IEEE Access
Multidisciplinary : Rapid Review : Open Access Journal

Received 12 August 2022, accepted 6 October 2022, date of publication 9 January 2023, date of current version 12 January 2023.

Digital Object Identifier 10.1109/ACCESS.2023.3235205

RESEARCH ARTICLE

Deep Domain Adaptation Using Cascaded Learning Networks and Metric Learning

ZHIYONG ZENG, DAWEI LI, AND XIUJUAN YANG
College of Computer and Cyber Security, Fujian Normal University, Fuzhou 350117, China
Digital Fujian Institute of Big Data Security Technology, Fujian Normal University, Fuzhou 350117, China

Corresponding author: Zhiyong Zeng (zzyong@fjnu.edu.cn)

This work was supported in part by the Guiding Project of Fujian Science and Technology Plan under Grant 2021Y0011, in part by the National Natural Science Foundation of China under Grant 61502103, and in part by the Natural Science Foundation of Fujian Province under Grant 2022J01187.

ABSTRACT How can we bridge efficiently distribution difference between the source and target domains in an isomorphic latent feature space by using metric learning. This study introduces metric learning on manifolds which combine a cascaded learning network and a metric learning model to form a Unified Domain Adaptation Model. Our approach is based on formulating a transfer from the source to the target as a geometric mean metric learning problem on manifolds. The solution of symmetric positive definite covariance matrices not only reduces the statistical differences between the source and target domains, but also the underlying geometry of the source and target domains using diffusions on the underlying source and target manifolds. By retaining both the nonlinear structure of the Riemannian geometry of the open cone of symmetric positive definite matrices and cascaded learning networks, we improve the state-of-the-art results on the Amazon (A), Caltech256 (C), DSLR (D), Webcam (W) and VisDA benchmark datasets by knowledge transfer, while achieving comparable performances to competing methods on domain adaptation modeling.

INDEX TERMS Cascaded learning networks, transfer learning, pca filters, metric learning, domain adaptation, symmetric positive definite covariance matrices, geodesic distance.

I. INTRODUCTION

When we use machine learning to solve real world problems, for example, in pattern recognition and image retrieval tasks, a significant challenge is that these algorithms require massive amounts of labeled training data, which may not always be available. Therefore, designing a good machine learning method that can transfer knowledge across domains plays an important role. Although many hand-crafted feature-based machine learning methods have been proposed in the past decades, there are many shortcomings of these algorithms: (1) Most of them use linear metrics to transform samples into a linear feature space, so that the nonlinear relationship of samples can be neglected. (2) Most of them assume that the distribution of the test and training samples are the same. This assumption does not hold in many computer vision applications because samples are captured across different scenarios. Hence, learning deep metrics directly from training data can achieve better performance than hand-crafted feature-based metric learning [1], [2], [3], [4], [5]. However, learning a deep network involves backward propagation, which incurs a high computational cost. In addition, training a deep network that is robust for image classification depends on the expertise in parameter tuning and tricks.

Fortunately, a combination of cascaded network and metric learning can be an alternative solution. The benefit of cascading is that it shrink the background while keeping the objects in the scene, which accelerates the training procedure of feature learning. In addition, cascading enables the integration of multiple sources of variation between the source and the target in a unified framework with a theoretically optimal solution. Metric learning can optimize deep neural networks for learning a good feature representation, which maximizes inter-class variations and minimizes intra-class variations, while also reducing the distribution divergence between the source and target.

Inspired by cascaded networks and metric learning, to solve the above-mentioned problems, we hope to design

The associate editor coordinating the review of this manuscript and approving it for publication was Chuan Li.

a simple deep learning network that can train and adapt to different data and tasks. Meanwhile, such a network could serve as a good baseline for people to justify the use of sophisticated architectures for their deep neural networks. In addition, we use a nonlinear distance to transform samples into a nonlinear feature space; thus, we can obtain the nonlinear relationship of the samples. Meanwhile, we introduced a mapping to bridge the distribution discrepancy between the source and target domain data.

To achieve these goals, on the one hand, we use basic and easy operations to emulate the processing layers in a convolutional neural network(CNN), PCA filters are used as data-adapting convolution filter banks, ReLu operations serve as activation layers, binary quantization serve as the nonlinear layers, and nonlinear operations are only used in the last output layer; we use the block-wise histograms of the binary codes to replace the pooling layers of deep neural networks. On the other hand, we model covariance matrices of samples as a nonlinear feature space of symmetric positive definite matrices, so that we could obtain the nonlinear relationship of samples. We used the maximum mean discrepancy (MMD) metric to align the source and target data.

In this study, we propose a novel deep domain adaptation algorithm using metric learning on a manifold. Unlike other deep-domain adaptation methods, we use a simple yet novel method; we do not exploit back propagation to update the model parameters. Specifically, we make the following main contributions: (1) We use cascaded principal components analysis filters to build a deep learning network, which not only extracts robust hierarchical visual features of an image, but also avoids a large amount of computational complexity. (2) We model a space of covariance matrices as a curved Riemannian manifold of symmetric positive definite (SPD) matrices for adapting deep representations, which reduces the distance between domains by projecting data onto a learned transfer subspace and largely enhances adaptation effectiveness compared to most metric learning methods. (3) We demonstrate both the uniqueness and global optimality of our method because we reduce all domain adaptation computations to nested equations, which involves solving the well-known Riccati equation.

The architecture of the proposed algorithm is illustrated in Figure 1. There are two components in this architecture: a cascaded learning network and cascaded metric learning on a Riemannian manifold. The justification and advantages that we introduce such a strategy are as follows: (1) In a cascaded learning network, although we only use forward propagation to update the parameters, we minimize the reconstruction error of PCA to obtain a family of orthogonal filters as convolution filter banks, which are much better than the filter parameters randomly initialized in CNN. In addition, the goal of using backward propagation in a CNN is to minimize classification error. (2) Stochastic gradient descent (SGD) was used for filter learning in the CNN, and the training time of deep network was much longer than that of the proposed method. For example, training a cascaded learning network on a training set (i.e., approximately 100000 images of 300 $\times$ 300 pixel dimension) takes approximately an hour, but the CNN take about 15 hours. (3) In cascaded metric learning on the Riemannian manifold, by introducing a secondary criterion for aligning the source and target domains, the proposed approach exploits both geometric and statistical information of the source and target domain data, and multiple sources of alignment are integrated by solving nested sets of Riccati equations. (4) We reduce all domain adaptation computations to nested equations that involve solve the Riccati equation. Hence, our method is both unique and globally optimal. (5) The experimental results confirm the remarkable benefits of the proposed method. (6) Our method is attractive because of its simplicity and tremendous speedup over widely used metric learning methods.

