Neural Comput & Applic (2018) 30:1211–1223
https://doi.org/10.1007/s00521-016-2758-x

ORIGINAL ARTICLE

Representation learning with deep extreme learning machines for efficient image set classification

Muhammad Uzair[1,3] · Faisal Shafait[2] · Bernard Ghanem[4] · Ajmal Mian[3]

Received: 31 May 2016 / Accepted: 23 November 2016 / Published online: 9 December 2016

Abstract Efficient and accurate representation of a collection of images, that belong to the same class, is a major research challenge for practical image set classification. Existing methods either make prior assumptions about the data structure, or perform heavy computations to learn structure from the data itself. In this paper, we propose an efficient image set representation that does not make any prior assumptions about the structure of the underlying data. We learn the nonlinear structure of image sets with deep extreme learning machines that are very efficient and generalize well even on a limited number of training samples. Extensive experiments on a broad range of public datasets for image set classification show that the proposed algorithm consistently outperforms state-of-the-art image set classification methods both in terms of speed and accuracy.

✉ Muhammad Uzair
uzair@ciitwah.edu.pk

Faisal Shafait
faisal.shafait@uwa.edu.au

Bernard Ghanem
bernard.ghanem@kaust.edu.sa

Ajmal Mian
ajmal.mian@uwa.edu.au

[1] COMSATS Institute of Information Technology, Wah Cantonment, Pakistan

[2] National University of Science and Technology, Islamabad, Pakistan

[3] Computer Science and Software Engineering, The University of Western Australia, Crawley, Australia

[4] King Abdullah University of Science and Technology, Thuwal, Saudi Arabia

Keywords Extreme learning machine · Image set classification · Representation learning · Face recognition

1 Introduction

Image set classification has received significant interest from the computer vision research community because of its wide range of applications in multi-view object classification [4, 6, 22, 38, 41–43] and face recognition [3, 5, 12, 13, 32–34]. The problem of image set classification arises in many computer vision applications where a given collection of images are known to belong to one class but with unknown identity. In contrast to the traditional single image-based classification, image set classification algorithms model the given image collection as a whole to obtain a more accurate estimate of the class identity. The images in a set usually cover a diverse range of image variations such as illumination, pose and scale changes. Image set classification algorithms have the capability to explicitly or implicitly model these variations for improved classification accuracy [12, 13, 22, 32]. Image set classification is also applicable as a generalized form of video-based classification. However, it is not necessary for the images in a set to have any temporal relationship [13, 42].

An image set classification algorithm must essentially address two core challenges; how to represent an image set to effectively capture image level as well as set level variations and how to define a distance/similarity measure between two image sets. Defining a suitable distance between two sets of images is often tied to the representation used to model the image sets in the first place. Hence, most of the research in this area has concentrated on developing image set representations by making certain assumptions about the set structure. Some techniques assume the set data follows a Gaussian distribution [32, 37, 41, 42] which is unlikely to be true for all types of images. Image sets have also been represented by linear subspaces [10, 22] even though there is evidence that they are more likely to lie on complex manifolds [12]. To model more complex data structures, several techniques have been proposed to model image sets as a convex or affine hulls of the data samples [3, 13, 30]. These techniques are conceptually similar to nearest neighbour classification and must impose certain constraints to avoid finding the neighbours in some low-dimensional space where image sets might intersect. However, the ability to model more complex image set structures comes at the cost of added algorithm complexity [12, 13, 30, 32, 41, 43]. Therefore, these algorithms cannot be efficiently scaled to handle large image set classification tasks [28].

In this work, we have focused on developing an efficient and accurate representation of image sets that can model arbitrarily complex image set structures on one hand, and scale to large problem sizes on the other. We employ extreme learning machines (ELM) for this purpose primarily due to their computational efficiency [8, 15–17, 29]. An ELM trains a single hidden layer feedforward neural network (SLFN) by randomly initializing the weights of the input layer and calculating the weights for the output layer analytically. Deep ELM have the potential of effectively learning the underlying structure of the image set without any prior assumption on the distribution or structure of image set data. Our algorithm learns a Deep ELM (DELM) model for each class in the gallery (training classes) through unsupervised feature learning with an ELM-based auto-encoder (ELM-AE) (Fig. 1). The probe (test) set is assigned a label based on the lowest reconstruction error.

The key contributions of this paper are threefold: (1) An effective image set representation scheme based on deep extreme learning machines that does not make any assumption about the structure of the set but implicitly learns it from training data. (2) The proposed algorithm does not require a large amount of training data. (3) The proposed framework is extremely fast both in training and testing, i.e. training is 6000 folds faster than the best-performing method, whereas the testing is 9 times faster. We evaluate the proposed algorithm on the problems of image set-based face recognition and object categorization on five benchmark datasets including Honda/UCSD [25], CMU Mobo [7], YouTube Celebrities [23], Celebrity-1000 [28] and ETH-80 [26]. Results demonstrate that the proposed algorithm consistently outperforms existing methods in terms of accuracy, while achieving substantial speedups at the same time.

2 Related work

Image set classification methods can be divided into two major categories. The first one is sample based, whereas the second one is structure based. The former uses the nearest neighbours of two image sets to compute the set-to-set distance under some predefined constraints. For example, Cevikalp and Triggs [3] defined affine hull image set distance (AHISD) and convex hull image set distance (CHISD) that measured the affine and convex hull distances, respectively, between two image sets. They used a convex or affine geometric region to represent image sets. For AHISD, the distance between the models was minimized by using least squares, whereas in the case of CHISD, an SVM was used to separate the two sets. Hu et al. [13] modelled image sets jointly as affine hulls and image samples. Nearest points on the affine hulls are calculated through convex optimization such that each point was also a *sparse* combination of the respective image set samples. Distance between the two sparse approximated nearest points (SANPs) was used for classification. Each SANP is situated close to some facet of its affine hull. Thus, similar sets have smaller distance. Later, Mian et al. [34] introduced the constraints of self-regularization and non-negativity to define more accurate between set distance. Mahmood et al. [33] performed spectral clustering on the combined gallery and test samples. The class-cluster distributions of the set samples were then used for classification. Lu et al. [30] jointly learn a structured dictionary and projection matrix to map set samples into a low-dimensional subspace. The low-dimensional samples were then represented using sparse codes and classification was performed based on the traditional minimum reconstruction error and majority voting scheme. In general, sample-based methods are highly susceptible to outliers and have high computational cost for large galleries.

The second category of image set classification techniques is structure based. The techniques in this category model image set structures with linear subspaces and measure the distance between subspaces for classification. The discriminant canonical correlation (DCC) [22] method used the canonical correlations between the sets to perform linear discriminant analysis. Manifold–manifold distance (MMD) [43] modelled an image set as more than one local clusters where the clusters are computed such that each one can be approximated by a linear subspace. Sparse approximated nearest subspaces (SANS) [4] extracted local clusters, via sparse representation, from the training image sets. The test image set clusters are forced to resemble those in the training sets and only corresponding clusters are matched through the subspace-based distance.

