Blind Domain Adaptation With Augmented Extreme Learning Machine Features

Muhammad Uzair and Ajmal Mian, *Member, IEEE*

***Abstract*—In practical applications, the test data often have different distribution from the training data leading to suboptimal visual classification performance. Domain adaptation (DA) addresses this problem by designing classifiers that are robust to mismatched distributions. Existing DA algorithms use the unlabeled test data from target domain during training time in addition to the source domain data. However, target domain data may not always be available for training. We propose a blind DA algorithm that does not require target domain samples for training. For this purpose, we learn a global nonlinear extreme learning machine (ELM) model from the source domain data in an unsupervised fashion. The global ELM model is then used to initialize and learn class specific ELM models from the source domain data. During testing, the target domain features are augmented with the reconstructed features from the global ELM model. The resulting enriched features are then classified using the class specific ELM models based on minimum reconstruction error. Extensive experiments on 16 standard datasets show that despite blind learning, our method outperforms six existing state-of-the-art methods in cross domain visual recognition.**

***Index Terms*—Blind domain adaptation (DA), extreme learning machines (ELMs), object recognition, visual classification.**

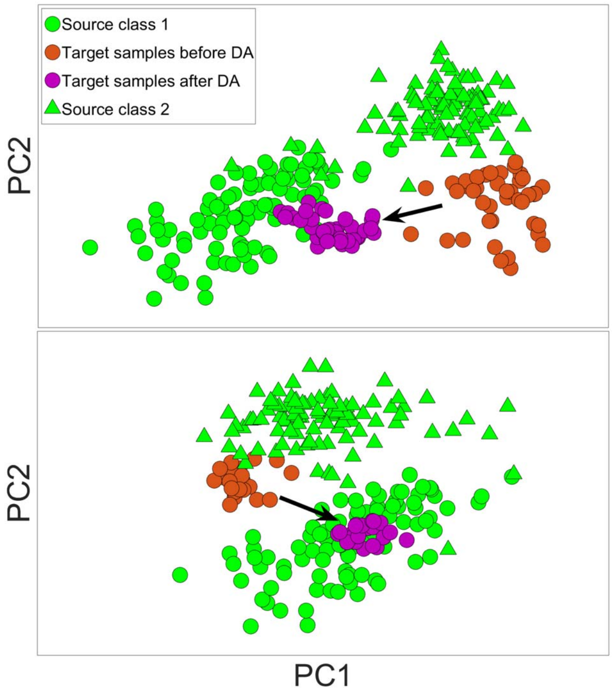


Fig. 1. Two real examples from the Office dataset [33]. The target domain samples belong to class 1 but are closer to class 2 due to the distribution mismatch between the domains. The proposed blind DA shifts the target samples closer to the correct class (better visualized in color).

I. Introduction

Conventional classification algorithms rely on the assumption that the training and test data come from the same distribution. However, this assumption is challenged by many real world computer vision applications [34]. For example, a face recognition system trained on high resolution laboratory images may be applied to recognize low resolution and noisy surveillance images. Similarly, an object recognition system trained on images downloaded from Internet may be deployed to recognize objects in personal photo-collections [1], [33]. In these situations, the magnitude of interclass differences is overshadowed by the magnitude of distribution shift between training and test datasets. Hence, a classifier designed without paying attention to the distribution mismatch is unlikely to give good performance (see Fig. 1). Therefore, it is important to design classifiers that are robust to the mismatch between distributions. This problem, generally referred to as domain adaptation (DA), has recently drawn significant attention from the computer vision community [6], [9], [10], [23], [25], [28], [30], [33].

DA aims to decrease the difference between domains and learn classifiers that are robust to mismatched distributions [10]. Learning is generally performed by utilizing plenty of labeled source domain data and limited target domain data. DA methods can be classified into two categories: 1) semi-supervised and 2) unsupervised. In the semi-supervised setting, the training data consists of a large number of labeled source domain data and limited labeled target domain data. In the unsupervised case, training data consists of labeled source domain data and unlabeled target domain data. The unsupervised case is more challenging but, at the same time, more representative of real-world scenarios.

Existing unsupervised and semi-supervised DA methods have limitations as they cannot be applied in scenarios where a large amount of target domain data is not available for training.

Manuscript received September 10, 2015; revised December 10, 2015; accepted January 26, 2016. This work was supported by the Australian Research Council Discovery under Grant DP110102399. This paper was recommended by Associate Editor X. Wang.

The authors are with the School of Computer Science and Software Engineering, University of Western Australia, Crawley, WA 6009, Australia (e-mail: muzair27@gmail.com; ajmal.mian@uwa.edu.au).

Color versions of one or more of the figures in this paper are available online at http://ieeexplore.ieee.org.

