This CVPR2014 paper is the Open Access version, provided by the Computer Vision Foundation.
The authoritative version of this paper is available in IEEE Xplore.

Learning Non-Linear Reconstruction Models for Image Set Classification

Munawar Hayat, Mohammed Bennamoun, Senjian An
School of Computer Science and Software Enginnering
The University of Western Australia
munawar.hayat@research.uwa.edu.au, {mohammed.bennamoun, senjian.an}@uwa.edu.au

Abstract

We propose a deep learning framework for image set classification with application to face recognition. An Adaptive Deep Network Template (ADNT) is defined whose parameters are initialized by performing unsupervised pre-training in a layer-wise fashion using Gaussian Restricted Boltzmann Machines (GRBMs). The pre-initialized ADNT is then separately trained for images of each class and class-specific models are learnt. Based on the minimum reconstruction error from the learnt class-specific models, a majority voting strategy is used for classification. The proposed framework is extensively evaluated for the task of image set classification based face recognition on Honda/UCSD, CMU Mobo, YouTube Celebrities and a Kinect dataset. Our experimental results and comparisons with existing state-of-the-art methods show that the proposed method consistently achieves the best performance on all these datasets.

1. Introduction

Face recognition has traditionally been considered as a single image classification problem. With the recent advances in imaging technology, multiple images of a person are becoming readily available in numerous scenarios such as video based surveillance, multi-view camera networks, personal albums and images of a person acquired over a long period of time. Face recognition from these multiple images is formulated as an image set classification problem and has gained a significant attention from the research community in recent years [17, 27, 25, 4, 8, 14, 26].

Compared with single image based classification methods, face recognition from image sets offers more promises as it can effectively handle a wide range of variations that are commonly present in the facial images of a person. These variations include changing illumination conditions, view point variations, expression deformations, occlusions and disguise. Facial images of a person under different variations are commonly modeled on a non-linear manifold geometry such as Grassmannian manifold [27, 25, 8] or Lie Group of Riemannian manifold [26]. This modeling of images on manifolds requires prior assumptions related to the specific category of the manifold on which face images are believed to lie. In contrast, this paper introduces a deep learning based framework which makes no prior assumption regarding the underlying geometry of face images and can automatically learn and discover the structure of the complex non-linear surface on which face images of a person (under different variations) are present. The proposed framework first defines an Adaptive Deep Network Template (ADNT) whose weights are initialized by unsupervised layer-wise pre-training using Gaussian Restricted Boltzmann Machines (GRBMs). The pre-initialized ADNT is then separately trained for images of each class to learn class-specific models. The training is performed in a way that the ADNT learns to reconstruct images of that class. A class-specific model is therefore made to learn the structure and the geometry of complex non-linear surface on which face images of that class are present. For classification, a reconstruction error and majority voting based strategy is devised. The proposed framework is evaluated for video based face recognition on Honda/UCSD [18], CMU Mobo [16] and YouTube Celebrities datasets [6] as well as a Kinect dataset [19, 5] and achieves state of the art performance.

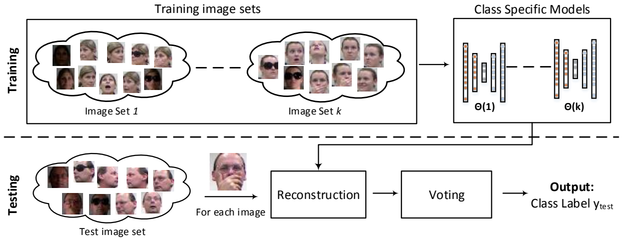

Figure 1. The block diagram of the proposed method. During training, class-specific models are learned from images of each person. These models are then used by a reconstruction error based voting strategy to decide about the class of a test image set.

2. Related Work

Image set classification generally involves two major steps: 1). to find a representation of the images in the set, and 2). to define suitable distance metrics for the computation of the similarity between these representations. Based on the used type of representation, existing image set classification methods can be categorized into parametric-model and non-parametric-model methods. The parametric-model methods [1] approximate an image set in terms of the parameters of a certain statistical distribution model and then measure the similarity between two image sets (two distribution parameters) using *e.g.* KL-divergence. These methods fail to produce a desirable performance if there is no strong statistical relationship between the test and the training image sets. The other type of image set representation methods *i.e.* non-parametric methods do not make any assumption about the statistical distribution of the data. These methods have shown promising results and are being actively developed recently.

The non-parametric model based methods represent an image set either by its representative exemplars or on a geometric surface. Based upon the type of representation, different distance metrics have been developed to determine the between-set distance. For example, for the image sets represented in terms of representative exemplars, the set-set distance can be defined as the Euclidean distance between the set representatives. These can simply be the set mean [27] or adaptively learnt set samples [4]. Cevikalp *et al.* [4] learn the set samples from the affine hull or convex hull models of the set images. The set to set distance is then termed as Affine Hull Image Set Distance (AHISD) or Convex Hull Image Set Distance (CHISD). Hu *et al.* [14] define set-set distance as the distance between their Sparse Approximated Nearest Points (SANPs). SANPs of two sets are determined from the mean image and the affine hull model of the corresponding set and are sparse approximated from the set images while simultaneously searching for the closest points in the respective sets. As set representative based methods require the computation of a one-to-one set distance, these methods are capable of handling intra set variations very effectively. However, their performance is highly prone to outliers. They are also computationally very expensive as a one-to-one match of the query set with all sets in the galley is required. These methods could therefore be very slow in the case of a large gallery size.

