcal{D}_s - \mathcal{D}_t} = 1 - \frac{P(\mathcal{D}_s \rightarrow \mathcal{D}_t) + P(\mathcal{D}_t \rightarrow \mathcal{D}_s)}{2} \tag{8}$$

where $\mathcal{D}_s, \mathcal{D}_t$ are the source and the target domains, respectively; $P(\mathcal{D}_s \rightarrow \mathcal{D}_t)$ is the testing performances on the $\mathcal{D}_t$ domain using the model $P$ trained on the $\mathcal{D}_s$ domain; the arrow "$\rightarrow$" is the direction from source to target domain. For example, "USPS$\rightarrow$MNIST" denotes that USPS

**TABLE 2.** Statistics of the seven benchmark data sets.

| Sets | Domains | #Example | Type | #Feature | #Class(shared) | Domain Dissimilarity ($DS_{\mathcal{D}_s-\mathcal{D}_t}$) |
|---|---|---|---|---|---|---|
| COILA-B | COILA | 720 | Object | $32 \times 32$ | 20 | 0.17 |
| | COILB | 720 | | | | |
| USPS-MNIST | MNIST | 2,000 | Digit | $16 \times 16$ | 10 | 0.44 |
| | USPS | 1800 | | | | |
| SYN-GTSRB | SYN. Signs | 100,000 | Traffic Signs | $40 \times 40$ | 43 | 0.48 |
| | GTSRB | 51,839 | | | | |
| MSRC-VOC | MSRC | 1269 | Object | $300 \times 200$ | 6 | 0.63 |
| | VOC2007 | 1530 | | | | |

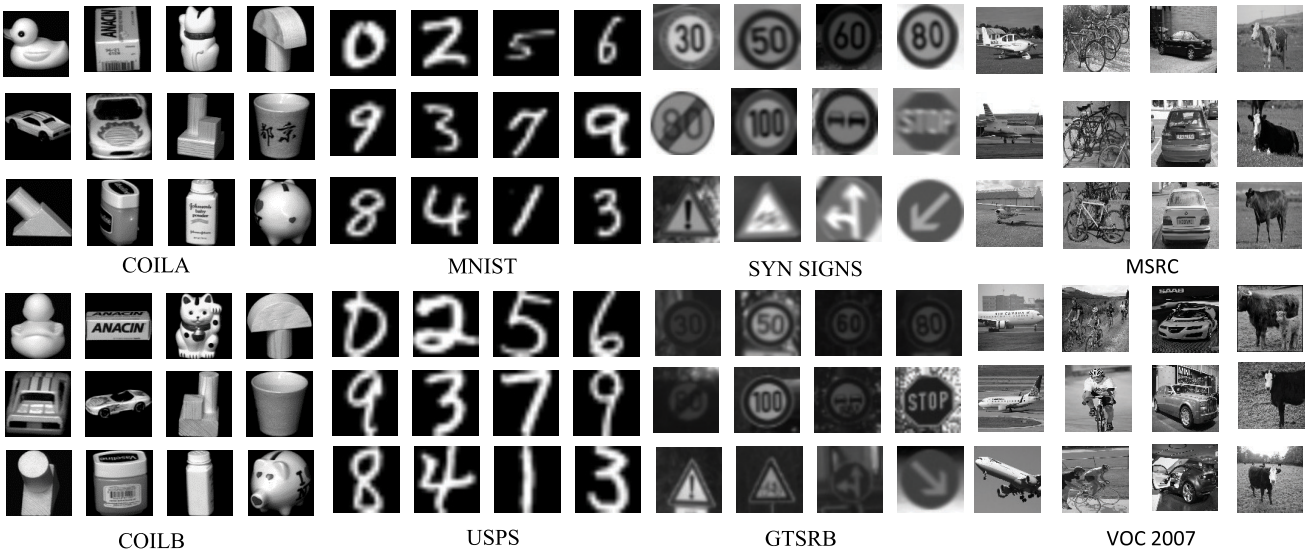


**FIGURE 2.** Image samples from COIL20, MNIST, USPS, SYN SIGNS, GTSRB, MSRC and VOC 2007 data sets, respectively. The data in two panels of the same column share the same class labels but have different data distributions.

is the source domain and MNIST is the target domain. We used a non-DA method, *i.e. principle component analysis + SVM*, as the model $P$ to calculate the $DS_{\mathcal{D}_s-\mathcal{D}_t}$ in Tab. 2.

The image examples of the seven data sets illustrated in Fig. 2 also provide intuitive explanations for the domain dissimilarity measured by the $DS_{\mathcal{D}_s-\mathcal{D}_t}$ in each transferable set. The images in MSRC-VOC contain the most complex background and objects, the domain-specific noises in MSRC-VOC are caused not only by different environmental conditions but also by the variations of the object itself. Compared to the domain-specific noises in USPS-MNIST that are caused by different writing styles, the noises in SYN-GTSRB, which are caused by different illumination and camera noises, are more complicated; the noises in COILA-B, which are caused by different orientations, could be considered as the simplest case. The detailed information of the seven benchmark data sets is described as follows.