Digital Object Identifier 10.1109/TCYB.2016.2523538

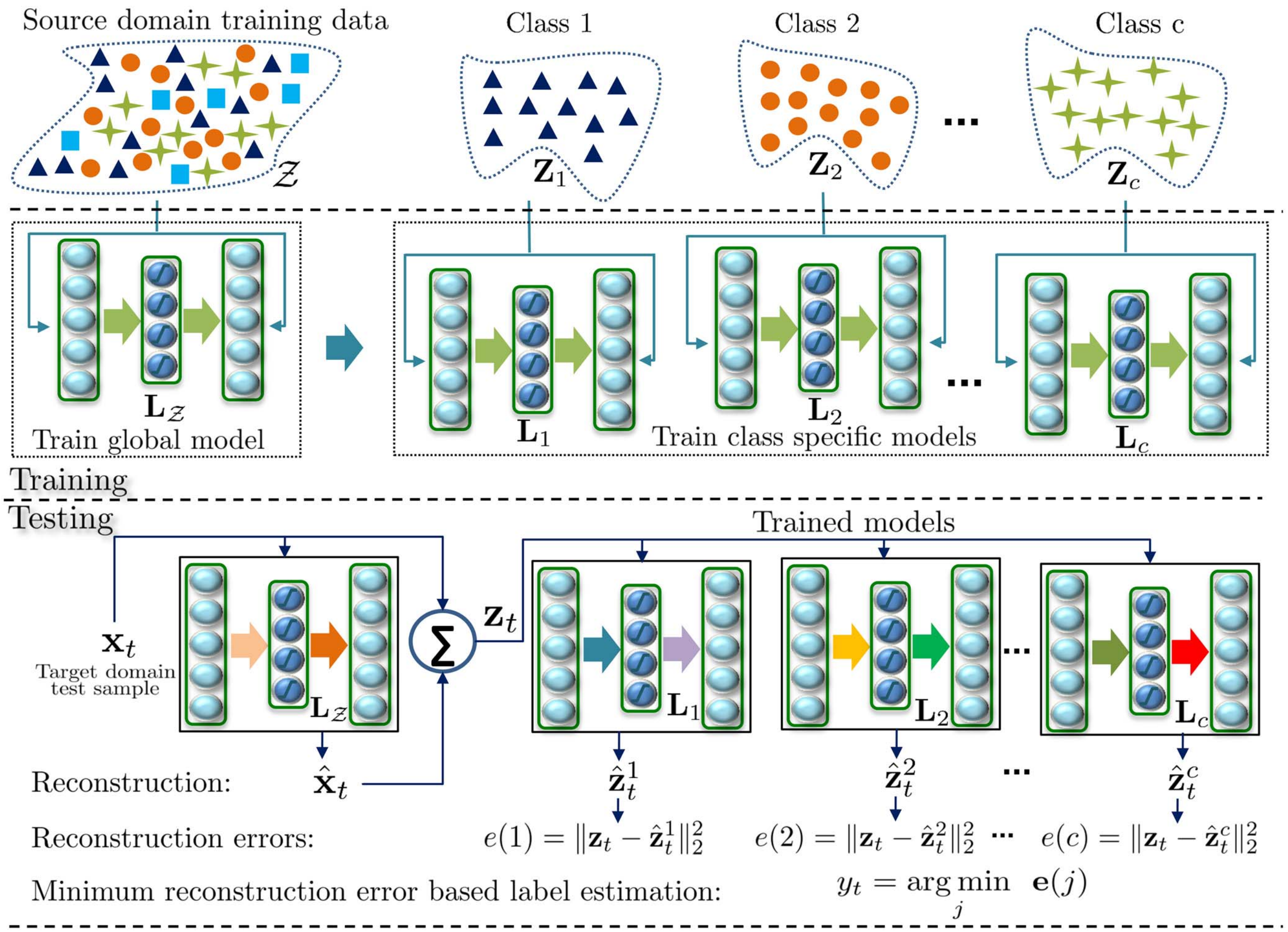

Source domain training data
$\mathcal{Z}$
Class 1
$\mathbf{Z}_1$
Class 2
$\mathbf{Z}_2$
Class c
$\mathbf{Z}_c$
$\mathbf{L}_{\mathcal{Z}}$
Train global model
$\mathbf{L}_1$
$\mathbf{L}_2$
Train class specific models
$\mathbf{L}_c$
Training
Testing
Trained models
$\mathbf{x}_t$
Target domain test sample
$\mathbf{z}_t$
Reconstruction:
$\hat{\mathbf{x}}_t$
$\hat{\mathbf{z}}_t^1$
$\hat{\mathbf{z}}_t^2$
$\hat{\mathbf{z}}_t^c$
Reconstruction errors:
$e(1) = \|\mathbf{z}_t - \hat{\mathbf{z}}_t^1\|_2^2$
$e(2) = \|\mathbf{z}_t - \hat{\mathbf{z}}_t^2\|_2^2$
$e(c) = \|\mathbf{z}_t - \hat{\mathbf{z}}_t^c\|_2^2$
Minimum reconstruction error based label estimation:
$y_t = \arg\min_j \ \mathbf{e}(j)$

Fig. 2. Illustration of the AELM algorithm. During training, two types of ELM models are learned. A global domain specific model using all source domain training samples and local class specific models for each source class. During testing, a target domain test sample is augmented with its reconstructed version from the global ELM model. Finally, the label of the test sample is estimated by the smallest reconstruction error from the class specific models.

For example, the subspace-based methods minimize the discrepancy between the source and target subspaces by interpolating intermediate subspaces such that they model the shift from the source to target subspaces [9], [10] or by aligning the source and target subspaces [6]. Therefore, if the target subspace is not accurately estimated due to under-sampling of the target domain, the performance of these methods will suffer. Moreover, these methods use linear subspaces to represent data which are unable to encode the nonlinear data structure. Similarly, methods [23], [25], [30] that explicitly try to reduce the distribution divergence between source and target domains usually adopt the empirical maximum mean discrepancy [11] as the nonparametric distance measure to compare different distributions. Sufficiently large amount of target domain data is required in this case as well in order to accurately estimate the probability distribution of the target data.

In this paper, we focus on the problem of DA where data from the target domain is not available during training time. This problem is also known as blind DA [21]. We propose a method to reduce the distribution shift implicitly by transferring source specific knowledge to target samples through feature augmentation. This is illustrated in Fig. 1 by two examples from the Amazon versus DSLR cross domain visual recognition task in the Office [33] dataset. Fig. 2 shows the block diagram of the proposed algorithm. Our method learns the nonlinear structure of the source domain using multiple parameters. This rich modeling of the structure in multiple parameters can unfold multiple factors of the source data that are useful for an accurate representation of the target domain. A target sample represented with the proposed nonlinear model is likely to encode both class specific as well as source specific knowledge. This representation is then used to enrich the original target sample by feature augmentation. For classification, we first learn class specific nonlinear models from the source classes and then reconstruct the enriched target sample using these models. Finally, the class labels are estimated based on the smallest reconstruction error.

The main contributions of this paper are: 1) a blind DA algorithm that does not require even a single sample from the target domain for training and 2) an effective strategy of feature augmentation to improve the representation of the target domain sample using source domain models. The proposed algorithm is extensively evaluated on 16 standard datasets and compared to three baseline and six state-of-the art

DA methods. Our results show that, even though our method performs blind learning, it outperforms existing state-of-the-art methods that perform nonblind learning, i.e., use target domain data for training.

II. Related Work

DA is an active research area in computer vision [1], [4], [6], [9], [10], [20], [33], [40]–[42] and detailed reviews are available in [26] and [29]. A common strategy of many DA methods is to transform the data representations of the source and target domains in a way that would reduce the mismatch between the source distribution and target distribution. Methods that employ this strategy either perform subspace learning [6], [9], [13], [28], [30] or instance reweighting [2], [3]. The subspace-based methods aim to find a shared feature space where the distributions of the two datasets are similar. For example, Gong *et al.* [9] computed a geodesic flow between the source and target subspaces and integrated an infinite number of subspaces along the path. For classification, a geodesic flow kernel (GFK) is computed by projecting the features of both domains on these subspaces. Ni *et al.* [28] interpolated intermediate subspaces between source and target domains using dictionary learning. Similarity, Cui *et al.* [49] interpolated intermediate covariance features between source and target domains on the Riemannian manifold. Pan *et al.* [30] proposed domain invariant representations by learning transfer components to bring the data distributions in different domains closer to each other. Recently, Fernando *et al.* [6] proposed domain invariant feature representations by first aligning the target domain subspace to the source domain subspace and then projecting raw features to the aligned subspaces.

Instance reweighting approaches reduce the distribution discrepancy by source sample reweighting and by training a classifier on the weighted samples. Gong *et al.* [8] exploited landmarks which are a subset of source domain labeled data samples having similar distribution to the target domain. Landmarks are then used to bridge the source domain to the target. More recently, Long *et al.* [25] simultaneously perform subspace learning and instance reweighting in order to achieve state-of-the-art results.

Model-based approaches directly design a classifier for DA by incorporating distribution adaptation within the model regularization [5], [25], [27], [39]. These methods transform the model parameters learned from the source domain data to the target domain without changing the feature space. For example, Yang *et al.* [38] proposed the adapted support vector machine (SVM) for learning a new decision boundary by introducing a new regularizer into the SVM objective function to minimize both the classification error over the training examples and the discrepancy between the adapted and original classifiers. Similarly, Long *et al.* [24] proposed an adaptive classifier by simultaneously optimizing the structural risk functional, the joint distribution matching between domains, and the manifold consistency underlying marginal distribution. The model-based methods may not generalize well if sufficiently large target data is not available during training.

We propose an efficient blind DA algorithm by augmenting the target domain features with the source domain representation. The target features are represented from the nonlinear representations of the source domain and therefore, encode source specific knowledge. Note that [21] also proposed a blind DA algorithm. However, they assume access to samples from more than one earlier time steps whereas our algorithm requires source domain data from a single time step. A related approach is the self-taught learning [31], where one learns representations from unlabeled examples from a larger set of unrelated categories in an unsupervised fashion. This representation can then be transferred to labeled data to perform domain invariant classification. Our approach is close to the methods that use nonlinear autoencoders for DA [7], [43]–[46]. However, these methods are nonblind and rely on the availability of target domain data during training. Compared to the previous subspace based methods, our algorithm does not involve expensive computations of the intermediate subspaces on the geodesic path between source and target domain subspaces. Compared to the methods that explicitly reduce the distribution mismatch, our algorithm does not require expensive iterative optimization and distribution estimation steps. Furthermore, our method learns one model from one source domain for application to multiple target domains whereas previous nonblind methods learn different models for different source-target domain combinations.

III. Proposed Method

In this section, we give a brief overview of extreme learning machines (ELMs) and show how they differ from other learning paradigms. Next, we explain how ELM learns the nonlinear data structure in multiple parameters without prior assumptions. Finally, we show how domain adaptive classification can be formulated using the learned ELM models.

A. Extreme Learning Machines

Let $\{\mathbf{X}, \mathbf{T}\} = \{\mathbf{x}_j, \mathbf{t}_j\}_{j=1}^{N}$ represent our training data for supervised learning where $\mathbf{x}_j \in \mathbb{R}^d$ and $\mathbf{t}_j \in \mathbb{R}^q$ are the input and output training samples, respectively. The output samples $\mathbf{t}_j$ are the class labels for classification or the desired output features for regression. In either case, a regression function that determines the outputs from the inputs is estimated. The standard single hidden layer feed-forward network (SLFN) is a popular form of such a regression function. In SLFN, n_h hidden nodes fully connect the d inputs to the q outputs. The output vector $\mathbf{o}_j$ (predicted $\mathbf{t}_j$) is generated by feeding forward $\mathbf{x}_j$ through an SLFN as, $\mathbf{o}_j = \sum_{i=1}^{n_h} \boldsymbol{\beta}_i g(\mathbf{w}_i^\top \mathbf{x}_j + b_i)$, where $\mathbf{w}_i \in \mathbb{R}^d$ is the weight vector connecting the input nodes and the ith hidden node, $\boldsymbol{\beta}_i \in \mathbb{R}^q$ is the weight vector connecting the ith hidden node and the output nodes, and b_i is the bias of the ith hidden node. The activation function $g(u)$ is a nonlinear piecewise continuous function, such as the sigmoid function $g(u) = (1/(1 + e^{-u}))$.