Unlike set representative based methods, the second category of non-parametric methods model a complete image set as a point on a geometric surface [27, 25, 8, 26, 10, 23]. The image set can be represented either by a subspace, mixture of subspaces or on a complex non-linear manifold. Principal angles have been very commonly used to determine the distance between image sets represented by a linear subspace. The d principal angles $0 \leq \theta_1 \leq \cdots \leq \theta_d \leq \frac{\pi}{2}$ between two subspaces are defined as the smallest angles between any vector in one subspace and any other vector in the second subspace. The similarity between sets is then defined as the sum of the cosines of the principal angles. For image set representations on manifolds, appropriate distance metrics have been adopted such as the geodesic distance [22] and the projection kernel metric [7] on the Grassmann manifold, and the log-map distance metric [9] on the Lie group of Riemannian manifold. In order to discriminate image sets on the manifold surface, different learning strategies have been developed. Mostly, Linear Discriminant Analysis (LDA) is contrived for different set representations. Examples include Discriminative Canonical Correlations (DCC) [17], Manifold Discriminant Analysis (MDA) [25], Graph Embedding Discriminant Analysis (GEDA) [8] and Covariance Discriminative Learning (CDL) [26].

The methods which model an image set on a geometric surface make prior assumption about the underlying surface on which the face data lies. For example, [17] assumes that face images lie on a linear surface and represents the image set as a linear subspace. Methods including MMD, MDA and GEDA represent an image set on a non-linear Grassmannian manifold, whereas, CDL [26] represents an image set in terms of the covariance matrix of pixel values on Lie Group of Riemannian manifold. For our proposed method, we do not make any prior assumptions about the structure of the surface on which the facial images of a person lie. We instead define a deep learning based framework which incorporates non-linear activation functions to automatically learn the underlying manifold structure. Deep learning has recently gained significant research attention in a number of areas [2, 13, 15]. Ours is the first method which incorporates deep learning for image set classification. The detailed description about our method is presented next.

3. Proposed Technique

We first define an Adaptive Deep Network Template (ADNT) which will be used to learn the underlying structure of the data. The architecture of our ADNT is summarized in Fig 2 and the details are presented in Sec 3.1. For such a deep network to perform well, an appropriate initialization of the weights is required. We initialize the weights of the ADNT by performing pre-training in a greedy layer wise fashion using Gaussian Restricted Boltzmann Machines (details in Sec 3.2). The ADNT with pre-initialized weights is then separately fine-tuned for each of the k classes of the training image sets. We therefore end up with a total of k fine-tuned deep network models, each corresponding to one of the k classes. The fine-tuned models are then used for image set classification (details in Sec 3.3)

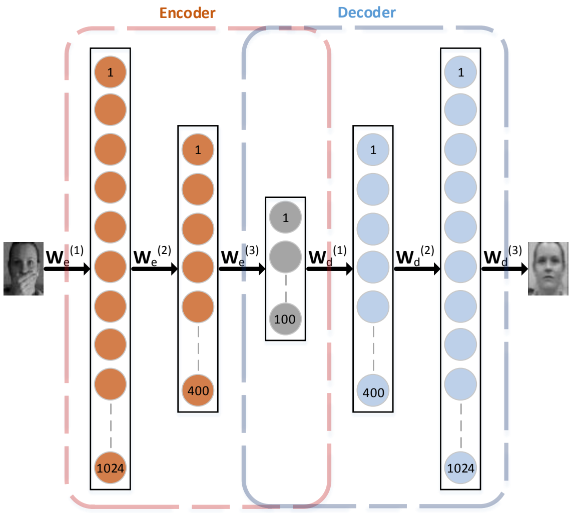

Figure 2. Structure of the Adaptive Deep Network Template (ADNT). The parameters of the template are initialized by unsupervised pre-training. The initialized template is then used to learn class specific models

3.1. The Adaptive Deep Network Template (ADNT)

As depicted in Fig 2, our ADNT is an Auto-Encoder (AE), consisting of two parts: an encoder and a decoder. Both the encoder and the decoder have three hidden layers each, with a shared third layer (the central hidden layer). The encoder part of the AE finds a compact low dimensional meaningful representation of the input data. We can formulate the encoder as a combination of non-linear functions $s(.)$ used to map the input data $\mathbf{x}$ to a representation $\mathbf{h}$ given by,

$$\begin{aligned} \mathbf{h} &= s(\mathbf{W}_e^{(3)}\mathbf{h}_2 + \mathbf{b}_e^{(3)}) \\ \mathbf{h}_2 &= s(\mathbf{W}_e^{(2)}\mathbf{h}_1 + \mathbf{b}_e^{(2)}) \\ \mathbf{h}_1 &= s(\mathbf{W}_e^{(1)}\mathbf{x} + \mathbf{b}_e^{(1)}) \end{aligned} \tag{1}$$

