Sparse Softmax Vector Coding based Deep Cascade Model

Ji Liu, Lei Zhang*

College of Communication Engineering, Chongqing University,
No. 174 Shazheng street, Shapingba district, Chongqing 400044, China
{jiliu,leizhang}@cqu.edu.cn

Abstract. Recently, many sparse coding techniques like sparse representation based classification (SRC) have been proposed to deal with face recognition (FR) problem. In SRC, a test image is linearly coded by the training images to calculate the sparse coefficients by l_1-norm minimization. Then, SRC needs to compute the representation error of each category when classifying the test image. The corresponding category of the test image would have the minimum representation error. In other words, representation errors of all classes show class discrimination. In this paper, we take advantage of this distinct representation errors that are transformed into softmax vector and find that the sub-pattern of the whole image is sometimes more discriminative than the whole image. Sparse softmax vector coding based deep cascade model (SSVD) is proposed to improve the pattern classification performance. The experiments demonstrate that the proposed model is much more effective than state-of-the-art methods.

Keywords: Sparse coding, softmax vector, spatial pyramid, deep cascade model

1 Introduction

Face recognition can be viewed as one of the most popular and challenging topic in computer vision and pattern recognition. In the past 20 years, substantial face recognition methods [1–13] have been developed by numerous researchers. Among these methods, sparse coding and discriminative methods have yielded significant results.

Nassem *et al.* [1] proposed the linear regression classifier (LRC) for face recognition. The main idea of LRC is representing a testing face by a suitable way and classifying it to one class, which can represent it better than other classes. One after another, l_1-norm regularization term is imposed upon the LRC model to avoid over-fitting by Wright *et al.* [2] who proposed a sparse representation based classification (SRC) framework to solve FR problems. In SRC, a testing image is coded by a sparse linear combination of training samples via the l_1-norm minimization. SRC classifies the testing image through estimating which class of training samples could generate the smallest reconstruction error of it with the corresponding class coding coefficients. Zhang *et al.* [3] illustrate that not only l_1-norm but also l_2-norm could achieve parallel results on coding coefficients and proposed the collaborative representation classifier (CRC) scheme. Among the above models, the fidelity terms are measured by the l_2-norm or l_1-norm, which follows the assumption that the pixels of error obey Gaussian or Laplacian distribution independently. Nevertheless, if there were some illumination variation, occlusion, or disguise in the images, the above assumption might be unconscionable.

Subsequently, several scholars enhanced the sparse coding based models and proposed some new methods. Typically, to obtain more robustness, Yang *et al.* [4] proposed a robust sparse coding (RSC) model for FR, in which the residual of the test image and the estimated one is assumed independently and identically distributed according to some probability density function (PDF), where the parameter characterizes the distribution. Then, RSC finds an maximum likelyhood estimation solution of the sparse coding, which can be viewed as a weighted LASSO problem. He *et al.* [5] took advantage of the correntropy induced robust error metric and proposed the correntropy based sparse representation (CESR) model. What is interesting is that RSC and CESR can be viewed similar work of M-estimator with different kernel size. Recently, He *et al.* [6] proposed a new model of using different half-quadratic functions to measure the error image, which combines the ideas of SRC, CESR and RSC. In addition, to make the LRC more robust to random pixel disguise, occlusion, or illumination, Nassem *et al.* [7] extended the LRC to the robust linear regression classification (RLRC) by making use of Huber estimator. Zhou *et al.* [8] borrowed the markov random field model into the sparse coding scheme and proposed sparse error correction with MRF model. Jia *et al.* [9] utilized structured sparsity-inducing norm into the SRC model and presented a structured sparse representing classifier (SSRC).