An ELM [14], [15], [47], [48] performs random feature mapping followed by linear parameter solving to learn the $\{\mathbf{w}_i, b_i, \boldsymbol{\beta}_i\}_{i=1}^{n_h}$) parameters of an SLFN. The hidden layer parameters ($\{\mathbf{w}_i, b_i\}_{i=1}^{n_h}$) are randomly initialized to project the input data into a random feature space using the mapping function $g(.)$. This random projection stage is the main difference between ELM and other learning paradigms that generally perform deterministic feature mapping. Another major difference is that these parameters are never updated through back propagation. Thus, the input parameters are decoupled from the output parameters $\{\boldsymbol{\beta}_i\}_{i=1}^{n_h}$. This property of ELM makes the learning process extremely efficient and feasible for online applications that require fast learning compared to the neural network architectures that learn all network parameters iteratively.

The parameters connecting the hidden layer and the output layer (i.e., $\{\boldsymbol{\beta}_i\}_{i=1}^{n_h}$) are learned efficiently using regularized least squares which has a closed form solution. Let $\psi(\mathbf{x}_j) = [g(\mathbf{w}_1^\top \mathbf{x}_j + b_1) \dots g(\mathbf{w}_{n_h}^\top \mathbf{x}_j + b_{n_h})] \in \mathbb{R}^{1\times n_h}$ be the response vector of the hidden layer to the input $\mathbf{x}_j$ and $\mathbf{B} \in \mathbb{R}^{n_h \times q}$ as the output parameters connecting the hidden and output layers. ELM finds $\mathbf{B}$ that minimizes the sum of the squared losses of the prediction errors

$$\min_{\mathbf{B}\in\mathbb{R}^{n_h\times q}} \quad \frac{1}{2}\|\mathbf{B}\|_F^2 + \frac{C}{2}\sum_{j=1}^{N}\|\mathbf{e}_j\|_2^2$$
$$\text{s.t.} \quad \psi(\mathbf{x}_j)\mathbf{B} = \mathbf{t}_j^\top - \mathbf{e}_j^\top, \quad j = 1, \dots, N \tag{1}$$

where the first term is a regularizer to avoid over-fitting, $\mathbf{e}_j \in \mathbb{R}^q$ is the error vector with respect to the jth training sample ($\mathbf{e}_j = \mathbf{t}_j - \mathbf{o}_j$), and C is a tradeoff coefficient. We can rewrite (1) as an unconstrained optimization problem

$$\min_{\mathbf{B}\in\mathbb{R}^{n_h\times q}} \frac{1}{2}\|\mathbf{B}\|_F^2 + \frac{C}{2}\|\mathbf{T} - \mathbf{H}\mathbf{B}\|_2^2 \tag{2}$$

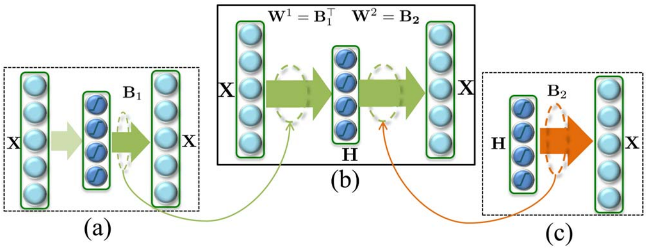
where $\mathbf{H} = [\psi(\mathbf{x}_1)^\top \cdots \psi(\mathbf{x}_N)^\top]^\top \in \mathbb{R}^{N\times n_h}$ and $\mathbf{T} = [\mathbf{t}_1 \cdots \mathbf{t}_N]^\top \in \mathbb{R}^{N\times q}$. Equation (2) has the following closed form solution:

$$\mathbf{B}^* = \begin{cases} \left(\mathbf{H}^\top\mathbf{H} + \dfrac{\mathbf{I}_{n_h}}{C}\right)^{-1}\mathbf{H}^\top\mathbf{T} & \text{if } N > n_h \\ \mathbf{H}^\top\left(\mathbf{H}\mathbf{H}^\top + \dfrac{\mathbf{I_N}}{C}\right)^{-1}\mathbf{T} & \text{if } N < n_h. \end{cases} \tag{3}$$

The parameters of the hidden layer can be randomly initialized according to any continuous probability distribution such as the uniform distribution. The only parameters to be learned are the output weights between the hidden nodes and the output nodes. These parameters are determined in a closed form by solving (3). These two features make ELMs more flexible than SVMs and much more computationally efficient than conventional feed-forward neural networks that use back-propagation to iteratively optimize the parameters [14]. Liu *et al.* [22] have recently proved that ELMs can attain the theoretical generalization bound of the feed-forward neural networks even when the connections within hidden neurons are randomly fixed.

B. Learning Nonlinear Structure With ELM-AE

In this paper, we learn the nonlinear structure of the input data in an unsupervised way using a feed-forward network whose parameters are learned by ELM algorithm [18]. This ELM-based auto-encoder has three attractive properties. First, it can learn the nonlinear data structures without prior assumptions. Second, it is highly efficient to train. Finally, it has good generalization capabilities as demonstrated by our experimental results.


$\mathbf{W}^1 = \mathbf{B}_1^\top$
$\mathbf{W}^2 = \mathbf{B}_2$
X
B₁
X
X
H
X
H
B₂
X

(a) (b) (c)

Fig. 3. ELM-AE training [18]. (a) Input features $\mathbf{X}$ are projected to the randomly initialized hidden layer and passed through the sigmoid function. Reconstruction matrix $\mathbf{B}_1$ is calculated using (3) and its transpose is set to be the input weight matrix, i.e., $\mathbf{W}^1 = \mathbf{B}_1^\top$. (b) Input features are projected again using the learned $\mathbf{W}^1$ and passed through the sigmoid function to obtain $\mathbf{H}$. (c) Output weights $\mathbf{W}^2$ are recomputed using (3).

The ELM auto-encoder (ELM-AE) is trained in an unsupervised fashion by using the input samples to be the same as the outputs, i.e., $\mathbf{T} = \mathbf{X}$ (see also Fig. 3). Denoting the parameters of the ELM model to be learned as $\mathbf{L} = \{\mathbf{W}^1, \mathbf{W}^2\}$, where $\mathbf{W}^i = [\mathbf{w}_1^i, \dots, \mathbf{w}_{n_i}^i]^\top \in \mathbb{R}^{n_{i+1}\times n_i}$. To learn $\mathbf{W}^1$ we use orthogonal weight vectors to connect the input layer to each unit of the hidden layer. Using orthogonal weight vectors results in a more effective projection of the input data to a random subspace. Orthogonalization of the random weights better preserves pairwise distances in the random ELM feature space [17] and subsequently leads to better generalization of the ELM-AE.

Next, we calculate $\mathbf{B}$ from (3) depending on the number of nodes in the hidden layer. Note that, $\mathbf{B}$ reprojects the lower dimensional representation of the input data back to its original space while minimizing the reconstruction error. Since this projection matrix is data-driven, it can be used as the weights of the first layer ($\mathbf{W}^1 = \mathbf{B}^\top$). Finally, the parameters $\mathbf{W}^2$ are learned by projecting $\mathbf{X}$ on the learned $\mathbf{W}^1$, passing the projected features through the sigmoid nonlinearity and then estimating $\mathbf{W}^2$ using (3). In ELM-AE the random feature mapping can preserve the structure of the original input data [37] and can be viewed as a less-structured counterpart to classical feature mapping such as principal component analysis (PCA). The projected features are further nonlinearly mapped and then reconstructed. Thus, the overall network can recover the structure of the data in multiple parameters. In contrast to auto-associative neural networks trained with back-propagation, the ELM-AE also does not require expensive iterative fine tuning of the weights.