Where $\mathbf{W}_e^{(i)} \in \mathbb{R}^{d_{i-1}\times d_i}$ is the encoder weight matrix for layer i having d_i nodes, $\mathbf{b}_e^{(i)} \in \mathbb{R}^{d_i}$ is the bias vector and $s(.)$ is the non-linear activation function (typically a sigmoid or tangent hyperbolic). The encoder parameters are learnt by combining the encoder with the decoder and jointly training the encoder-decoder structure to reconstruct the input data by minimization of a cost function. The decoder can therefore be defined as a combination of non-linear functions which reconstruct the input $\mathbf{x}$ from the encoder output $\mathbf{h}$. The reconstructed output $\tilde{\mathbf{x}}$ of the decoder is given by,

$$\begin{aligned} \tilde{\mathbf{x}} &= s(\mathbf{W}_d^{(3)}\mathbf{x}_2 + \mathbf{b}_d^{(3)}) \\ \mathbf{x}_2 &= s(\mathbf{W}_d^{(2)}\mathbf{x}_1 + \mathbf{b}_d^{(2)}) \\ \mathbf{x}_1 &= s(\mathbf{W}_d^{(1)}\mathbf{h} + \mathbf{b}_d^{(1)}) \end{aligned} \tag{2}$$

We can represent the complete encoder-decoder structure (the ADNT) by its parameters $\theta_{\text{ADNT}} = \{\theta_{\mathbf{W}}, \theta_{\mathbf{b}}\}$, where $\theta_{\mathbf{W}} = \left\{\mathbf{W}_e^{(i)}, \mathbf{W}_d^{(i)}\right\}^3$ and $\theta_{\mathbf{b}} = \left\{\mathbf{b}_e^{(i)}, \mathbf{b}_d^{(i)}\right\}^3$. Later (in Sec. 3.3) we will use this template and separately train it for all classes of the training image sets to learn class specific models.

3.2. ADNT's Parameter Initialization

The above defined ADNT is used to learn class specific models. This is accomplished by separate training of the ADNT for images of each class of the training image sets. The training is performed with stochastic gradient descent through back propagation. The training fails if the ADNT is initialized with inappropriate weights. More specifically, if the initialized weights are too large, the network gets stuck in local minima. On the other hand, if the initialized weights are too small, the vanishing gradient problem is encountered during back propagation in the initial layers and the network becomes infeasible to train. The weights of the template are therefore initialized by performing unsupervised pre-training [12]. For that, a greedy layer-wise approach is adopted and Gaussian RBMs are used. Below, we first present a brief overview of binary and Gaussian RBMs and then explain their use for our ADNT's parameter initialization.

An RBM is a generative undirected graphical model with a bipartite structure of two sets of binary stochastic nodes termed as the visible ($\{v_i\}_1^{N_v}, v_i \in \{0,1\}$) and the hidden layer nodes ($\{h_j\}_1^{N_h}, h_j \in \{0,1\}$). The nodes of the visible layer are symmetrically connected with the nodes of the hidden layer through a weight matrix $\mathbf{W} \in \mathbb{R}^{N_v \times N_h}$ but there are no intra layer node connections. The joint probability $p(\mathbf{v},\mathbf{h})$ of the RBM structure is given by,

$$p(\mathbf{v},\mathbf{h}) = \frac{1}{Z}\exp(-E(\mathbf{v},\mathbf{h})) \tag{3}$$

Z is the partition function (used as a normalization constant) and $E(v,h)$ is the energy function of the model defined as:

$$E(\mathbf{v},\mathbf{h}) = \sum_i b_i v_i - \sum_j c_j h_j - \sum_{ij} w_{ij} v_i h_j \tag{4}$$

Where $\mathbf{b}$ and $\mathbf{c}$ are the biases of the visible and hidden layer nodes respectively. The training of an RBM for learning its model parameters $\{\mathbf{W},\mathbf{b},\mathbf{c}\}$ is performed by Contrastive Divergence (CD), a numerical method proposed by Hinton *et al.* [11, 3] for efficient approximation to gradient computation and RBM parameter learning.

The standard RBM developed for binary stochastic data can be generalized to the real valued data by appropriate modifications in its energy function. Guassian RBM

(GRBM) is one such popular extension whose energy function is defined by modifying the bias term of the visible units as:

$$E_{\text{GRBM}}(\mathbf{v},\mathbf{h}) = \sum_i \frac{(v_i - b_i)^2}{2\sigma_i^2} - \sum_j c_j h_j - \sum_{ij} w_{ij} \frac{v_i}{\sigma_i} h_j \tag{5}$$

σ_i is the standard deviation of the real valued Gaussian distributed inputs to the visible node v_i. It is possible to learn σ_i for each visible unit but this becomes difficult when using CD for GRBM parameter learning. We instead adopt an alternative approach and fix σ_i to a unit value in the data pre-processing stage.

