Single Sample Face Recognition via Learning Deep Supervised Autoencoders

Shenghua Gao, Yuting Zhang, Kui Jia, Jiwen Lu, *Member, IEEE*, and Yingying Zhang

***Abstract*— This paper targets learning robust image representation for single training sample per person face recognition. Motivated by the success of deep learning in image representation, we propose a supervised autoencoder, which is a new type of building block for deep architectures. There are two features distinct our supervised autoencoder from standard autoencoder. First, we enforce the faces with variants to be mapped with the canonical face of the person, for example, frontal face with neutral expression and normal illumination; Second, we enforce features corresponding to the same person to be similar. As a result, our supervised autoencoder extracts the features which are robust to variances in illumination, expression, occlusion, and pose, and facilitates the face recognition. We stack such supervised autoencoders to get the deep architecture and use it for extracting features in image representation. Experimental results on the AR, Extended Yale B, CMU-PIE, and Multi-PIE data sets demonstrate that by coupling with the commonly used sparse representation-based classification, our stacked supervised autoencoders-based face representation significantly outperforms the commonly used image representations in single sample per person face recognition, and it achieves higher recognition accuracy compared with other deep learning models, including the deep Lambertian network, in spite of much less training data and without any domain information. Moreover, supervised autoencoder can also be used for face verification, which further demonstrates its effectiveness for face representation.**

***Index Terms*— Single training sample per person, face recognition, supervised auto-encoder, deep architecture.**

I. Introduction

SINGLE sample per person (SSPP) face recognition [1], [2][1] is a very important research topic in computer vision because of its potential applications in many realistic scenarios like passport identification, gate ID identification, video surveillance, *etc*. However, as shown in Fig. 1, there is only one training image (gallery image) in SSPP, and the faces to be recognized may contain lots of variances in, for example, illumination, expression, occlusion, pose, *etc*. Therefore, SSPP face recognition is an extremely challenging task.

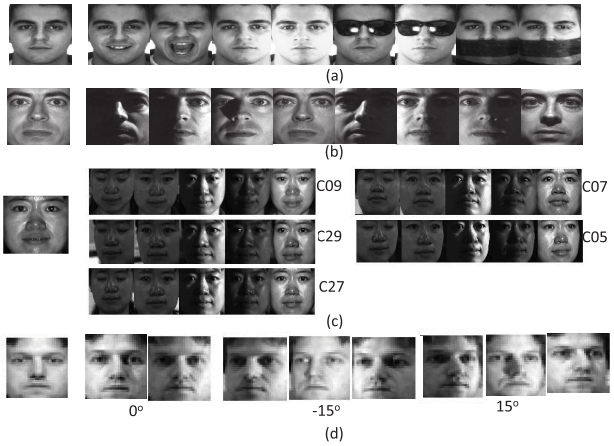

Fig. 1. Samples of gallery images (the first column) and probe images (the rest faces) on the AR, Extended Yale B, CMU-PIE, and Multi-PIE datasets. (a) The AR dataset. (b) The Extended Yale B dataset. (c) The CMU-PIE dataset. (d) The Multi-PIE dataset.

Because of the presence of such challenging intra-class variances, a robust face representation which can overcome the effect of these variances is extremely desirable, and will greatly facilitate SSPP face recognition. However, restricted by having only one training sample for each person, the commonly used subspace analysis based face representation methods, like Eigenfaces [4] and Fisherfaces [5], are no longer suitable nor applicable to SSPP. To seek a good SSPP image representation, lots of endeavors have been made which achieve some good performance under certain settings. For example, with the help of manually generated virtual faces [6] or an external dataset [1], traditional face representation methods can be extended to the SSPP scenario. However, the performance of these methods is still not unsatisfactory for challenging real data. After representing each face with a feature vector, a classification technique, like Nearest Neighbor, sparse representation based classification (SRC) [7], can be used to predict the labels of the probe images.

Deep neural networks have demonstrated their great successes in image representation [8], [9], and their fundamental ingredient is the training of a nonlinear feature extractor at each layer [10]–[12]. After the layer-wise training of each building block and the building of a deep architecture, the output of the network is used for the image representation in the subsequent task. As a typical building block in deep neural networks, Denoising Auto-Encoder [11] extracts the

Manuscript received December 9, 2014; revised March 16, 2015; accepted May 30, 2015. Date of publication June 16, 2015; date of current version August 6, 2015. This work was supported by the Shanghai Pujiang Program under Grant 15PJ1405700. The associate editor coordinating the review of this manuscript and approving it for publication was Prof. Patrizio Campisi.

S. Gao and Y. Zhang are with ShanghaiTech University, Shanghai 200444, China (e-mail: gaoshh@shanghaitech.edu.cn; zyt@zju.edu.cn).

K. Jia is with the University of Macau, Macau 999078, China (e-mail: kuijia@gmail.com).

J. Lu is with Advanced Digital Sciences Center, Singapore 138632 (e-mail: jiwen.lu@adsc.com.sg).

Y. Zhang is with Zhejiang University, Hangzhou 310027, China (e-mail: zhangyy2@shanghaitech.edu.cn).

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

Digital Object Identifier 10.1109/TIFS.2015.2446438

[1]SSPP is one specific task of one shot learning [3]

features through a deterministic nonlinear mapping, and it is robust to the noises of the data. Image representations based on the denoising auto-encoder have shown good performance in many tasks, like object recognition, digit recognition, *etc*. Motivated by the success of denoising auto-encoder based deep neural networks, and driven by the SSPP face recognition, we propose a supervised auto-encoder to build the deep neural network. We first treat the faces with all type of variants (for example, illumination, expression, occlusion, or poses) as images contaminated by noises. With a supervised auto-encoder, we can recover the face without the variant, meanwhile, such a supervised auto-encoder also extracts robust features for image representation in SSPP scenario, i.e., the features corresponding to the same person should have the same (ideally) or similar features, and such features should be stable to the intra-class variances which commonly exist in the SSPP scenario.

The contributions of this paper are two-fold. Firstly, we propose a new type of building block – supervised auto-encoder, for building the deep neural network. Different from standard Auto-Encoder, on the one hand, we enforce all the faces with variances to be mapped with the canonical face of the person, for example, frontal face with neutral expression and normal illumination. Such strategy helps remove the variances in face recognition. On the other hand, by imposing the similarity preservation constraints on the extracted features, the supervised auto-encoder makes the features corresponding to the same person similar, therefore it extracts more robust features for face representation. Secondly, by leveraging the supervised auto-encoder, robust features can be extracted for image representation in SSPP face recognition, therefore improves the recognition accuracy of SSPP.

The rest of the paper is organized as follows: we will review the related work, including the commonly used image representation methods for SSPP scenario as well as the auto-encoder and its variants in Section II. In Section III, we will introduce our supervised auto-encoder, and discuss its application in SSPP. We will experimentally evaluate the proposed technique and its parameters in Section IV, and conclude our work in Section V.

II. Related Work

A. Work Related to SSPP Face Representation

Though subspace analysis based methods are usually adopted for effective and efficient face representation in general face recognition, they are no longer suitable nor applicable to the SSPP scenario. On the one hand, because of the limited number of gallery images and uncertainty of the variances between probe images and gallery image, it is not easy to estimate the data distribution and get the proper projection matrix for unsupervised methods, like Eigenfaces [4], 2DPCA [13], *etc*. To make these unsupervised methods more suitable for SSPP, by taking advantage of some virtual faces generated by different methods, Projection-Combined Principal Component Analysis ($(PC)^2A$) [14], Enhanced $(PC)^2A$ ($E(PC)^2A$) [15], *etc.*, have been proposed. On the other hand, to make FLDA [5] based image representation able to be used in the SSPP case where the intra-class variance is impossible to be directly estimated because only one training sample is provided for each person, virtual samples are usually generated by using small perturbation [16], transformation [17], SVD decomposition [6], or subimages generated by dividing each image into small patches [18]. Moreover, the intra-class variances can also be estimated from a generic dataset [1], [19], where each subject contains images with variances in pose, expression, illumination, occlusion, *etc*. Recently, with the emergence of deep learning technique, Tang *et al.* propose a Deep Lambertian Network (DLN) [20] which combines Deep Belief Nets [10] with Lambertian reflection assumption. Therefore DLN extracts illumination invariant features for face representation and it also shows good performance under SSPP setting, but it is not able to handle other variances, like expressions, poses, occlusions, *etc.*, which restricts its application in real world problems.