- COIL20 (Columbia Object Image Library) object data set consists of a total of 1440 images evenly distributed in 20 different objects, *i.e., each object has 72 images*. We followed the setting used in [52] to split the data set into 2 subsets for the experiments, *i.e., COILA and COILB*. COILA includes the images of angle ranges $[0°, 85°] \cup [180°, 265°]$ and COILB includes the images of angle ranges $[90°, 175°] \cup [270°, 355°]$. All images are uniformly rescaled into $32 \times 32$ size in the experiments.
- USPS and MNIST are two commonly used image data sets of handwritten digits, which share the same ten digit classes. The USPS data set consists of 7,291 training images and 2,007 testing images. The MNIST dataset contains a training set of 60,000 images and a testing set of 10,000 images. We followed the setting used in [52]: 1800 and 2000 images are randomly selected from USPS and MNIST, and all images are uniformly resized into $16 \times 16$ in the experiments.
- Synthetic Signs and GTSRB are traffic sign image data sets, which have 43 common traffic signs, *e.g. speed limit signs, mandatory signs etc*. The Synthetic Sign data set consists of 100,000 stimulated images, which were generated by the artificial transformations algorithm proposed in [48]; the GTSRB data set consists of 39,209 training images and 12,630 testing images, which were captured by a camera installed on a vehicle from the real world. We followed the experimental setting described in [53]: all images are resized using the bi-linear interpolation method to the uniform size of 40x40 pixels.

- MSRC and VOC2007 are object image data sets, which share 6 class labels, *i.e. aeroplane, bicycle, bird, car, cow and sheep*. The MSRC dataset is provided by Microsoft Research Cambridge, which contains 4323 images labeled by 18 classes. The VOC2007 dataset contains 5011 images of 20 classes. We followed the experimental setting in [52], and randomly selected 1269 images and 1530 images from MSRC and VOC2007 respectively in the experiments. All images are resized using the bi-linear interpolation method to the uniform size of 300x200 pixels.

### B. COMPARATIVE STUDIES

In this sub-section, the seven DA tasks, *i.e., MNIST (M) → UPSP (U), UPSP (U) → MNIST (M), COILA (CA) → COILB (CB), COILB (CB) → COILA (CA), SYN Signs (S) → GTSRB (G), MSRC (MS)→ VOC (V) and VOC (V) → MSRC (MS)* , were used to evaluate the performance of the SLC–AE and the following twelve state-of-the-art DA methods.

- Principle Component Analysis (PCA) [54]: PCA is an unsupervised learning algorithm. In the experiments, PCA was first trained on the source data to learn a low-dimensional feature space. In this feature space, a SVM was then trained on the source data and used to classify the target data.
- Information-Theoretic Metric Learning (ITML) [55]: ITML is a supervised learning algorithm. In the experiments, ITML was trained on the labeled source data to learn a distance metric, which was then used to classify the target data using nearest neighbor (NN) classifier.
- Geodesic Flow Kernel (GFK) [40]: GFK is a DA method. In the experiments, GFK was first trained on the union of the source and target data to learn a geodesic flow kernel, and then a NN classifier was used to classify the target data using the learned kernel space.
- Joint Domain Adaptation (JDA) [52]: JDA is a DA method. In the experiments, JDA was first used to reduce the differences in both marginal and conditional distributions between the source and the target domains for learning a feature space. Then a NN classifier was used to classify the target data in the feature space.
- Transfer Component Analysis (TCA) [56]: TCA is a DA method. In the experiments, TCA was used to learn a sub-space supported by the transfer components at first. Then a NN classifier was used to classify the target data in the feature space.
- Marginalized Denoting Auto-encoder (mSDA) [19]: mSDA is a DA method based on SDA. In the experiment, mSDA was first trained on the union of the source and target data sets to learn a feature space. Then a NN classifier was used to classify the target data in the feature space.
- Deep Robust Encoder (DLRC) [4]: DLRC is a DA method based on SDA. In the experiment, DLRC was first trained to learn a low-rank feature space. Then a NN classifier was used to classify the target data in the feature space.
- Deep Robust Encoder (DRE) [24]: DRE is a DA method based on SDA. In the experiment, DRE was first trained to jointly optimize a low-rank dictionary and a regularized deep auto-encoder to learn a feature space. Then a NN classifier was used to classify the target data in the feature space.
- Robust Transfer Metric Learning (RTML) [57]: RTML is a DA method. In the experiment, RTML was trained to mitigate the domain shift in raw data space and feature space to learn a low-rank metric, which was used to classify the target data using NN classifier.
- Scatter Component Analysis (SCA) [58]: SCA is a DA method. In the experiment, SCA was first used to minimize the mismatch between domains to learn a feature space. Then a NN classifier was used to classify the target data in the feature space.
- Domain-Irrelevant Class clustEring (DICE) [59]: DICE is a DA method. In the experiment, DICE was first used to maximize the inter-class as well as minimize the cross-domain distribution divergence and the intra-domain structure to learn a low-dimensional feature space.Then a NN classifier was used to classify the target data in the feature space.
- Domain Invariant and Class Discriminative (DICD) [60]: DICD is a DA method. In the experiment, DICD was first used to maximize the inter-class dispersion and minimize the intra-class scatter to learn a low-dimensional feature space. Then a NN classifier was used to classify the target data in the feature space.