C. Feature Augmentation via ELM-AE for Blind Domain Adaptation

Let $\mathcal{Z} = \{\mathbf{Z}_m\}_{m=1}^c \in \mathbb{R}^{d\times N}$ be the labeled source domain training data containing c different classes and N samples: $N = \sum_{m=1}^c s_m$, where s_m is the number of samples in the mth class. Let $\mathbf{Z}_m = \{\mathbf{z}_m^i\}_{i=1}^{s_m} \in \mathbb{R}^{d\times s_m}$ be the mth class, where $\mathbf{z}_m^i \in \mathbb{R}^d$ is a d-dimensional feature representation of the ith sample. Let $\mathbf{Y} = \{y_m\}_{m=1}^c$ be the class labels of the classes in $\mathcal{Z}$. The problem of blind domain adaptive classification involves estimating the label y_t of a sample $\mathbf{x}_t$ in the target domain by using the learned classifier from $\mathcal{Z}$ without any prior observation of the target domain samples.

1) Training: We learn two types of ELM-AE for domain adaptive classification. The first type is a global ELM-AE which is learned in an unsupervised fashion from all the training samples $\mathcal{Z}$ of the sources domain without using their labels. The global ELM-AE model is represented as $\mathbf{L}_{\mathcal{Z}} = \{\mathbf{W}^1_{\mathcal{Z}}, \mathbf{W}^2_{\mathcal{Z}}\}$. The global ELM-AE $\mathbf{L}_{\mathcal{Z}}$ encodes the source domain since it has been optimized to reconstruct any sample from the source domain. Next, we exploit the extremely fast learning time of ELM to learn class specific ELM-AEs from the labeled training samples of the source domain. Thus, for c training classes we learn c class specific ELM-AEs. The class specific ELM-AE models are denoted by $\{\mathbf{L}_j\}_{j=1}^c$, where a model for class j is represented by $\mathbf{L}_j = \{\mathbf{W}^1_j, \mathbf{W}^2_j\}$. For training a class specific model $\mathbf{L}_j$, we start from the previously learned global model $\mathbf{L}_{\mathcal{Z}}$ and fine tune it for class j only. In other words, instead of initializing the hidden layer weights $\mathbf{W}^1_j$ randomly, we initialize them with $\mathbf{W}^1_{\mathcal{Z}}$. Thus, the class specific ELM-AEs encode both domain specific and class specific information.

The learned ELM models are able to encode complex nonlinear structure of the source domain training data due to the nonlinear architecture of ELM. Compared to the previous DA methods that represent data using linear structure such as GFK [9], subspace alignment (SA) [6], transfer component analysis (TCA) [30], and transfer sparse coding [23], our proposed ELM models learn the structure of the training classes in multiple parameters, therefore, it is capable of learning more complex nonlinear manifold structures. Moreover, learning ELM models is computationally more efficient.

2) Testing: Given a test sample $\mathbf{x}_t$ from target domain, we predict its label as follows. We first represent $\mathbf{x}_t$ using the source domain specific model $\mathbf{L}_{\mathcal{Z}}$. This representation is given by the features of the last layer of the global ELM model

$$\hat{\mathbf{x}}_t = \mathbf{W}^2_{\mathcal{Z}} g\Big(\mathbf{W}^1_{\mathcal{Z}} \mathbf{x}_t\Big) \tag{4}$$

where g is chosen to be the sigmoid function. $\hat{\mathbf{x}}_t$ encodes both class specific and domain specific knowledge of the source domain. Then, we enrich the features of the test sample $\mathbf{x}_t$ with its new representation $\hat{\mathbf{x}}_t$ by augmentation, i.e., $\mathbf{z}_t = \mathbf{x}_t + \hat{\mathbf{x}}_t$.

Since the global ELM model $\mathbf{L}_{\mathcal{Z}}$ is trained on the source samples only, it will provide a better representation of the source data. Although, $\mathbf{L}_{\mathcal{Z}}$ may not be able to accurately reconstruct a target domain sample $\mathbf{x}_t$, the reconstructed $\hat{\mathbf{x}}_t$ is the source domain representation of $\mathbf{x}_t$. In other words, reconstruction with $\mathbf{L}_{\mathcal{Z}}$ provides a domain shift to $\mathbf{x}_t$. While $\hat{\mathbf{x}}_t$ better matches the source domain, $\mathbf{x}_t$ contains more identity specific information. Therefore, our augmentation of $\mathbf{x}_t$ with $\hat{\mathbf{x}}_t$ simultaneously achieves identity discrimination ability and compatibility with the source domain. As illustrated in Fig. 1, feature augmentation shifts the target domain toward its correct identity in the source domain.

We then normalize $\mathbf{z}_t$ using its ℓ_2 norm. Next, the enriched feature representation $\mathbf{z}_t$ is reconstructed using the class-specific models $\{\mathbf{L}_j\}_{j=1}^c$ and assigning to the class that incurs the least reconstruction error. Formally, the reconstruction of $\mathbf{z}_t$ from a model $\mathbf{L}_j$ is given by

$$\hat{\mathbf{z}}^j_t = f\big(\mathbf{z}_t, \mathbf{L}_j\big) = \mathbf{W}^2_j g\Big(\mathbf{W}^1_j \mathbf{z}_t\Big) \tag{5}$$

where f is the reconstruction and g is chosen to be the sigmoid function. The reconstruction error is computed as the squared Euclidean distance between $\mathbf{z}_t$ and $\hat{\mathbf{z}}^j_t$ as $e(j) = \|\mathbf{z}_t - \hat{\mathbf{z}}^j_t\|_2$. Finally, the predicted label y_t is chosen to be the class that incurs the minimum reconstruction error over all the classes

$$y_t = \arg\min_j \ \mathbf{e}(j). \tag{6}$$

Algorithm 1 Proposed Blind DA

```
Input: :
  Source domain training data $\mathcal{Z} = \{\mathbf{Z}_m\}_{m=1}^c \in \mathbb{R}^{d \times N}$ containing $c$ classes
  Class labels $\mathbf{Y} = \{y_m\}_{m=1}^c$
  Test sample from target domain $\mathbf{x}_t \in \mathbb{R}^d$
  Number of neurons in hidden layers $n_h$ and parameter $C = \{C_{\mathbf{W}^1}, C_{\mathbf{W}^2}\}$
Output: : Label $y_t$ of $\mathbf{x}_t$
  Training:
  $\mathbf{L}_{\mathcal{Z}} = \{\mathbf{W}^1_{\mathcal{Z}}, \mathbf{W}^2_{\mathcal{Z}}\}$ ▷ Learn a domain-specific global ELM from $\mathcal{Z}$
  for $j = 1 : c$ do
      $\mathbf{L}_j = \{\mathbf{W}^1_j, \mathbf{W}^2_j\}$ ▷ Learn class specific ELM models
  end for
  Testing:
  $\hat{\mathbf{x}}_t = \mathbf{W}^2_{\mathcal{Z}} g(\mathbf{W}^1_{\mathcal{Z}} \mathbf{x}_t)$ ▷ Domain specific representation
  $\mathbf{z}_t = \mathbf{x}_t + \hat{\mathbf{x}}_t$ ▷ Feature augmentation
  for $j = 1 : c$ do
      $\hat{\mathbf{z}}^j_t = f(\mathbf{z}_t; \mathbf{L}_j)$ {Reconstruct from model $\mathbf{L}_j$ (5)}
      $e(j) = \|\mathbf{z}_t - \hat{\mathbf{z}}^j_t\|_2^2$
  end for
  $y_t \triangleq \arg\min_j \ \mathbf{e}(j)$
```

The overall procedure of the proposed algorithm is summarized in Algorithm 1. The proposed algorithm is termed as augmented ELM (AELM).

IV. Experimental Results

We performed extensive experiments on 16 cross domain datasets generated by permuting six public datasets (Fig. 4). Results are compared to three baseline classification algorithms and six state-of-the-art DA methods.