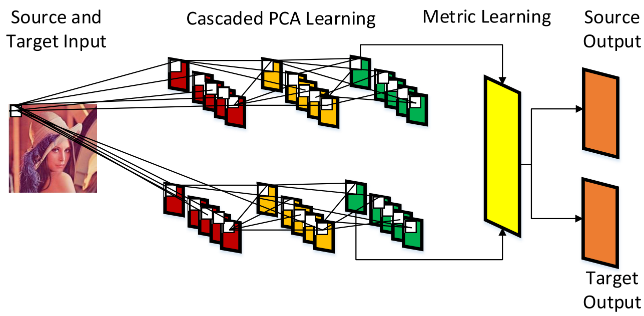


FIGURE 1. **The architecture of the proposed cascaded principal component analysis network and metric learning. The proposed framework for deep domain adaptation. Transferable feature will finish transiting from general to specific along the network. (1) the features extracted from cascaded principal component analysis network are general. (2) the features extracted from metric learning are going to fit specific tasks by Riemannian manifold of symmetric positive definite (SPD) matrices and MMD.**

II. RELATED WORK

Domain adaptation involves learning a discriminative model in the presence of a data shift between the source and the target. Domain adaptation aims to build learning machines that generalize across domains with different probability distributions. The main challenge of domain adaptation is the reduction of discrepancies in the data distribution across domains. Most existing domain adaptation methods learn a shallow representation model by which domain discrepancy is minimised, which cannot disentangle the explanatory factors of the variations. Deep neural networks can learn nonlinear relationships of samples that explain factors of variations behind data and extract transferable factors underlying different domains [6], [7]. Hence, deep neural networks have been widely studied in the computer vision [8], [9], [10], natural language processing communities [11]. In recent years, many deep neural network models have been presented, and representative models include IBN-Net, ResNet, SeNet, DenseNet [12], [13], [14], [15], and generative adversarial networks [16], [17], [18]. However, most of them aim to learn hierarchical feature representations via deep learning rather than domain adaptation.

More recent studies have shown that deep learning has been exploited in metric learning. Schroff proposed a triplet measure of face similarity to learn mapping from facial images to a compact Euclidean space [19]. Li introduced a progressive and nonlinear deep metric learning approach to obtain a better metric space from a learnt metric space [20]. Although these approaches have achieved satisfactory performance, they assume that the training and the test samples are captured in the same scenarios, which is not always satisfied in real world. Meanwhile, to train network model, deep neural networks is also to need a large amount of labeled data, when data samples are small, deep networks are usually overfitted.

To overcome the shortcomings of deep metric learning, several different transfer learning methods have been proposed, including deep adaptation network [21], [22], [23], domain adversarial network [24], [25], [26]. For example, Ganin augmented a deep architecture with few standard layers and a gradient reversal layer to achieve domain adaptation behaviour [22]. Sun extended the CORAL method [27] to learn nonlinear transformations in deep neural network [23]. Tzeng first proposed learning a discriminative representation in the source domain and then a separate encoding that maps the target data to the same space using an asymmetric mapping through a domain adversarial loss [25]. Hoffman proposed a cycle-consistent adversarial domain adaptation model to adapt representations at both pixel-level and feature-level [26]. Although these approaches have achieved reasonably good results, most of them consider only minimising the distribution difference between the source and target domains, they do not consider how to minimise intra-class distance and maximise inter-class distance. Furthermore, if the distribution difference is large and transfer functions can not be explicitly obtained, the above methods cannot effectively transfer knowledge. While deep transfer metric learning techniques have been proposed to overcome the above issues [3], [28], [29], these methods use backward propagation to update the parameters of the deep network, which leads to tremendous computational complexity. For example, French et al. developed a mean teacher model for domain adaptation and designed confidence thresholding and class balancing to achieve state-of-the-art results for variety of benchmarks [30]. Pan et al. proposed a transferrable prototypical network for unsupervised domain adaptation, in which the prototypes for each class in the source and target domains were close in the embedding space, and the score distributions predicted by prototypes separately on source and target data were similar [31]. To extend semi-supervised learning techniques for unsupervised domain adaptation, Zhang et al. proposed an algorithm for label propagation with augmented anchors (A^2LP), which improves label propagation by generating of unlabeled virtual instances with high-confidence label predictions [32]. To address the open-set domain adaptation problem, Pan et al. proposed a self-ensembling with category-agnostic cluster architecture, which steered domain adaptation with the additional guidance of category-agnostic clusters that are specific to the target domain [33]. To reduce the computation cost, You et al. proposed a logarithm of the maximum evidence method to assess pretrained models for transfer learning, which is fast, accurate, and general [34]. Shu et al. proposed a zoo-tuning method to learn to adaptively transfer the parameters of pre-trained models to the target task, which promotes knowledge transfer by simultaneously adapting multiple source models to downstream tasks [35].

However, under a distributional shift in UDA, pseudo-labels can be unreliable because of their large discrepancy from the ground labels. To improve the quality of target pseudo-labels, Liu et al. developed a cycle-self-training algorithm for UDA, which forced pseudo-labels to generalize across domains [36]. Chen et al. proposed closing the domain gap method through orthogonal bases of the representation spaces, which used a new geometrical distance over the representation subspace and learned deep transferable representations by minimizing it [37]. Wang et al. presented a self-tuning method to enable data-efficient deep learning by unifying the exploration of labeled and unlabeled data and the transfer of a pre-trained model, as well as a pseudo-group contrast mechanism to reduce the reliance on pseudo-labels, which boosted the tolerance to false labels [38]. Long et al. designed a distance-based sampling criterion to assign label for each unlabelled sample by its nearest labelled sample, and proposed a self-training semi-supervised deep learning method to train a fault diagnosis model, which exploited a gradually mechanism to increase the number of selected pseudo-labelled candidates [39]. Later, Long et al. used a 2-D edge embedding and a multihead masked attention mechanism to realize a self-adaptation graph structure, which adopted GNN as a meta-learner to make the model exploit well the relationships among the samples in data sets [40].