The NN classifier was employed after all DA methods for a fair comparison. For the non-DA methods, we followed the experimental protocol commonly used in the DA literature [4], [60], *i.e. employing SVM classifier for PCA and using NN classifier for ITML*. The recognition rate (fraction of correct matches) is used as a quality measurement for evaluating the performances of DA methods, which implies that the higher the recognition rate, the better the DA method is. Since it is hard to tune the optimal parameters through cross validation in DA experiments, we empirically searched the optimal parameters, and reported the best recognition rate for each method. Note that this experimental protocol is commonly employed in the DA literature, *e.g.,* [19], [24], [57].

Tab. 3 illustrates each model's DA performance on the four transferable sets. We can see that most DA models can achieve better performance than the non-DA models, especially on the DA tasks having a large domain shift. For example, in MSRC-VOC, which contains domain shift $DS_{\mathcal{D}_s-\mathcal{D}_t} = 0.63$, all of the listed DA models performed better, on average, than the non-DA models. Due to the characteristics of each model and the amount of images used for training, we also noticed that a few of the DA models, *e.g. GFK and TCA*, did not achieve better (average) performances than the non-DA models on the seven data sets. The proposed SLC–AE was significantly better than the non-DA

**TABLE 3. Comparison of models' recognition performances.**

| Method | Recognition Rate (%) | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|
| | COILA-B | | USPS-MNIST | | SYN-GTSRB | MSRC-VOC | | Average | DA Model | Encoder -based |
| | CA→CB | CB→CA | M→U | U→M | S→G | MS→V | V→MS | | | |
| PCA | 83.52 | 83.13 | 66.22 | 44.95 | 62.76 | 32.94 | 41.06 | 59.22 | × | × |
| ITML | 85.30 | 86.86 | 67.24 | 46.87 | 66.54 | 32.35 | 43.30 | 61.20 | × | × |
| GFK | 72.50 | 74.17 | 61.22 | 46.45 | 70.34 | 34.18 | 44.47 | 57.61 | ✓ | × |
| JDA | 89.31 | 88.47 | 67.48 | 59.65 | 73.35 | 37.4 | 58.20 | 68.12 | ✓ | × |
| TCA | 88.47 | 85.83 | 58.78 | 44.15 | 68.4 | 32.55 | 45.78 | 60.56 | ✓ | × |
| mSDA | 82.87 | 85.00 | 60.05 | 48.22 | 77.68 | 35.10 | 53.50 | 63.2 | ✓ | ✓ |
| DLRC | 90.35 | 89.15 | 69.52 | 57.82 | 78.56 | 38.62 | 53.56 | 68.2 | ✓ | ✓ |
| DRE | 93.17 | 92.47 | 71.83 | 60.35 | 80.4 | 39.62 | 53.84 | 70.24 | ✓ | ✓ |
| RTML | 91.23 | - | - | 61.82 | - | 38.63 | - | - | ✓ | × |
| SCA | - | - | 65.11 | 48.00 | - | 32.75 | 48.94 | - | ✓ | × |
| DICE | 92.5 | 94.4 | **79.7** | 59.8 | - | - | - | - | ✓ | × |
| DICD | 93.33 | **95.69** | 77.83 | 65.20 | - | - | - | - | ✓ | × |
| SLC–AE | **96.77** | 95.27 | 72.02 | **68.89** | **90.47** | **48.32** | **60.71** | **76.06** | ✓ | ✓ |

models on all four of the transferable sets. Moreover, the improvements brought by the SLC–AE were (approximately) positively related to the domain shifts in the four transferable sets. For example, compared to PCA, the improvements achieved by the SLC–AE were 12.69% (COILA-B), 14.87% (USPS-MNIST),27.71% (SYN-GTSRB) and 17.51% (MSRC-VOC); accordingly, the domain shifts in the four transferable sets were 0.17 (COILA-B), 0.44(USPS-MNIST), 0.48 (SYN-GTSRB) and 0.63 (MSRC-VOC). The abnormal performance improvement in SYN-GTSRB was commonly achieved by the deep models, such as mSDA, DLRC and DRE, rather than by the shallow models, which was attributed to the large number of images in SYN-GTSRB.

In the experiments, four deep models based on the encoder were evaluated on the four transferable sets. The DLRC uses the same training strategy as the mSDA model, *i.e. the training strategy to reduce man-made noises*. Instead of reconstructing the input, DRE learns a low-rank dictionary to generate a new sample for reconstruction. Since the low-rank constraint has proven to be effective in noise reduction [22], [61], the purpose of DRE could be considered as similar to that of the SLC–AE, *i.e., training SDA to reduce the domain-specific noises from the real-word*. Although these four models are all based on SDA, the objective function designed for reducing the domain-specific noises from the real world (see Eq. 2) rather than the man-made noises (see Eq. 1) made the performance of DRE and the SLC–AE superior to the mSDA and DLRC on all of the four transferable sets. Compared to DRE, the discrimination and locality constraints made the SLC–AE a better model, especially for MSRC-VOC and SYN-GTSRB, which contain the largest and the second largest domain shifts in the four transferable sets.

Finally, we compared the performance of the SLC–AE with the four recently published DA models, *i.e. RTML, SCA, DICE,* and *DICD*. Since the programs of these models are yet to be released in the public domain, the models' performances can be cited from the published literature, but only on the parts of the DA tasks used in the experiments. The comparison results showed that the SLC–AE achieved the best average performances, but did not perform well on the M→U task. The DICE and DICD achieved the best and the second best performance on the M→U task, which were much better than the others. The strategy commonly used in DICE and DICD, but not in the methods that performed poorly on the M→U task, is to maximize the inter-class dispersion of each domain. This provides positive evidence in favor of the assumption that the major problem in performing DA on the M→U task is caused by the inter-class dispersion. How to use the SLC–AE to maximize the inter-class dispersion of each domain will be addressed in our future work.