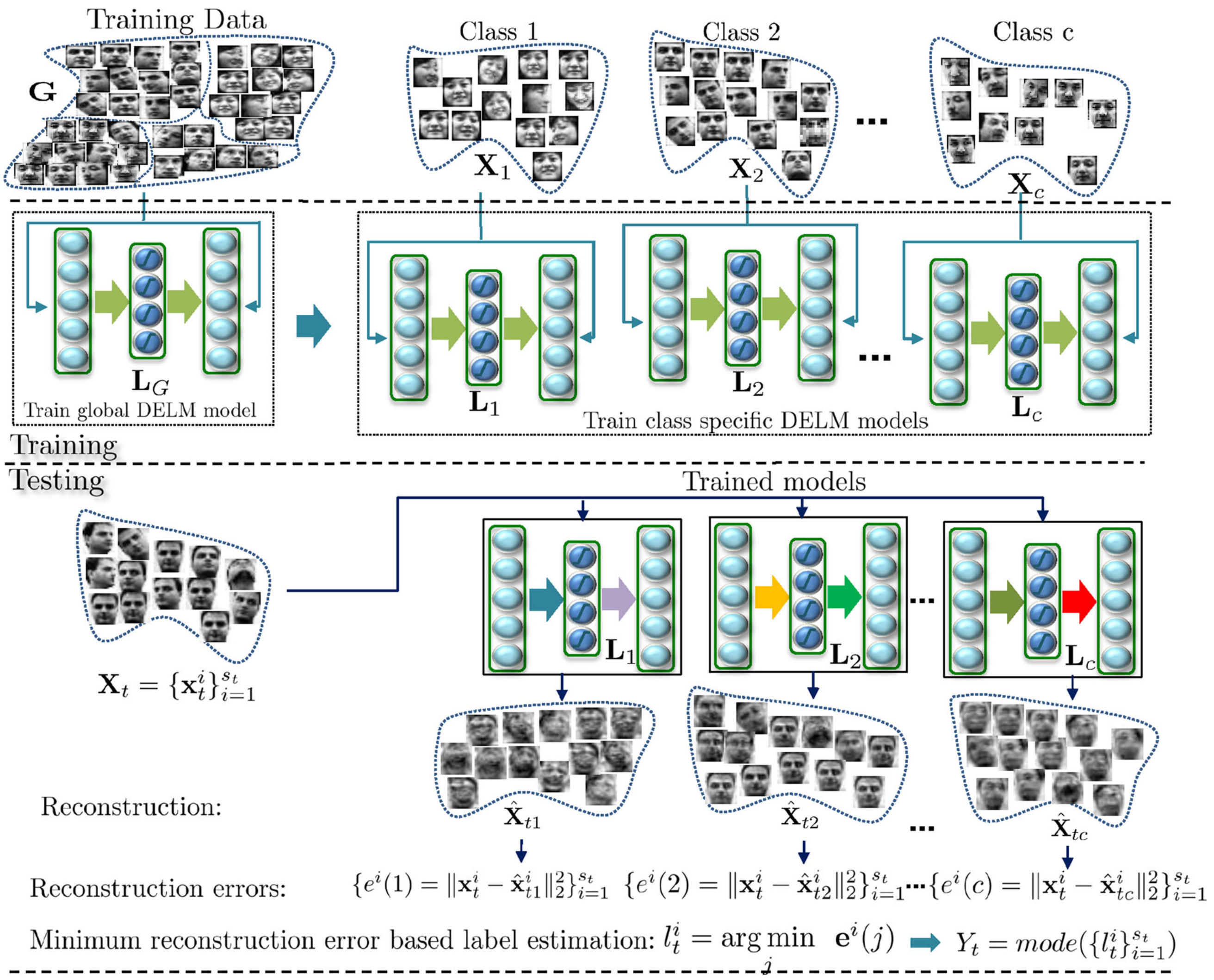
Training Data
G
Class 1
Class 2
Class c
X_1
X_2
X_c
L_G
Train global DELM model
L_1
L_2
L_c
Train class specific DELM models
Training
Testing
Trained models
$X_t = \{x_t^i\}_{i=1}^{s_t}$
Reconstruction:
$\hat{X}_{t1}$
$\hat{X}_{t2}$
$\hat{X}_{tc}$
Reconstruction errors: $\{e^i(1) = \|x_t^i - \hat{x}_{t1}^i\|_2^2\}_{i=1}^{s_t}$ $\{e^i(2) = \|x_t^i - \hat{x}_{t2}^i\|_2^2\}_{i=1}^{s_t}$ $\{e^i(c) = \|x_t^i - \hat{x}_{tc}^i\|_2^2\}_{i=1}^{s_t}$
Minimum reconstruction error based label estimation: $l_t^i = \arg\min_j \mathbf{e}^i(j)$ $Y_t = mode(\{l_t^i\}_{i=1}^{s_t})$

Fig. 1 Proposed image set classification algorithm. During training, we first learn a domain-specific deep extreme learning machines (DELM) model $\mathbf{L}_G$. Starting from the domain-specific model, we then learn class-specific DELM models $\mathbf{L}_j$ for the gallery sets of each class separately. Each sample of a probe image set $\mathbf{X}_t$ is first reconstructed with all the learned DELM models and its label is estimated using on the minimum reconstruction error. The label of the image set as a whole is estimated using majority voting

Manifold discriminant analysis (MDA) [41] modelled an image set using multiple locally linear clusters which were then transformed by a linear discriminant operator for class separation. Image set structure was also modelled by Harandi et al. [10] with linear subspaces which were considered as points on the Grassmannian manifold. This method defined kernels that mapped points lying in the Grassmannian manifold back to the Euclidean space where graph-embedding discriminant analysis is used for classification. Similarly, covariance discriminative learning (CDL) [42] modelled the image set structure with a covariance matrix and mapped the covariance matrix from the Riemannian manifold to the Euclidean space using the Log-Euclidean distance kernel function. A regression function was then learned using kernel partial least squares to perform image set classification.

Deep learning has also been employed for learning the image set structure. For example, Hayat et al. [12] learned the structure of each gallery image set using a deep learning model. The label of the probe set was then estimated using the minimum reconstruction error and majority voting scheme. Lu et al. [31] also represented image sets with a deep model and used metric learning to further maximize the margin between different classes in a shared nonlinear feature space. Some recent structure-based techniques are based on learning a Riemannian metric directly on the Riemannian manifold without kernel mapping, where the image set models are subspaces or covariance matrices to characterize the set data structure. For example, Harandi et at. [11] proposed a manifold-based dimensionality reduction method. Unlike previous methods that flatten the manifold through kernel embedding, this method works on the original manifold to produce better low-dimensional representations of image sets. Huang et al. [18] learned a projection metric directly on Grassmann manifold without mapping the manifold in Hilbert space. Thus, the image sets can be represented on a more discriminative Grassmannian manifold. Similarity, Huang et al. [19] represented image sets with symmetric positive definite (SPD) matrices and learned a log-Euclidean metric by directly manipulating the original SPD matrix logarithm without its vectorization. Some structure-based methods exploit more flexible statistical models to learn the set structure. For example, Wang et al. [44] represented image sets with the more flexible Gaussian mixture models (GMM) and proposed discriminant analysis on the Riemannian manifold of Gaussian distributions for classification. Harandi et al. [9] modelled the set structure with probability distribution functions (PDFs) via kernel density estimation. The models are then matched using the Csiszar f-divergences. Structure-based techniques are more powerful, however, compared to sample-based techniques; they generally require larger number of samples per set (dense sampling) for the accurate modelling of the underlying set structure.

We propose a structure-based image set classification algorithm that neither makes prior assumptions about the set structure nor incur a heavy computational burden to learn the structure from the data. The proposed representation is based on deep extreme learning machines and is capable of automatically learning the *nonlinear* structure of image sets. The proposed algorithm is extremely efficient to train and generalizes very well even with a small number of training samples.

3 Proposed methodology

We first give a brief overview of extreme learning machines (ELMs) and how they differ from other learning paradigms. Then, we discuss how to extend the traditional ELM idea to multiple layers, thus, allowing a deeper representation. Finally, we show how image set classification can be formulated using the deep ELM (DELM) models and how it can benefit from ELM's attractive properties, namely very efficient learning (easily scalable to large datasets) and generalizability (no prior assumptions on the set data).

3.1 Extreme learning machines

Consider a supervised learning problem with N training samples, $\{\mathbf{X}, \mathbf{T}\} = \{\mathbf{x}_j, \mathbf{t}_j\}_{j=1}^{N}$ where $\mathbf{x}_j \in \mathbb{R}^d$ and $\mathbf{t}_j \in \mathbb{R}^q$ are the j^{th} input and target samples, respectively. d and q are the input and target feature dimensions, respectively. For the task of classification, $\mathbf{t}_j$ is the class label vector while for regression $\mathbf{t}_j$ represents the desired output feature. In either case, we seek a regressor function from the inputs to the targets. A well-known type of this function is the single hidden layer feedforward neural network (SLFN), where n_h hidden nodes fully connect the d inputs to the q outputs. This is done through an activation function $g(u)$. The predicted output vector $\mathbf{o}_j$ generated by feeding forward $\mathbf{x}_j$ through an SLFN is mathematically modelled as

$$\mathbf{o}_j = \sum_{i=1}^{n_h} \boldsymbol{\beta}_i g(\mathbf{w}_i^{\top}\mathbf{x}_j + b_i) \tag{1}$$

where $\mathbf{w}_i \in \mathbb{R}^d$ is the weight vector that connects the i-th hidden node to the input nodes, $\boldsymbol{\beta}_i \in \mathbb{R}^q$ is the weight vector that connects the i-th hidden node to the output nodes, and b_i is the bias of the i-th hidden node. The activation function $g(u)$ can be any nonlinear piecewise continuous function [27], e.g. the sigmoid function $g(u) = \frac{1}{1+e^{-u}}$.