A. Dataset Specifications

The six datasets that we use in our experiments include U.S. Postal Service (USPS),[1] Mixed National Institute of Standards and Technology (MNIST),[2] Microsoft Research Cambridge (MSRC),[3] visual object classes (VOC)2007,[4]

[1] www-i6.informatik.rwth-aachen.de/~keysers/usps.html

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

[3] http://research.microsoft.com/en-us/projects/objectclassrecognition

[4] http://pascallin.ecs.soton.ac.uk/challenges/VOC/voc2007

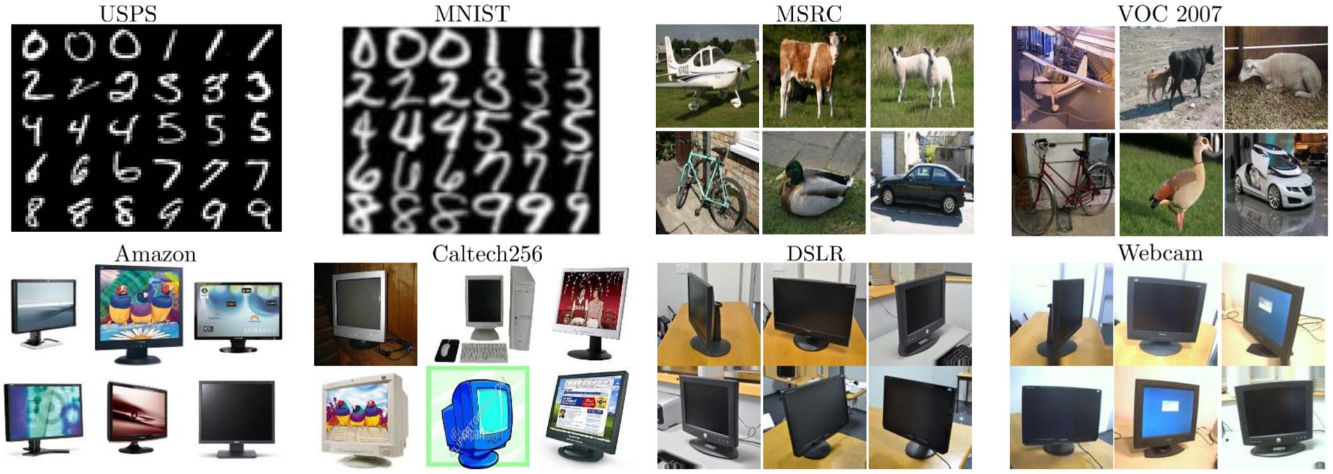


Fig. 4. Samples images from six datasets used in our experiments. Each dataset represents a different domain.

Office [9], [33], and Caltech-256 [12] (Fig. 4). These are benchmark datasets widely used for evaluating visual DA algorithms [6], [9], [23], [30].

1) Cross Domain Digit Recognition: For this task we use USPS and MNIST datasets. USPS digit dataset contains 7291 training images and 2007 test images of size 16 × 16. MNIST digit dataset contains 60 000 training images and 10 000 test images of size 28 × 28. The images in the USPS and MNIST databases follow very different distributions (see Fig. 4). Both datasets share ten classes of digits. Following [25], we construct one dataset (USPS→MNIST) by randomly sampling 1800 images in USPS to form the source data (training data), and randomly sampling 2000 images from MNIST to form the target data (testing data). A second dataset is created by switching the source and target (MNIST→USPS). All images of USPS and MNIST datasets are converted to grayscale, uniformly resized to 16 × 16 and vectorized to form the feature representation.

2) Cross Domain Object Categorization: For this task we use MSRC, VOC2007, Office, and Caltech-256 datasets. MSRC dataset is provided by Microsoft Research Cambridge and consists of 4323 images of 18 object classes. VOC dataset (training/validations subset) consists of 5011 images of 20 object classes. The two datasets share six object categories, including airplane, bicycle, bird, car, cow, and sheep. We follow [23] and [25] and construct one dataset MSRC→VOC by selecting all 1269 images (shared category) in MSRC to form the source domain, and all 1530 images (shared category) in VOC2007 to form the target domain. We switch the source/target domains to get another dataset VOC→MSRC. All images are uniformly resized to 256 × 256, and for every image 128-dimensional dense scale invariant feature transform (DSIFT) features are extracted using the VLFeat [35] computer vision library. A 240-dimensional codebook is learned, where *K*-means clustering is used to construct the codewords. Office [9], [33] is a popular benchmark for visual DA. The database contains images of objects from three different domains, Amazon (images downloaded from on-line merchants), Webcam (low-resolution images by a Web camera), and DSLR (high-resolution images by a digital single-lens reflex (SLR) camera). The database contains 46 522 images of 31 object categories. Caltech-256 [12] is a standard database for object category recognition. The dataset contains 30 607 images of 256 object categories. We perform experiments on the Office and Caltech datasets provided by Gong *et al.* [9]. Ten object categories common to all four datasets were extracted for experiments. We follow similar feature extraction and experiment protocols used in previous work [9], [25]. Specifically, SURF features are extracted and quantized into an 800-bin histogram with codebooks learned with *K*-means on a subset of images from Amazon. The four different domains are represented by C (Caltech-256), A (Amazon),→ W (Webcam), and D (DSLR). By randomly selecting different source and target domains, we construct 4 × 3 = 12 cross domain object datasets C→A, C→W, C→D, . . . , D→W.

B. Experimental Setup

We compare the results of the proposed algorithm with three baseline methods and six state-of-the art DA methods. The baseline methods include 1-nearest neighbor (NN) classifier, PCA + NN and the basic ELMs without feature augmentation. The six DA methods include joint feature selection and subspace learning (FSSL) [13] + NN, TCA [30] + NN, GFK [9] + NN, marginalized stacked denoising autoencoders (mSDA) [43], SA [6] + NN, and transfer joint matching (TJM) [25] + NN. The results of NN, PCA, FSSL, TCA, GFK, and TJM are reported by Long *et al*. [25] whereas the results of SA [6] SA + NN are computed using the original implementation provided by its authors.

We follow the same evaluation protocol as [9], [23], and [25] for a fair comparison. Similar to [25], NN is trained on the labeled source data, and tested on the unlabeled target data. PCA, FSSL, TCA, GFK, and TJM are performed as dimensionality reduction on all datasets and then a 1-NN classifier is trained on the labeled source data and used to classify the unlabeled target data. Note that these methods utilize the target domain data in computing the subspace. Since labeled training data and unlabeled test data are sampled from different distributions,

TABLE I

COMPARISON OF ACCURACIES (%) ON CROSS DOMAIN DIGIT RECOGNITION AND OBJECT CATEGORIZATION. THE PROPOSED AELM HAS THE BEST AVERAGE PERFORMANCE EVEN THOUGH IT IS THE ONLY METHOD THAT PERFORMS BLIND DA

	NN	PCA	ELM	FSSL	TCA	GFK	SA	mSDA	TJM	AELM
USPS→MNIST	44.70	44.95	57.70	51.45	44.15	46.45	40.15	43.20	52.25	**57.77**
MNIST→USPS	65.94	66.22	61.11	57.44	58.78	61.22	48.22	**66.94**	63.28	62.33
MSRC→VOC	31.96	32.94	25.03	29.74	32.55	34.18	34.37	28.62	32.75	**34.77**
VOC→MSRC	41.06	42.79	56.33	37.93	32.75	44.47	46.57	49.40	49.41	**56.58**
C→A	23.70	36.95	49.37	35.88	45.82	41.02	41.02	45.92	46.76	**53.13**
C→W	25.76	32.54	37.97	32.32	30.51	40.68	40.34	37.96	39.98	**49.49**
C→D	25.48	38.22	45.22	37.53	35.67	38.58	47.13	46.49	44.59	**50.96**
A→C	26.00	34.73	40.07	33.91	40.07	40.25	40.16	40.96	39.45	**41.14**
A→W	29.83	35.59	33.56	34.35	35.25	38.98	39.66	40.33	**42.03**	35.25
A→D	25.48	27.39	34.31	26.37	34.39	36.31	35.03	36.30	**45.22**	36.94
W→C	19.86	26.36	31.17	25.85	29.92	30.72	31.17	31.96	30.19	**34.11**
W→A	22.96	29.35	33.85	29.53	28.81	29.75	33.82	33.61	29.96	**38.93**
W→D	59.24	77.07	88.54	76.79	85.99	80.89	85.99	87.26	89.17	**89.81**
D→C	26.27	29.65	28.23	27.89	32.06	30.28	31.26	30.89	31.43	**33.83**
D→A	28.50	32.05	28.50	30.61	31.42	32.05	**35.80**	35.59	32.78	33.09
D→W	63.39	75.93	73.22	74.99	86.44	75.59	84.75	**87.45**	85.42	80.33
Average	35.01	41.42	45.27	40.16	42.79	43.86	44.71	46.43	47.17	**49.28**