In this paper, borrowing the idea of deep learning, we propose a novel deep domain adaptation method by learning hierarchical feature representation of domain data and the geodesic distance of curved Riemannian manifold of symmetric positive definite (SPD) with some information transferred from the source domain. This algorithm only use forward propagation to update the parameters of deep network, and solve the task of learning a SPD matrix by formulating it as a smooth, strictly convex optimization problem.

III. DEEP LEARNING NETWORK USING METRIC LEARNING

A. DEEP LEARNING NETWORK USING CASCADED PRINCIPAL COMPONENT ANALYSIS

Denote $\{I_i\}_{i=1}^N$ be N training images of size $m \times n$, $k_1 \times k_2$ be the 2D filter size or patch size at all hierarchical feature learning stages. The proposed model is outlined in Figure 1, and hierarchical features will be learned from the input images $\{I_i\}_{i=1}^N$ in the cascaded principal components analysis (PCA) network, transfer metric learning will be learned from feature space. Next, we present each component of the proposed framework more detail.

Center on each pixel, we extract a $k_1 \times k_2$ patch, and we extract all patches of the ith image $x_{i,1}, x_{i,2}, \ldots, x_{i,mn} \in R^{k_1 \times k_2}$, where $x_{i,j}$ represents the jth vectorized patch of I_i, then we subtract patch mean from each patch and obtain a mean removed patch $\overline{x}_{i,j}$, We construct a matrix $\overline{X}_i = [\overline{x}_{i,1}, \overline{x}_{i,2}, \ldots, \overline{x}_{m,n}]$ for I_i. By constructing the same matrix for all input samples and getting them together, we obtain $X = [\overline{X}_1, \overline{X}_2, \ldots, \overline{X}_N] \in R^{k_1 \times k_2 \times Nmn}$. Suppose that the number of filters in stage i is N_i, the minimization reconstruction loss of PCA within a family of orthonormal filters is as follows:

$$\min_{V \in R^{k_1 k_2 \times N_1}} ||X - VV^T||_F^2, s.t. V^T V = I_{N1}. \tag{1}$$

where I_{N_1} is identity matrix of size $N_1 \times N_1$, the solution is the N_1 principal eigenvectors of XX^T. The PCA filters are define as:

$$W_l^1 = mat_{k_1,k_2}(q_l(XX^T)) \in R^{k_1 \times k_2}, l = 1, 2, \ldots, N_1. \tag{2}$$

where $mat(p)$ is a map function that maps $p \in R^{k_1 \times k_2}$ to a matrix $W \in R^{k_1 \times k_2}$, and $q_l(XX^T)$ is the lth principal eigenvector of XX^T. The main variation of all the mean-removed training patches is characterized by the leading principal eigenvectors. Hence, the lth filter output of the first stage is:

$$I_i^l = I_i * W_l^l, i = 1, 2, \ldots, N. \tag{3}$$

where * represents 2D convolution operation. Next, in order to enhance nonlinear separability, we impose a activation operation to I_i^l, the activation function is ReLu. Then we take all the patches of I_i^l, subtract patch mean from each patch, and get $\overline{Y}_i^l = [\overline{y}_{i,l,1}, \overline{y}_{i,l,2}, \ldots, \overline{y}_{i,l,mn}] \in R^{k_1 k_2 \times mn}$, where $\overline{y}_{i,l,j}$ denotes the jth mean-removed patch in I_i^l. Finally, we obtain the lth filter output of the matrix collecting all mean-removed patches $\overline{Y}^l = [\overline{Y}_{1,l}, \overline{Y}_{2,l}, \ldots, \overline{Y}_{N,l}] \in R^{k_1 k_2 \times Nmn}$, now we concatenate Y^l for all filter outputs as:

$$Y = [Y^1, Y^2, \ldots, Y^{N_1}] \in R^{k_1 k_2 \times N_1 Nmn}. \tag{4}$$

So we obtain the second stage PCA filters

$$W_l^2 = mat_{k_1,k_2}(q_l(YY^T)) \in R^{k_1 \times k_2}, l = 1, 2, \ldots, N_2. \tag{5}$$

For each I_i^l of the second stage, we convolve I_i^l with W_l^2 for $l = 1, 2, \ldots, N_2$ and get N_2 outputs:

$$O_i^l = \{I_i^l * W_l^2\}_{l=1}^{N_2}, i = 1, 2, \ldots, N. \tag{6}$$

Similar to DNN, we can repeat the above process to construct a deeper architecture.

For all these outputs in O_i^l, we use step function to binarize them and get $\{H(I_i^l * W_l^2)\}_{l=1}^{N_2}$. Around each pixel, we convert N_2 binary bits into a decimal number. We convert the N_2 outputs in O_i^l back into a single integer-valued image:

$$T_i^l = \sum_{i=1}^{N_2} 2^{l-1} H(I_i^l * W_i^1), i = 1, 2, \ldots, N. \tag{7}$$

After we get N_1 images T_i^l, we partition each of N_1 images into B block. We compute the histogram in each block, and concatenate all the B histograms into one vector, and denote as Bhist(T_i^l). We obtain the feature of the input images I_i by this encoding process:

$$f_i = [Bhist(T_i^1), \ldots, Bhist(T_i^{N_1})]^T \in R^{(2^{N_2})N_1 B}. \tag{8}$$

Our local blocks are non-overlapping for local histogram extraction. The parameters of cascaded principal component analysis network include the number of filters in each stage N_1, N_2, the filter size k_1, k_2, the number of stages, and the block size for local histogram computation.