An ELM learns the parameters of an SLFN (i.e. $\{\mathbf{w}_i, b_i, \boldsymbol{\beta}_i\}_{i=1}^{n_h}$) in two sequential stages: random feature projection and linear parameter solving [17, 24, 35, 40, 45]. In the first ELM stage, the hidden layer parameters ($\{\mathbf{w}_i, b_i\}_{i=1}^{n_h}$) are randomly initialized to project the input data to a random ELM feature space using the mapping function $g()$. It is this random projection stage that differentiates ELM from most existing learning paradigms, which perform deterministic feature mapping. For example, an SVM uses kernel functions, while deep neural networks [1] use restricted Boltzmann machines (RBM) for feature mapping/learning. By randomizing the feature mapping stage, the ELM can discover nonlinear structures in the data without the need for priors, which are inherently the case for deterministic feature mapping schemes. Also, these parameters are set randomly and are not subsequently updated, thus decoupling them from the output parameters $\{\boldsymbol{\beta}_i\}_{i=1}^{n_h}$, which can be learned in a very efficient manner as we will see next. This decoupling strategy significantly speeds up the parameter learning process in ELM, thus, making it much more computationally attractive than deep neural network architectures that learn *all* network parameters *iteratively*.

In the second ELM stage, the parameters that connects the hidden layer to the output layer (i.e.$\{\boldsymbol{\beta}_i\}_{i=1}^{n_h}$) are learned efficiently using regularized least squares. Here, we denote $\psi(\mathbf{x}_j) = [g(\mathbf{w}_1^{\top}\mathbf{x}_j + b_1)\ldots g(\mathbf{w}_{n_h}^{\top}\mathbf{x}_j + b_{n_h})] \in \mathbb{R}^{1\times n_h}$ as 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. An ELM aims to solve for $\mathbf{B}$ by minimizing the sum of the squared losses of the prediction errors:

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

In (2), the first term is a regularizer against over-fitting, $\mathbf{e}_j \in \mathbb{R}^q$ is the error vector for the j-th training example (i.e. $\mathbf{e}_j = \mathbf{t}_j - \mathbf{o}_j$), and C is a tradeoff coefficient. By concatenating $\mathbf{H} = [\psi(\mathbf{x}_1)^\top \cdots \psi(\mathbf{x}_N)^\top]^\top \in \mathbb{R}^{N\times n_h}$ and $\mathbf{T} = [\mathbf{t}_1 \cdots \mathbf{t}_N]^\top \in \mathbb{R}^{N\times q}$, we obtain an equivalent unconstrained optimization problem, which is widely known as ridge regression or regularized least squares.

$$\min_{\mathbf{B}\in\mathbb{R}^{n_h\times q}} \frac{1}{2}\|\mathbf{B}\|_F^2 + \frac{C}{2}\|\mathbf{T} - \mathbf{H}\mathbf{B}\|_2^2, \quad (3)$$

Since the above problem is convex, its global solution needs to satisfy the following linear system:

$$\mathbf{B} + C\mathbf{H}^\top(\mathbf{T} - \mathbf{H}\mathbf{B}) = \mathbf{0}. \quad (4)$$

The solution to this system depends on the nature and size of matrix $\mathbf{H}$. If the number of rows of $\mathbf{H}$ is greater than its number of columns and $\mathbf{H}$ is of full column rank (which is usual when $N > n_h$), the system is overdetermined and a closed form solution exists for (3) in (5), where $\mathbf{I}_{n_h}\mathbb{R}^{n_h\times n_h}$ is an identity matrix. Note that in practice, rather than explicitly inverting the $n_h \times n_h$ matrix, we obtain $\mathbf{B}^*$ by solving the linear system in a more efficient and numerically stable manner.

$$\mathbf{B}^* = \left(\mathbf{H}^\top\mathbf{H} + \frac{\mathbf{I}_{n_h}}{C}\right)^{-1}\mathbf{H}^\top\mathbf{T} \quad (5)$$

If $N < n_h$, $\mathbf{H}$ will be having more columns than rows. This leads to an under-determined least squares problem and $\mathbf{B}$ may have infinite number of solutions. In such case, we can restrict $\mathbf{B}$ to be a linear combination of the rows of $\mathbf{H} : \mathbf{B} = \mathbf{H}^\top\boldsymbol{\alpha}$ ($\boldsymbol{\alpha} \in \mathbb{R}^{N\times q}$). Note that when the number of columns of $\mathbf{H}$ is greater than its rows and $\mathbf{H}$ is of full row rank, then $\mathbf{H}\mathbf{H}^\top$ is invertible. By multiplying both sides of (4) by $(\mathbf{H}\mathbf{H}^\top)^{-1}\mathbf{H}$, we obtain a closed form solution for $\mathbf{B}^*$

$$\mathbf{B}^* = \mathbf{H}^\top\boldsymbol{\alpha}^* = \mathbf{H}^\top\left(\mathbf{H}\mathbf{H}^\top + \frac{\mathbf{I_N}}{C}\right)^{-1}\mathbf{T} \quad (6)$$

To summarize, ELMs have two major attractive properties. Firstly, the parameters of the hidden mapping function can be randomly generated according to any continuous probability distribution, e.g. the uniform distribution on $[-1, 1]$. Secondly, as such, the only parameters that are to be learned during training are the weights between the hidden nodes and the output nodes. This is efficiently done by solving a single linear system or even in closed form. These two properties make ELMs more flexible than SVMs and much more computationally attractive than the conventional feed-forward neural networks that use back-propagation [15].

The theoretical foundations of ELMs have been explored recently by many researchers. Liu et al. [29] have recently explained that ELM has the capability to achieve the theoretical generalization bound of the feedforward neural networks even when the weights of the hidden layer neurons are set randomly. Huang [14] have recently provided a detailed theoretical explanation of the key characteristics of ELMs which differentiates them from other learning algorithms for the feedforward neural networks. Moreover, ELM has also been extended by Huang et al. [14] for learning powerful deep features in a hierarchical manner with low computational burden. Therefore, ELM has found many successful applications in feature learning, clustering, regression and classification [14].

3.2 Learning representations with ELMs

Learning rich representations efficiently is crucial for achieving high generalization performance, especially at large scales. This form of learning can usually be done using auto-encoders, where a parametric regressor function is learned to map the input to itself. Although deep neural networks can be learned for this purpose and have been shown to achieve exceptional performance in many computer vision tasks [1, 2], they are generally very slow in training. We use ELM-based auto-encoders [21] to perform unsupervised learning of image set representations. ELMs are computationally very fast to train. A deep ELM is essentially a multiple-layer neural network whose parameters are learned by training a cascade of multiple ELM layers. Such a learning procedure is highly efficient in learning time and has good generalization capabilities.