Due to the restriction that there are no intra-layer node connections, inference becomes readily tractable for the RBM as opposed to most directed graphical models. The probability distributions for GRBM are given by,

$$\begin{aligned} p(h_j = 1|\mathbf{v}) &= \textbf{sigmoid}\left(\textstyle\sum_i w_{ij} v_i + c_j\right) \\ p(v_i|\mathbf{h}) &= \tfrac{1}{\sigma_i \sqrt{2\pi}} \exp\left(\tfrac{-(v_i - u_i)^2}{2\sigma_i^2}\right) \end{aligned} \tag{6}$$

where

$$u_i = b_i + \sigma_i^2 \sum_j w_{ij} h_j \tag{7}$$

Since our data is real valued, we use GRBMs to initialize the weights of our ADNT. Two layers are considered at a time and the GRBM parameters are learnt. Initially, the nodes of the input layer are considered to be visible units $\mathbf{v}$ and the nodes of the first hidden layer as the hidden units $\mathbf{h}$ of the first GRBM and its parameters are learnt. The activations of the first GRBM's hidden units are then used as an input to train the second GRBM. The process is repeated for all three hidden layers of the encoder part of the ADNT structure. The weights learnt for the encoder layers are then tied to the corresponding decoder layers *i.e.* $\mathbf{W}_d^{(3)} = \mathbf{W}_e^{(1)^T}, \mathbf{W}_d^{(2)} = \mathbf{W}_e^{(2)^T}, \mathbf{W}_d^{(1)} = \mathbf{W}_e^{(3)^T}$ (See Fig. 2 for notations)

3.3. Image Set Classification Algorithm

We are now ready to describe our reconstruction error based image set classification algorithm. The complete algorithm is summarized in Alg 1. The details are presented below.

Problem Formulation: Given k training image sets $\{X_c\}_{1\times k}$ and their corresponding class labels $y_c \in [1, 2, \cdots k]$, where the image set $X_c = \left\{\mathbf{x}^{(t)}\right\}_{1\times N_c}$ has N_c images $\mathbf{x}^{(t)} \in \mathbb{R}^{d_x \times d_y}$ belonging to class c, the problem of image set classification can be formulated as follows: given a test image set $X_{test} = \left\{\mathbf{x}^{(t)}\right\}_{1\times N_{test}}$, find the class y_{test} to which X_{test} belongs to?

Unsupervised Pre-Training: We first define our ADNT and initialize its weights by performing unsupervised pre-training. Our ADNT is a multi-layer neural network with 1024-400-100-400-1024 nodes. In order to initialize the weights of the ADNT by GRBMs, we generate an unsupervised training data set. Face images from all training image sets are gathered into a data set $\mathcal{X}_u = \cup\left\{\forall \mathbf{x}^{(t)} \in X_c; \forall c \in [1, 2, \cdots k]\right\}$. The images in the resulting data set $\mathcal{X}_u$ are randomly shuffled and used for layer-wise GRBM training of all layers of the encoder part of the template (1024-400-100). The weights of the decoder layers (100-400-1024) are then initialized with their corresponding tied weights of the encoder layers. Using pre-training for weights initialization has several advantages over random initialization. Since the ADNT is pre-trained for face images, the initialized weights are very close to the actual weights [2]. Therefore, it is highly unlikely that the network gets stuck in a local optima. Moreover, with properly initialized weights, the gradient computation becomes feasible resulting in the convergence of the weights to optimal values.

Learning Class Specific Models: Now that we have the ADNT structure with pre-initialized weights, we separately fine tune its parameters $\theta_{\text{ADNT}} = \{\theta_{\mathbf{W}}, \theta_{\mathbf{b}}\}$ for each of the k training image sets. We therefore learn k class-specific models. The learning of a class-specific model $\theta(c)$ is carried out by performing stochastic gradient descent through back propagation for the minimization of the reconstruction error, over all examples $\mathbf{x}^{(t)}$ of a training image set X_c ,

$$J\left(\theta_{\text{ADNT}}; \mathbf{x}^{(t)} \in X_c\right) = \sum_{\mathbf{x}^{(t)}} \left\| \mathbf{x}^{(t)} - \tilde{\mathbf{x}}^{(t)} \right\|^2 \tag{8}$$

Since the model is being trained to reconstruct the input data, it might end-up learning an identity function and reproduce the input data. Appropriate settings in the configurations of the ADNT are therefore required to ensure that a class specific model learns the underlying structure of the data and produces useful representations. For our ADNT, since the number of nodes in the first hidden layer are larger than the dimensions of the input data, we first learn an over-complete representation of the data by mapping it to a high dimensional space. This high dimensional representation is then followed by a bottleneck *i.e.* the data is mapped back to a compact, abstract and low dimensional representation in the subsequent layers of the encoder. With such mapping, the redundant information in the data is discarded and only the required useful content of the data is retained.

In order to avoid over-fitting and improve generalization of the learnt model to unknown test data, we introduce regularization terms into the cost function of ADNT. A weight decay penalty term J_{wd} and a sparsity constraint J_{sp} are added and the modified cost function becomes,

$$J_{\text{reg}}\left(\theta_{\text{ADNT}};\mathbf{x}^{(t)}\in X_c\right)=\sum_{\mathbf{x}^{(t)}}\left\|\mathbf{x}^{(t)}-\tilde{\mathbf{x}}^{(t)}\right\|^2+\lambda_{\text{wd}}J_{\text{wd}}+\lambda_{\text{sp}}J_{\text{sp}} \tag{9}$$

λ_{wd} and λ_{sp} are regularization parameters. J_{wd} ensures small values of weights for all hidden units. It is defined as the summation of the Frobenius norm of all weight matrices,

$$J_{\text{wd}}=\sum_{i}^{3}\left\|\mathbf{W}_e^{(i)}\right\|_F^2+\sum_{i}^{3}\left\|\mathbf{W}_d^{(i)}\right\|_F^2 \tag{10}$$