To improve the recognition rate of sparse coding methods, we propose a deep cascade model based on sparse softmax vector coding (SSVD) in this paper, inspired by [23]. The main contributions of our work are as follows. (1) The use of discriminative softmax vector. SRC codes a testing image by sparse linear combination of all training images and classifies it to the class which has minimum representation error. In other words, representation errors of all classes show class discrimination. Most existing sparse coding based methods only focus on the original or extracted image feature. To further explore the effectiveness of sparse coding method on the discriminative representation errors, we propose the SSVD method, in which the softmax vectors transformed by representation errors are used to do sparse representation repeatedly. (2) Three-level spatial pyramid structure is used to enhance class discrimination. Most of the sparse coding methods are based on the whole images, which ignores the local information of the subregion. Because the subregions of the whole image show more detailed local information and more discriminative than the whole image, SSVD combines the whole image and its subregions to obtain softmax vectors by using three-level spatial pyramid structure as shown in the image coding part of Fig. 1. (3) Deep cascade model based on concatenated softmax vectors is pro-

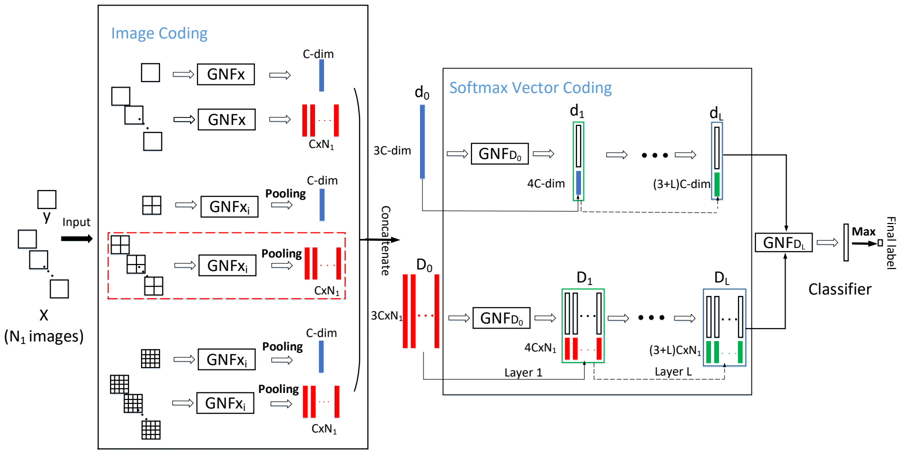

Fig. 1. An example is given to illustrate how the deep model works when classifying a test image **y** under all training images **X**.

posed. As the cascade model goes deep, the concatenated softmax vectors obtain more class discrimination, which is in favour of classification. Our extensive experiments in benchmark databases show that the proposed deep model achieves better performance than many existing sparse coding methods.

The rest of this paper is organized as follows. Section 2 presents the proposed deep model. Section 3 presents the solving algorithm of sparse representation. The experiment results are shown in Section 4. Section 5 concludes this paper.

2 The Proposed Approach

In this section, we illustrate how we classify the testing image by giving all training images. First, we define a procedure getting new feature in the first part. Then, in the second part, we present a detailed illustration that how the deep cascade model goes as shown in Fig. 1

2.1 Getting New Feature

According to SRC, suppose that we have C classes of subjects and define that $\mathbf{d}$ represents one of testing sample and $\mathbf{D} = [\mathbf{D}_1, \mathbf{D}_2, \cdots, \mathbf{D}_C]$ represents the dictionary. The representation model can be transformed into following problem [15]:

$$\min_{\boldsymbol{\alpha}} \parallel \mathbf{d} - \mathbf{D}\boldsymbol{\alpha} \parallel_2^2 + \lambda \parallel \boldsymbol{\alpha} \parallel_1 \tag{1}$$

where λ is a scalar constant. After solving the above function, we compute the representation error of each class as follow:

$$r_c = \parallel \mathbf{d} - \mathbf{D}^c \boldsymbol{\alpha}^c \parallel_2^2 \tag{2}$$

where $\mathbf{D}^c$ is the *c-th* class samples, and $\boldsymbol{\alpha}^c$ is the coefficient vector associated with *c-th* class. Softmax vector $\mathbf{r}$ is computed by softmax function as follow:

$$\mathbf{r} = \frac{e^{-r_c}}{\sum_{c=1}^{C} e^{-r_c}} \tag{3}$$

where $\mathbf{r} = [r_1, r_2, \cdots, r_C] \in \mathbb{R}^C$. If the test sample $\mathbf{d}$ belonged to class $i(\leq C)$, r_i should be bigger than other atoms in softmax vector $\mathbf{r}$, which is called class discrimination. The above process of obtaining softmax vector $\mathbf{r}$ is named as Getting New Feature on dictionary $\mathbf{D}$ (GNF_D)

2.2 Sparse Softmax Vector Coding based Deep Cascade Model

Without loss of generality, we let $\mathbf{X}$ represents the training images and $\mathbf{Y}$ represents the testing images. The class number is C. The numbers of training images and testing images are N_1 and N_2. For each image, a three-level spatial pyramid is used to compute the softmax vector. We take one testing image $\mathbf{y}$ and all training images $\mathbf{X}$ as an example to explain how to obtain the softmax vectors and classify the testing image $\mathbf{y}$ as shown in Fig. 1.