B. Work Related to Auto-Encoder

Auto-encoder which is also termed as autoassociator or Diabolo network, is one of the commonly used building blocks in deep neural networks. It contains two modules. (i) A encoder maps the input x to the hidden nodes through some deterministic mapping function f: $h = f(x)$. (ii) A decoder maps the hidden nodes back to the original input space through another deterministic mapping function g: $x' = g(h)$. For real-valued input, by minimizing the reconstruction error $\|x - g(f(x))\|_2^2$, the parameters of encoder and decoder can be learnt. Then the output of the hidden layer is used as the feature for image representation. It has been shown that such a nonlinear auto-encoder is different from PCA [21], and it has been proven that "training an auto-encoder to minimize reconstruction error amounts to maximizing a lower bound on the mutual information between input and the learnt representation" [11]. To further boost the ability of auto-encoder for image representation in building deep networks, Vincent *et al.* [11] propose a denoising auto-encoder which enhances its generalization by training with locally corrupted inputs. Rafai *et al.* enhance the robustness of auto-encoder to noises by adding the Jacobian [22], or Jacobian and Hessian at the same time [23], into the objective of basic auto-encoder. Zou *et al.* [24] use the pooling operation after the Reconstruction Independent Component Analysis [25] encoding process, and enforce the pooled features to be similar for instances with the same class label. These extensions improve the performance of auto-encoder based neural networks for image representation in object and digit recognition.

C. Deep Learning Based Face Verification Systems

In [26], a Siamese Networks is proposed for face verification and such SN is based on Convolutional Neural Network in [26]. In the experiment, we have tried different settings and use the one with the best performance. As the SN is proposed for face verification, to do the face recognition, we run face verification over all pairs of faces in the

test gallery set and choose the pair that's most similar. In this way, we can predict the label of the probe. Recently, Taigman *et al.* also propose a Convolutional Neural Networks based face verification system [27], which significantly outperforms the existing hand-crafted features based systems on the Labeled Face in the Wild (LFW) database. Similarly, Fan *et al.* also use another variant of convolutional neural network which is termed as Pyramid CNN, for face verification, and also achieve similar performance on the LFW database. But these works are proposed for face verification, where a pair of faces are given and the algorithm identifies whether the two faces belonging to the same person or not (a random guess is 50%). Our task solves face recognition in which, given a test image, it tries to identify precisely the correct person among many. Random guess gives one over the number of subjects. Besides the tasks to be solved are different, the network architecture of our paper is also different with these all existing works. Moreover, our work is also closely related to the work of Zhu *et al.* [28], [29]. In [28], a supervised training is deployed to train a network consisting of encoding layer and pooling layer. In [29], multiple CNN networks trained on different regions of face and a regression layer are trained to solve the face verification task. Similar to these work, our work also makes faces of the same person be represented similarly, but we are based on different architectures. Moreover, different from [28], [29], features learnt by our method is not designed for some specific task. As shown later, our model can also be used for face recognition or face verification.

III. Supervised Auto-Encoder for SSPP Face Representation

The motivation of our supervised auto-encoder for SSPP face representation comes from the denoising auto-encoder [11]. In this section, we will first revisit the denoising auto-encoder. Then we will propose our supervised auto-encoder, its formulation, its optimization, its differences with denoising auto-encoder, and its application in SSPP face representation.

A. A Revisit of Denoising Auto-Encoder

A denoising auto-encoder tries to reconstruct the clean input data by using the manually corrupted version of it. Mathematically, denote the input as x. In denoising auto-encoder, input x is first corrupted by some predefined noise, for example, Additive Gaussian noise ($\tilde{x}|x \sim N(x, \sigma^2 I)$), masking noise (a fraction of x is forced to 0), or salt-and-pepper noise (a fraction of x is forced to be 0 or 1). Such a corrupted $\tilde{x}$ is used as the input of the encoder $h = f(\tilde{x}) = s_f(W\tilde{x} + b_f)$. Then the output of encoder h is input into the decoder $\hat{x} = g(h) = s_g(W'h + b_g)$. Here s_f and s_g are predefined activation functions of encoder and decoder respectively, which can be sigmoid functions, hyperbolic tangent functions, or rectifier functions [30], *etc.* $W \in \mathbb{R}^{d_h \times d_x}$ and $b_f \in \mathbb{R}^{d_h}$ are the parameters of encoder, and $W' \in \mathbb{R}^{d_x \times d_h}$ and $b_g \in \mathbb{R}^{d_x}$ are the parameters of decoder. d_x and d_h are the dimensionality of input data and the number of hidden nodes respectively. Based on the above definitions, the objective of denoising auto-encoder is given as follows:

$$\min_{W, W', b_f, b_g} \sum_{x \in X} L(x, \hat{x}) \tag{1}$$

Here L is the reconstruction error, typically squared error $L(x, \hat{x}) = \|x - \hat{x}\|^2$ for real-valued inputs. After learning f and g, the output of clean input ($f(x)$) is used as the input of the next layer. By training such denoising auto-encoder layer by layer, stacked denoising auto-encoders are built. Experimental results show that stacked denoising auto-encoders greatly improve the generalization performance of the neural network. Even when the fraction of corrupted pixels (corrupted by zero masking noises) reaches up to 55%, the recognition accuracy is still better or comparable with that of a network trained without corruptions.

B. Supervised Auto-Encoder

In real applications, like passport or Gate ID identification, the only training sample (gallery image) for each person is usually a frontal face with frontal/uniform lighting, neutral expression, and no occlusion. However, the test faces (probe images) are usually accompanied by variances in illumination, occlusion, expression, pose, *etc*. *Compared to the denoising auto-encoder, these gallery images can be seen as clean data and these probe images can be seen as corrupted data.* For robust face recognition, we desire to learn the features which are robust to these variances. The success of denoising auto-encoder convinces us of the possibility to learn such features. Then the problem becomes: How do we learn a mapping function which captures the discriminative structures of the faces of different persons , while staying robust to the possible variances of these faces? Once such a function is learnt, robust features can be extracted for image presentation, with an expected improvement to the performance of SSPP face recognition.

Given a set of data which contain the gallery images (clean data), probe images (corrupted data) as well as their labels, we use them to train a deep neural network for feature extraction. We denote each probe image in this dataset as $\tilde{x}_i$, and its corresponding gallery image as x_i ($i = 1, \ldots, N$). It is desirable that x_i and $\tilde{x}_i$ should be represented similarly. Therefore the following formulation is proposed (following the work [11], [22], only the tied weights case is explored in this paper, i.e., $W' = W^T$):

$$\begin{aligned}\min_{W, b_f, b_g} \quad & \frac{1}{N} \sum_i \left(\|x_i - g(f(\tilde{x}_i))\|_2^2 + \lambda \|f(x_i) - f(\tilde{x}_i)\|_2^2 \right) \\ & + \alpha \left(\mathrm{KL}(\rho_x || \rho_0) + \mathrm{KL}(\rho_{\tilde{x}} || \rho_0) \right)\end{aligned} \tag{2}$$

where

$$\begin{aligned}\rho_x &= \frac{1}{N} \sum_i \frac{1}{2}(f(x_i) + 1), \\ \rho_{\tilde{x}} &= \frac{1}{N} \sum_i \frac{1}{2}(f(\tilde{x}_i) + 1), \\ \mathrm{KL}(\rho || \rho_0) &= \sum_j \left(\rho_0 \log(\frac{\rho_0}{\rho_j}) + (1 - \rho_0) \log(\frac{1 - \rho_0}{1 - \rho_j}) \right).\end{aligned} \tag{3}$$