J_{sp} enforces that the mean activation $\bar{\rho}_j^{(i)}$ (over all training examples) of the jth unit of the ith hidden layer is as close as possible to a sparsity target ρ (typically a small value, set to 10^{-3} in our experiments in Sec. 4). It is defined in terms of the KL divergence as,

$$\begin{aligned} J_{\text{sp}} &= \sum_{i}^{5}\sum_{j}\text{KL}\left(\rho\|\bar{\rho}_j^{(i)}\right) \qquad (11)\\ &= \sum_{i}^{5}\sum_{j}\rho\log\frac{\rho}{\bar{\rho}_j^{(i)}}+(1-\rho)\log\frac{1-\rho}{1-\bar{\rho}_j^{(i)}} \end{aligned}$$

A class-specific model $\theta(c)$ is achieved by training the regularized ADNT over all images of the set X_c,

$$\theta(c)=\min_{\theta_{\text{ADNT}}}J_{\text{reg}}\left(\theta_{\text{ADNT}};\mathbf{x}^{(t)}\in X_c\right) \tag{12}$$

A class-specific model $\theta(c)$ is therefore made to learn the underlying structure of the manifold on which face images of that class lie. Since the activation functions used are non-linear and a number of layers are stacked together, the AE structure is capable of learning very complex non-linear manifold structures.

Classification: Given a test image set $X_{test}=\left\{\mathbf{x}^{(1)},\mathbf{x}^{(2)},\cdots\mathbf{x}^{(N_{test})}\right\}$, we separately reconstruct (using Eqs. 1 & 2) each of its image $\mathbf{x}^{(t)}\in X_{test}$ from all class specific models $\theta(c), c=1\cdots k$. If $\tilde{\mathbf{x}}^{(t)}(c)$ is the reconstruction of the image $\mathbf{x}^{(t)}$ from model $\theta(c)$ (the model finetuned with images of X_c), then the reconstruction error is given by,

$$r^{(t)}(c)=\left\|\mathbf{x}^{(t)}-\tilde{\mathbf{x}}^{(t)}(c)\right\|_2 \tag{13}$$

After computing the reconstruction errors for all k models, the decision about the class $y^{(t)}$ of the image $\mathbf{x}^{(t)}$ is made based upon the criteria of minimum reconstruction error,

$$y^{(t)}=\arg\min_c r^{(t)}(c) \tag{14}$$

Here the idea is that the unseen image $\mathbf{x}^{(t)}$ will be reconstructed with the least error only from the model trained from images with the same label. Following this procedure, the class labels of all N_{test} images of the test set are computed. The label y_{test} of the test image set X_{test} is then defined as the most recurring label amongst all images of X_{test}. This is given by,

$$\begin{aligned} &y_{test}=\arg\max_c \textstyle\sum_t \delta_c(y^{(t)}) \text{ ,where}\\ &\delta_c(y^{(t)})=\begin{cases}1, & y^{(t)}=c\\ 0, & \text{otherwise}\end{cases} \end{aligned} \tag{15}$$

Algorithm 1 Proposed Image Set Classification Method

```
Input: Training data:
            $k$ Image Sets $\{X_c\}_{1\times k}$ s.t. $X_c=\left\{\mathbf{x}^{(t)}\right\}_{1\times N_c}$
            labels: $y_c\in[1,2,\cdots k]$
        Testing Data:
            Image Set $X_{test}=\left\{\mathbf{x}^{(t)}\right\}_{1\times N_{test}}$
Output: Label $y_{test}$ of $X_{test}$
  Training
  Define ADNT structure
  Unsupervised data: $\mathcal{X}_u\leftarrow\cup\left\{\forall\mathbf{x}^{(t)}\in X_c;\forall c\in[1,2,\cdots k]\right\}$
  Train GRBMs using $\mathcal{X}_u$ to initialize $\theta_{\text{ADNT}}=\{\theta_{\mathbf{W}},\theta_{\mathbf{b}}\}$
  for $c=1\cdots k$ do
    $\theta(c)\leftarrow\min_{\theta_{\text{ADNT}}}J_{\text{reg}}\left(\theta_{\text{ADNT}};\mathbf{x}^{(t)}\in X_c\right)$
  end for
  Testing
  for each image $\mathbf{x}^{(t)}\in X_{test}$ do
    for $\theta(c)=\theta(1)\cdots\theta(k)$ do
      $\mathbf{h}^{(t)}\leftarrow s(\mathbf{W}_e^{(3)}s(\mathbf{W}_e^{(2)}s(\mathbf{W}_e^{(1)}\mathbf{x}^{(t)}+\mathbf{b}_e^{(1)})+\mathbf{b}_e^{(2)})+\mathbf{b}_e^{(3)})$
      $\tilde{\mathbf{x}}^{(t)}(c)\leftarrow s(\mathbf{W}_d^{(3)}s(\mathbf{W}_d^{(2)}s(\mathbf{W}_d^{(1)}\mathbf{h}^{(t)}+\mathbf{b}_d^{(1)})+\mathbf{b}_d^{(2)})+\mathbf{b}_d^{(3)})$
      $r^{(t)}(c)\leftarrow\left\|\mathbf{x}^{(t)}-\tilde{\mathbf{x}}^{(t)}(c)\right\|_2$
    end for
    Assign label to image $\mathbf{x}^{(t)}$: $y^{(t)}\leftarrow\arg\min_c r^{(t)}(c)$
  end for
  Label of $X_{test}$: $y_{test}\leftarrow\arg\max_c\sum_t\delta_c(y^{(t)})$. See Eq. 15
```

4. Experiments

The performance of our proposed method is evaluated on four data sets for the task of image set classification for face recognition. These datasets include three gray scale face video datasets: Honda/UCSD dataset [18], CMU Mobo dataset [6], YouTube Celebrities dataset [16]; and an RGB-D Kinect dataset obtained by combining three Kinect datasets. The detailed description of each of these datasets and their performance evaluation using our method and state-of-the-art methods is presented in Sec 4.2. Here, we first describe the pre-processing steps and the common experimental configurations followed for all datasets.