Figure 2 illustrates the deep extreme learning machine (DELM) learning process given the training set samples $\mathbf{X}$. A DELM auto-encoder is designed by setting the targets of the multi-layer network to the input, i.e. $\mathbf{T} = \mathbf{X}$. Here, a fully connected multi-layer network with h hidden layers is considered. Let $\mathbf{L} = \{\mathbf{W}^1, \ldots, \mathbf{W}^{h+1}\}$ denote the DELM parameters, where $\mathbf{W}^i = [\mathbf{w}_1^i, \ldots, \mathbf{w}_{n_i}^i]^\top \in \mathbb{R}^{n_{i+1}\times n_i}$. Each layer is decoupled from the network and processed as an ELM to simplify its training. To train individual ELM-AE, the targets are set the same as the inputs. For example in Fig. 2, $\mathbf{W}^1$ is learned using the corresponding ELM with $\mathbf{T} = \mathbf{X}$. The weight vectors that connect the input layer to the first hidden layer are orthonormal, effectively projecting the input data to a random subspace. Orthogonalization of these random weights tends to better preserve pairwise distances in the random ELM feature space [20] compared to initializing random weights independently, and at the same time improves the ELM auto-encoder generalization

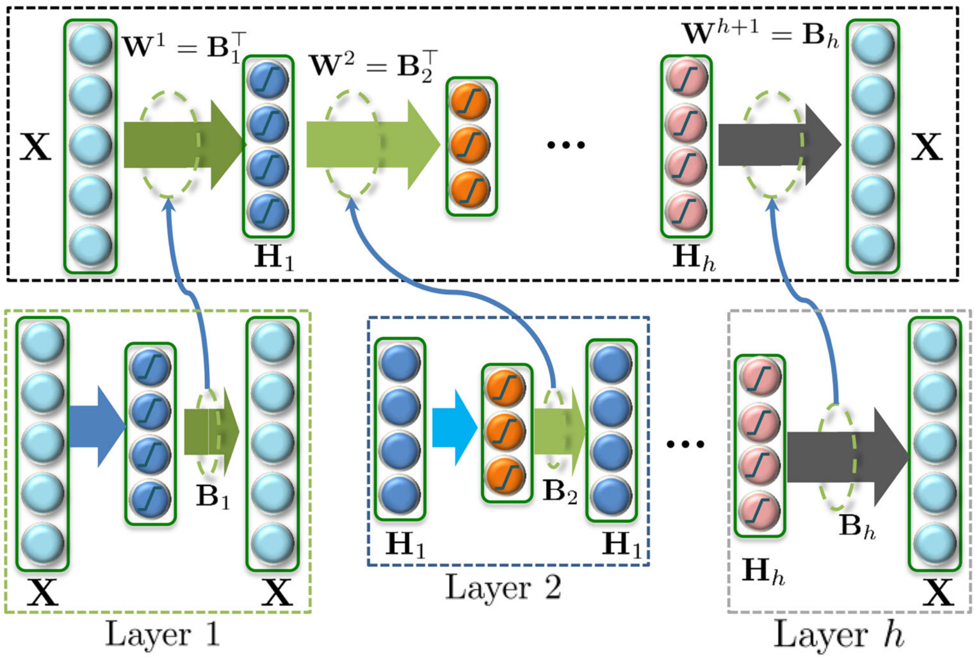

Fig. 2 Layerwise training of a deep ELM model with h hidden layers and input $\mathbf{X}$

performance. In the next step, depending on the number of hidden layer nodes, (5) or (6) is used to calculate $\mathbf{B}_1$. Note that, $\mathbf{B}_1$ re-projects the low-dimensional representation of the input data back to its original space while minimizing the reconstruction error. Therefore, this projection matrix is data-driven and hence used as the weights of the first layer ($\mathbf{W}^1 = \mathbf{B}_1^\top$). Similarly, $\mathbf{W}^2$ is learned by setting the input and output of Layer 2 to $\mathbf{H}_1$ i.e. the output of Layer 1. In this manner, all parameters of the DELM are computed sequentially. However, when the number of nodes in two consecutive layers is equal, the random projection obtained in the second layer is in the same feature space as the input of the first layer. Using (5) or (6) does not ensure orthogonality of the computed weight matrix $\mathbf{B}$. Imposing orthogonality in this case results in a more accurate solution since the data always lie in the same space. Therefore, the output weights $\mathbf{B}$ are calculated as the solution to the Orthogonal Procrustes problem

$$\begin{aligned} &\mathbf{B}^* = \min_{\mathbf{B}\in\mathbb{R}^{n_h\times q}} \|\mathbf{HB} - \mathbf{T}\|_F^2, \\ &s.t.\ \ \mathbf{B}^\top\mathbf{B} = \mathbf{I}. \end{aligned} \tag{7}$$

The closed form solution is obtained by finding the nearest orthogonal matrix to the given matrix $\mathbf{M} = \mathbf{H}^\top\mathbf{T}$. To find the orthogonal matrix $\mathbf{B}^*$, we use the singular value decomposition $\mathbf{M} = \mathbf{U\Sigma V}^\top$ to compute $\mathbf{B}^* = \mathbf{UV}^\top$.

In ELM-AE, the orthogonal random weights and biases of the hidden nodes project the input data to a different or equal dimension space. The DELM models can automatically learn the nonlinear structure of data in a very efficient manner. In contrast to deep networks, DELM also does not require expensive iterative fine tuning of the weights.

3.3 Deep ELM models for image set classification

DELM-based image set classification has two main steps. Firstly, we learn a global domain-specific DELM model using all the training image data and then build class-specific DELM models using the global representation as an initialization. In doing so, we encode both domain level and class-specific properties of the data.

Define $\mathbf{G} = \{\mathbf{X}_m\}_{m=1}^c \in \mathbb{R}^{d\times N}$ as the gallery of c image sets (c classes) with a total of N images: $N = \sum_{m=1}^c s_m$, where s_m is the number of samples in the m-th image set defined by $\mathbf{X}_m = \{\mathbf{x}_m^i\}_{i=1}^{s_m} \in \mathbb{R}^{d\times s_m}$ (where $\mathbf{x}_m^i \in \mathbb{R}^d$ is a d-dimensional features obtained by vectorizing the pixels of the i-th image). The vector $\mathbf{x}_m^i$ may also contain features such as Histogram of Oriented Gradients (HOG) or PCA coefficients instead of pixel values. Note that s_m can be different for different image sets; however, the $\mathbf{x}_m^i$ dimensionality is the same. Let $\mathbf{Y} = \{y_m\}_{m=1}^c$ be the class labels of the image sets in $\mathbf{G}$. For a probe (test) image set $\mathbf{X}_t = \{\mathbf{x}_t^i\}_{i=1}^{s_t} \in \mathbb{R}^{d\times s_t}$, the problem of image set classification involves estimating the label Y_t of $\mathbf{X}_t$ given the gallery $\mathbf{G}$.

Training We learn a global domain-specific DELM model by initializing its weights using the ELM auto-encoding procedure described earlier. This global DELM is a multi-layer neural network with h hidden layers. Its parameters are learned using the images in $\mathbf{G}$ in an unsupervised manner. The global DELM model is represented as $\mathbf{L}_G = \{\mathbf{W}_G^1, \ldots, \mathbf{W}_G^{h+1}\}$, where $\mathbf{W}_G^i$ denotes the weight matrix of the i^{th} layer learned using the auto-encoding method in Sect. 3.2. The global DELM model serves as a starting point, from which we learn class-specific DELM models.

Since $\mathbf{L}_G$ encodes domain-specific representation (as it has been trained to reconstruct any sample from that domain), we use it to learn a separate DELM model for each of the c training classes. In other words, instead of randomly initializing the hidden layers weights, as in the conventional ELM, we use the weights in $\mathbf{L}_G$ to initialize the class-specific models. Thus, we have c DELM models for c classes $\{\mathbf{L}_j\}_{j=1}^{c}$, where each class-specific model is represented as $\mathbf{L}_j = \{\mathbf{W}_j^1, \ldots, \mathbf{W}_j^{h+1}\}$.

The learned ELM models are able to encode complex nonlinear structure of the training data due to their deep architecture with multiple nonlinear layers. Compared to the previous structure-based algorithms such as DCC [22], GGDA [10] and CDL [42], our proposed DELM models learn the structure of the image data in multiple parameters, therefore, it is capable of learning more complex structure on nonlinear manifolds. Moreover, this DELM model is more computationally efficient than previous methods.