B. DEEP DOMAIN ADAPTATION USING METRIC LEARNING

To represents source and target domain datasets in metric learning, we define the two sets as follows:

$$S \subseteq D_s \times D_s = \{(f_i, f_j) | f_i \in D_s, f_j \in D_s\}. \tag{9}$$
$$D \subseteq D_t \times D_t = \{(f_i, f_j) | f_i \in D_t, f_j \in D_t\}. \tag{10}$$

Inspired by geometric mean metric learning (GMML) algorithm [41], we model the distance bewteen similar points in the S set by Mahalanobis distance and the distance bewteen dissimilar points in the D set by the inverse metric, we add their contribution to the overall objective and propose a new objective function:

$$\sum_{(f_i,f_j) \in S} d_A(f_i, f_j) + \sum_{(f_i,f_j) \in D} d_{A^{-1}}(f_i, f_j). \tag{11}$$

where A is a set of symmetric positive definite (SPD) matrices, it forms a Riemannian manifold of nonpositive curvature. we describe (28) using trace and turn it into the optimization problem:

$$\min_{A \succ 0} \sum_{(f_i,f_j) \in S} tr(A(f_i - f_j)(f_i - f_j)^T) + \sum_{(f_i,f_j) \in D} tr(A^{-1}(f_i - f_j)(f_i - f_j)^T). \tag{12}$$

We define the source and target domain covariance matrics M and N:

$$M := \sum_{(f_i,f_j) \in S} (f_i - f_j)(f_i - f_j)^T. \tag{13}$$
$$N := \sum_{(f_i,f_j) \in D} (f_i - f_j)(f_i - f_j)^T. \tag{14}$$

We define the optimization problem (28) using M and N,namely

$$L = \min_{A \succ 0} \sum_{(f_i,f_j) \in S} tr(AM) + \sum_{(f_i,f_j) \in D} tr(A^{-1}N). \tag{15}$$

Since L is both strictly convex and strictly geodesically convex, therefore, if $\nabla L = 0$, equation (28) has a solution, and the solution will be the global optimal. So we differentiate L with respect to A, we get

$$\nabla L = M - A^{-1}NA^{-1} = 0. \tag{16}$$
$$AMA = N. \tag{17}$$

Hence, equation (28) is a Riccati equation, whose solution is the midpoint of the geodesic joining M^{-1} to N, where length is measured on the Riemannian manifold of SPD matrices. namely

$$A = M^{-1}\sharp_{\frac{1}{2}}N = M^{-\frac{1}{2}}(M^{\frac{1}{2}}NM^{\frac{1}{2}})^{\frac{1}{2}}M^{-\frac{1}{2}}. \tag{18}$$

where $\sharp_{\frac{1}{2}}$ is the geometric mean of SPD matrices [41].

To exploit the multiple source and target data of the different distribution, we can use the maximum mean discrepancy metric (MMD) [42] to incorporate the statistical alignment constraint.

$$||\frac{1}{n}\sum_{i=1}^{n} Wf_i^s - \frac{1}{m}\sum_{i=1}^{m} Wf_i^t||^2 = tr(AXLX^T). \tag{19}$$

where $X = \{f_1^s, f_2^s, \ldots, f_n^s, f_1^t, f_2^t, \ldots, f_m^t\} \in R^{m+n}$, $L \in R^{(m+n)\times(m+n)}$, L is a symmetric positive-semidefinite matrix, if $f_i, f_j \in X_s$, $L(i,j) = \frac{1}{n^2}$, if $f_i, f_j \in X_t$, $L(i,j) = \frac{1}{m^2}$, otherwise, $L(i,j) = -\frac{1}{mn}$. We add the MMD objective to the previous overall objective in equation (28) and obtain the following manifold objective function:

$$L = \min_{A \succ 0} tr(AM) + tr(A^{-1}N) + tr(AXLX^T). \tag{20}$$

We differentiate once again L with respect to A, we will obtain the solution of the modified objective function:

$$\nabla L = M - A^{-1}NA^{-1} + XLX^T 0. \tag{21}$$
$$A = M_m^{-1}\sharp_{\frac{1}{2}}N. \tag{22}$$

where $M_m = M + XLX^T$.

To model the nonlinear manifold geometry of the source and target domains, geometrical constraints can be imposed on the solution. We model the source and target domains as a nonlinear manifold, and use a random walk on a nearest neighbor graph connecting nearby points to build a diffusion on a discrete graph approximation of the continuous manifold. asymptotic convergence of the graph Laplacian to the underlying manifold Laplacian has been shown for established standard results [43]. We exploit the above method to find two graph kernels, K_s and K_t using the eigenvectors of the random walk on the source and target domain manifolds, respectively.

$$K_s = \sum_{i=1}^{m} e^{-\frac{-\sigma_s^2}{2\lambda_i^s}} v_i^s(v_i^s)^T. \tag{23}$$
$$K_t = \sum_{i=1}^{n} e^{-\frac{-\sigma_t^2}{2\lambda_i^t}} v_i^t(v_i^t)^T. \tag{24}$$

where v_i^s and v_i^t are the eigenvectors of the random walk diffusion matrix on the source and target manifolds, respectively, λ_i^s and λ_i^t are the corresponding eigenvalues.

Now we integrate the source and target domain manifold geometry into a new objective function:

$$L = \min_{A \succ 0} tr(AM) + tr(A^{-1}N) + tr(AX(K + \mu L)X^T). \tag{25}$$

where $K \succ 0$, $K = \begin{pmatrix} K_s^{-1} & 0 \\ 0 & K_t^{-1} \end{pmatrix}$, and μ is a weighting term that combines the geometric and statistical constraints over A.

Now differentiating L with respect to A once again, we obtain the solution to equation (28) $A = M_{gs}\sharp_{\frac{1}{2}}N$, where $M_{gs} = M + X(K + \mu L)X^T$.