There are 3 parallel channels that are designed to process the input images. In the first channel, the original testing image $\mathbf{y}$ is represented by all training images $\mathbf{X}$ and go through the GNF_X procedure to get a softmax vector. Similarly, a softmax vector set of training images will be obtained after each training image goes through GNF_X procedure. In the second channel, all the input images are equally divided into 4 subregions. Let $\mathbf{y}_i$ denote the *i-th*$(i = 1, \cdots, 4)$ subregion of test image $\mathbf{y}$ and $\mathbf{X}_i$ denote the *i-th*$(i = 1, \cdots, 4)$ subregion set of all the training images $\mathbf{X}$. Similar to the first channel, $\mathbf{y}_i$ goes through GNF_{X_i} procedure, then 4 softmax vectors will be generated. Those 4 softmax vectors is transformed into one vector after max pooling or average pooling. Like the testing image, each subregion of per training image goes through the corresponding GNF_{X_i} procedure, and 4 softmax vectors will be generated. After the max pooling or average pooling, the 4 softmax vectors are transformed into 1 vector. Then the transformed vector of each image is parallel integrated into one matrix as shown in Fig. 2 that is an instance presented in red dashed part of the second channel in Fig. 1 to illustrate max pooling and average pooling. In the third channel, each input images are equally divided into 16 subregions. Using the same approach in the second channel, a transformed softmax vector of testing image $\mathbf{y}$ and a transformed softmax vector set of training images $\mathbf{X}$ will be generated.

After those 3 channels, 3 softmax vectors (tinted with blue) of testing image are concatenated into one vector $\mathbf{d}_0$ and 3 softmax vector sets (tinted with red) are concatenated into one vector set $\mathbf{D}_0$. Then, $\mathbf{d}_0$ goes through GNF_{D_0} procedure to compute the softmax vector that is concatenated with $\mathbf{d}_0$ to construct input sample $\mathbf{d}_1$ of second layer. Similarly, each column in $\mathbf{D}_0$ also goes through GNF_{D_0} procedure to compute the softmax set that is concatenated with $\mathbf{D}_0$ to construct input dictionary $\mathbf{D}_1$ of second layer. Using the same way, we can

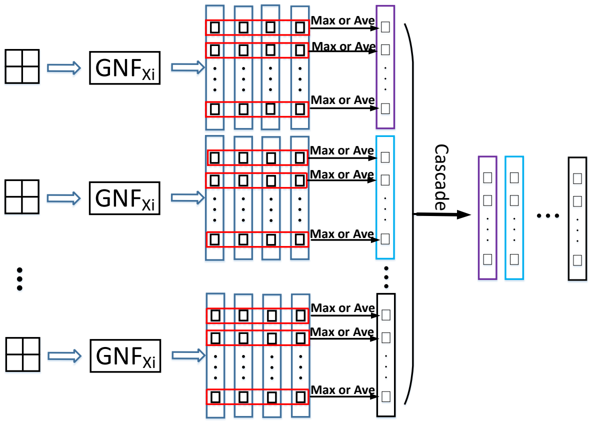

Fig. 2. Illustration of the max pooling and average pooling. 4 softmax vectors will be obtained after each subregion of per image goes through GNF_{X_i} procedure. Then, we compute the maximum value or average of the 4 values in the corresponding dimension to construct a new vector. Finally, the new vector of each image is parallel integrated into one matrix.

obtain the testing sample $\mathbf{d}_L$ and dictionary $\mathbf{D}_L$ of level L. Finally, $\mathbf{d}_L$ goes through GNF_{D_L} to get the softmax vector. The prediction will be obtained by taking the class with the maximum value in softmax vector.

3 Solving Algorithm of Sparse Representation

In recent years, many algorithms have been proposed for sparse representation. In particular, the alternating direction method of multipliers (ADMM), first proposed in 1970s [14], has drawn a lot of attention. Yang and Zhang [15] integrated the proximal methods into ADMM when solving l_1-norm minimization problems.