### C. PARAMETER STUDIES

In this sub-section, we evaluate the properties of the SLC–AE, *e.g., robustness to noise, parameter influence, and layer size impact*, to achieve a better understanding.

First, we investigated the impacts of different corruption ratios on six DA models. As shown in Fig. 3, the six models were evaluated on the CA→CB task with 0%, 10%, 20%, 30%, 40% and 50% corruption (Gaussian noise), respectively. We can see that the SDA-based models, such as mSDA, DRE and SLC–AE, achieved more robust performance with various levels of corruption than the non-SDA-based models, such as the PCA, TCA, and JDA models, due to the denoising strategy used in the training procedure of SDA. The SLC–AE model outperformed other competitors by achieving better and more robust performance, which demonstrated that it could be used as a robust feature extractor, especially for data with large amount of corruption.

Second, the influence of parameters $k$ and $\mu$ on the performance of the SLC–AE was evaluated, where $k$ denotes the size of the reconstruction set and $\mu$ the trade-off between the reconstruction error and locality. Fig. 4 illustrates the performance of the SLC–AE on the CA→CB task using different $k$ and $\mu$. We can see that the parameter $k$ was very important to the SLC–AE, which is understandable because the more the neighbor samples are used to generate the 'clean' data, the higher the possibility that the domain-specific noises

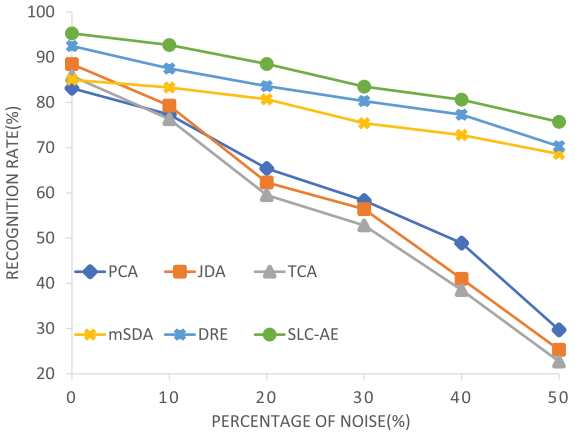


**FIGURE 3.** Recognition rates of models on the COIL20 database with different levels of noise.

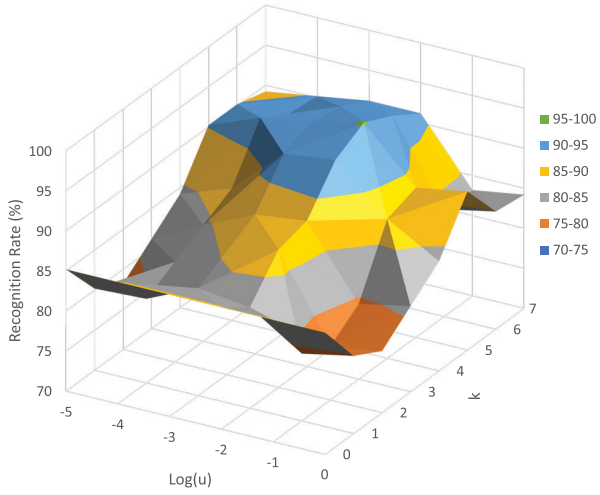


**FIGURE 4.** Parameter analysis on $k$ and $\mu$.

can be removed. The experimental results demonstrated that the performance of the SLC–AE was stable when $k > 4$.

We let $k = 0$ denote that the input is 'clean', a training strategy of the conventional SDA. When $k = 0$, since the parameter $\mu$ was not used, it did not affect the performances of the SLC–AE. It should be noted that a larger $k$ decreases the importance of $\mu$. Without loss of generality, we set $k = 5$ and $\mu = 10^{-2}$ throughout the experiments.

Finally, we evaluated the impact of the layer size for the SLC–AE on the five DA tasks, *i.e. M→U, U→M, C1→C2, C2→C1 and S→G*. Fig. 5 illustrates the performance of the SLC–AE using different layer sizes. We can see that the SLC–AE achieved better performance when the layer size increased, especially in the DA task having a large domain shift. The experimental results demonstrated that the shift between two domains could be minimized by the SLC–AE from coarse to fine in the manner of stacking auto-encoder. Considering the additional training time consumed by the deeper structure, we used a six-layer structure to generate the evaluation results in the experiments.