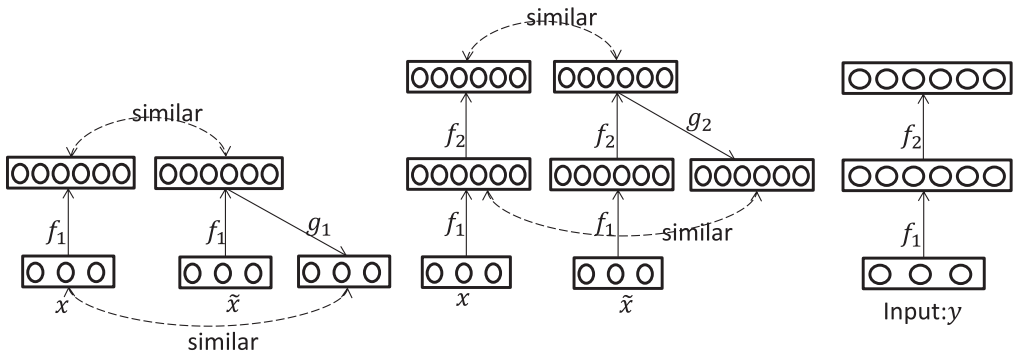

Fig. 2. Architecture of Staked Supervised Auto-Encoders. The left figure: The basic supervised auto-encoder, which is comprised of the clean/"corrupted" faces, there features (hidden layer), as well as the reconstructed clean face by using the "corrupted face". The middle figure: The output of previous hidden layer is used as the input to train the next supervised auto-encoder. We repeat such training several times until the desired number of hidden layers is reached. In this paper, only two hidden layers are used. The right figure: Once the network is trained, given any input face, the output of the last hidden layer is used as the feature for image representation.

In this paper, the activation functions used are the hyperbolic tangent, i.e., $h = f(x) = \tanh(Wx + b_f)$, and $g(h) = \tanh(W^T h + b_g)$.

We list the properties of the supervised auto-encoder as follows:

1) The first term in equation (2) is the reconstruction error. It means that though gallery images contain some variances, after passing through the encoder and the decoder, they will be repaired. In this way, our learnt model is robust to the variances of expression, occlusion, pose, *etc.*, which are quite different to the noises in the Denoising Auto-encoder.
2) The second term in equation (2) is the similarity preservation term. Since the output of the hidden layer is used as the feature, $f(x_i)$ and $f(\tilde{x}_i)$ correspond to the features of the same person. It is desirable that they should be the same (ideally) or similar. Such a constraint enforces the learning of a nonlinear mapping robust to the variances that commonly appear in SSPP.
3) The third and the fourth term, the Kullback-Leiber divergence (KL divergence) terms in equation (2) introduce sparsity in the hidden layer. Work from biological studies shows that the percentage of the activated neurons of human brain at the same time is around 1% to 4% [31]. Therefore the sparsity constraint on the activation of the hidden layer is commonly used in the auto-encoder based neural networks and results show that sparse auto-encoder often achieves better performance [32] than that trained without the sparsity constraint. Since the activation function we used is the hyperbolic tangent, its output is between -1 and 1, where the value of -1 is regarded as non-activated [33]. Therefore we map the output of the encoder to the range (0,1) first in equation (3). Here ρ_x and $\rho_{\tilde{x}}$ are the mapped average activations of the clean data and the corrupted data respectively. By choosing a small ρ_0, the KL divergence regularizer enforces that only a few fraction of neurons are activated [33], [34]. Following the work [33], [34], we also set ρ_0 to 0.05.
4) Weighting the first and the second term with $\frac{1}{N}$ (N is the total number of training samples) helps balance the contributions of the the first two terms and the last terms in the optimization. Otherwise we may need to tune α on different datasets because the training samples for different datasets may be different.
5) We aim at learning an feature extractor to represent faces corresponding to the same person similarly, but [26] and [35] focus on learning a distance metric in the last layer of their respective DNN architectures. Our work is different from [26] and [35] in terms of network architecture and application.
6) Since the labels of the faces are used while training this building block, we term our proposed formulation as the **Supervised Auto-Encoder (SAE)**. We illustrate the idea of such supervised auto-encoder in Fig 2 (the left figure).

C. Optimization of Supervised Auto-Encoder

The optimization method is very important for good performance of deep neural networks. Following the work [36], [37], the entries of W are randomly sampled from the uniform distribution between $[-\sqrt{\frac{6}{d_h+d_x}}, \sqrt{\frac{6}{d_h+d_x}}]$, and b_f and b_g are initialized with zero vectors. It is worth noting that such normalized initialization is very important for the good performance of Auto-Encoder based method "presumably because the layer-to-layer transformations maintain magnitudes of activations (flowing upward) and gradients (flowing backward) [36]". Then the Limited memory BroydenCFletcherCGoldfarbCShanno (L-BFGS) algorithm is used for learning the parameters because of its faster convergence rate and better performance compared to stochastic gradient descent methods and conjugate gradient [38] methods.

As the computational cost of the similarity preservation term (the 2^{nd} term in equation (2)) and the calculation of its gradient with respect to the unknown parameters is almost the same with that of the reconstruction error term, the overall computational complexity is still $O(d_x \times d_h)$ [22] in our supervised auto-encoder.

D. Stacking Supervised Auto-Encoders to Build Deep Architecture

Deep neural networks demonstrate better performance for image representation than shallow neural networks [32].

Therefore we also train the supervised auto-encoder in a layer-wise manner and get the deep architecture. Here we term such an architecture as the **Stacked Supervised Auto-Encoders (SSAE)**. After learning the activation function of the encoder in previous layer, it is applied to the clean data and corrupted data respectively, and the outputs serve as the clean input data and corrupted input data to train the supervised auto-encoder in the next layer. Once the whole network is trained, the output of the highest layer will be used as the feature for image representation. We illustrate this learning strategy in Fig. 2.

E. The Differences Between Stacked Supervised Auto-Encoders and Stacked Denoising Auto-Encoders

Our stacked auto-encoders are different from the stacked denoising auto-encoders in the following three aspects:

1) Input: Denoising auto-encoder is an unsupervised feature learning method, and the corrupted data are generated by manually corrupting the clean data with predefined noises. Therefore the trained model is robust to these predefined noises. However, in our supervised denoising auto-encoder, the corrupted data are the photos taken under different settings, and they are physically meaningful data. By enforcing the reconstructed probe image to be similar to its gallery image, we can overcome the possible variances which appear in SSPP face recognition, and which are quite different from the noises in denoising auto-encoder. This is why our supervised auto-encoder needs the labels of data to train the network.

2) Formulation: The similarity preservation term is used to further emphasize the desired property of SSPP image representation. Though we enforce the reconstructed probe image to be similar to the gallery image, it cannot be guaranteed that the extracted features of them are similar because the variances in pose, expression, illumination, *etc.* can make the probe and the gallery quite different. But in SSPP face recognition, it is desirable that the images of the same person have similar representation. To this end, the similarity preservation term is added to the objective of SAE in equation (2). Such term further makes the extracted features be robust to the variances in face recognition. As shown in Section IV-F, this term greatly improves the recognition accuracy, especially for the cases with large intra-class variances.

3) Building a Deep Architecture: In stacked denoising auto-encoders, once the activation function of the encoder in previous layer is trained, it maps the clean data to the hidden layer, and the outputs serve as the clean input data of next layer SAE. Then the noises which are generated in the same way with that in the previous layer are added on the clean data to serve as the corrupted data. As claimed in [34], since the features in the hidden layer are no longer in the same feature space with the input data, the meaning of applying the noises generated in the same way with that in previous layer to the output of the hidden layer is no longer clear. Similar to [34], we apply the activation function of the encoder to both the clean data and the corrupted data, and their outputs serve as the clean and corrupted input data of the next layer. Hence this way of stacking the basic supervised auto-encoders is more natural for building a deep architecture.

IV. Experiments

In this section, we will experimentally evaluate the stacked supervised auto-encoder for extracting the features in SSPP face recognition task on Extended Yale B, CMU-PIE, and AR datasets. Important parameters will also be experimentally evaluated. Besides face recognition, we also use stacked supervised auto-encoder for face verification on the LFW dataset.