In this paper, we also use ADMM method to solve sparse representation problem. In the SSVD model, sparse representation problems need to be solved in different stages. We just take the first channel in Fig. 1 for an instance to illustrate how we solve the sparse coding coefficients of testing samples based on the dictionary $\mathbf{X}$. Let $\mathbf{X} = [\mathbf{x_1}, \mathbf{x_2}, \cdots, \mathbf{x_{N_1}}] \in \mathbb{R}^{d \times N_1}$ denote training samples and $\mathbf{Y} = [\mathbf{y_1}, \mathbf{y_2}, \cdots, \mathbf{y_{N_2}}] \in \mathbb{R}^{d \times N_2}$ denote testing samples. Each column represents a sample. To learn the representation coefficients, a general sparse representation model is formulated as

$$\min_{\mathbf{W}} \parallel \mathbf{Y} - \mathbf{X}\mathbf{W} \parallel_F^2 + \lambda \parallel \mathbf{W} \parallel_1 \tag{4}$$

where λ is the regularization parameter for balancing respective term. We introduce $\mathbf{Z} = \mathbf{W}$ to solve model (4) by using augmented Lagrangian function according to ADMM method. The augmented Lagrangian function of problem (4) is formulated as

$$\begin{aligned} \text{Ł}_{\mu}(\mathbf{W},\mathbf{Z},\mathbf{\Lambda}) = \min_{\mathbf{W},\mathbf{Z},\mathbf{\Lambda}} \parallel \mathbf{Y}-\mathbf{X}\mathbf{W} \parallel_F^2 + \lambda \parallel \mathbf{Z} \parallel_1 + < \mathbf{\Lambda}, \mathbf{W}-\mathbf{Z} > \\ + \frac{\mu}{2} \parallel \mathbf{W}-\mathbf{Z} \parallel_F^2 \end{aligned} \tag{5}$$

where $< \mathbf{P}, \mathbf{Q} >= tr(\mathbf{P}^{\mathbf{T}}\mathbf{Q})$, $\mathbf{\Lambda}$ is a Lagrange multiplier and μ is a scalar constant. The augmented Lagrangian is minimized alone one coordinate direction at each iteration. ADMM consists of the following iterations.
(i) Given $\mathbf{Z} = \mathbf{Z}^t, \mathbf{\Lambda} = \mathbf{\Lambda}^t$, updating $\mathbf{W}$ by

$$\mathbf{W}^{t+1} = arg \min_{\mathbf{W}} L_{\mu}(\mathbf{W},\mathbf{Z},\mathbf{\Lambda}) \tag{6}$$

(ii) Given $\mathbf{W} = \mathbf{W}^{t+1}, \mathbf{\Lambda} = \mathbf{\Lambda}^k$, updating $\mathbf{Z}$ by

$$\mathbf{Z}^{t+1} = arg \min_{\mathbf{Z}} L_{\mu}(\mathbf{W},\mathbf{Z},\mathbf{\Lambda}) \tag{7}$$

(iii) Given $\mathbf{W} = \mathbf{W}^{t+1}, \mathbf{Z} = \mathbf{Z}^{t+1}$, updating $\mathbf{\Lambda}$ by

$$\mathbf{\Lambda}^{t+1} = \mathbf{\Lambda}^t + \mu(\mathbf{W}^{t+1} + \mathbf{Z}^{t+1}) \tag{8}$$

The key steps are to solve the optimization problems in Eqs.(6) and (7). Based on the augmented Lagrangian function in Eq.(5), Eq.(6) can be expressed as

$$\mathbf{W}^{t+1} = arg \min_{\mathbf{W}} (\parallel \mathbf{Y}-\mathbf{X}\mathbf{W} \parallel_F^2 + < \mathbf{\Lambda}, \mathbf{W}-\mathbf{Z} > + \frac{\mu}{2} \parallel \mathbf{W}-\mathbf{Z} \parallel_F^2) \tag{9}$$