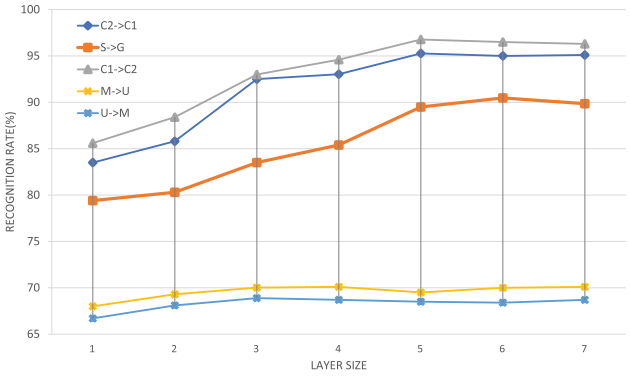


**FIGURE 5.** The impact of layer sizes on the performance of the SLC–AE.

**TABLE 4.** Ablation study for the different neighbor selection strategies of the SLC–AE.

| Method | Recognition Rate (%) | | | | |
|---|---|---|---|---|---|
| | C1→C2 | C2→C1 | M→U | U→M | S→G |
| SLC–AE$_{ns}$ | 86.5 | 87.17 | 67.22 | 52.55 | 75.34 |
| SLC–AE$_{am}$ | 89.3 | 88.6 | 68.4 | 56.7 | 79.56 |
| SLC–AE | **96.77** | **95.27** | **72.02** | **68.89** | **90.47** |

### D. ABLATION STUDY

In this sub-section, we investigate the impact of the innovative component, *i.e. the selector*, on the performance of the SLC–AE. The selector consists of two innovative strategies, *i.e estimating the affinity matrix in the low-dimensional space and refining the matrix using discriminative information*. The following two models are designed to compare with the SLC–AE for the ablation study,

- SLC–AE$_{ns}$: No selector was used in the SLC–AE. The neighbors were directly selected according to the affinity matrix estimated in the high-dimensional space using Eq. 7.
- SLC–AE$_{am}$: No refinement was used for the affinity matrix in the SLC–AE. The affinity matrix was estimated by the selector in the low-dimensional space using Eq. 3, but no additionally discriminative information was used to refine it.

Tab. 4 illustrates the performances of the three models on the five DA tasks. We can see that estimating the affinity matrix in the low-dimensional space plays an important role for the DA tasks containing large domain shifts. For example, compared to the SLC–AE$_{ns}$, the performance improvement brought by the SLC–AE$_{am}$ on the S→G task (4.22%) is larger than that on the C1→C2 task (2.8%) and the C2→C1 task (1.43%). This is because the image illumination between domains significantly changes in SYN-GTSRB, and the neighbor samples sharing similar pixel intensities have a low possibility of being semantically similar. The experimental results demonstrated that estimating the affinity matrix in the low-dimensional space is an effective strategy in such scenarios.

In the proposed SLC–AE, the discriminative information contained in the source and target data is used to refine the

affinity matrix. This is important in the early stages of model training, when the neighbor samples in the low-dimensional manifold have different class labels, due to the randomly initialized model parameters. Selecting the neighbor samples sharing the same class label for 'clean' data generation leads the feature learning procedure to minimize the intra-class divergence of each domain as well as the conditional distributions between domains. From the experimental results, we can see that the discriminative information contributed 3%–10.44% to the performance improvement achieved by the SLC–AE.

## V. CONCLUSION

In this paper, we present our research in DA with a focus in an area that assumes that the originally collected data in different domains are 'corrupted' by the domain-specific noises from the real world. The proposed deep DA architecture, SLC–AE, was designed to learn the robust and domain-invariant features through training the SDA model to reduce the domain-specific noises. The core problem of SLC–AE is how to generate the 'clean' data from the 'corrupted' inputs for the supervised training of SDA. To address this problem, a novel component, the selector, is tailor-made to learn a low-dimensional manifold for selecting semantically similar neighbor samples to reduce the domain-specific noises. We evaluated the performance of the SLC–AE on seven data sets. Compared to twelve state-of-the-art methods, the experimental results demonstrated that the SLC–AE achieved the best average performance on the seven data sets. Moreover, the SLC–AE was robust to the domain-specific noises from the real world, which was attributed to the locality and discrimination constraints used in dynamically exploring the low-dimensional manifold.

## REFERENCES

[1] S. Maji and J. Malik, "Fast and accurate digit classification," Dept. EECS, Univ. California, Berkeley, CA, USA, Tech. Rep. UCB/EECS-2009-159, Nov. 2009. [Online]. Available: http://www2.eecs.berkeley.edu/Pubs/TechRpts/2009/EECS-2009-159.html

[2] M. Al-Shedivat, J. J.-Y. Wang, M. Alzahrani, J. Z. Huang, and X. Gao, "Supervised transfer sparse coding," in *Proc. 28th AAAI Conf. Artif. Intell.* Menlo Park, CA, USA: AAAI Press, 2014, pp. 1665–1672.

[3] S. J. Pan, V. W. Zheng, Q. Yang, and D. H. Hu, "Transfer learning for WiFi-based indoor localization," in *Proc. Assoc. Advancement Artif. Intell. Workshop*. Palo Alto, CA, USA: Assoc. Advancement Artif. Intell., 2008, pp. 1–6.

[4] Z. Ding, M. Shao, and Y. Fu, "Deep low-rank coding for transfer learning," in *Proc. 24th Int. Conf. Artif. Intell.* Menlo Park, CA, USA: AAAI Press, 2015, pp. 3453–3459.

[5] L. Duan, D. Xu, I. W.-H. Tsang, and J. Luo, "Visual event recognition in videos by learning from Web data," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 34, no. 9, pp. 1667–1680, Sep. 2012.

[6] H. Shimodaira, "Improving predictive inference under covariate shift by weighting the log-likelihood function," *J. Statist. Planning Inference*, vol. 90, no. 2, pp. 227–244, 2000.