under this experimental setup, it is impossible to tune the optimal parameters via cross validation. Thus the parameters of all the compared algorithms are manually tuned and the results of the parameters which give the best accuracy on all datasets are reported. For subspace learning methods (PCA, GFK, FSSL, and SA), the number of subspace basis vectors are selected by searching in the range $\{10, 20, \dots, 200\}$. For TCA, the adaptation regularization parameter λ is searched in the range $\{0.01, 0.1, 1, 10, 100\}$. For mSDA the number of hidden layers is searched in the range 1–5, and the corruption probability is searched in the range $\{0.01, \dots, 1\}$. The proposed algorithm requires two parameters: 1) number of neurons in the hidden layers n_h and 2) the regularization constant C. These parameters are chosen through tenfold cross validation using only the source domain training data. For object categorization datasets (Office, MSRC, VOC, and Caltech) $n_h = 12$ and $C = \{4, 10^8\}$ for learning $\mathbf{W}^1$ and $\mathbf{W}^2$, respectively. For digit recognition datasets $n_h = 19$ and $C = \{1.5, 10^8\}$.

V. RESULTS AND ANALYSIS

The classification accuracies of AELM and the nine compared methods on the 16 cross domain visual object datasets are listed in Table I. The NN classifier achieved an average classification accuracy of 35%. PCA+NN only performed well on the MNIST→USPS dataset indicating that the adaptation difficulty varies across the 16 datasets. The baseline ELM classifier without adaptation also suffers from the domain shift present in the datasets. TCA performed better than PCA because it jointly performs feature transformation and matching. Since it relies on the linear structure of the data, its performance is still lower than that of the nonadapted baseline ELM.

FSSL performed well on the digit datasets, but did not achieve good results on object categorization. A possible explanation is that FSSL performs joint FSSL to learn a shared subspace by automatically selecting the relevant features for adaptation. In the case of digits, the black background pixels make it easy to select the foreground pixels as relevant features [25]. However, the automatic feature selection in FSSL is unable to cope with large domain differences in the object categorization tasks.

GFK and SA methods achieved good accuracies on object categorization but performed lower on the digit datasets. Their average accuracies were also lower than the basic ELM because these methods use linear subspaces to represent the data. On the other hand, ELM based auto-encoding learns the nonlinear structure which improves the accuracies on most datasets. Moreover, the subspace dimension in GFK needs to be small to ensure that the source subspace can transit smoothly along the geodesic flow toward the target subspace. This limits the representation ability of GFK. The stacked autoencoders method mSDA [43] achieved the third best accuracy. Note that this method was originally designed for cross domain sentiment classification whereas we applied it to cross domain visual object recognition.

The instance reweighting based technique (TJM) achieved very good results on both digit recognition and object categorization tasks. TJM learns a more accurate shared subspace and outperforms other feature transformation methods such as TCA and FSSL by introducing a structured sparsity penalty on the source instances, which can adaptively reweight the source instances according to their relevance to the target instances.

The proposed AELM achieved the best performance on 11 out of 16 datasets. The average classification accuracy of AELM on the 16 datasets is the highest (49.3%) outperforming the nearest competitive method (TJM) by 2.1%. TJM could outperform AELM on only two datasets. Recall that AELM is the only blind DA method in Table I and all other nonbaseline methods require target domain data for training. The improved accuracy of AELM is attributed to its ability to better represent nonlinear data structure and the feature augmentation. Apart from TJM, previous methods either perform well on digit recognition or object categorization whereas the proposed AELM performs equally well on both tasks indicating its ability to generalize to different cross domain image recognition problems.

TABLE II
COMPARISON OF AVERAGE ACCURACIES (%) USING CNN FEATURES FOR THE 12 CROSS DOMAIN VISUAL RECOGNITION TASKS (USING THE OFFICE AND CALTECH DATASETS)

	NN	ELM	GFK	SA	mSDA	TJM	AELM
C→A	87.05	89.07	87.27	87.06	89.67	88.10	**89.46**
C→W	72.20	70.51	75.93	75.59	68.47	72.20	**79.32**
C→D	80.89	78.98	**83.44**	80.25	82.17	74.52	**81.53**
A→C	78.54	79.61	**80.32**	79.61	78.81	77.65	79.96
A→W	77.31	74.58	76.95	78.31	**78.98**	75.25	77.63
A→D	80.25	80.25	80.89	81.53	79.62	82.80	**85.35**
W→C	68.21	70.61	67.76	68.83	69.46	**71.42**	71.24
W→A	73.07	75.37	74.32	75.16	76.62	**80.27**	76.83
W→D	**100.0**	**100.0**	**100.0**	**100.0**	**100.0**	**100.0**	**100.0**
D→C	70.08	68.21	69.10	69.99	73.29	72.57	**75.60**
D→A	75.89	80.79	75.78	73.49	81.32	78.60	**83.19**
D→W	97.97	98.31	98.64	98.98	98.64	98.31	**98.98**
Avg	80.12	80.52	80.87	80.73	81.42	80.97	**83.25**

A. Evaluation Using Off-the-Shelf CNN Features

It is important to emphasize that the proposed AELM is not a feature extraction technique but a blind DA technique. It can improve the performance of any feature type extracted from different domains. To show this, we apply AELM to convolutional neural network (CNN) features. For this purpose, we use the VLFeat MatConvNet [36] library which offers different pretrained CNN models. We use the Caffe [16] implementation of AlexNet [19] which is trained on the ImageNet dataset. The output of the first fully connected layer is used as 4096 dimensional image features. Table II summarizes the average accuracy for the 12 cross domain visual recognition tasks of Office+Caltech datasets. As expected, CNN features significantly outperform the hand crafted features (Table I) by a large margin. This is because CNN features are robust to domain changes and generalize well to novel tasks [32].

A domain shift is still likely to be present in the CNN features and thus, DA algorithms can be employed on top of CNN features to minimize this shift. Table II shows our results on the CNN features. Note that we only compare those methods for which the code is available. Despite using the target domain data in training their models, all compared methods give insignificant improvement over the baseline NN. The proposed AELM achieves the overall best accuracy of 83.3% which is an improvement of 3.1% over the baseline. Recall that AELM does not use any target domain data for training. This shows that the proposed AELM is independent of the input features and consistently improves performance through a blind, yet more accurate domain transfer.

B. Blind Domain Transfer Analysis

The experimental protocol used in Tables I and II favor previous methods as the target domain data (that was meant to be classified) was also used to train their models. In this section, we test the generalization ability of two of the best performing methods (GFK and SA) to unseen target data by reducing the number of target samples used for training. We are unable to test TJM in this setting because TJM requires retraining of the model every time a new target sample arrives for classification. We perform this experiment on the 12 cross domain object categorization datasets generated from the Office and Caltech datasets. For each dataset, we randomly select $r = \{1, \dots, 20\}$ target samples per class for training. A domain adaptive classifier is then learned using GFK and SA from this new target data along with the full source data.

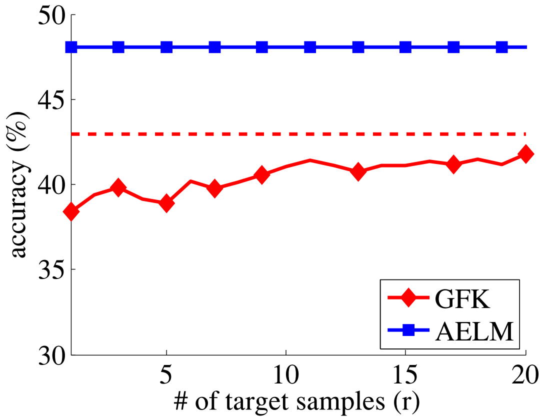


Fig. 5. Comparing AELM with GFK based on the average accuracy versus the number of target samples r per class for the 12 cross domain visual recognition tasks listed in Table II. The dotted lines show the average accuracy when all the target domain samples are used for training. Since AELM does not require any target samples, its performance remains constant whereas the performance of GFK fluctuates.