When we compute the solution of equation (28), the geodesic viewpoint is important because it decides how to assign different weight to the matrics M and N. Hence, we should refine the equation (28) using weighted geometric mean. To explain the idea of weighted geometric mean, for two SPD matrices M and N, we introduce the following Riemannian distance metric on the nonlinear manifold of SPD matrices.

$$\delta_R^2(X, Y) := ||log(Y^{-\frac{1}{2}}XY^{-\frac{1}{2}})||_F^2. \tag{26}$$

where $\| \bullet \|_F$ denotes the Frobenius norm. Using this metric, we can generalize the equation (28) to the weighted form. We import a parameter that characterizes the degree of balance between the cost terms of the source and target domain data. The weighted formulation is then

$$L = \min_{A \succ 0} L_t(A) = t\delta_R^2(A, N) + (1-t)\delta_R^2(A, (M + X(K + \mu L)X^T))^{-1}. \tag{27}$$

The unique solution to equation (28) is $M_{gs}^{-1}\sharp_t N$, where $M_{gs} = M + X(K + \mu L)X^T$, $\sharp_t$ denotes a geodesic between two points in a manifold. For the SPD manifold, the geodesic $\gamma(t)$ for a weight scalar $0 \leq t \leq 1$ between M^{-1} and N is defined by (28):

$$\gamma(t) = M^{-\frac{1}{2}}(M^{\frac{1}{2}}NM^{\frac{1}{2}})^t M^{-\frac{1}{2}}. \tag{28}$$

IV. DEEP DOMAIN ADAPTATION ALGORITHM

We present the proposed deep domain adaptation algorithm in Algorithm 1. The algorithm is based on learning deep feature representations from source and target domain datasets, computing the Mahalanobis distance matrix A depending on the source and target covariances, incorporating a MMD metric, and combining the source and target manifold geometry. Thus, we ensure the uniqueness and optimality of the proposed algorithm by modeling the Riemannian manifold underlying SPD matrices.

V. EXPERIMENTS

We compare the proposed model to state-of-the-art transfer learning methods on unsupervised domain adaptation problems. We focus on the effectiveness of domain adaptation using metric learning.

Algorithm 1 Algorithm for Deep Domain Adaptation Using Cascaded PCANet and Metric Learning

Input:

The set of source domain data, D_s, with labels $L_S = \{y_i\}$, and the set of unlabelled target domain data, D_t;
Parameters: step length of geodesic t, hyperparameters parameter μ;

Output:

Distance matrix A, Use the learned A matrix to transfer source features to the target domain, and perform classification;

1: Do forward propagation to all data points;
2: Compute the feature of all data points by (28);
3: Compute the source and target matrices M and N by (28) and (28);
4: obtain MMD term by (28);
5: Compute the weighted geometric mean taking into account the MMD term by (28);
6: Compute the weighted geometric mean taking into account the source and target manifold geometry by (28);
7: Compute the cascaded weighted geometric mean taking into account the source and target manifold geometry by (28);
8: **return** Return distance matrix $A = M_{gs}^{-1}\sharp_t N$;

A. SETUP

This evaluation was conducted on two public datasets: Office-31 [44], Office+Caltech [45], and VisDA [46].

1) OFFICE-31

This dataset is a evaluative benchmark for domain adaptation. It comprises 4652 images in 31 classes collected from three different domains: Amazon (A), which contains images downloaded from amazon.com, Webcam (W) and DSLR (D), which contain images taken by a web camera and digital DSLR camera in an office with different settings, respectively. For completeness, we evaluated the proposed method across six transfer tasks, A $\rightarrow$ W, D $\rightarrow$ W, W $\rightarrow$ D, A $\rightarrow$ D, D $\rightarrow$ A, and W $\rightarrow$ A.

2) OFFICE-10+CALTECH-10

This dataset is built by using Caltech-256 as source domain and the 3 domains in Office 31 as target domain. We used the 10 categories shared by Office 31 and Caltech-256 and selected images of the ten categories in each domain of Office 31 as the target domain. We evaluated our method across six transfer tasks, A $\rightarrow$ C, W $\rightarrow$ C, D $\rightarrow$ C, C $\rightarrow$ A, C $\rightarrow$ W and C $\rightarrow$ D.

3) VisDA

VisDA is the largest dataset for challenging domain adaptation, containing over 280k images from the training, validation and testing domains. All the three domains included the same 12 categories. The training domain includes 152k synthetic images generated by 3D CAD model of the same categories from different angles and under different lighting conditions. The validation domain consists of 55k real images from COCO [47]. The testing domain contains 72k images from video frames in YTBB [48]. We evaluate the accuracy and accuracy averaged over all known and unknown of all the 12 categories for adaptation.

4) COMPARED METHODS

We compare the performance of our method with state of the art transfer learning and deep learning methods: Transfer component anaysis (TCA) [49] is a classical transfer learning method based on MMD regularized PCA. The geodesic flow kernel (GFK) [34] is a widely evaluative method for our datasets that transforms intermediate representation to bridge the distribution discrepancy between the source and target. Convolutional neural network (CNN) [50] is a good method which turns out a strong model for transferable representation. A deep adaptation network (HAN) [21] embeds hidden representations of all task-specific layers in a reproducing kernel Hilbert space where different domain distributions optimally using multi-kernel MMD can be explicitly matched. Gradient reversal (RevGrad) [23] is a new unsupervised domain adaptation method that uses backward propagation training. Deep CORAL [22] is an extended CORAL method for learning nonlinear transformations in deep networks. Adversarial discriminative domain adaptation (ADDA) [25] firstly learn a discriminative representation in the source domain, then learn a separate encoding that maps the target data to the same space using an asymmetric mapping through a domain adversarial loss. Cycle-consistent adversarial domain adaptation (CyCADA) [26] is a excellent method that adapts representations at both pixel-level and feature-level. Self-ensembling for domain adaptation (SE) [30] is an effective algorithm that uses a self-ensembling mechanism. Self-ensembling with category-agnostic clusters (SE-CC) [33] is a novel architecture that leads to domain adaptation under the guidance of category-agnostic clusters in the target domain. Transferrable prototypical networks (TPN) [31] are frameworks of prototypical networks for both general-purpose and task-specific adaptation. Label propagation with augmented anchors (A^2LP) [32] is an improved label propagation algorithm that generates unlabeled virtual instances. Cycle self-training (CST) [36] is a principled self-training algorithm that forces pseudo-labels to generalize across domains.

We followed standard evaluation protocols for unsupervised transfer learning and exploited all labeled source examples and all unlabeled target examples. The average classification accuracy of each transfer task was compared using three random experiments. For baseline methods, we followed the standard procedures to select the model, as explained in their respective papers. For MMD-based methods (i.e., TCA, DAN and our method), we used Gaussian

TABLE 1. Classification accuracy on Office-31 dataset for unsupervised adaptation.