[7] N. Japkowicz and S. Stephen, "The class imbalance problem: A systematic study," *Intell. Data Anal.*, vol. 6, no. 5, pp. 429–449, Oct. 2002.

[8] J. J. Heckman, "Sample selection bias as a specification error," *Econometrica*, vol. 47, pp. 61–153, Jan. 1979.

[9] B. Zadrozny, "Learning and evaluating classifiers under sample selection bias," in *Proc. 21st Int. Conf. Mach. Learn.*. New York, NY, USA: ACM, 2004, pp. 114–122.

[10] J. Donahue, Y. Jia, O. Vinyals, J. Hoffman, N. Zhang, E. Tzeng, and T. Darrell, "DeCAF: A deep convolutional activation feature for generic visual recognition," in *Proc. 31st Int. Conf. Int. Conf. Mach. Learn.* New York, NY, USA: JMLR.org, 2014, pp. I-647–I-655.

[11] S. Saxena and J. Verbeek, "Heterogeneous face recognition with CNNs," in *Proc. Eur. Conf. Comput. Vis.* Cham, Switzerland: Springer, 2016, pp. 483–491.

[12] B. Sun, J. Feng, and K. Saenko, "Return of frustratingly easy domain adaptation," in *Proc. 30th AAAI Conf. Artif. Intell.* Menlo Park, CA, USA: AAAI Press, 2016, pp. 2058–2065.

[13] G. Csurka, B. Chidlowskii, S. Clinchant, and S. Michel, "Unsupervised domain adaptation with regularized domain instance denoising," in *Proc. Eur. Conf. Comput. Vis.* Cham, Switzerland: Springer, 2016, pp. 458–466.

[14] M. D. Zeiler and R. Fergus, "Visualizing and understanding convolutional networks," in *Proc. Eur. Conf. Comput. Vis.* Cham, Switzerland: Springer, 2014, pp. 818–833.

[15] M. Oquab, L. Bottou, I. Laptev, and J. Sivic, "Learning and transferring mid-level image representations using convolutional neural networks," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.* Piscataway, NJ, USA: IEEE, Jun. 2014, pp. 1717–1724.

[16] A. Babenko, A. Slesarev, A. Chigorin, and V. Lempitsky, "Neural codes for image retrieval," in *Proc. Eur. Conf. Comput. Vis.* Cham, Switzerland: Springer, 2014, pp. 584–599.

[17] B. Chu, V. Madhavan, O. Beijbom, J. Hoffman, and T. Darrell, "Best practices for fine-tuning visual classifiers to new domains," in *Proc. Eur. Conf. Comput. Vis.* Cham, Switzerland: Springer, 2016, pp. 435–442.

[18] X. Glorot, A. Bordes, and Y. Bengio, "Domain adaptation for large-scale sentiment classification: A deep learning approach," in *Proc. 28th Int. Conf. Int. Conf. Mach. Learn.* Madison, WI, USA: Omnipress, 2011, pp. 513–520.

[19] M. Chen, Z. Xu, K. Q. Weinberger, and F. Sha, "Marginalized denoising autoencoders for domain adaptation," in *Proc. 29th Int. Conf. Int. Conf. Mach. Learn.* Madison, WI, USA: Omnipress, 2012, pp. 1627–1634.

[20] P. Vincent, H. Larochelle, Y. Bengio, and P.-A. Manzagol, "Extracting and composing robust features with denoising autoencoders," in *Proc. 25th Int. Conf. Mach. Learn.* New York, NY, USA: ACM, 2008, pp. 1096–1103.

[21] A. Gogna and A. Majumdar, "Discriminative autoencoder for feature extraction: Application to character recognition," *Neural Process. Lett.*, vol. 49, pp. 1723–1735, Jun. 2018.

[22] K. Gupta and A. Majumdar, "Learning autoencoders with low-rank weights," in *Proc. IEEE Int. Conf. Image Process.* Piscataway, NY, USA: IEEE, Sep. 2017, pp. 3899–3903.

[23] F. Yuan, L. Yao, and B. Benatallah, "Adversarial collaborative auto-encoder for Top-N recommendation," Aug. 2018, *arXiv:1808.05361*. [Online]. Available: https://arxiv.org/abs/1808.05361

[24] Z. Ding, M. Shao, and Y. Fu, "Deep robust encoder through locality preserving low-rank dictionary," in *Proc. Eur. Conf. Comput. Vis.* Cham, Switzerland: Springer, 2016, pp. 567–582.

[25] T. Hongshen, R. Ni, Y. Zhao, and X. Li, "Median filtering detection of small-size image based on CNN," *J. Vis. Commun. Image Represent.*, vol. 51, pp. 162–168, Feb. 2018.

[26] C.-S. Lee, Y.-H. Kuo, and P.-T. Yu, "Weighted fuzzy mean filters for image processing," *Fuzzy Sets Syst.*, vol. 89, pp. 157–180, Jul. 1997.

[27] A. B. Said, A. Mohamed, T. Elfouly, K. Abualsaud, and K. Harras, "Deep learning and low rank dictionary model for mhealth data classification," in *Proc. 14th Int. Wireless Commun. Mobile Comput. Conf. (IWCMC)*. Piscataway, NJ, USA: IEEE, Jun. 2018, pp. 358–363.