A. Experimental Setup

We use the Extended Yale B, AR, and CMU-PIE, and Multi-PIE datasets for evaluation of the effectiveness of the proposed SSAE model. All the images are gray-scale images and are manually aligned.[2] The image size is 32×32. Thus the dimensionality of the input vector is 1024. Each input feature is normalized with the ℓ_2 normalization. We set $\lambda = \lambda_0 \times \frac{d_x}{d_h}$. Further, we set $\lambda_0 = 10$ on the CMU-PIE and Multi-PIE dataset, and set $\lambda_0 = 5$ on the AR and the Extended Yale B datasets. The reason for this is that the variances are more significant on the CMU-PIE and Multi-PIE dataset than that of the AR or the Extended Yale B datasets. The weight corresponding to the KL divergency term (α) is fixed to be 10^{-4}. Following the work [11], we fix the number of the hidden nodes in all the hidden layers to be 2048. In our experiments, we found that two hidden layers already give a sufficiently good performance. After extracting features with stacked supervised auto-encoders for face representation, following the work [7], sparse representation based classification is used for face recognition. The features are normalized to make their ℓ_2 norms equal to 1 before sparse coding.

1) Baselines: We compare our Stacked Supervised Auto-Encoders (SSAE) with the following work because of their close relationships. It is also worth noting that all the comparisons are based on the same training/test set, and the same generic data if they are used.

1) Sparse Representation for Classification (SRC) [7] with raw pixels;
2) LBP feature followed by SRC;
3) Collaborative Representation for Classification (CRC) [39].
4) AGL [1];
5) One-Shot Similarity Kernel (OSS) [40]. The generic data are used as the negative data in OSS, i.e., we use the same generic/training/testing split for both OSS and our SAE.

[2]In all our experiments, faces are cropped based on manually labeled landmark points, and aligned based landmark points. We also tested our work with the faces cropped with the OpenCV face detector on the AR dataset, but the performance of our work on such data is less than 50% on AR. One possible reason for such poor performance is that we don't have enough data to train a deep network to be robust to the misalignment. The combination of our work with automatic face alignment method is our future work along this direction.

6) Denoising Auto-Encoder (DAE) [11] with 10% masking noises followed by SRC;
7) Modified Denoising Auto-Encoder (MDAE). We propose to use the reconstruction error term only in equation (2) ($\lambda = \alpha = 0$), and term such baseline method as Modified Denoising Auto-Encoder (MDAE);
8) Siamese network (SN) [26].[3]

In all these baseline methods, 1-3 correspond to the very popular sparse coding related methods. 4-5 are specially designed for SSPP, and 6-8 are most related deep learning methods.[4]

B. Dataset Description

The CMU-PIE dataset [42] contains 41,368 images of 68 subjects. For each subject, the images are taken under 13 different poses, 4 different illumination conditions, and 4 different expressions. For each subject, we use the face images taken with the frontal pose, neutral expression, and normal lighting condition as the galleries, and use the rest of the images taken with the poses C27, C29, C07, C05, C09 as probes. We use images of 20 subjects to learn the SSAE, and use the remaining 48 subjects for evaluation.

The AR dataset [43] contains over 4,000 frontal faces taken from 126 subjects (70 men and 56 women) in two different sessions, and the images contain variances in occlusion (sunglasses or scarves), expression (neutral expression, smile, angry, scream), and illumination. Some images contains both occlusion and illumination variances. In our experiments, 20 subjects from session 1 are used as the generic set for training the SSAE, and another 80 subjects also from session 1 are used for evaluation.

The Extended Yale B dataset [44] contains 38 categories. For each subject, we use the frontal faces whose light source direction with respect to the camera axis is 0 degree azimuth ('A + 000') and 0 degree elevation ('E + 00') as gallery images, and use the rest of the images with different lighting conditions as the probe images. Following the work of deep Lambertian networks (DLN) [20], 28 categories are used to train the SSAE and the remaining 10 categories which are from the original Yale B dataset are used for evaluation.

The Multi-PIE dataset [45] contain images of 337 persons taken under the four sessions over the span of 5 months. For each persons, images are taken under 15 different view angles and 19 different illuminations while displaying different facial expressions. In our experiments, only the images with the neutral expression, near frontal pose (0°, 15°, −15°), and different illuminations are used. For each person, the frontal face with neutral expression, frontal illumination is used as the gallery and all the images are used as the probe. The 249 persons in session 1 are used as the evaluation, and the first time appearance images corresponding to the other 88 persons which only appear in session 2-4 are used to train the network.

TABLE I
PERFORMANCE COMPARISON BETWEEN DIFFERENT METHODS ON THE EXTENDED YALE B AND THE AR DATASETS (%)

Method	AR	Extended Yale B
DAE	37.60	42.81
MDAE	80.63	80.95
DLN	NA	81
SN	81.15	78.89
AGL	58.86	72.22
CRC	54.16	49.21
SRC	54.99	49.37
OSS	78.75	65.40
LBP	80.83	74.76
ESRC	81.25	75.56
SSAE	**85.21**	**82.22**

TABLE II
PERFORMANCE COMPARISON BETWEEN DIFFERENT METHODS ON THE CMU-PIE DATASET (%)

Method	C27	C05	C07	C09	C29
DAE	27.03	18.20	22.19	25.95	23.70
MDAE	78.40	42.35	54.31	63.19	44.27
SN	38.54	25.95	24.05	38.19	45.75
AGL	69.23	40.43	44.39	48.61	48.87
CRC	51.98	26.96	35.77	42.36	30.38
SRC	51.46	29.08	35.16	42.01	31.25
OSS	52.06	39.03	42.12	45.23	36.55
LBP	73.27	43.62	50.04	62.50	37.24
ESRC	81.83	67.35	62.4	70.23	65.28
SSAE	**82.79**	**67.52**	**71.45**	**71.96**	**68.06**

TABLE III
PERFORMANCE COMPARISON BETWEEN DIFFERENT METHODS ON THE MULTI-PIE DATASET (%)

	0	-15	15
DAE	59.06	19.1	22.49
MDAE	71.34	20.46	24.12
AGL	87.82	58.37	56.1
CRC	62.88	8.47	10.42
SRC	70.13	9.66	9.46
LBP	64.24	26.89	29.04
SSAE	97.93	67.19	63.25

C. Performance Evaluation

The performance of different methods on the AR, Extended Yale B, CMU-PIE, and Multi-PIE datasets is listed in Table I, Table II, and Table III. We can see that our SSAE outperforms all the rest of the methods including those specially designed methods for SSPP image representation, and it achieves the

[3]We tried several different network architectures in order to get the best experimental result for the Siamese network. Surprisingly, the one of the best performers is the two-layer network which take a fully connected layer (with 500 hidden units and the tanh non-linearity) as the first layer and the Siamese as the second one. The original deep architecture proposed in [26] (i.e. "the basic architecture is C1-S2-C3-S4-C5-F6") doesn't work as well as this simpler ones (With the architecture listed in [1], the accuracy of Siamese Network on AR is below 60%). Probably, the limited size of our generic training set make it difficult to train a good convolutional neural network with deep architecture. Also as the image we used are well-aligned which is different from [26], the convolutional layers in the original architecture might become redundant.

[4]Because the codes of deepface [27] and Pyramid CNN [41] are not available, and the implementation and preprocessing involves lots of tricks, say very sophisticated alignment method and lots of outside data required for network training, here we don't compare our method with these works. Another reason for not comparing with [27] and [41] is that the task solved by these methods is face verification, but our work solves the face verification task.

best performance, which proves the effectiveness of SSAE for extracting robust features for face representation. Needless to say, SSAE based face representation can also be combined with ESRC if another set of labeled data are provided.