4.1. Experimental Settings

The face from each frame in the videos of Honda/UCSD and Mobo datasets is automatically detected using Viola and Jones face detection algorithm [24]. It was observed that face detection by [24] failed in a significant number of frames in the case of YouTube Celebrities dataset due to its poor image resolution and large head rotations. We used [21] to track the face region across every video sequence given the location of the face window in the first frame (provided with the dataset). In the case of Kinect face datasets, the random regression forrest based classifier proposed in [5] is used to automatically detect faces from depth images. As depth data is pre-aligned with RGB, the same location of the detected face in the depth image is used for the corresponding RGB image. After a successful detection, the face region is cropped and all colored images are converted to gray scale levels. The cropped gray scale images are then resized to 20×20, 40×40 and 30×30 for Honda/UCSD, Mobo and YouTube celebrities datasets respectively. The depth and the gray scale images of the Kinect datasets are resized to 20×20. Histogram equalization is applied on all images to minimize illumination variations. No other pre-processing such as background removal or alignment is applied. Each cropped and histogram equalized face image is then divided into 4×4 (5×5 in case of CMU Mobo dataset, as in [14, 4]) distinct non-overlapping uniformly spaced rectangular blocks and $\mathbb{R}^{59}$ histograms of $LBP^{u2}_{8,1}$ [20] are computed for every block. Histograms from all blocks are concatenated into a single vector which is used as a face feature vector in all of our experiments. In case of the Kinect dataset, the LBP feature vectors for gray scale and depth images are concatenated and the resulting feature vector is used.

Compared Methods We compare our proposed method with a number of recently proposed state of the art image set classification methods. These include Discriminant Canonical Correlation Analysis (DCC) [17], Manifold-to-Manifold Distance (MMD) [27], Manifold Discriminant Analysis (MDA) [25], the Linear version of the Affine Hull-based Image Set Distance (AHISD) [4], the Convex Hull-based Image Set Distance (CHISD) [4], Sparse Approximated Nearest Points (SANP) [14], Covariance Discriminant Learning (CDL) [26] and Set to Set Distance Metric Learning (SSDML) [29]. The implementations provided by the respective authors are used for all methods except CDL which was carefully implemented by us. The parameters for all methods are optimized for best performance. Specifically, for DCC, we set the dimensions of the embedding space to 100. The number of retained dimensions for a subspace are set to 10 (90% energy is preserved) and the corresponding 10 maximum canonical correlations are used to compute set-set similarity. The parameters for MMD and MDA are adopted from [27] and [25] respectively. No parameter settings are required for AHISD. For CHISD, the same error penalty term ($C = 100$) as in [4] is adopted. For SANP, same weight parameters as in [14] are adopted for convex optimization. No parameter settings are required for CDL and SSDML.

4.2. Results and Analysis

Honda/UCSD Dataset: The Honda/UCSD dataset [18] contains 59 video sequences of 20 different subjects. The number of frames for each video sequence varies from 12 to 645. For our experiments, we consider each video as an image set. Similar to [18, 14, 27, 25], we use 20 video sequences for training and the remaining 39 for testing. In order to achieve consistency in the results, we repeat our experiments ten times with different random selections of the training and testing sets.

The achieved performance in terms of average identification rates and standard deviations of our method and the compared methods is presented in Table 1. The results show that the proposed method achieves perfect classification on the Honda/UCSD data set.

CMU Mobo Dataset: The Mobo (Motion of Body) dataset [6] was originally created for human body pose identification. The dataset contains a total of 96 sequences of 24 subjects walking on a treadmill. Similar to [14, 27, 4], we randomly select one sequence of a subject for training and the remaining three sequences are used for testing. We repeat our experiments 10 times for different random selections of the training and the testing sets. The average identification rates of our proposed method along with a comparison with other methods is provided in Table 1. The results suggest that the proposed method achieves a very high performance of 97.96% and outperforms the other methods.

YouTube Celebrities Dataset: YouTube Celebrities [16] is the largest and the most challenging dataset used for image set classification based face recognition. The dataset contains 1910 videos of 47 celebrities collected from YouTube. The face images of the dataset exhibit a large diversity and variations in the form of pose, illumination and expressions. Moreover, the quality and resolution of the images is very low due to the high compression rate. Since the face regions in the videos are cropped by tracking [21], the low image quality introduces many tracking errors and the region of the cropped face is not uniform across frames of even the same video. We directly use the face region automatically extracted from tracking and do not refine its cropping by enforcing constraints as in [16].

For performance evaluation, we use five fold cross validation experimental settings as proposed in [14, 25, 25].