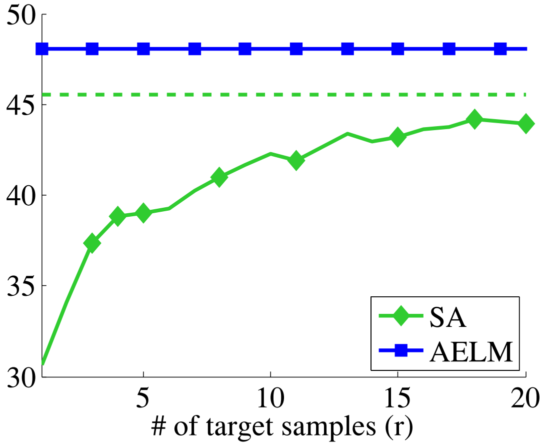


Fig. 6. Comparing AELM with SA based on the average accuracy versus the number of target samples r per class for the 12 cross domain visual recognition tasks listed in Table II. The dotted lines show the average accuracy when all the target domain samples are used for training. Since AELM does not require any target samples, its performance remains constant whereas the performance of SA reduces when the number of target samples used for training is reduced.

Figs. 5 and 6 compare the performance of AELM with GFK and SA based on the average accuracies on the 12 cross domain datasets against different values of target samples r used for training. As expected, the accuracy of GFK fluctuates and that of SA reduces significantly when the number of target training samples is reduced. Since these methods are subspace based, they require dense sampling of the target domain classes in order to estimate accurate linear structure of the target domain. Notice that the performance of AELM remains the same (and better) throughout as it is independent of the number of target samples.

TABLE III
COMPARISON OF AVERAGE EXECUTION TIMES (IN SECONDS) USING CNN FEATURES FOR THE 12 CROSS DOMAIN VISUAL RECOGNITION TASKS, I.E., USING THE OFFICE AND CALTECH DATASETS AS SOURCE AND TARGET DOMAINS, RESPECTIVELY

	GFK	SA	mSDA	TJM	AELM
Training time	32.26	2.5	10.44	6.52	0.15
Testing time	0.54	0.03	0.56	0.03	0.19

C. Statistical Analysis of the Results

For statistical analysis of our results, we used the nonparametric Friedman test which is based on ranking the performance of algorithms on multiple datasets. We consider the accuracies of the seven algorithms using CNN features reported in Table II for this test. In our analysis, the Friedman test gives a p value of 1.5×10^{-4} which rejects the null hypothesis that the performance of the seven algorithms is the same. Hence our results are statistically significant. As per Friedman test, the mean ranks of the NN, ELM, GFK, SA, msDA, TJM, and AELM on the 12 datasets are 2.41, 3.04, 3.70, 3.79, 4.70, 4.00, and 6.34, respectively. Note that the mean rank of the proposed AELM is significantly higher than the remaining algorithms and has a higher improvement over its nearest competitor compared to the rest. In addition to accuracy and its statistical significance, our algorithm has the advantages of being blind (i.e., it does not require target domain data for training) and extremely fast training time as detailed in the next section.

D. Execution Time

Table III compares the average execution times of our method with other DA algorithms for 12 cross domain visual recognition tasks on the Office and Caltech datasets using CNN features. MATLAB implementations on a Core i5 3.3 GHz CPU with 16 GB RAM were used for calculating the execution times. The proposed AELM algorithm is significantly faster than the other algorithms in the training phase which makes it feasible for online learning applications. The testing speeds of SA and TJM are faster than AELM because they perform NN classification in a low dimensional subspace while AELM performs multiple reconstructions in the original high dimensional feature space. However, the proposed AELM achieves the highest accuracy.

VI. CONCLUSION

We presented a blind domain adaptive classification algorithm which does not require any target domain data for training and still outperforms existing state-of-the-art methods that are nonblind. The proposed AELM has good generalization abilities and performs well on different tasks such as digit recognition and object categorization as well as different feature types such as raw pixel values, SURF, DSIFT, and CNN features. Extensive experiments on 16 standard datasets demonstrated that despite blind learning, the proposed AELM outperforms five existing state-of-the-art methods in cross domain visual recognition.

REFERENCES

[1] A. Bergamo and L. Torresani, “Exploiting weakly-labeled Web images to improve object classification: A domain adaptation approach,” in *Proc. Adv. Neural Inf. Process. Syst.*, Vancouver, BC, Canada, 2010, pp. 181–189.

[2] L. Bruzzone and M. Marconcini, “Domain adaptation problems: A DASVM classification technique and a circular validation strategy,” *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 32, no. 5, pp. 770–787, May 2010.

[3] W.-S. Chu, F. De la Torre, and J. F. Cohn, “Selective transfer machine for personalized facial action unit detection,” in *Proc. IEEE Int. Conf. Comput. Vis. Pattern Recognit.*, Portland, OR, USA, 2013, pp. 3515–3522.

[4] W. Dai, G.-R. Xue, Q. Yang, and Y. Yu, “Transferring naive Bayes classifiers for text classification,” in *Proc. Nat. Conf. Artif. Intell. (AAAI)*, Vancouver, BC, Canada, 2007, pp. 540–545.

[5] L. Duan, D. Xu, and I. W. Tsang, “Domain adaptation from multiple sources: A domain-dependent regularization approach,” *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 23, no. 3, pp. 504–518, Mar. 2012.

[6] B. Fernando, A. Habrard, M. Sebban, and T. Tuytelaars, “Unsupervised visual domain adaptation using subspace alignment,” in *Proc. IEEE Int. Conf. Comput. Vis.*, Sydney, NSW, Australia, 2013, pp. 2960–2967.

[7] X. Glorot, A. Bordes, and Y. Bengio, “Domain adaptation for large-scale sentiment classification: A deep learning approach,” in *Proc. Int. Conf. Mach. Learn.*, Bellevue, WA, USA, 2011, pp. 513–520.

[8] B. Gong, K. Grauman, and F. Sha, “Connecting the dots with landmarks: Discriminatively learning domain-invariant features for unsupervised domain adaptation,” in *Proc. Int. Conf. Mach. Learn.*, Atlanta, GA, USA, 2013, pp. 153–159.

[9] B. Gong, Y. Shi, F. Sha, and K. Grauman, “Geodesic flow kernel for unsupervised domain adaptation,” in *Proc. IEEE Int. Conf. Comput. Vis. Pattern Recognit.*, Providence, RI, USA, 2012, pp. 2066–2073.

[10] R. Gopalan, R. Li, and R. Chellappa, “Domain adaptation for object recognition: An unsupervised approach,” in *Proc. IEEE Int. Conf. Comput. Vis.*, Barcelona, Spain, 2011, pp. 999–1006.

[11] A. Gretton, K. Borgwardt, M. Rasch, B. Schölkopf, and A. Smola, “A kernel method for the two-sample problem,” in *Proc. Adv. Neural Inf. Proc. Systems*, Vancouver, BC, Canada, 2006, pp. 513–520.

[12] G. Griffin, A. Holub, and P. Perona, “Caltech-256 object category dataset,” Dept. Eng. Appl. Sci., California Inst. Technol., Pasadena, CA, USA, Tech. Rep. CNS-TR-2007-001, 2007.

[13] Q. Gu, Z. Li, and J. Han, “Joint feature selection and subspace learning,” in *Proc. Int. Joint Conf. Artif. Intell.*, Barcelona, Spain, 2011, pp. 1294–1299.

[14] G.-B. Huang, L. Chen, and C.-K. Siew, “Universal approximation using incremental constructive feedforward networks with random hidden nodes,” *IEEE Trans. Neural Netw.*, vol. 17, no. 4, pp. 879–892, Jul. 2012.

[15] G.-B. Huang, Q.-Y. Zhu, and C.-K. Siew, “Extreme learning machine: Theory and applications,” *Neurocomputing*, vol. 70, nos. 1–3, pp. 489–501, 2006.

[16] Y. Jia *et al.*, “Caffe: Convolutional architecture for fast feature embedding,” in *Proc. Int. Conf. Multimedia*, Orlando, FL, USA, 2014, pp. 675–678.

[17] W. B. Johnson and J. Lindenstrauss, “Extensions of Lipschitz mappings into a Hilbert space,” in *Proc. Conf. Modern Anal. Probab.*, vol. 26. Providence, RI, USA, 1984, pp. 189–206.