1) SAE vs. Hand-Crafted Features: The improvement of the features learnt from our method over the popular LBP features is between about 10% (C27) and 31% (C29) on the CMU-PIE dataset, with larger improvements for subsets of larger poses. On the AR dataset and the Extended Yale B dataset where there is little pose variation, the improvement of our method over LBP is still around 5%. Our method also outperforms ESRC, which introduces a pre-trained intra-class variance dictionary to extend the SRC method to the SSPP case, and the improvement is very evident for cases of large pose variance (e.g., CMU-PIE C07, C09, C29, Multi-PIE 15, −15). Compared with ESRC, another advantage of our method is speed. ESRC relies on the intra-class variance dictionary to achieve good performance, and the size of such dictionary is usually big. Such a big dictionary will increase computational costs, but in our method, the dictionary size in the sparse coding is the same with the number of gallery faces. Thus our method is significantly faster than ESRC in testing. For example, ESRC costs about 4.63 seconds to solve sparse coding for each probe face on Extended Yale B, while our method can solve the sparse coding in about 1.8×10^{-3} seconds. Here the number of atoms in intra-class variance dictionary is 1746 on the Extended Yale B. Therefore the total atoms in the dictionary is 1756 for ESRC. In our method, the number of atoms in the dictionary of SRC is only 10. All the methods are based on Matlab implementations, and run on a Windows Server (64bit) with a 2.13GHz CPU and 16GB RAM. It is worth noting that as the intra-class variance dictionary is very large on Multi-PIE, the optimization is very slow, therefore we don't include its performance on the Multi-PIE dataset.

2) SAE vs. Other DNNs: Compared with DAE, the improvement of our method is over 30% on all the datasets. The reason for the poor performance of DAE for SSPP face representation is that the training of DAE is unsupervised, and zero masking noise is used to train the DAE. Therefore it is natural that the trained DAE cannot handle the variances in poses, expressions, *etc*. Different from DAE, MDAE enforces the reconstructed faces of the probe images, which contain the variances in expression, pose, *etc*, to be well aligned to their gallery images (frontal faces), therefore its performance is better than DAE, but it is still inferior to that of stacked SAE, especially for the cases with variance in pose and expression on AR and CMU-PIE. In contrast, besides using more appropriate reconstruction error term, our SAE also enforces extracted features with the same person to be similar. Therefore, SAE can overcome these variances in SSPP and is more suitable for extracting features for face recognition. Moreover, on the Extended Yale B dataset, the performance of our method (83.97% when the number of hidden nodes is 1024, and 82.22% when the number of hidden nodes is 2048) is better than the 81% obtained by Deep Lambertian Network (DLN) [20], which is specially designed for removing the illumination effect in SSPP face recognition [20]. In addition to the 28 categories from the Extended Yale B, DLN also uses the Toronto Face Database, which is a very large dataset to train the network. Although we use much less data to train the SAE and it does not use any domain information about illumination model as in DLN, performance of SAE is still better.[5] In addition, our SAE also outperforms SN on all the datasets. These experiments clearly demonstrate the superiority of SAE for recognition tasks over other DNN building blocks such as DAE and DLN.

3) More Observations: We notice that as the poses of probe images change, the performance drops notably on the CMU-PIE. For example, the recognition accuracy drops at least 10.83% (C09) compared with that of the frontal face (C27) pose. Interestingly, we also notice that the performance of C07 (look up) and C09 (look down) is slightly better than C05 (look right) and C29 (look left). A possible reason is that the missing parts of face are larger for C05 and C29 than that in C07 and C09 (please refer to Fig. 1). Similar observations can also be found on the Multi-PIE dataset in Table III where the recognition accuracy also drops by around 30% if the poses are non-frontal.

Moreover, some probe images and their reconstructed images on the AR dataset are also shown in Fig. 3. We can see our method can remove the illumination, and recover the neutral face from the faces with different expressions. For the faces with occlusion, our method can also simulate the faces without occlusion, but compared with illumination and expression, it is more difficult to recover the face from the occluded face because too much information is lost. Such results are natural because human can infer the faces with normal illumination and neutral expression from the experience (For the deep neural networks, the experience is learnt from the generic set). But it is also almost impossible for our human to infer the occluded face parts because too much information is missing.

Moreover, some probe images and their reconstructed images on the AR dataset are also shown in Fig. 3. We can see our method can remove the illumination, and recover the neutral face from the faces with different expressions. For the faces with occlusion, our method can also simulate the faces without occlusion, but compared with illumination and expression, it is more difficult to recover the face from the occluded face because too much information is lost. Such results are natural because human can infer the faces with normal illumination and neutral expression from the experience (For the deep neural networks, the experience is learnt from the generic set). But it is also almost impossible for our human to infer the occluded face parts because too much information is missing.

[5]Because the results on the AR and CMU-PIE datasets and the codes for DLN are not available for comparison, we don't report its performance on the AR and CMU-PIE datasets. But it is unlikely that DLN works well on the AR and CMU-PIE as it is not designed to handle other variations in the data such as poses, occlusions, expressions, *etc*. Moreover, the state-of-the-art performance on Extended Yale B has reached 93.6% in terms of accuracy under the SSPP setting in [46]. But the experimental setup in [46] is a little different from that in our paper.

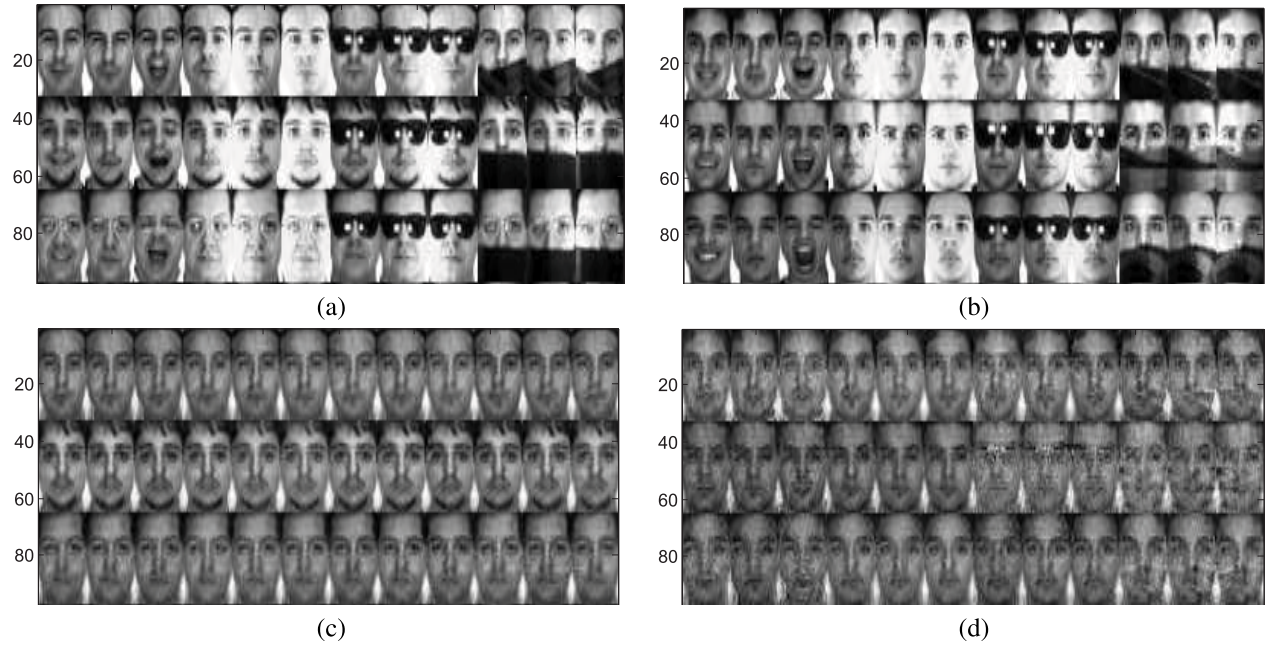

Fig. 3. A comparison between the original images and the reconstructed ones on the AR dataset. (a): Corrupted faces used for training the network. (b): Corrupted faces used for evaluation. (c): Reconstructed faces of (a). (d): Reconstructed faces of (b).

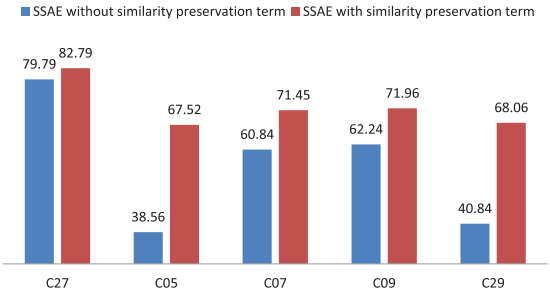

Fig. 4. The Effect of the Similarity Preservation Term on the CMU-PIE Dataset (%).