[28] W. Wang, Y. Huang, Y. Wang, and L. Wang, "Generalized autoencoder: A neural network framework for dimensionality reduction," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. Workshops*. Piscataway, NY, USA: IEEE, Jun. 2014, pp. 490–497.

[29] S. J. Pan and Q. Yang, "A survey on transfer learning," *IEEE Trans. Knowl. Data Eng.*, vol. 22, no. 10, pp. 1345–1359, Oct. 2010.

[30] J. Yosinski, J. Clune, Y. Bengio, and H. Lipson, "How transferable are features in deep neural networks?" in *Proc. 27th Int. Conf. Neural Inf. Process. Syst.* Cambridge, MA, USA: MIT Press, 2014, pp. 3320–3328.

[31] E. Tzeng, J. Hoffman, N. Zhang, K. Saenko, and T. Darrell, "Deep domain confusion: Maximizing for domain invariance," *CoRR*, vol. abs/1412.3474, pp. 1–9, Dec. 2014.

[32] M. Long, Y. Cao, J. Wang, and M. I. Jordan, "Learning transferable features with deep adaptation networks," in *Proc. 32nd Int. Conf. Int. Conf. Mach. Learn.* New York, NY, USA: JMLR.org, 2015, pp. 97–105.

[33] M. Long, H. Zhu, J. Wang, and M. I. Jordan, "Deep transfer learning with joint adaptation networks," in *Proc. 34th Int. Conf. Mach. Learn.* New York, NY, USA: JMLR.org, 2017, pp. 2208–2217.
[34] I. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. Courville, and Y. Bengio, "Generative adversarial nets," in *Proc. 27th Int. Conf. Neural Inf. Process. Syst.* Cambridge, MA, USA: MIT Press, 2014, pp. 2672–2680.
[35] Y. Ganin, E. Ustinova, H. Ajakan, P. Germain, H. Larochelle, F. Laviolette, M. Marchand, and V. Lempitsky, "Domain-adversarial training of neural networks," *J. Mach. Learn. Res.*, vol. 17, no. 1, pp. 2030–2096, May 2015.
[36] G. E. Hinton and R. R. Salakhutdinov, "Reducing the dimensionality of data with neural networks," *Science*, vol. 313, no. 5786, pp. 504–507, 2006.
[37] M. Baktashmotlagh, M. T. Harandi, B. C. Lovell, and M. Salzmann, "Domain adaptation on the statistical manifold," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2014, pp. 2481–2488.
[38] J. Li, J. Zhao, and K. Lu, "Joint feature selection and structure preservation for domain adaptation," in *Proc. 25th Int. Joint Conf. Artif. Intell.* Menlo Park, CA, USA: AAAI Press, 2016, pp. 1697–1703.
[39] M. Kan, S. Shan, and X. Chen, "Bi-shifting auto-encoder for unsupervised domain adaptation," in *Proc. IEEE Int. Conf. Comput. Vis.* Piscataway, NJ, USA: IEEE, Dec. 2015, pp. 3846–3854.
[40] B. Gong, Y. Shi, F. Sha, and K. Grauman, "Geodesic flow kernel for unsupervised domain adaptation," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.* Piscataway, NJ, USA: IEEE, Jun. 2012, pp. 2066–2073.
[41] D. Hong, N. Yokoya, and X. X. Zhu, "Learning a robust local manifold representation for hyperspectral dimensionality reduction," *IEEE J. Sel. Topics Appl. Earth Observ. Remote Sens.*, vol. 10, no. 6, pp. 2960–2975, Jun. 2017.
[42] X. Wei, H. Shen, and M. Kleinsteuber, "Trace quotient meets sparsity: A method for learning low dimensional image representations," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.* Piscataway, NJ, USA: IEEE, Jun. 2016, pp. 5268–5277.
[43] K. Yu and T. Zhang, "Improved local coordinate coding using local tangents," in *Proc. 27th Int. Conf. Int. Conf. Mach. Learn.* Madison, WI, USA: Omnipress, 2010, pp. 1215–1222.
[44] K. Yu, T. Zhang, and Y. Gong, "Nonlinear learning using local coordinate coding," in *Proc. 22nd Int. Conf. Neural Inf. Process. Syst.* Red Hook, NY, USA: Curran Associates, 2009, pp. 2223–2231.
[45] M. Long, J. Wang, G. Ding, J. Sun, and P. S. Yu, "Transfer joint matching for unsupervised domain adaptation," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.* Piscataway, NJ, USA: IEEE, Jun. 2014, pp. 1410–1417.
[46] M. Belkin and P. Niyogi, "Laplacian eigenmaps and spectral techniques for embedding and clustering," in *Proc. 14th Int. Conf. Neural Inf. Process. Syst.* Cambridge, MA, USA: MIT Press, 2001, pp. 585–591.
[47] S. A. Nene, S. K. Nayar, and H. Murase, "Columbia object image library (coil-20)," Columbia Univ., Columbia, SC, USA, Tech. Rep. CUCS-005-96, Feb. 1996. [Online]. Available: http://www.cs.columbia.edu/CAVE/software/softlib/coil-20.php
[48] B. Moiseev, A. Konev, A. Chigorin, and A. Konushin, "Evaluation of traffic sign recognition methods trained on synthetically generated data," in *Proc. Int. Conf. Adv. Concepts Intell. Vis. Syst.* Berlin, Germany: Springer, 2013, pp. 576–583.
[49] J. Stallkamp, M. Schlipsing, J. Salmen, and C. Igel, "Man vs. computer: Benchmarking machine learning algorithms for traffic sign recognition," *Neural Netw.*, vol. 32, pp. 323–332, Aug. 2012.
[50] M. Everingham, L. Van Gool, C. K. I. Williams, J. Winn, and A. Zisserman. *The PASCAL Visual Object Classes Challenge 2007 (VOC2007) Results*. [Online]. Available: http://host.robots.ox.ac.uk/pascal/VOC/voc2007/index.html
[51] J. Winn, A. Criminisi, and T. Minka, "Object categorization by learned universal visual dictionary," in *Proc. IEEE Int. Conf. Comput. Vis.* Piscataway, NJ, USA: IEEE, Oct. 2005, pp. 1800–1807.
[52] M. Long, J. Wang, G. Ding, J. Sun, and P. S. Yu, "Transfer feature learning with joint distribution adaptation," in *Proc. IEEE Int. Conf. Comput. Vis.* Piscataway, NJ, USA: IEEE, Dec. 2013, pp. 2200–2207.
[53] Y. Ganin and V. Lempitsky, "Unsupervised domain adaptation by backpropagation," in *Proc. 32nd Int. Conf. Int. Conf. Mach. Learn.* New York, NY, USA: JMLR.org, 2015, pp. 1180–1189.
[54] S. Wold, K. Esbensen, and P. Geladi, "Principal component analysis," *Chemometrics Intell. Lab. Syst.*, vol. 2, nos. 1–3, pp. 37–52, 1987.
[55] J. V. Davis, B. Kulis, P. Jain, S. Sra, and I. S. Dhillon, "Information-theoretic metric learning," in *Proc. 24th Int. Conf. Mach. Learn.* New York, NY, USA: ACM, 2007, pp. 209–216.
[56] S. J. Pan, I. W. Tsang, J. T. Kwok, and Q. Yang, "Domain adaptation via transfer component analysis," *IEEE Trans. Neural Netw.*, vol. 22, no. 2, pp. 199–210, Feb. 2011.
[57] Z. Ding and Y. Fu, "Robust transfer metric learning for image classification," *IEEE Trans. Image Process.*, vol. 26, no. 2, pp. 660–670, Feb. 2017.
[58] M. Ghifary, D. Balduzzi, W. B. Kleijn, and M. Zhang, "Scatter component analysis: A unified framework for domain adaptation and domain generalization," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 39, no. 7, pp. 1414–1430, Jul. 2017.
[59] J. Liang, R. He, Z. Sun, and T. Tan, "Aggregating randomized clustering-promoting invariant projections for domain adaptation," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 41, no. 5, pp. 1027–1042, May 2019.
[60] S. Li, S. Song, G. Huang, Z. Ding, and C. Wu, "Domain invariant and class discriminative feature learning for visual domain adaptation," *IEEE Trans. Image Process.*, vol. 27, no. 9, pp. 4260–4273, Sep. 2018.
[61] P. Gao, R. Wang, M. Wang, and J. H. Chow, "Low-rank matrix recovery from noisy, quantized, and erroneous measurements," *IEEE Trans. Signal Process.*, vol. 66, no. 11, pp. 2918–2932, Jun. 2018.