Since Eq.(9) is a standard regression model, we can get its closed-form solution as follows

$$\mathbf{W}^{t+1} = (\mathbf{X}^T\mathbf{X} + \mu\mathbf{I})^{-1}(\mathbf{X}^T\mathbf{Y} - \mathbf{\Lambda}^t + \mu\mathbf{Z}^t) \tag{10}$$

where $\mathbf{I}$ is a identity matrix. Based on the augmented Lagrangian function in Eq.(5), Eq.(7) can be rewritten as

$$\mathbf{Z}^{t+1} = arg \min_{\mathbf{Z}} (\lambda \parallel \mathbf{Z} \parallel_1 + < \mathbf{\Lambda}, \mathbf{W}-\mathbf{Z} > + \frac{\mu}{2} \parallel \mathbf{W}-\mathbf{Z} \parallel_F^2) \tag{11}$$

Because l_1-norm problem is indifferentiable, the shrinkage technique [15] is used to solve this problem. The optimal solution presents as

$$\mathbf{W}^{t+1} = shrinkage_{\frac{\lambda}{\mu}}(\mathbf{W}^{t+1} + \frac{\mathbf{\Lambda}^t}{\mu}) \tag{12}$$

According to ADMM algorithm, the objective function value will be convergence until certain optimality conditions and stopping criteria are satisfied. In this paper, to simplify this problem, we set a max iteration instead. The detailed process for solving problem (4) is summarized in Algorithm 1.

Algorithm 1. The proposed SSVD

```
Input: Training samples $\mathbf{X}$ and testing samples $\mathbf{Y}$ normalized with $l_2$-norm,
    parameters $\lambda = 10^{-4}, \mu = 10^{-1}$, identity matrix $\mathbf{I}$
Output: $\mathbf{W}$, $\mathbf{Z}$, $\mathbf{\Lambda}$
Initialize: $\mathbf{W}^0 = \mathbf{Z}^0 = \mathbf{\Lambda}^0 = 0$
repeat
    update $\mathbf{W}$: $\mathbf{W}^{t+1} = (\mathbf{X}^{\mathbf{T}}\mathbf{X} + \mu\mathbf{I})^{-1}(\mathbf{X}^{\mathbf{T}}\mathbf{Y} - \mathbf{\Lambda}^t + \mu\mathbf{Z}^t)$
    update $\mathbf{Z}$: $\mathbf{Z}^{t+1} = \mathit{shrinkage}_{\frac{\lambda}{\mu}}(\mathbf{W}^{t+1} + \frac{\mathbf{\Lambda}^t}{\mu})$
    update $\mathbf{\Lambda}$: $\mathbf{\Lambda}^{t+1} = \mathbf{\Lambda}^t + \mu(\mathbf{W}^{t+1} - \mathbf{Z}^{t+1})$
until convergence
```

4 Experimental Results

In this section, we present the experimental results of our proposed SSVD method on publicly available databases, following the same experimental settings in [16]. We randomly split the databases into two part. To avoid special case, all the experiments are run 10 times, and the average recognition rates are reported. Different from [16], we just validate our proposed framework on three face databases (Extended Yale B [17], CMU PIE [18], AR [19]) and one object database (COIL-100 [20]). We compare the proposed method with the popular methods such as LLC, LRC, CRC, SRC, SVM [21] and three methods (ENLR, DENLR, MENLR) proposed in [16]. Our (Max) and Our (Ave) respectively represent the methods to obtain the final softmax vectors in the Image Coding part by using max pooling and average pooling.

In the experiments, we reshape each image into one vector or extract the random feature of image. The l_2-normalization is used for all the samples. The experimental results shows that our method can achieve more significant results than many compared methods especially on face databases. The bold numbers represent the best recognition rate. In the following experiments, we let λ_1, μ_1 represent the parameters in image coding part and λ_2, μ_2 represent the parameters in softmax vector coding part in Fig. 1. The number of layers is set as 10 on all database.

1) Extended Yale B Database: The Extended Yale B database contains 2414 frontal face images of 38 individuals each of them has around 64 near frontal images under different illuminations. We randomly select 15, 20, 25, 30 images per person for training, and the rest for testing. We set $\lambda_1 = 10^{-4}$, $\mu_1 = 10^{-1}$, $\lambda_2 = 10^{-4}$, and $\mu_2 = 1.7$. The recognition rates of different methods on this database are summarized in Table 1. Note that the mean recognition rate are reported, and the bold numbers represent the best recognition rates. It is worth noting that our method can achieve the best recognition rates. Typically, when the number of training samples is 15, the recognition rate of our method is 4 percent higher than MENLR that achieves the best result among the compared methods. Besides, it meas that our method can achieve good recognition rate when there is less training samples on this database.