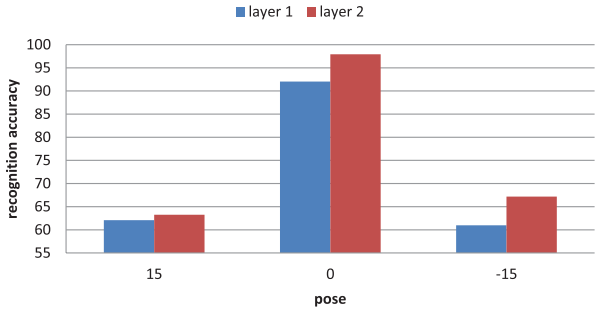

Fig. 5. The effect of SSAE with different depth on the Multi-PIE dataset.

D. Evaluation of Similarity Preservation Term

The similarity preservation term, the second term in equation (2), is very important in the formulation of SAE. Here we list the performance of different networks trained with the formulation with and without this term in Fig. 4. It can be seen that this term improves the recognition by 3% for the benign frontal face case (C27), about 10% for C09 and C07, and about 29% for C05 and C29. We can see that improvement in performance increases with the increase of the poses (the misalignment of faces is larger for C05 and C29 than that in C07 and C09). This validates that the similarity preservation term indeed enhances the robustness of face representation, especially for pose variations.

E. The Effect of Deep Architecture

As an example, we show the performance of the SSAE with different layers on the Multi-PIE dataset in Fig. 5. We can see that 2-layer SSAE network outperforms the single layer network, which demonstrates the effectiveness of the deep architecture.

F. Parameter Evaluation

1) Number of Hidden Nodes in Hidden Layers: We change the number of hidden nodes from 512 to 4096 on the Extended Yale B, and show their performance in Fig. 6 (top left). Results show that the recognition accuracy is higher when the number of hidden nodes is 1024 or 2048, which is the same or larger than the dimensionality of the data. The reason for this is that more hidden nodes usually increase the expressiveness of the auto-encoder [32]. We also note that too many hidden nodes actually decreases the accuracy. A possible reason is that the data used for learning the network in our setting is not enough (only images of 28 subjects are used to learn the parameters), which may limit stability of the learnt network.

2) Weight of KL Terms (α): We plot the performance with different α in Fig. 6 (top right). The poor performance when α is too small (10^{-6}) proves the importance of the sparsity term. But if we impose too large weight on α (10^{-2}), more hidden nodes will hibernate for a given input, which affects

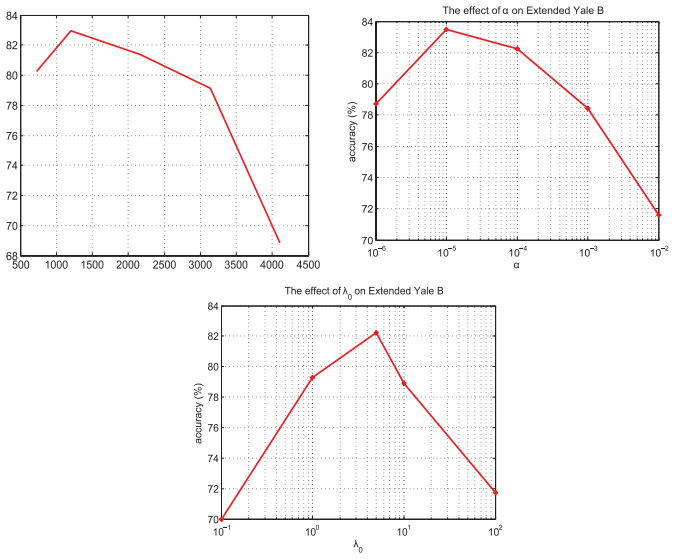

Fig. 6. The effect of different parameters in supervised auto-encoder on the Extended Yale B dataset.

the sensitiveness of the model to the input, and may make faces of different persons be represented similarly. This may be the reason that for a value of 10^2, our model has the lowest performance. So properly setting α is a requirement for the good performance of our model. Similarly phenomenon also happens in sparse representation based face recognition [47], too small or too large sparsity both will reduce the recognition accuracy.

3) Weight of Similarity Preservation Term (λ): We plot the performance of SSAE with different λ in Fig. 6 (bottom). The poor performance with smaller λ_0 (0.1 or 1) also demonstrates the importance of the similarity preservation term. But if λ is too big, the learnt network might become less discriminative for different subjects as it enforces too strongly the similarity of the learnt features for diverse samples. That leads to a drop in performance for very large λ. Moreover, from the plot, we see that good performance can be obtained for a fairly large range of λ.

G. The Comparison of Different Activation Functions

Besides hyperbolic tangent, we also evaluate the performance of SSAE with other activation functions, including sigmoid and Rectified Linear Units (ReLU) [48]. To achieve the sparsity in the hidden layer, different strategies are used for different activation functions. Specifically, if the activation function is sigmoid, the objective function is rewritten as follows:

$$\min_{W,b_f,b_g} \frac{1}{N}\sum_i \left(\|x_i - g(f(\tilde{x}_i))\|_2^2 + \lambda\|f(x_i) - f(\tilde{x}_i)\|_2^2\right) + \alpha\left(\mathrm{KL}(\rho_x||\rho_0) + \mathrm{KL}(\rho_{\tilde{x}}||\rho_0)\right) \quad (4)$$

where

$$\rho_x = \frac{1}{N}\sum_i f(x_i),$$

TABLE IV
PERFORMANCE COMPARISON WITH DIFFERENT ACTIVATION FUNCTIONS (%)

	sigmoid	ReLU	tanh
AR	62.81	53.96	85.21
Extended Yale B	74.76	63.02	82.22

TABLE V
PERFORMANCE COMPARISON BETWEEN DIFFERENT METHODS ON THE LFW DATASETS. (%)

Method	AUC
V1-like/MKL, funneled[51]	0.7935
APEM (fusion), funneled[52]	0.8408
MRF-MLBP[53]	0.7908
Fisher vector faces[54]	0.8747
Eigen-PEP[55]	0.8897
MRF-Fusion-CSKDA[56]	0.9589
SSAE	**0.8548**

$$\rho_{\tilde{x}} = \frac{1}{N}\sum_i f(\tilde{x}_i),$$

$$\mathrm{KL}(\rho||\rho_0) = \sum_j \left(\rho_0 \log(\frac{\rho_0}{\rho_j}) + (1-\rho_0)\log(\frac{1-\rho_0}{1-\rho_j})\right). \quad (5)$$

If the activation function is ReLU, the objective function is rewritten as follows:

$$\min_{W,b_f,b_g} \frac{1}{N}\sum_i \left(\|x_i - g(f(\tilde{x}_i))\|_2^2 + \lambda\|f(x_i) - f(\tilde{x}_i)\|_2^2\right) + \alpha\left(\|f(x_i)\|_1 + \|f(\tilde{x}_i)\|_1\right) \quad (6)$$

We list the performance of our SAE based on different activations on the AR dataset in Table IV. We can see that hyperbolic tangent usually achieves the best performance. It is worth noting that ReLU is usually used for Convolutional Neural Networks (CNN) and demonstrate good performance for image classification. But many tricks, including the momentum, weight decay, early stopping, are used to optimize the objective function in CNN. In our objective function optimization, we simply use the L-BFGS to optimize the objective function. Many existing works [38], [49] have shown that different optimization methods will greatly affect the performance of deep neural networks. Maybe more advanced optimization method and more tricks help improve the performance of ReLU.

H. Extension: SSAE for Face Verification

Besides face recognition, our model can also be used for face verification. Specially, we tested our work on the LFW dataset under the constrained without outside data protocol. The performance of different methods under such protocol is listed in Table V. Interestingly, though our work is designed for learning features to make faces of the same person represented similarly, the performance of our model for face verification is not bad. By learning more sophisticated distance metric [50] with our face representation simultaneously, the performance of our method on LFW probably can be further boosted.