Method	A → W	D → W	W → D	A → D	D → A	W → A	Avg
TCA	58.9	90.2	88.1	57.6	51.7	47.8	65.7
GFK	58.4	93.5	91.1	58.7	52.2	46.0	66.7
CNN	62.0	95.5	99.0	64.2	51.5	50.3	70.4
DAN	68.5	96.0	98.9	66.8	50.4	49.9	71.8
D-CORAL	66.6	95.8	99.2	67.1	52.8	51.6	72.2
RevGrad	73.4	96.6	99.3	67.4	53.5	52.2	73.7
ADDA	73.8	96.7	99.1	71.9	54.8	53.6	75.0
CyCADA	79.2	99.3	99.8	78.4	61.1	59.9	79.6
SE	87.6	82.9	83.2	92.7	93.4	91.5	88.6
TPN	83.2	89.5	90.8	85.3	73.6	73.2	82.6
A^2LP	87.8	98.1	99.1	87.9	75.8	75.9	87.4
SE-CC	85.2	97.7	99.4	85.4	87.9	86.9	90.4
OURS	86.2	95.4	98.2	84.6	80.8	80.1	87.6

TABLE 2. Classification accuracy on Office-Caltech dataset for unsupervised adaptation.

Method	A → C	W → C	D → C	C → A	C → W	C → D	Avg
TCA	42.6	34.1	35.5	54.6	50.4	50.3	44.6
GFK	41.4	26.5	36.2	56.3	43.6	42.0	41.0
CNN	84.0	76.3	81.1	91.2	83.4	89.2	84.2
DAN	86.3	81.6	82.2	92.1	92.3	90.6	87.5
D-CORAL	87.6	82.9	83.2	92.7	93.4	91.5	88.6
RevGrad	88.4	83.3	83.8	93.5	94.0	93.8	89.5
ADDA	89.5	84.2	84.9	94.6	94.8	94.3	90.4
CyCADA	90.4	85.3	85.6	95.2	95.4	95.9	91.3
SE	90.6	88.9	88.4	96.6	96.8	97.0	93.1
TPN	90.8	89.5	89.6	93.3	94.0	94.6	91.9
A^2LP	89.5	90.2	90.4	94.9	95.1	96.3	92.7
SE-CC	92.6	92.8	91.0	97.2	97.8	98.6	95.0
OURS	89.3	89.9	90.2	94.1	95.3	96.0	92.5

TABLE 3. Classification accuracy on VisDA dataset for unsupervised adaptation.

Method	aero	bike	bus	car	horse	knife	mbike	person	plant	skbrd	train	truck	Avg
SE	96.3	87.8	84.3	66.5	96.1	96.2	90.4	81.6	95.4	91.5	87.4	51.5	85.3
TPN	93.7	85.2	69.2	81.5	93.6	61.8	89.3	81.4	93.6	81.6	84.4	49.9	80.4
A^2LP	95.5	82.9	77.8	77.1	95.2	96.0	86.5	65.4	87.4	42.8	86.5	53.2	78.6
SE-CC	96.3	86.5	82.5	81.2	96.1	97.2	91.3	84.6	94.4	94.1	88.3	53.5	87.1
CST	96.0	82.3	78.4	72.0	93.1	95.5	88.7	62.6	89.2	35.9	85.7	54.2	77.8
OURS	96.1	83.0	78.8	91.5	94.2	93.8	89.7	67.4	90.5	43.3	87.4	50.8	80.6

kernel with bandwidth c to set the median pairwise squared distances on the training data. For all methods, we executed standard cross-validation on the labeled source data to choose their hyper-parameters. For our method, the parameters of the cascaded principal component analysis network are set as follows: the filter size of the networks is $k_1 = k_2 = 5$, the block size 8×6, and the number of filters is $N_1 = N_2 = 5$. The hyperparameter t of cascaded metric learning is set to 0.6 for all domain adaptation tasks.

B. RESULTS AND DISCUSSION

1) PERFORMANCE COMPARISON

We show the unsupervised adaptation results for the first six Office-31 transfer tasks in Table 1, and the results for the other six Office-10 + Caltech-10 transfer tasks in Table 2. From these results, we can observe that our method outperforms most of the compared methods on most transfer tasks, but slightly falls behind the SE and SE-CC methods on some tasks, and achieves comparable performance on the easy transfer tasks, D → W, and W → D, where the source and target domains are similar. This is reasonable because domain adaptability may vary across different transfer tasks.

A performance comparison of VisDA for domain adaptation is presented in Table 3. The proposed method performs consistently better than most of the compared methods, but slightly falls behind the SE and SE-CC methods on the mean accuracy and some tasks. In particular, the mean accuracy averaged over all 12 classes can achieve 80.6%, making an absolute improvement over TPN by 0.2%.

The performance improvement demonstrates that our architecture of deep domain adaptation via cascaded principal component analysis networks and cascaded metric learning can transfer domain knowledge across different domains.

We attemp to use a very deep cascaded learning network to learn more transferable features. However, when the number of network layers exceeds three, the performance improvement is trivial.

From the experimental results, we can make the following observations. (1) Deep transfer learning methods

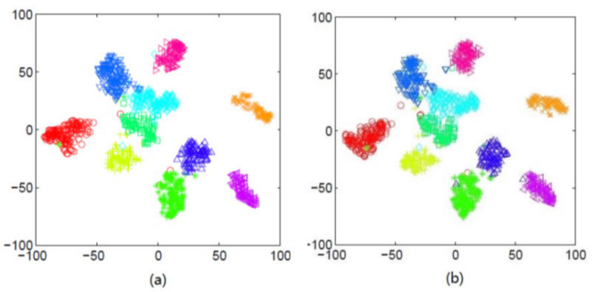


FIGURE 2. The t-SNE visualization of feature learnt by the proposed method on (a) Source, (b) Target of Office-10+Caltech-10.

significantly outperform traditional shallow transfer learning methods by a large margin. (2) Deep transfer learning methods that reduce the distribution discrepancy between the source and target domain data using deep adaptation network outperform the standard CNN and conventional shallow transfer learning methods. (3) In contrast to all the previous deep transfer learning methods in which transferable features are implemented by forward propagation and back propagation, the proposed transfer network only use forward propagation to learn transferable features, which can significantly speed up deep transfer learning and substantially boost the domain adaptation performance.