[18] L. L. C. Kasun, H. Zhou, and G.-B. Huang, “Representational learning with ELMs for big data,” *IEEE Intell. Syst.*, vol. 28, no. 6, pp. 31–34, Nov. 2013.

[19] A. Krizhevsky, I. Sutskever, and G. E. Hinton, “ImageNet classification with deep convolutional neural networks,” in *Proc. Adv. Neural Inf. Process. Syst.*, 2012, pp. 1106–1114.

[20] B. Kulis, K. Saenko, and T. Darrell, “What you saw is not what you get: Domain adaptation using asymmetric kernel transforms,” in *Proc. IEEE Int. Conf. Comput. Vis. Pattern Recognit.*, Colorado Springs, CO, USA, 2011, pp. 1785–1792.

[21] C. Lampert. (2014). *Blind Domain Adaptation: An RKHS Approach.* [Online]. Available: http://arxiv.org/abs/1406.5362v1

[22] X. Liu, S. Lin, J. Fang, and Z. Xu, “Is extreme learning machine feasible? A theoretical assessment (part I),” *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 26, no. 1, pp. 7–20, Jan. 2015.

[23] M. Long *et al.*, “Transfer sparse coding for robust image representation,” in *Proc. IEEE Int. Conf. Comput. Vis. Pattern Recognit.*, Portland, OR, USA, 2013, pp. 407–414.

[24] M. Long, J. Wang, G. Ding, S. J. Pan, and P. S. Yu, "Adaptation regularization: A general framework for transfer learning," *IEEE Trans. Knowl. Data Eng.*, vol. 26, no. 5, pp. 1076–1089, May 2014.

[25] M. Long, J. Wang, G. Ding, J. Sun, and P. S. Yu, "Transfer joint matching for unsupervised domain adaptation," in *Proc. IEEE Int. Conf. Comput. Vis. Pattern Recognit.*, Columbus, OH, USA, 2014, pp. 1410–1417.

[26] A. Margolis, "A literature review of domain adaptation with unlabeled data," Dept. Elect. Eng., Univ. Washington, Seattle, WA, USA, Tech. Rep., 2011.

[27] F. Mirrashed and M. Rastegari, "Domain adaptive classification," in *Proc. IEEE Int. Conf. Comput. Vis.*, Sydney, NSW, Australia, 2013, pp. 2608–2615.

[28] J. Ni, Q. Qiu, and R. Chellappa, "Subspace interpolation via dictionary learning for unsupervised domain adaptation," in *Proc. IEEE Int. Conf. Comput. Vis. Pattern Recognit.*, Portland, OR, USA, 2013, pp. 692–699.

[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. 2009.

[30] 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.

[31] R. Raina, A. Battle, H. Lee, B. Packer, and A. Y. Ng, "Self-taught learning: Transfer learning from unlabeled data," in *Proc. Int. Conf. Mach. Learn.*, Corvallis, OR, USA, 2007, pp. 759–766.

[32] A. S. Razavian, H. Azizpour, J. Sullivan, and S. Carlsson, "CNN features off-the-shelf: An astounding baseline for recognition," *CoRR*, 2014. [Online]. Available: http://arxiv.org/abs/1403.6382

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

[34] A. Torralba and A. A. Efros, "Unbiased look at dataset bias," in *Proc. IEEE Int. Conf. Comput. Vis. Pattern Recognit.*, Providence, RI, USA, 2011, pp. 1521–1528.

[35] A. Vedaldi and B. Fulkerson, "VLfeat—An open and portable library of computer vision algorithms," in *Proc. ACM Int. Conf. Multimedia*, Florence, Italy, 2010, pp. 1469–1472.

[36] A. Vedaldi and K. Lenc, "MatConvNet—Convolutional neural networks for MATLAB," *CoRR*, 2014. [Online]. Available: http://arxiv.org/abs/1412.4564

[37] J. Wright, A. Y. Yang, A. Ganesh, S. S. Sastry, and Y. Ma, "Robust face recognition via sparse representation," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 31, no. 2, pp. 210–227, Feb. 2009.

[38] J. Yang, R. Yan, and A. G. Hauptmann, "Adapting SVM classifiers to data with shifted distributions," in *Proc. IEEE Int. Conf. Data Mining Workshops*, Washington, DC, USA, 2007, pp. 69–76.

[39] Y. Aytar and A. Zisserman, "Tabula Rasa: Model transfer for object category detection," in *Proc. IEEE Int. Conf. Comput. Vis.*, Barcelona, Spain, 2011, pp. 2252–2259.

[40] T. Tommasi and B. Caputo, "Frustratingly easy NBNN domain adaptation," in *Proc. IEEE Int. Conf. Comput. Vis.*, Sydney, NSW, Australia, 2013, pp. 897–904.

[41] N. Farajidavar, T. de Campos, and J. Kittler, "Transductive transfer machine," in *Proc. Asian Conf. Comput. Vis.*, Singapore, 2014, pp. 623–639.

[42] N. Patricia and B. Caputo, "Learning to learn, from transfer learning to domain adaptation: A unifying perspective," in *Proc. IEEE Int. Conf. Comput. Vis. Pattern Recognit.*, Columbus, OH, USA, 2014, pp. 1442–1449.

[43] M. Chen, Z. Xu, K. Q. Weinberger, and F. Sha, "Marginalized stacked denoising autoencoders for domain adaptation," in *Proc. Int. Conf. Mach. Learn.*, Edinburgh, U.K., 2012, pp. 156–162.

[44] Y. Bengio, "Deep learning of representations for unsupervised and transfer learning," in *Proc. ICML Workshop Unsupervised Transfer Learn.*, Washington, DC, USA, 2011, pp. 17–36.

[45] G. Mesnil *et al.*, "Unsupervised and transfer learning challenge: A deep learning approach," in *Proc. Unsupervised Transfer Learn. Challenge Workshop*, 2012, pp. 97–110.

[46] S. Rifai, P. Vincent, X. Muller, X. Glorot, and Y. Bengio, "Contractive auto-encoders: Explicit invariance during feature extraction," in *Proc. Int. Conf. Mach. Learn.*, Bellevue, WA, USA, 2011, pp. 455–462.

[47] G. Huang, S. Song, J. N. Gupta, and C. Wu, "Semi-supervised and unsupervised extreme learning machines," *IEEE Trans. Cybern.*, vol. 44, no. 12, pp. 2405–2417, Dec. 2014.

[48] Z. Bai, G.-B. Huang, D. Wang, H. Wang, and M. B. Westover, "Sparse extreme learning machine for classification," *IEEE Trans. Cybern.*, vol. 44, no. 10, pp. 1858–1870, Oct. 2014.

[49] Z. Cui *et al.*, "Flowing on Riemannian manifold: Domain adaptation by shifting covariance," *IEEE Trans. Cybern.*, vol. 44, no. 12, pp. 2264–2273, Dec. 2014.

Muhammad Uzair received the B.Sc. degree in computer systems engineering from the University of Engineering and Technology Peshawar, Peshawar, Pakistan, in 2006, the M.S. degree in electronics and computer engineering from Hanyang University, Seoul, Korea, in 2009, and the Ph.D. degree in computer engineering from the University of Western Australia (UWA), Crawley, WA, Australia, in 2016.

His current research interests include computer vision, machine learning, domain adaptation, image set modeling, and hyperspectral image analysis.

Mr. Uzair was a recipient of the Higher Education Commission Pakistan Scholarship for M.S. study and the UWA SURF Scholarship for Ph.D. study.

Ajmal Mian (M'13) received the Ph.D. degree with distinction from the University of Western Australia, Crawley, WA, Australia, in 2006.

He is currently with the School of Computer Science and Software Engineering, University of Western Australia. He has secured five Australian Research Council grants worth over $2.3 million and one National Health and Medical Research grant. His current research interests include computer vision, action recognition, 3-D shape analysis, hyperspectral image analysis, machine learning, image set modeling, and multimodal biometrics.

Prof. Mian was a recipient of the Australasian Distinguished Doctoral Dissertation Award from Computing Research and Education Association of Australasia, two prestigious nationally competitive fellowships namely the Australian Postdoctoral Fellowship in 2008 and the Australian Research Fellowship in 2011, the UWA Outstanding Young Investigator Award in 2011, the West Australian Early Career Scientist of the Year Award in 2012, and the Vice-Chancellor's Mid-Career Research Award in 2014.