Testing Given a test image set $\mathbf{X}_t = \{\mathbf{x}_t^i\}_{i=1}^{s_t}$, we predict its label by first representing each image in this set using each of the class-specific representations $\{\mathbf{L}_j\}_{j=1}^{c}$ and assigning each image to the class that incurs the least reconstruction error. Then, majority voting on the predicted image-level classes is performed to predict the class of the image set. The overall procedure is summarized in Algorithm 1.

We reconstruct each test image $\mathbf{x}_t^i$ in the set using each of the class-specific models $\{\mathbf{L}_j\}_{j=1}^{c}$. The reconstructed sample from model $\mathbf{L}_j$ is denoted by $\hat{\mathbf{x}}_{tj}^i$ and is given by

$$\hat{\mathbf{x}}_{tj}^i = f(\mathbf{x}_{tj}^i, \mathbf{L}_j) = \mathbf{W}_j^{h+1} g(\mathbf{W}_j^h, \ldots, g(\mathbf{W}_j^1 \mathbf{x}_t^i)) \tag{8}$$

where f is the reconstruction and g is chosen to be the sigmoid function. The reconstruction error of sample $\mathbf{x}_t^i$ is computed as the Euclidean distance between $\mathbf{x}_t^i$ and $\hat{\mathbf{x}}_{tj}^i$ as $e^i(j) = \|\mathbf{x}_t^i - \hat{\mathbf{x}}_{tj}^i\|_2$. The predicted label l_t^i for sample $\mathbf{x}_t^i$ is chosen to be the class that incurs the minimum reconstruction error

$$l_t^i = \arg\min_j \mathbf{e}^i(j). \tag{9}$$

Finally, the test image set $\mathbf{X}_t$ is labelled using majority voting on the set of predicted image-level labels. Formally, we set the image set label $Y_t = \text{mode}(\{l_t^i\}_{i=1}^{s_t})$.

4 Experiments and results

We perform extensive experiments on five public datasets (see Fig. 3) and compare our results to 14 state-of-the-art image set classification methods. These datasets have been widely used in the literature to evaluate image set-based classification algorithms. Details of the datasets used, experimental protocol, and results obtained are provided next.

Algorithm 1 Proposed Image Set Classification Algorithm

```
Input: :
  Gallery $\mathbf{G} = \{\mathbf{X}_m\}_{m=1}^{c}$ containing $c$ image sets $\mathbf{X}_s = \{\mathbf{x}_m^i\}_{i=1}^{s_m} \in \mathbb{R}^{d \times s_m}$ belonging to $c$ classes
  Class labels $\mathbf{Y} = \{y_m\}_{m=1}^{c}$
  Probe set $\mathbf{X}_t = \{\mathbf{x}_t^i\}_{i=1}^{s_t} \in \mathbb{R}^{d \times s_t}$
  Number of hidden layers $h$
Output: : Label $Y_t$ of $\mathbf{X}_t$
  Training:
  $\mathbf{L}_G = \{\mathbf{W}_G^1, ..., \mathbf{W}_G^{h+1}\}$ {Learn a domain-specific global DELM model with $h$ hidden layers from $\mathbf{G}$}
  for $j = 1 : c$ do
    $\mathbf{L}_j = \{\mathbf{W}_j^1, ...\mathbf{W}_j^{h+1}\}$ { Learn DELM models for each class}
  end for
  Testing:
  for $i = 1 : s_t$ do
    for $j = 1 : c$ do
      $\hat{\mathbf{x}}_{tj}^i = f(\mathbf{x}_t^i; \mathbf{L}_j)$ {Reconstruct from model $\mathbf{L}_j$ (8)}
      $e^i(j) = \|\mathbf{x}_t^i - \hat{\mathbf{x}}_{tj}^i\|_2$
    end for
    $l_t^i \triangleq \arg\min_j \ \mathbf{e}^i(j)$
  end for
  $Y_t \triangleq \text{mode}(\{l_t^i\}_{i=1}^{s_t})$
```

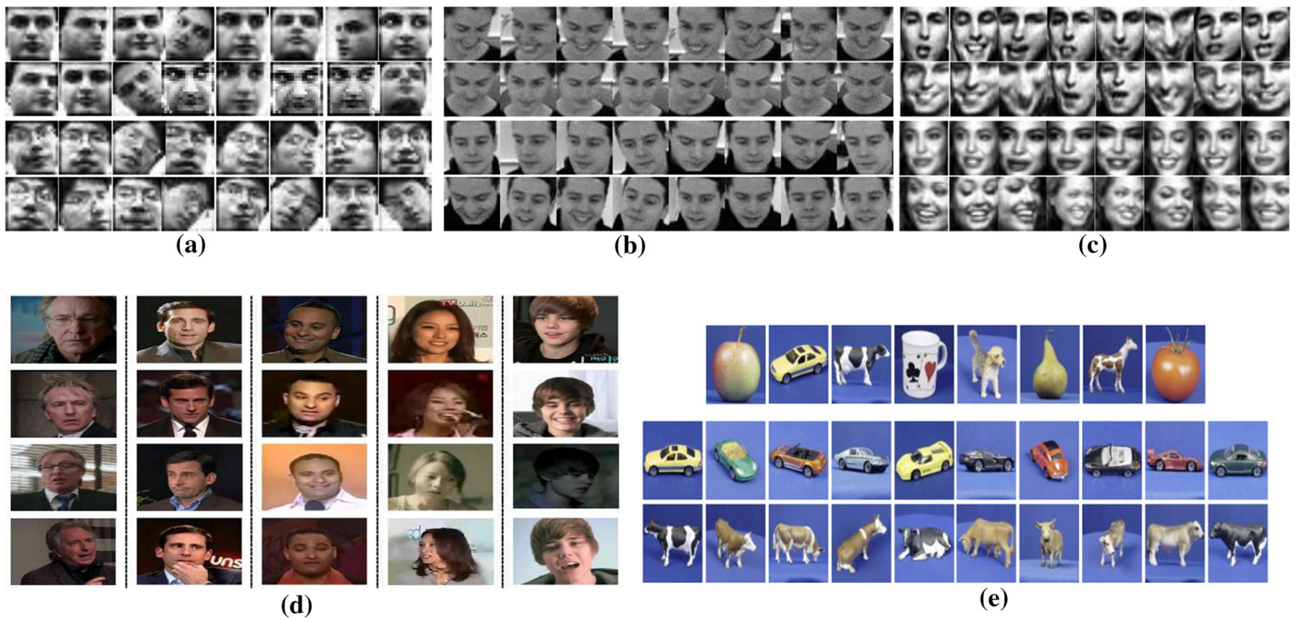

Fig. 3 Example image sets from **a** Honda, **b** CMU Mobo, **c** YouTube Celebrities, **d** exemplar video frames from the Celebrity-1000 dataset, **e** eight object categories and 10 different objects in one category of the ETH-80 dataset

4.1 Dataset specifications

The *Honda/UCSD dataset* [25] comprises 59 videos containing faces of 20 different individuals. Each video contains one face. This database was collected for evaluating the performance of face tracking and recognition methods. The videos are recorded indoor at 15 frames per second. The minimum length of a video sequence is around 15 s and a minimum of two videos are available per individual. The faces in the videos contain significant pose and illumination variations due to head rotations of the subjects. For consistency with prior research that used the Honda/UCSD data for face recognition experiments, we use 20×20 histogram equalized face images extracted from the videos using the Viola and Jones face detection method [39]. In our experiments, the face images detected in each video sequence form an image set.

The *CMU Motion of Body (MoBo) dataset* [7] consists of video sequences of individuals walking on a treadmill. This dataset contains 96 videos of 24 different individuals. Since the subjects walk in four different styles (slow, fast, incline, walk while holding an object), their faces contain significant pose, illumination and image resolution variations. We detect face images in the videos using the Viola and Jones face detection algorithm [39] and use LBP features of the face images similar to [3] in our experiments.

The *YouTube Celebrities* [23] is a challenging dataset that was initially collected for benchmarking the performance of face tracking methods but has also been widely used since then for performance evaluation of face recognition algorithms. This dataset was collected from YouTube and consists of 1910 videos of 47 celebrities such as actors, actresses, players and politicians. Most videos are of low resolution and contain significant compression artefacts, facial pose, illumination and expression variations. There are upto 400 frames per video sequence. To detect faces in YouTube Celebrities, we track a face in each video sequence with the algorithm in [36] due to its high accuracy. To initialize tracking, we use the location of the face window in the first frame which was provided with this dataset. After successful detection and tracking, the face regions are cropped, converted to grey scale and then resized to 20×20. We use the LBP features ($d = 928$) of 20×20 face images for image set classification.