Methods	Honda/UCSD	CMU Mobo	YouTube	Kinect
DCC CVPR'07 [17]	92.56 ± 2.25	88.89 ± 2.45	51.42 ± 4.95	92.52 ± 2.00
MMD CVPR'08 [27]	92.05 ± 2.25	92.50 ± 2.87	54.04 ± 3.69	93.90 ± 2.25
MDA CVPR'09 [25]	94.36 ± 3.38	80.97 ± 12.28	55.11 ± 4.55	93.46 ± 3.57
AHISD CVPR'10 [4]	91.28 ± 1.79	92.92 ± 2.12	61.49 ± 5.63	91.60 ± 2.18
CHISD CVPR'10 [4]	93.62 ± 1.63	96.52 ± 1.18	60.42 ± 5.95	92.73 ± 1.91
SANP CVPR'11 [14]	95.13 ± 3.07	97.64 ± 0.94	65.60 ± 5.57	93.83 ± 3.12
CDL CVPR'12 [26]	98.97 ± 1.32	90.00 ± 4.38	56.38 ± 5.31	94.59 ± 0.96
SSDML ICCV'13 [29]	86.41 ± 3.64	95.14 ± 2.20	66.24 ± 5.21	86.88 ± 3.39
Our Method	**100.0 ± 0.0**	**97.96 ± 0.28**	**71.35 ± 5.10**	**98.12 ± 1.69**

Table 1. Experimental Results on Honda, CMU, YouTube and Kinect datasets for different methods

The whole dataset is equally divided (with minimum overlap) into five folds with 9 image sets per subject in each fold. Three of these image sets are randomly selected for training, whereas the remaining six sets are used for testing. Table 1 summarizes the average identification rates and the standard deviations of different methods. It can be observed that the achieved identification rates for all methods are low for this dataset compared with the Honda/UCSD and Mobo dataset. This is owing to the challenging nature of the dataset. The videos have been captured in real life scenarios and they exhibit a wide range of appearance variations. The results suggest that our proposed method significantly outperforms the existing methods and achieves a relative performance improvement of 9.0% over the second best method.

Figure 3. Example images from gray scale datasets: Honda/UCSD (top), CMU/Mobo (center) and YouTube (bottom). Each row corresponds to images of one identity.

Figure 4. Sample images from Kinect datasets: CurtinFaces (top), Biwi (center) and our dataset (bottom)

Kinect Dataset: We also evaluate the performance of our proposed method for RGB-D based face recognition from Kinect data. Face recognition from Kinect data is still in its infancy and only a few work have addressed this problem [19]. The method by Li *et al*. [19] first pre-processes Kinect depth images to achieve a canonical frontal view for faces with profile and non-frontal views. The sparse representation based classification method of [28] is used for recognition. Once evaluated on CurtinFaces, the method achieves a classification rate of 91.1% for RGB , 88.7% for D and 96.7% for fusion of RGB-D data. The proposed method is single frame based and does not make use of the plentitude of data which can be instantly acquired from a Kinect sensor (30 frames per second). Here we formulate face recognition from Kinect data as an RGB-D based image set classification problem. Our formulation avoids expensive pre-processing steps (such as hole filling, spike removal and canonical view estimation; otherwise required for single image based classification) and effectively makes use of the abundant and readily available Kinect data.

The method in [19] is evaluated on CurtinFaces (a Kinect RGB-D database of 52 subjects). For our image set classification experiments, we combine three Kinect datasets: CurtinFaces [19], Biwi Kinect [5] and an in-house dataset acquired at our lab. The number of subjects in each of these datasets is 52 (5000 RGB-D images), 20 (15,000 RGB-D images) and 48 (15000 RGB-D images) respectively. Sample RGB images from these datasets are shown in Figure 4. Each row corresponds to images of a person taken from CurtinFaces (top row), Biwi (middle row) and our Kinect dataset (last row). These datasets are combined into a single dataset of 120 subjects. The images in the joint dataset have a large range of variations in the form of changing illumination conditions, head pose rotations, expression deformations, sunglass disguise, and occlusions by hand. For performance evaluation, RGB-D images of each subject are randomly divided into five uniform folds. Considering each fold as an image set, we select one set for training and the remaining sets for testing. All experiments are repeated five times for different selections of training and testing sets. The results averaged over five iterations are summarized in Table 1. The results show that the proposed method achieves a very high performance. The results suggest that image set classification proves to be a better choice for Kinect based face recognition. It avoids computationally expensive pre-processing steps and the achieved identification rates with all image set classification techniques in Table 1 are comparable or better than the single image based technique (96.7%) of [19].

Acknowledgements

This work is supported by SIRF scholarship from The University of Western Australia (UWA) and Australian Research Council (ARC) grant DP110102166. Thanks to Salman H. Khan for useful discussions.

5. Conclusion

We proposed a novel deep learning framework for image set classification. An adaptive multi-layer auto-encoder structure has been introduced which is first pre-trained for appropriate parameter initialization and then used for learning class specific models. A class specific model automatically learns the underlying non-linear complex geometric surface of the images of that class. These learnt models are then used for a minimum reconstruction error based classification strategy during testing. The proposed framework was extensively evaluated on three benchmark gray scale datasets as well as an RGB-D Kinect dataset and state of the art performance has been achieved.