2) FEATURE VISUALIZATION

We visualize the features learnt by the proposed method with the t-SNE [51] on source and target of Office-10+Caltech-10 in Figure 2(a)-(b). Through domain adaptation by the proposed method, the distributions of two domains are brought closer, making the target distribution similar to the source one.

VI. CONCLUSION

In this study, we propose a new deep domain adaptation approach to bridge the distribution discrepancy between the source and target domain data. The cascaded deep network does not require regularized parameters and backward propagation, which consist of only a cascaded linear map and a nonlinear output, this simplicity and tremendous speedup can be an alternative framework to deep CNN. Using the geodesic distance between the source and target domain data covariances, we built the nonlinear Riemannian geometry of the symmetric positive definite matrices (SPDs), which enables the integration of geometry and statistical information by introducing the Riccati equation, which simplifies domain adaptation computation because the Riccati equation is a mathematically elegant solution to the domain adaptation problem. A cascaded weighted geometric mean strategy further improves domain adaptation effectiveness. An extensive experimental evaluations of the standard domain adaptation benchmarks show that the proposed method is effective.

CONFLICTS OF INTEREST

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

REFERENCES

[1] Y. Cui, F. Zhou, Y. Lin, and S. Belongie, "Fine-grained categorization and dataset bootstrapping using deep metric learning with humans in the loop," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2016, pp. 1153–1162.

[2] J. Wang, Y. Song, T. Leung, C. Rosenberg, J. Wang, J. Philbin, B. Chen, and Y. Wu, "Learning fine-grained image similarity with deep ranking," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2014, pp. 1386–1393.

[3] D. Yi, Z. Lei, S. C. Liao, and S. Z. Li, "Deep metric learning for person re-identification," in *Proc. ICPR*, Aug. 2014, pp. 34–39.

[4] J. Hu, J. Lu, and Y.-P. Tan, "Discriminative deep metric learning for face verification in the wild," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2014, pp. 1875–1882.

[5] J. Hu, J. Lu, and Y.-P. Tan, "Deep metric learning for visual tracking," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 26, no. 11, pp. 2056–2068, Nov. 2016.

[6] Y. Bengio, A. Courville, and P. Vincent, "Representation learning: A review and new perspectives," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 35, no. 8, pp. 1798–1828, Aug. 2013.

[7] J. Yosinski, J. Clune, Y. Bengio, and H. Lipson, "How transferable are features in deep neural networks?" in *Proc. Adv. Neural Inf. Process. Syst.*, 2014, pp. 3320–3328.

[8] 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," 2013, *arXiv:1310.1531*.

[9] L.-C. Chen, G. Papandreou, I. Kokkinos, K. Murphy, and A. L. Yuille, "DeepLab: Semantic image segmentation with deep convolutional nets, atrous convolution, and fully connected CRFs," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 40, no. 4, pp. 834–848, Apr. 2018.

[10] F. Chollet, "Xception: Deep learning with depthwise separable convolutions," 2016, *arXiv:1610.02357*.

[11] K. Greff, R. K. Srivastava, J. Koutnìk, B. R. Steunebrink, and J. Schmidhuber, "LSTM: A search space Odyssey," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 28, no. 10, pp. 2222–2232, Oct. 2017.

[12] X. Pan, P. Luo, J. Shi, and X. Tang, "Two at once: Enhancing learning and generalization capacities via IBN-net," 2018, *arXiv:1807.09441*.

[13] S. Xie, R. Girshick, P. Dollar, Z. Tu, and K. He, "Aggregated residual transformations for deep neural networks," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jul. 2017, pp. 5987–5995.

[14] J. Hu, L. Shen, S. Albanie, G. Sun, and E. Wu, "Squeeze-and-excitation networks," 2017, *arXiv:1709.01507*.

[15] G. Huang, Z. Liu, L. Van Der Maaten, and K. Q. Weinberger, "Densely connected convolutional networks," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jul. 2017, pp. 4700–4708.

[16] I. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. Courville, and Y. Bengio, "Generative adversarial nets," in *Proc. Adv. Neural Inf. Process. Syst.*, 2014, pp. 2672–2680.

[17] A. Radford, L. Metz, and S. Chintala, "Unsupervised representation learning with deep convolutional generative adversarial networks," 2015, *arXiv:1511.06434*.

[18] M. Arjovsky, S. Chintala, and L. Bottou, "Wasserstein generative adversarial networks," in *Proc. Int. Conf. Mach. Learn.*, 2017, pp. 214–223.

[19] F. Schroff, D. Kalenichenko, and J. Philbin, "FaceNet: A unified embedding for face recognition and clustering," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2015, pp. 815–823.

[20] W. Li, J. Huo, Y. Shi, Y. Gao, L. Wang, and J. Luo, "Online progressive deep metric learning," 2018, *arXiv:1805.05510*.

[21] M. Long, Y. Cao, J. Wang, and M. I. Jordan, "Learning transferable features with deep adaptation networks," 2015, *arXiv:1502.02791*.

[22] Y. Ganin and V. Lempitsky, "Unsupervised domain adaptation by back propagation," in *Proc. ICML*, 2015, pp. 1180–1189.

[23] B. Sun and K. Saenko, "Deep coral: Correlation alignment for deep domain adaptation," in *Proc. Eur. Conf. Comput. Vis.* Cham, Switzerland: Springer, 2016, pp. 443–450.

[24] 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, 2016.

[25] E. Tzeng, J. Hoffman, K. Saenko, and T. Darrell, "Adversarial discriminative domain adaptation," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jul. 2017, pp. 7167–7176.

[26] J. Hoffman, E. Tzeng, T. Park, J.-Y. Zhu, P. Isola, K. Saenko, A. A. Efros, and T. Darrell, "CyCADA: Cycle-consistent adversarial domain adaptation," 2017, *arXiv:1711.03213*.

[27] B. Sun, J. Feng, and K. Saenko, "Return of frustratingly easy domain adaptation," in *Proc. AAAI*, 2016, pp. 1–8.

[28] J. Hu, J. Lu, Y.-P. Tan, and J. Zhou, "Deep transfer metric learning," *IEEE Trans. Image Process.*, vol. 25, no. 12, pp. 5576–5588, Dec. 2016.