The *Celebrity-1000 database* [28] is a large-scale unconstrained video database downloaded from YouTube and Youku. It contains 159,726 face video sequences of 1000 individuals covering a wide range of poses, illuminations, expressions and image resolutions. We follow the standard closed-set test protocol defined in [28] where four overlapping subsets of the dataset are created with increasing complexity containing 100, 200, 500, 1000 subjects. Each subset is further divided into training and test partitions with disjoint video sequences. Approximately 70% of the sequences are randomly selected to construct the gallery and the rest are used as test sets. We use the PCA reduced LBP+Gabor features provided by Liu et al. [28]. The feature dimension d is 1651, 1790, 1815 and 1854 for the subsets 100, 200, 500 and 1000, respectively.

The *ETH-80 Object Categorization dataset* [26] consists of eight different object categories (apple, cow, dog, cup, pear, tomato, horse and car). Each object category contains 10 different instances of the same class. Each object instance has 41 images, captured from multiple viewpoints, to make an image set. The images are cropped to 256×256 so that the object is in the centre with 20% border area. We re-scale the images to 20×20 and convert them to greyscale in our experiments for image set-based object categorization. ETH-80 is challenging because the number of images in each set is low. Moreover, the objects have significant within class appearance variations due to large differences in viewing angle.

4.2 Experimental setup

We follow the standard experimental protocol [3, 12, 13, 41–43] for a fair comparison with 14 state-of-the-art algorithms including discriminant canonical correlation (DCC) [22], manifold-manifold distance [43], manifold discriminant analysis (MDA) [41], affine and convex hull-based image set distance (AHISD, CHISD) [3], sparse approximated nearest points (SANP) [13], covariance discriminative learning (CDL) [42], graph-embedding Grassmannian discriminant analysis (GGDA) [10], set-to-set distance metric learning (SSDML) [46], nonlinear reconstruction models (NLRM) [12], geometry-aware dimensionality reduction (GADR) [11], projection metric learning on Grassmann manifold (PMLGM) [18] and log-Euclidean metric learning (LEML) [19]. We use the source codes supplied by the original authors of all compared algorithms.

We tuned the parameters of all the algorithms empirically to optimize their performances. In the case of DCC [22], a subspace dimension is set to 10, leading to the maximum canonical correlations of 10. The parameters of MMD and MDA are chosen as recommended by the original authors [41, 43]. More precisely, the Euclidean to geodesic distance ratio is chosen within the range of $\{1.0–5.0\}$ for different datasets and maximum canonical correlation is used to define MMD. Twelve connected NNs are used for calculating the geodesic distances in MMD and MDA. The PCA energy in CHISD, AHISD and SANP is selected from $\{80, 85, 90, 95, 99\%\}$ and the best performances are reported for each dataset. The error penalty parameter C is set to 100 in CHISD. We used $k^{[cc]} = 1\, k^{[proj]} = 100$ and $v = 3$ in GGDA. The number of eigenvectors used to represent an image set in Mobo dataset was 9, in YouTube Celebrities dataset was 6 and for all other datasets, it was 10. CDL and SSDML do not require any parameter tuning. PLS was used as a classifier with CDL. For NLRM [12], we used the network depth and model parameters as recommended by the authors. For GADR [11], the number of nearest neighbours v_w were set to the minimum number of samples in each class and v_b was set to 6. We used the Stein kernel-based NN classifier on the low-dimensional SPD manifold. For PMLGM [18], we used the Grassmannian graph-embedding discriminant analysis for classification and searched the parameter β in the range of $\{1e^2 - 1e^6\}$ and set α to 0.2. For LEML [19], we used the CDL-PLS model and set the parameters as recommended by the authors. The parameter η was searched in the range $\{0.1,1,10\}$ and ζ in the range $\{0.1–0.5\}$. The parameters of our algorithm include the number of hidden layers h, the number of neurons in each hidden layer n_h and the parameters C. We set the number of hidden layers $h = 2$ for all datasets. The parameter C was chosen in the range $\{10^4–10^8\}$ for the first layer and $\{10^{16}–10^{20}\}$ for the last layer. The number of neurons in each hidden layer n_h was 20 for Honda, Mobo and Celebrity-1000, 40 for YouTube, 150 for ETH80 dataset.

One video sequence per subject was chosen to construct the gallery, and the rest of the video sequences were chosen as probes for the Honda and MoBo datasets. However, DCC learning requires at least two image sets for each class in the gallery. Therefore, we randomly partitioned single gallery image sets into two non-overlapping subsets. We conducted 10 repeated experiments with different gallery and probe combinations in each experiment (fold). For the YouTube Celebrities dataset, we conduct fivefold cross-validation experiments similar to [13]. The videos are divided into nine image sets per subject and each time, three image sets are randomly selected per subject for training and the rest are used for testing. ETH-80 dataset has five image sets per class in the gallery for training and the remaining five sets for testing.

4.3 Results and analysis

Table 1 reports the average and standard deviation recognition rate (%) for tenfold experiments on Honda, Mobo and ETH datasets and fivefold experiments on the YouTube dataset. Our approach performs better than competing algorithms on YouTube celebrities, CMU Mobo and ETH-80 datasets and achieves perfect results on the Honda dataset. Recall that our algorithm involves no supervised discriminative analysis as in DCC, MDA, CDL, GGDA,-GADR, PMLGM and LEML, yet it performs better in both accuracy and execution time. On the ETH-80 dataset, structure-based algorithms [10–12, 19, 22, 41–43] achieve better accuracy than the sample-based ones [3, 13, 46] because the individual samples cannot model significant intra-class pose and object appearance variations.

Table 2 summarizes the image set classification results on all the splits of the Celebrity-1000 dataset. On the

Table 1 Comparison of the average classification accuracies and standard deviations (%) (results are obtained by performing tenfold experiments for Honda, Mobo and ETH datasets and fivefold for YouTube Celebrities dataset)

	Honda	MoBo	ETH-80	Youtube
DCC [22] (TPAMI 2007)	94.67 ± 1.32	93.61 ± 1.76	90.91 ± 5.31	66.75 ± 4.47
MMD [43] (CVPR 2008)	94.87 ± 1.16	93.19 ± 1.66	85.73 ± 8.33	65.12 ± 4.36
MDA [41] (CVPR 2009)	97.44 ± 0.91	95.97 ± 1.90	80.50 ± 6.81	68.12 ± 4.85
GGDA [10] (CVPR 2011)	94.61 ± 2.07	85.75 ± 1.82	85.75 ± 6.41	62.81 ± 4.42
CDL [42] (CVPR 2012)	100.0 ± 0.00	95.83 ± 2.07	88.20 ± 6.80	68.96 ± 5.29
GADR [11] (ECCV 2014)	96.78 ± 2.32	96.11 ± 1.34	95.50 ± 4.04	69.83 ± 4.39
PMLGM [18] (CVPR 2015)	100 ± 0.00	96.25 ± 1.18	94.50 ± 5.32	70.89 ± 4.66
LEML [19] (ICML 2015)	100 ± 0.00	95.56 ± 1.01	92.75 ± 5.94	70.23 ± 4.71
AHISD [3] (CVPR 2010)	89.74 ± 1.85	94.58 ± 2.57	74.76 ± 3.31	71.92 ± 4.55
CHISD [3] (CVPR 2010)	92.31 ± 2.12	96.52 ± 1.18	71.00 ± 3.93	73.17 ± 4.69
SANP [13] (TPAMI 2012)	93.08 ± 3.43	97.08 ± 1.03	72.43 ± 4.98	74.01 ± 4.68
SSDML [46] (ICCV 2013)	89.41 ± 3.64	95.14 ± 2.20	81.00 ± 6.58	70.81 ± 3.42
NLRM [12] (CVPR 2014)	100.0 ± 0.0	97.92 ± 1.76	95.25 ± 4.77	73.55 ± 4.74
DELM (this paper)	**100.0** ± **0.0**	**98.00** ± **0.67**	**96.00** ± **3.51**	**75.31** ± **4.63**