References

[1] O. Arandjelovic, G. Shakhnarovich, J. Fisher, R. Cipolla, and T. Darrell. Face recognition with image sets using manifold density divergence. In *CVPR*, volume 1, pages 581–588. IEEE, 2005.

[2] Y. Bengio, A. Courville, and P. Vincent. Representation learning: A review and new perspectives. *TPAMI*, 35(8):1798–1828, 2013.

[3] M. A. Carreira-Perpinan and G. E. Hinton. On contrastive divergence learning. In *Artificial Intelligence and Statistics*, volume 2005, page 17, 2005.

[4] H. Cevikalp and B. Triggs. Face recognition based on image sets. In *CVPR*, pages 2567–2573. IEEE, 2010.

[5] G. Fanelli, J. Gall, and L. Van Gool. Real time head pose estimation with random regression forests. In *CVPR*, pages 617–624. IEEE, 2011.

[6] R. Gross and J. Shi. The cmu motion of body (mobo) database. Technical report, 2001.

[7] J. Hamm and D. Lee. Grassmann discriminant analysis: a unifying view on subspace-based learning. In *ICML*, pages 376–383. ACM, 2008.

[8] M. Harandi, C. Sanderson, S. Shirazi, and B. Lovell. Graph embedding discriminant analysis on grassmannian manifolds for improved image set matching. In *CVPR*, pages 2705–2712. IEEE, 2011.

[9] M. T. Harandi, C. Sanderson, A. Wiliem, and B. C. Lovell. Kernel analysis over riemannian manifolds for visual recognition of actions, pedestrians and textures. In *WACV*, pages 433–439. IEEE, 2012.

[10] M. Hayat, M. Bennamoun, and A. A. El-Sallam. Clustering of video-patches on grassmannian manifold for facial expression recognition from 3d videos. In *WACV*, 2013.

[11] G. Hinton, S. Osindero, M. Welling, and Y.-W. Teh. Unsupervised discovery of nonlinear structure using contrastive backpropagation. *Cognitive science*, 30(4):725–731, 2006.

[12] G. E. Hinton, S. Osindero, and Y.-W. Teh. A fast learning algorithm for deep belief nets. *Neural computation*, 18(7):1527–1554, 2006.

[13] G. E. Hinton and R. R. Salakhutdinov. Reducing the dimensionality of data with neural networks. *Science*, 313(5786):504–507, 2006.

[14] Y. Hu, A. S. Mian, and R. Owens. Sparse approximated nearest points for image set classification. In *CVPR*, pages 121–128. IEEE, 2011.

[15] S. H. Khan, M. Bennamoun, F. Sohel, and R. Togneri. Automatic feature learning for robust shadow detection. In *CVPR*, 2014.

[16] M. Kim, S. Kumar, V. Pavlovic, and H. Rowley. Face tracking and recognition with visual constraints in real-world videos. In *CVPR*, pages 1–8. IEEE, 2008.

[17] T. Kim, J. Kittler, and R. Cipolla. Discriminative learning and recognition of image set classes using canonical correlations. *IEEE TPAMI*, 29(6):1005–1018, 2007.

[18] K.-C. Lee, J. Ho, M.-H. Yang, and D. Kriegman. Video-based face recognition using probabilistic appearance manifolds. In *CVPR*, volume 1, pages I–313. IEEE, 2003.

[19] B. Y. Li, A. S. Mian, W. Liu, and A. Krishna. Using kinect for face recognition under varying poses, expressions, illumination and disguise. In *WACV*, pages 186–192. IEEE, 2013.

[20] T. Ojala, M. Pietikainen, and T. Maenpaa. Multiresolution gray-scale and rotation invariant texture classification with local binary patterns. *TPAMI*, 24(7):971–987, 2002.

[21] D. A. Ross, J. Lim, R.-S. Lin, and M.-H. Yang. Incremental learning for robust visual tracking. *IJCV*, 77(1-3):125–141, 2008.

[22] P. Turaga, A. Veeraraghavan, A. Srivastava, and R. Chellappa. Statistical computations on grassmann and stiefel manifolds for image and video-based recognition. *IEEE TPAMI*, 33(11):2273 –2286, nov. 2011.

[23] M. Uzair, A. Mahmood, A. Mian, and C. McDonald. A compact discriminative representation for efficient image-set classification with application to biometric recognition. In *ICB*, pages 1–8, 2013.

[24] P. Viola and M. J. Jones. Robust real-time face detection. *IJCV*, 57:137–154, 2004.

[25] R. Wang and X. Chen. Manifold discriminant analysis. In *CVPR*, pages 429–436. IEEE, 2009.

[26] R. Wang, H. Guo, L. S. Davis, and Q. Dai. Covariance discriminative learning: A natural and efficient approach to image set classification. In *CVPR*, pages 2496–2503. IEEE, 2012.

[27] R. Wang, S. Shan, X. Chen, and W. Gao. Manifold-manifold distance with application to face recognition based on image set. In *CVPR*, pages 1–8. IEEE, 2008.

[28] J. Wright, A. Y. Yang, A. Ganesh, S. S. Sastry, and Y. Ma. Robust face recognition via sparse representation. *TPAMI*, 31:210–227, 2009.

[29] P. Zhu, L. Zhang, W. Zuo, and D. Zhang. From point to set: Extend the learning of distance metrics. In *ICCV*, 2013.