2) CMU PIE Database: The CMU PIE face database contains 41,368 face images from 68 subjects as a whole. The images under five near frontal poses (C05, C07, C09, C27 and C29) are used in our experiment. We randomly select 15, 20, 25, 30 images from each subject as training samples and the remaining images as test samples. We set $\lambda_1 = 10^{-4}$, $\mu_1 = 10^{-1}$, $\lambda_2 = 10^{-4}$, and $\mu_2 = 10^{-2}$. The classification rates of different methods are summarized in Table 2. It is clear that our method outperforms the compared methods in different cases.

Table 1. Recognition rates (%) on Extended Yale B database with different number of training samples

Alg.	15	20	25	30
LLC	88.63	91.52	94.20	95.21
LRC	89.47	92.52	93.50	94.62
CRC	91.39	94.26	95.91	97.04
SRC	91.72	93.71	95.56	96.37
SVM	89.35	92.74	95.07	96.20
ENLR	92.18	94.28	95.70	96.80
DENLR	94.34	96.66	97.70	98.51
MENLR	94.76	97.27	97.68	98.74
Our(Max)	**98.87**	**99.51**	**99.63**	**99.79**
Our(Ave)	98.68	99.44	99.62	99.73

Table 2. Recognition rates (%) on CMU PIE database with different number of training samples

Alg.	15	20	25	30
LLC	84.62	90.90	93.27	94.46
LRC	85.61	90.17	92.65	94.01
CRC	89.76	92.42	93.80	94.61
SRC	88.97	91.14	92.62	93.71
SVM	86.66	90.70	92.66	93.06
ENLR	90.47	92.82	93.94	94.67
DENLR	92.25	94.06	95.61	95.86
MENLR	93.21	94.88	95.74	96.18
Our(Max)	91.44	93.73	94.95	95.66
Our(Ave)	**93.79**	**95.59**	**96.37**	**96.84**

3) AR Database: The AR face database contains about 4,000 color face images of 126 subject, which consist of the frontal faces with different facial expressions, illuminations and disguises. In this experiment, we select a subset including 2600 images from 50 female and 50 male subjects. We randomly select 8, 11, 14, 17 images for each subject as training samples and the rest of images as test samples. Following the experiment in [22], each image and its subregion are projected onto a 540-dimensional feature vector with a randomly generated matrix from a zero-mean normal distribution. We set $\lambda_1 = 10^{-5}$, $\mu_1 = 2$, $\lambda_2 = 10^{-5}$, and $\mu_2 = 10^{-3}$. The recognition rates of different methods on this database are summarized in Table 3. From the table, we can see that our method achieves the best recognition rates.

4) COIL-100 Database: Columbia Object Image Library (COIL-100) database contains various views of 100 objects (72 images per object) with different lighting conditions. In our experiments, the images are converted to gray-scale images with the 32×32 pixels. We randomly select 15, 20, 25, 30 images per object to construct the training set, and the test set contains the rest of the images. We set $\lambda_1 = 10^{-2}$, $\mu_1 = 1$, $\lambda_2 = 10^{-4}$, and $\mu_2 = 10^{-2}$. The recognition rates of different methods on this database are summarized in Table 4. We can see that our method is inferior to the best MENLR, but still better than other methods.

Table 3. Recognition rates (%) on AR database with different number of training samples

Alg.	8	11	14	17
LLC	54.26	60.87	66.88	71.58
LRC	63.87	76.87	85.20	90.88
CRC	86.53	91.66	94.06	95.74
SRC	84.08	89.45	92.20	95.14
SVM	75.74	86.19	91.99	95.08
ENLR	90.42	93.80	95.41	96.31
DENLR	91.94	95.69	97.30	98.21
MENLR	92.61	95.63	97.16	98.56
Our(Max)	92.72	96.66	97.65	98.27
Our(Ave)	**95.15**	**97.31**	**98.13**	**98.70**

Table 4. Recognition rates (%) on COIL-100 database with different number of training samples