V. Conclusion and Future Work

In this paper, we propose a supervised auto-encoder, and use it to build deep neural network architecture for extracting robust features for SSPP face representation. By introducing a similarity preservation term, our supervised auto-encoder enforces faces corresponding to the same person to be represented similarly. Experimental results on the AR, Extended Yale B, and CMU-PIE datasets demonstrate clear superiority of this module over other conventional modules such as DAE or DLN.

In view of the size of images and training sets, we restrict the image size to be 32×32, and only images of a handful of subjects are used to train the network. For example, only 20 subjects are used on the CMU-PIE and AR datasets, and only 28 subjects on the Extended Yale B dataset. Obviously more training samples will improve the stability of the learnt network and larger images will improve the face recognition accuracy [57].

References

[1] Y. Su, S. Shan, X. Chen, and W. Gao, "Adaptive generic learning for face recognition from a single sample per person," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2010, pp. 2699–2706.

[2] X. Tan, S. Chen, Z.-H. Zhou, and F. Zhang, "Face recognition from a single image per person: A survey," *Pattern Recognit.*, vol. 39, no. 9, pp. 1725–1745, 2006.

[3] L. Fei-Fei, R. Fergus, and P. Perona, "One-shot learning of object categories," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 28, no. 4, pp. 594–611, Apr. 2006.

[4] M. A. Turk and A. P. Pentland, "Face recognition using eigenfaces," in *Proc. IEEE Comput. Soc. Conf. Comput. Vis. Pattern Recognit.*, Jun. 1991, pp. 586–591.

[5] P. N. Belhumeur, J. P. Hespanha, and D. J. Kriegman, "Eigenfaces vs. Fisherfaces: Recognition using class specific linear projection," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 19, no. 7, pp. 711–720, Jul. 1997.

[6] Q.-X. Gao, L. Zhang, and D. Zhang, "Face recognition using FLDA with single training image per person," *Appl. Math. Comput.*, vol. 205, no. 2, pp. 726–734, 2008.

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

[8] Q. V. Le *et al.*, "Building high-level features using large scale unsupervised learning," in *Proc. 29th Int. Conf. Mach. Learn.*, 2012, pp. 81–88.

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

[10] G. E. Hinton, S. Osindero, and Y.-W. Teh, "A fast learning algorithm for deep belief nets," *Neural Comput.*, vol. 18, no. 7, pp. 1527–1554, 2006.

[11] P. Vincent, H. Larochelle, I. Lajoie, Y. Bengio, and P.-A. Manzagol, "Stacked denoising autoencoders: Learning useful representations in a deep network with a local denoising criterion," *J. Mach. Learn. Res.*, vol. 11, no. 5, pp. 3371–3408, 2010.

[12] D. Erhan, Y. Bengio, A. Courville, P.-A. Manzagol, P. Vincent, and S. Bengio, "Why does unsupervised pre-training help deep learning?" *J. Mach. Learn. Res.*, vol. 11, pp. 625–660, Feb. 2010.

[13] J. Yang, D. Zhang, A. F. Frangi, and J.-Y. Yang, "Two-dimensional PCA: A new approach to appearance-based face representation and recognition," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 21, no. 1, pp. 131–137, Jan. 2004.

[14] J. Wu and Z.-H. Zhou, "Face recognition with one training image per person," *Pattern Recognit. Lett.*, vol. 23, no. 14, pp. 1711–1719, 2002.

[15] S. Chen, D. Zhang, and Z.-H. Zhou, "Enhanced $(PC)^2A$ for face recognition with one training image per person," *Pattern Recognit. Lett.*, vol. 25, no. 10, pp. 1173–1181, 2004.

[16] A. M. Martinez, "Recognizing imprecisely localized, partially occluded, and expression variant faces from a single sample per class," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 24, no. 6, pp. 748–763, Jun. 2002.

[17] S. Shan, B. Cao, W. Gao, and D. Zhao, "Extended Fisherface for face recognition from a single example image per person," in *Proc. IEEE Int. Symp. Circuits Syst.*, May 2002, pp. II-81–II-84.

[18] S. Chen, J. Liu, and Z.-H. Zhou, "Making FLDA applicable to face recognition with one sample per person," *Pattern Recognit.*, vol. 37, no. 7, pp. 1553–1555, 2004.

[19] T.-K. Kim and J. Kittler, "Locally linear discriminant analysis for multimodally distributed classes for face recognition with a single model image," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 27, no. 3, pp. 318–327, Mar. 2005.

[20] Y. Tang, R. Salakhutdinov, and G. E. Hinton, "Deep Lambertian networks," in *Proc. 29th Int. Conf. Mach. Learn.*, 2012, pp. 1623–1630.

[21] N. Japkowicz, S. J. Hanson, and M. A. Gluck, "Nonlinear autoassociation is not equivalent to PCA," *Neural Comput.*, vol. 12, no. 3, pp. 531–545, 2000.

[22] S. Rifai, P. Vincent, X. Muller, X. Glorot, and Y. Bengio, "Contractive auto-encoders: Explicit invariance during feature extraction," in *Proc. 28th Int. Conf. Mach. Learn.*, 2011, pp. 833–840.

[23] S. Rifai *et al.*, "Higher order contractive auto-encoder," in *Proc. Eur. Conf. Mach. Learn.*, 2011, pp. 645–660.

[24] W. Y. Zou, A. Y. Ng, S. Zhu, and K. Yu, "Deep learning of invariant features via simulated fixations in video," in *Proc. Adv. Neural Inf. Process. Syst.*, 2012, pp. 3203–3211.

[25] Q. V. Le, A. Karpenko, J. Ngiam, and A. Y. Ng, "ICA with reconstruction cost for efficient overcomplete feature learning," in *Proc. Adv. Neural Inf. Process. Syst.*, 2011.

[26] S. Chopra, R. Hadsell, and Y. LeCun, "Learning a similarity metric discriminatively, with application to face verification," in *Proc. IEEE Comput. Soc. Conf. Comput. Vis. Pattern Recognit.*, Jun. 2005, pp. 539–546.

[27] Y. Taigman, M. Yang, M. Ranzato, and L. Wolf, "Deepface: Closing the gap to human-level performance in face verification," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2014, pp. 1701–1708.

[28] Z. Zhu, P. Luo, X. Wang, and X. Tang, "Deep learning identity-preserving face space," in *Proc. IEEE Int. Conf. Comput. Vis. (ICCV)*, Dec. 2013, pp. 113–120.

[29] Z. Zhu, P. Luo, X. Wang, and X. Tang. (2014). "Recover canonical-view faces in the wild with deep neural networks." [Online]. Available: http://arxiv.org/abs/1404.3543

[30] V. Nair and G. E. Hinton, "Rectified linear units improve restricted Boltzmann machines," in *Proc. 27th Int. Conf. Mach. Learn.*, 2010, pp. 807–814.

[31] P. Lennie, "The cost of cortical computation," *Current Biol.*, vol. 13, no. 6, pp. 493–497, 2003.

[32] Y. Bengio, "Learning deep architectures for AI," *Found. Trends Mach. Learn.*, vol. 2, no. 1, pp. 1–127, 2009.

[33] A. Ng, "Sparse autoencoder," Stanford Univ., Stanford, CA, USA, Lecture Notes CS294A, 2011.

[34] J. Xie, L. Xu, and E. Chen, "Image denoising and inpainting with deep neural networks," in *Proc. Adv. Neural Inf. Process. Syst.*, 2012, pp. 350–358.

[35] R. Salakhutdinov and G. E. Hinton, "Learning a nonlinear embedding by preserving class neighbourhood structure," in *Proc. Int. Conf. Artif. Intell. Statist.*, 2007, pp. 412–419.

[36] X. Glorot and Y. Bengio, "Understanding the difficulty of training deep feedforward neural networks," in *Proc. Int. Conf. Artif. Intell. Statist.*, 2010, pp. 249–256.

[37] Y. Bengio, "Practical recommendations for gradient-based training of deep architectures," in *Neural Networks: Tricks of the Trade*. Berlin, Germany: Springer-Verlag, 2012, pp. 437–478.