**XISHUAI PENG** received the B.S. degree in computer science from the Nanjing University of Technology, Nanjing, Jiangsu, China, in 2010, and the M.S. degree in computer application from the Nanjing University of Science and Technology, Nanjing, in 2014. He is currently pursuing the Ph.D. degree with Shanghai Jiao Tong University, Shanghai, China. His research interests include image processing, deep learning, and computer vision.

**YUANXIANG LI** received the Ph.D. degree in signal and information processing from Tsinghua University, Beijing, China, in 2001. He was a Research Fellow with the Department of Computer Science, National University of Singapore, from 2002 to 2004, and a Visiting Professor with the Department of Electrical and Computer Engineering, University of Michigan–Dearborn, from 2015 to 2016. He is currently an Associate Professor with the School of Aeronautics and Astronautics, Shanghai Jiao Tong University, China. His current research interests include machine learning, image recognition, image restoration, image compression, fault diagnosis, and prediction.

**YI LU MURPHEY** received the M.S. degree in computer science from Wayne State University, Detroit, MI, USA, in 1983, and the Ph.D. degree with a major in computer engineering and a minor in control engineering from the University of Michigan, Ann Arbor, MI, USA, in 1989. She is currently a Professor with the College of Engineering and Computer Science, University of Michigan-Dearborn, Dearborn, MI, USA. Her current research interests include machine learning, pattern recognition, computer vision, and intelligent systems with applications to automated and connected vehicles, optimal vehicle power management, data analytics, and robotic vision systems. She is also a Senior Life Member of AAAI. She is an Editor of the *Journal of Pattern Recognition*.

**JIANHUA LUO** received the Ph.D. degree in biomedical engineering from Zhejiang University, Hangzhou, Zhejiang, China, in 1995. He is currently a Professor with the School of Aeronautics and Astronautics, Shanghai Jiao Tong University, China. His research interests include medical image processing and bioinformatics.