Bold values indicate the best accuracies

Table 2 Comparison of the classification accuracy on different subsets of Celeb-1000 dataset

	Subset-100	Subset-200	Subset-500	Subset-1000	Average
DCC [22]	25.24	10.38	10.18	–	–
MMD [43]	17.52	10.23	9.79	–	–
MDA [41]	15.93	9.21	9.87	–	–
GGDA [10]	11.95	8.24	9.64	–	–
CDL [42]	11.95	11.11	10.65	–	–
AHISD [3]	19.92	23.94	18.97	–	–
CHISD [3]	20.31	22.41	18.35	–	–
SANP [13]	20.71	21.64	19.12	–	–
SSDML [46]	18.32	17.62	9.96	–	–
NLRM [12]	34.66	31.81	27.68	–	–
MTJSR [28]	**50.59**	40.80	35.48	**30.03**	39.22
Proposed DELM	49.80	**45.21**	**38.88**	28.83	**40.68**

Bold values indicate the best accuracies

subset-100 (Celeb-100), our method obtains a 15% improvement in classification accuracy over the existing methods. As the feature dimension and dataset size is huge, the training and testing time of all other methods is very large on this dataset (for example on the Celeb-100, the NLRM [12] method took about 60 h for training and the MMD and MDA took more than 80 h using a Core i7 3.4GHz CPU with 8GB RAM). In contrast, our method takes only 5.02 s for training and achieves better classification accuracy than all previous methods. Similarly, on the subset-200, the NLRM method took about 5 days for training and the MMD and MDA took more than 8 days. On subset-200, DELM takes only 9.02 s for training and achieves better classification accuracy.

The subset-1000 contains 15 million frames in 1000 training image sets and 36 thousands frames in 2580 test image sets. Therefore, previous image set classification methods have a huge computational and memory requirement on this subset. This makes the experimental evaluation and the parameter tuning of these methods very difficult and extremely time consuming. Therefore, on the subset-1000, we only report the results of the proposed algorithm and compare to Multi-Task Joint Sparse Representation (MTJSR) [28]. Note that the accuracies of MTJSR in Table 2 are provided by the original author [28]. The proposed algorithm has comparable or better accuracy than the MTJSR on different subsets. However, the reported testing time of MTJSR in [28] is very high (3254 s) on the subset-1000. In contrast, DELM only takes 350 s during training and 1.7 during for testing. Thus, compared to previous image set classification algorithms, our DELM-based framework is more scalable to large-scale datasets.

Robustness Similar to the experimental protocol of [3, 42], we test the robustness of DELM to noise, i.e. outlier samples. We use Honda dataset in these experiments. First, from each set we randomly select 100 images to generate our clean data. This is done for both the gallery and the probe sets. Next, we added one randomly selected image from every class to other classes, thus corrupting each image set with 19 outlier images. The clean image sets and three noisy cases are labelled as N_c (clean), N_G (only gallery sets have noise), N_P (only probe sets have noise) and N_{G+P} (both gallery sets and probe sets have noise). From Fig. 4, we can see that our algorithm is more robust to outliers in comparison with other methods. Sample-based algorithms (AHISD, CHISD, SANP) are more adversely affected by outliers compared to the structure-based methods. This is not surprising because modelling the set structure as a whole can better resist the influence of outlier samples.

In the next experiment, we evaluate the robustness of our algorithm to the decreasing number of images in each image set. We used the YouTube Celebrities dataset for this experiment and randomly selected N_r samples from each image set for training and testing each. We used the maximum available samples in case there were less than N_r samples in a given set. The average accuracies of different methods for three values of N_r is depicted in Fig. 5. Note that our algorithm is comparatively more robust and consistently outperforms other methods for all values of N_r.

Parameter analysis We perform parameter sensitivity analysis of the proposed DELM using the YouTube dataset. First, we perform image set classification using only the domain-specific model $\mathbf{L}_G$ for the reconstruction. This achieved an average accuracy of ($62.41 \pm 4.21\%$) which is significantly lower than the accuracy ($75.31 \pm 4.63\%$) of using the class-specific models. This confirms that learning the class-specific models is important for improved accuracy. Next, we perform experiments by changing the number of hidden layers h. For $h = 1$ the accuracy on YouTube dataset is $68.25 \pm 4.21\%$, whereas for $h = 2$ the accuracy is 75.31 ± 4.63. By increasing h further, the execution time and memory requirement increase, but the improvement in accuracy was not significant. Finally, we

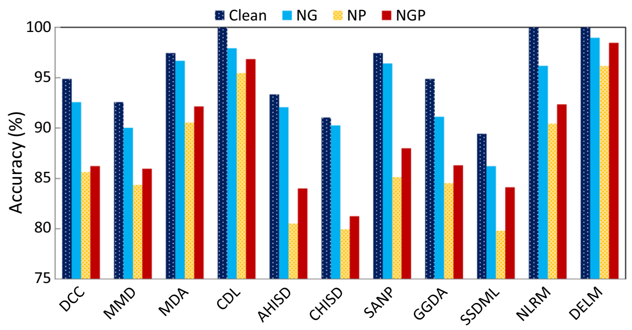

Fig. 4 Average accuracy of different image set classification algorithms when the image sets are corrupted by noise

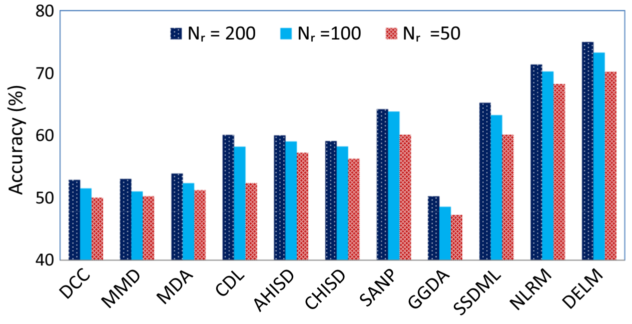

Fig. 5 Robustness of the accuracy to the number of images in each set. N_r samples are randomly selected

Table 3 Execution times (in s) and training memory requirements (in megabytes) on the YouTube Celebrities data

Method	Training	Testing	Memory (MB)
DCC [22]	167.49	8.08	20.8
MMD [43]	313.57	78.32	150.2
MDA [41]	580.70	201.48	$>4 \times 10^4$
CDL [42]	345.88	13.08	238.8
GADR [11]	335.71	20.27	250.5
PMLGM [18]	198.25	10.24	230.4
LEML [19]	135.41	17.38	280.7
GGDA [10]	450.92	20.24	200.0
AHISD [3]	–	18.10	93.7
CHISD [3]	–	190.61	971.4
SANP [13]	–	17.94	160.6
SSDML [46]	400.01	21.87	127.7
NLRM [12]	6542	0.54	523.7
Proposed DELM	**1.01**	**0.06**	**14.3**

Test time is for matching one test image set to 141 training image sets

Bold values indicate the best execution time

vary simultaneously the parameters n_h and C and observe the average accuracy. The accuracy remains stable for large values of parameter C. We vary n_h in the range $\{20, \ldots, 200\}$ and observed that the average accuracy is more stable in the a range of $n_h = \{30, \ldots, 60\}$ for YouTube dataset, and hence, we report the results with $n_h = 40$.

Execution time We compare execution times on the YouTube Celebrities dataset. Table 3 shows the average execution times over the fivefold experiments using a Core i7 3.4GHz CPU with 8GB RAM running MATLAB. The proposed algorithm is significantly faster than the compared state-of-the-art algorithms in both training and testing. For example, our method takes only 1.01 s in training compared to 6542 s for NLRM, while achieving better accuracy.

Memory requirement We also compare the training memory requirement of the proposed algorithm with other algorithms on the YouTube Celebrities dataset. DELM has lower training memory requirements (14.3 MB) to achieve better classification results than previous image set classification algorithms (Table 3).