[38] Q. V. Le, J. Ngiam, A. Coates, A. Lahiri, B. Prochnow, and A. Y. Ng, "On optimization methods for deep learning," in *Proc. 28th Int. Conf. Mach. Learn.*, 2011, pp. 265–272.

[39] L. Zhang, M. Yang, and X. Feng, "Sparse representation or collaborative representation: Which helps face recognition?" in *Proc. IEEE Int. Conf. Comput. Vis.*, Nov. 2011, pp. 471–478.

[40] L. Wolf, T. Hassner, and Y. Taigman, "The one-shot similarity kernel," in *Proc. IEEE 12th Int. Conf. Comput. Vis.*, Sep./Oct. 2009, pp. 897–902.

[41] H. Fan, Z. Cao, Y. Jiang, Q. Yin, and C. Doudou. (2014). "Learning deep face representation." [Online]. Available: http://arxiv.org/abs/1403.2802

[42] T. Sim, S. Baker, and M. Bsat, "The CMU pose, illumination, and expression (PIE) database," in *Proc. 5th IEEE Int. Conf. FG*, May 2002, pp. 46–51.

[43] A. Martínez and R. Benavente, "The AR face database," CVC Tech. Rep. #24, Jun. 1998.

[44] A. S. Georghiades, P. N. Belhumeur, and D. Kriegman, "From few to many: Illumination cone models for face recognition under variable lighting and pose," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 23, no. 6, pp. 643–660, Jun. 2001.
[45] R. Gross, I. Matthews, J. Cohn, T. Kanade, and S. Baker, "Multi-PIE," *Image Vis. Comput.*, vol. 28, no. 5, pp. 807–813, 2010.
[46] Y. Tang, R. Salakhutdinov, and G. E. Hinton, "Tensor analyzers," in *Proc. 30th Int. Conf. Mach. Learn.*, 2013, pp. 163–171.
[47] S. Gao, I. W.-H. Tsang, and L.-T. Chia, "Sparse representation with kernels," *IEEE Trans. Image Process.*, vol. 22, no. 2, pp. 423–434, Feb. 2013.
[48] X. Glorot, A. Bordes, and Y. Bengio, "Deep sparse rectifier neural networks," in *Proc. 14th Int. Conf. Artif. Intell. Statist.*, 2011, pp. 315–323.
[49] G. E. Hinton and R. R. Salakhutdinov, "Reducing the dimensionality of data with neural networks," *Science*, vol. 313, no. 5786, pp. 504–507, 2006.
[50] J. Hu, J. Lu, and Y.-P. Tan, "Discriminative deep metric learning for face verification in the wild," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2014, pp. 1875–1882.
[51] N. Pinto, J. J. DiCarlo, and D. D. Cox, "How far can you get with a modern face recognition test set using only simple features?" in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2009, pp. 2591–2598.
[52] H. Li, G. Hua, Z. Lin, J. Brandt, and J. Yang, "Probabilistic elastic matching for pose variant face verification," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2013, pp. 3499–3506.
[53] S. R. Arashloo and J. Kittler, "Efficient processing of MRFs for unconstrained-pose face recognition," in *Proc. IEEE 6th Int. Conf. Biometrics, Theory, Appl. Syst. (BTAS)*, Sep./Oct. 2013, pp. 1–8.
[54] K. Simonyan, O. M. Parkhi, A. Vedaldi, and A. Zisserman, "Fisher vector faces in the wild," in *Proc. Brit. Mach. Vis. Conf.*, 2013, pp. 8.1–8.12.
[55] H. Li, G. Hua, X. Shen, Z. Lin, and J. Brandt, "Eigen-PEP for video face recognition," in *Computer Vision* (Lecture Notes in Computer Science), vol. 9005. Berlin, Germany: Springer-Verlag, 2015, pp. 17–33.
[56] S. R. Arashloo and J. Kittler, "Class-specific kernel fusion of multiple descriptors for face verification using multiscale binarised statistical image features," *IEEE Trans. Inf. Forensics Security*, vol. 9, no. 12, pp. 2100–2109, Dec. 2014.
[57] M. D. Zeiler and R. Fergus, "Visualizing and understanding convolutional networks," in *Computer Vision* (Lecture Notes in Computer Science), vol. 8689. New York, NY, USA: Springer-Verlag, 2014, pp. 818–833.

Shenghua Gao received the B.E. (Hons.) degree from the University of Science and Technology of China, in 2008, and the Ph.D. degree from Nanyang Technological University, in 2012. He is currently an Assistant Professor with ShanghaiTech University, China. From 2012 to 2014, he was a Research Scientist with the Advanced Digital Sciences Center, Singapore. He has authored over 30 papers on object and face recognition related topics in many international conferences and journals, including the IEEE TRANSACTIONS ON PATTERN ANALYSIS AND MACHINE INTELLIGENCE, the *International Journal of Computer Vision*, the IEEE TRANSACTIONS ON IMAGE PROCESSING, the IEEE TRANSACTIONS ON NEURAL NETWORKS AND LEARNING SYSTEMS, the IEEE TRANSACTIONS ON MULTIMEDIA, the IEEE TRANSACTIONS ON CIRCUITS AND SYSTEMS FOR VIDEO TECHNOLOGY, *Computer Vision and Pattern Recognition*, and European Conference on Computer Vision. His research interests include computer vision and machine learning. He received the Microsoft Research Fellowship in 2010 and the ACM Shanghai Young Scientist Award in 2015.

Yuting Zhang received the B.E. degree in computer science from Zhejiang University, in 2009, where he is currently pursuing the Ph.D. degree with the Department of Computer Science, advised by G. Pan. He is currently a Visiting Student with the Department of Electronic Engineering and Computer Science, University of Michigan, USA. He was also a Junior Research Assistant with the Advanced Digital Sciences Center, Singapore, and the University of Illinois at Urbana–Champaign in 2012. His research interests include machine learning, computer vision, and in particular, application of deep graph model.

Kui Jia received the B.Eng. degree in marine engineering from Northwestern Polytechnic University, China, in 2001, the M.Eng. degree in electrical and computer engineering from the National University of Singapore, in 2003, and the Ph.D. degree in computer science from Queen Mary, University of London, London, U.K., in 2007. He is currently a Visiting Assistant Professor with the University of Macau, Macau, China. He also holds a Research Scientist position with the Advanced Digital Sciences Center, Singapore. His research interests are computer vision, machine learning, and image processing.

Jiwen Lu (S'10–M'11) received the B.Eng. degree in mechanical engineering and the M.Eng. degree in electrical engineering from the Xi'an University of Technology, Xi'an, China, and the Ph.D. degree in electrical engineering from Nanyang Technological University, Singapore. He is currently a Research Scientist with the Advanced Digital Sciences Center, Singapore. His research interests include computer vision, pattern recognition, and machine learning.

He has authored or coauthored over 100 scientific papers in these areas, with 23 papers published in IEEE Transactions journals (IEEE TRANSACTIONS ON PATTERN ANALYSIS AND MACHINE INTELLIGENCE, the IEEE TRANSACTIONS ON IMAGE PROCESSING, the IEEE TRANSACTIONS ON INFORMATION FORENSICS AND SECURITY, the IEEE TRANSACTIONS ON CIRCUITS AND SYSTEMS FOR VIDEO TECHNOLOGY, and IEEE TRANSACTIONS ON MULTIMEDIA), and 12 papers published in top-tier computer vision conferences (ICCV/CVPR/ECCV). He serves as an Area Chair for ICME 2015 and ICB 2015, and a Special Session Chair for VCIP 2015. He was a recipient of the First-Prize National Scholarship and the National Outstanding Student Award from the Ministry of Education of China in 2002 and 2003, the Best Student Paper Award from the PREMIA of Singapore in 2012, and the Top 10% Best Paper Award from MMSP 2014. Recently, he has given tutorials at some conferences, such as CVPR 2015, FG 2015, ACCV 2014, ICME 2014, and IJCB 2014.

Yingying Zhang received the B.E. degree from Xi'an Jiaotong University, China, in 2014. He is currently pursuing the master's degree with ShanghaiTech University, China. His research interests are computer vision and machine learning.