[29] M. Geng, Y. Wang, T. Xiang, and Y. Tian, "Deep transfer learning for person re-identification," 2016, *arXiv:1611.05244*.

[30] G. French, M. Mackiewicz, and M. Fisher, "Self-ensembling for visual domain adaptation," 2017, *arXiv:1706.05208*.

[31] Y. Pan, T. Yao, Y. Li, Y. Wang, C.-W. Ngo, and T. Mei, "Transferrable prototypical networks for unsupervised domain adaptation," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2019, pp. 2239–2247.

[32] Y. Zhang, B. Deng, K. Jia, and L. Zhang, "Label propagation with augmented anchors: A simple semi-supervised learning baseline for unsupervised domain adaptation," in *Proc. Eur. Conf. Comput. Vis.* Cham, Switzerland: Springer, 2020, pp. 781–797.

[33] Y. Pan, T. Yao, Y. Li, C.-W. Ngo, and T. Mei, "Exploring category-agnostic clusters for open-set domain adaptation," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2020, pp. 13867–13875.

[34] K. You, Y. Liu, J. Wang, and M. Long, "LogME: Practical assessment of pre-trained models for transfer learning," in *Proc. Int. Conf. Mach. Learn.*, 2021, pp. 12133–12143.

[35] Y. Shu, Z. Kou, Z. Cao, J. Wang, and M. Long, "Zoo-tuning: Adaptive transfer from a zoo of models," in *Proc. Int. Conf. Mach. Learn.*, 2021, pp. 9626–9637.

[36] H. Liu, J. Wang, and M. Long, "Cycle self-training for domain adaptation," in *Proc. Adv. Neural Inf. Process. Syst.*, vol. 34, 2021, pp. 22968–22981.

[37] X. Chen, S. Wang, J. Wang, and M. Long, "Representation subspace distance for domain adaptation regression," in *Proc. ICML*, 2021, pp. 1749–1759.

[38] X. Wang, J. Gao, M. Long, and J. Wang, "Self-tuning for data-efficient deep learning," in *Proc. Int. Conf. Mach. Learn.*, 2021, pp. 10738–10748.

[39] J. Long, Y. Chen, Z. Yang, Y. Huang, and C. Li, "A novel self-training semi-supervised deep learning approach for machinery fault diagnosis," *Int. J. Prod. Res.*, pp. 1–14, Feb. 2022.

[40] J. Long, R. Zhang, Z. Yang, Y. Huang, Y. Liu, and C. Li, "Self-adaptation graph attention network via meta-learning for machinery fault diagnosis with few labeled data," *IEEE Trans. Instrum. Meas.*, vol. 71, pp. 1–11, 2022.

[41] P. Zadeh, R. Hosseini, and S. Sra, "Geometric mean metric learning," in *Proc. Int. Conf. Mach. Learn.*, 2016, pp. 2464–2471.

[42] K. M. Borgwardt, A. Gretton, M. J. Rasch, H.-P. Kriegel, B. Scholkopf, and A. J. Smola, "Integrating structured biological data by kernel maximum mean discrepancy," *Bioinformatics*, vol. 22, no. 14, pp. 49–57, Jul. 2006.

[43] M. Belkin and P. Niyogi, "Convergence of Laplacian eigenmaps," in *Proc. Adv. Neural Inf. Process. Syst.*, 2007, pp. 129–136.

[44] K. Saenko, B. Kulis, M. Fritz, and T. Darrell, "Adapting visual category models to new domains," in *Proc. Eur. Conf. Comput. Vis.* Berlin, Germany: Springer, 2010, pp. 213–226.

[45] B. Gong, Y. Shi, F. Sha, and K. Grauman, "Geodesic flow kernel for unsupervised domain adaptation," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2012, pp. 2066–2073.

[46] X. Peng, B. Usman, K. Saito, N. Kaushik, J. Hoffman, and K. Saenko, "Syn2Real: A new benchmark for synthetic-to-real visual domain adaptation," 2018, *arXiv:1806.09755*.

[47] T.-Y. Lin, M. Maire, S. Belongie, J. Hays, P. Perona, D. Rammnan, P. Dollar, and C. L. Zitnick, "Microsoft COCO: Common objects in context," in *Proc. ECCV*, 2014, pp. 740–755.

[48] E. Real, J. Shlens, S. Mazzocchi, X. Pan, and V. Vanhoucke, "YouTube-BoundingBoxes: A large high-precision human-annotated data set for object detection in video," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jul. 2017, pp. 5296–5305.

[49] 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.

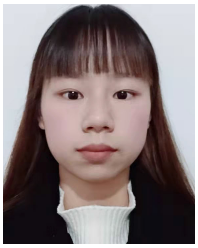
[50] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "ImageNet classification with deep convolutional neural networks," in *Proc. Adv. Neural Inf. Process. Syst. (NIPS)*, 2012, pp. 1097–1105.

[51] L. V. D. Maaten and G. Hinton, "Visualizing data using t-SNE," *J. Mach. Learn. Res.*, vol. 9, no. 11, pp. 1–27, 2008.

ZHIYONG ZENG received the B.S. degree in chemistry from Gannan Normal University, Ganzhou, in 1987, the M.S. degree in chemistry from Fuzhou University, Fuzhou, in 1998, and the Ph.D. degree in computer application technology from Xidian University, Xi'an, in 2006.

From 2007 to 2008, he was an Assistant Professor with the Faculty of Software, Fujian Normal University. Since 2008, he has been an Associate Professor with the College of Computer and Cyber Security, Fujian Normal University. He is the author of more than 50 articles, and more than five inventions. He holds one patents. His research interests include digital image processes and applications, computer vision, artificial intelligence, and machine learning.

DAWEI LI received the B.E. degree in computer science and technology from Shijiazhuang Tiedao University, in 2020. He is currently pursuing the M.S. degree in computer science and engineering with Fujian Normal University. His research interests include computer vision and pattern recognition.



XIUJUAN YANG received the bachelor's degree in network engineering from the Anhui University of Technology, China, in 2020. She is currently pursuing the master's degree with the School of Computer and Cyberspace Security, Fujian Normal University. Her research interests include object detection, image processing, pattern recognition, and artificial intelligence.