5 Conclusion

We presented an algorithm for efficient image set classification by learning the nonlinear structures of image sets data using deep extreme learning machines. Our algorithm does not make any assumptions about the underlying image set data and is scalable to large datasets. Nonlinear structure is learned with the deep extreme learning machines (DELM) that enjoy the very fast training times of ELMs while providing deeper representations. Moreover, DELM models can be accurately learned from smaller image sets containing only a few samples. Experiments on five benchmark datasets show that our algorithm consistently outperforms 14 existing state-of-the-art image classification methods in both accuracy and execution time.

Acknowledgements This work was supported by the Australian Research Council (ARC) Grant DP110102399 and UWA Research Collaboration Award 2014.

References

1. Bengio Y (2009) Learning deep architectures for AI. Found Trends Mach Learn 2(1):1–127
2. Bengio Y, Courville A, Vincent P (2013) Representation learning: a review and new perspectives. IEEE Trans PAMI 35(8):1798–1828
3. Cevikalp H, Triggs B (2010) Face recognition based on image sets. In: CVPR, pp 2567–2573
4. Chen S, Sanderson C, Harandi MT, Lovell BC (2013) Improved image set classification via joint sparse approximated nearest subspaces. In: CVPR, pp. 452–459
5. Cui Z, Chang H, Shan S, Ma B, Chen X (2014) Joint sparse representation for video-based face recognition. Neurocomputing 135:306–312
6. Du JX, Shao MW, Zhai CM, Wang J, Tang Y, Chen CLP (2015) Recognition of leaf image set based on manifoldmanifold distance. Neurocomputing 188:131–138
7. Gross R, Shi J (2001) The cmu motion of body database. Tech. Rep. CMU-RI-TR-01-18, Robotics Institute
8. Han B, He B, Sun T, Yan T, Ma M, Shen Y, Lendasse A (2016) HSR: $l_{1/2}$-regularized sparse representation for fast face recognition using hierarchical feature selection. Neural Comput Appl 27(2):305–320
9. Harandi M, Salzmannl M, Baktashmotlagh M (2015) Beyond gauss: image-set matching on the riemannian manifold of pdfs. In: ICCV
10. Harandi M, Sanderson C, Shirazi S, Lovell B (2011) Graph-embedding discriminant analysis on grassmannian manifolds for improved image set matching. In: CVPR, pp 2705–2712
11. Harandi MT, Salzmann M, Hartley R (2014) From manifold to manifold: geometry-aware dimensionality reduction for SPD matrices. In: ECCV, pp 17–32
12. Hayat M, Bennamoun M, An S (2014) Learning nonlinear reconstruction models for image set classification. In: CVPR, pp 1915–1922
13. Hu Y, Mian A, Owens R (2012) Face recognition using sparse approximated nearest points between image sets. IEEE Trans PAMI 34(10):1992–2004
14. Huang G (2015) What are extreme learning machines? Filling the gap between Frank Rosenblatt's dream and John von Neumann's puzzle. Cognit Comput 7(3):263–278
15. Huang GB, Chen L, Siew CK (2006) Universal approximation using incremental constructive feedforward networks with random hidden nodes. IEEE Trans Neural Netw 17(4):879–892
16. Huang GB, Zhou H, Ding X, Zhang R (2012) Extreme learning machine for regression and multiclass classification. IEEE Trans SMC Part B 42(2):513–529
17. Huang GB, Zhu QY, Siew CK (2006) Extreme learning machine: theory and applications. Neurocomputing 70(1–3):489–501
18. Huang Z, Wang R, Shan S, Chen X (2015) Projection metric learning on Grassmann manifold with application to video based face recognition. In: CVPR, pp 140–149
19. Huang Z, Wang R, Shan S, Li X, Chen X (2015) Log-euclidean metric learning on symmetric positive definite manifold with application to image set classification. In: ICML
20. Johnson W, Lindenstrauss J (1984) Extensions of Lipschitz mappings into a Hilbert space. Conference in modern analysis and probability 26:189–206
21. Kasun LLC, Zhou H, Huang GB (2013) Representational learning with ELMs for big data. IEEE Intell Syst 28(6):30–59
22. Kim TK, Kittler J, Cipolla R (2007) Discriminative learning and recognition of image set classes using canonical correlations. IEEE Trans PAMI 29(6):1005–1018
23. Kim M, Kumar S, Pavlovic V, Rowley H (2008) Face tracking and recognition with visual constraints in real-world videos. In: CVPR, pp 1–8
24. Lan Y, Hu Z, Soh YC, Huang GB (2013) An extreme learning machine approach for speaker recognition. Neural Comput Appl 22(3):417–425
25. Lee KC, Ho J, Yang MH, Kriegman D (2003) Video-based face recognition using probabilistic appearance manifolds. In: CVPR, pp I313–I320
26. Leibe B, Schiele B (2003) Analyzing appearance and contour based methods for object categorization. In: CVPR, pp 409–415
27. Li B, Li Y, Rong X (2013) The extreme learning machine learning algorithm with tunable activation function. Neural Comput Appl 22(3):531–539
28. Liu L, Zhang L, Liu H, Yan S (2014) Towards large-population face identification in unconstrained videos. IEEE Trans CSVT PP(99):1–1
29. Liu X, Lin S, Fang J, Xu Z (2015) Is extreme learning machine feasible? a theoretical assessment (part i). IEEE Trans Neural Netw Learn Syst 26(1):7–20
30. Lu J, Wang G, Deng W, Moulin P (2014) Simultaneous feature and dictionary learning for image set based face recognition. In: ECCV, pp 265–280
31. Lu J, Wang G, Deng W, Moulin P, Zhou J (2015) Multi-manifold deep metric learning for image set classification. In: CVPR, pp 1137–1145
32. Lu J, Wang G, Moulin P (2013) Image set classification using holistic multiple order statistics features and localized multi-kernel metric learning. In: ICCV, pp 329–336
33. Mahmood A, Mian A, Owens R (2014) Semi-supervised spectral clustering for image set classification. In: CVPR, pp 121–128
34. Mian A, Hu Y, Hartley R, Owens R (2013) Image set based face recognition using self-regularized non-negative coding and adaptive distance metric learning. IEEE Trans Image Process 22:5252–5262

35. Nian R, He B, Lendasse A (2013) 3D object recognition based on a geometrical topology model and extreme learning machine. Neural Comput Appl 22(3):427–433
36. Ross D, Lim J, Lin R, Yang M (2008) Incremental learning for robust visual tracking. Int J Comput Vis 77:125–141
37. Uzair M, Mahmood A, Mian A, McDonald C (2013) A compact discriminative representation for efficient image-set classification with application to biometric recognition. In: International conference on biometrics, pp 1–8
38. Uzair M, Mahmood A, Mian A, McDonald C (2014) Periocular region-based person identification in the visible, infrared and hyperspectral imagery. Neurocomputing 149(Part B):854–867
39. Viola P, Jones M (2004) Robust real-time face detection. Int J Comput Vis 57:137–154
40. Wang GG, Lu M, Dong YQ, Zhao XJ (2016) Self-adaptive extreme learning machine. Neural Comput Appl 27(2):291–303
41. Wang R, Chen X (2009) Manifold discriminant analysis. In: CVPR, pp 429–436
42. Wang R, Guo H, Davis L, Dai Q (2012) Covariance discriminative learning: a natural and efficient approach to image set classification. In: CVPR, pp 2496–2503
43. Wang R, Shan S, Chen X, Gao W (2008) Manifold-manifold distance with application to face recognition based on image set. In: CVPR, pp 1–8
44. Wang W, Wang R, Huang Z, Shan S, Chen X (2015) Discriminant analysis on Riemannian manifold of Gaussian distributions for face recognition with image sets. In: CVPR
45. Xie L, Lu C, Mei Y, Du H, Man Z (2016) An optimal method for data clustering. Neural Comput Appl 27(2):283–289
46. Zhu P, Zhang L, Zuo W, Zhang D (2013) From point to set: extend the learning of distance metrics. In: ICCV, pp 2664–2671