Alg.	15	20	25	30
LLC	86.93	90.25	92.50	93.84
LRC	85.33	88.79	91.09	92.63
CRC	81.36	84.33	86.33	87.72
SRC	86.10	89.47	91.99	93.91
SVM	84.89	88.10	90.80	92.44
ENLR	88.40	91.28	93.37	94.66
DENLR	91.92	94.36	95.80	96.87
MENLR	**92.75**	**94.88**	**96.34**	**97.36**
Our(Max)	89.51	92.77	94.55	95.90
Our(Ave)	91.09	93.95	95.48	96.89

In summary, the proposed SSVD model can achieve remarkable results on face databases. It is also worth noting that SSVD (Max) outperforms SSVD (Ave) on Extended Yale B database and is inferior to SSVD (Ave) on CMU PIE, AR and COIL-100 database. The important advantage of SSVD model is that each image is divided into 4 or 16 subregions, which means that one image can be represented 4 or 16 times. It is useful to amend the misclassified image. We take an example to explain the effect of max pooling and average pooling. Suppose that there is a four categories image set split into two parts training set and testing set. Given a misclassified testing image that is actually from class 1, we will obtain its softmax vector $r = [0.25 \quad 0.40 \quad 0.15 \quad 0.20]^T$. As for its subregions, there are two cases. (1) There exist one subregion (first subregion we suppose) which shows much more discriminative than the whole image and other subregions. We let $r_1 = [0.60 \quad 0.20 \quad 0.10 \quad 0.10]^T$, $r_2 = [0.30 \quad 0.45 \quad 0.10 \quad 0.15]^T$, $r_3 = [0.25 \quad 0.50 \quad 0.10 \quad 0.15]^T$, and $r_4 = [0.30 \quad 0.35 \quad 0.25 \quad 0.10]^T$ respectively represent the softmax vectors of the 4 subregions. After the max pooling, we will obtain the final softmax vector $r' = [0.60 \quad 0.50 \quad 0.25 \quad 0.15]^T$ which can amend the misclassified image. (2) The above case is unusual in reality. Instead, there more likely exist most subregions which show a lot discriminative than other subregions. The misclassified image and its softmax vector are the same as case (1). We let $r_1 = [0.35 \quad 0.25 \quad 0.15 \quad 0.25]^T$, $r_2 = [0.40 \quad 0.20 \quad 0.30 \quad 0.10]^T$, $r_3 = [0.20 \quad 0.50 \quad 0.10 \quad 0.20]^T$, and $r_4 = [0.45 \quad 0.15 \quad 0.20 \quad 0.20]^T$ respectively represent the softmax vectors of the 4 subregions. After the average pooling, we will obtain the final softmax vector $r' = [0.35 \quad 0.28 \quad 0.19 \quad 0.19]^T$ which can also amend the misclassified image.

Discussion of Layers: To better illustrate our methods, we give the sample curves (presented in Fig. 3) that shows the recognition rates with different layers in the deep model for each database. It is clear that as the number of layers increases, the recognition rate represents a rising tendency, which demonstrates the effectiveness of deep cascade model.

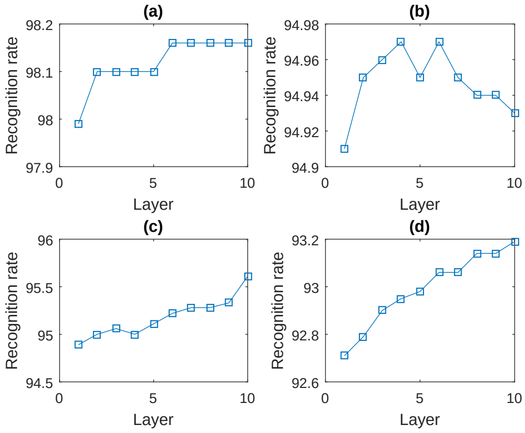

Fig. 3. The recognition rates with different layers on different database: (a) Extended Yale B database, (b) CMU PIE Database, (c) AR database, (d) COIL-100 database.

Convergence: To illustrate the effectiveness of our solving algorithm for problem (4), we show the objective function values with the varying iteration number (presented in Fig. 4) on the Extended Yale B database by using the Algorithm.1 to solve problem (4). It is easy to find that the objective function values present a convergence trend, which demonstrates the effectiveness of the Algorithm.1.

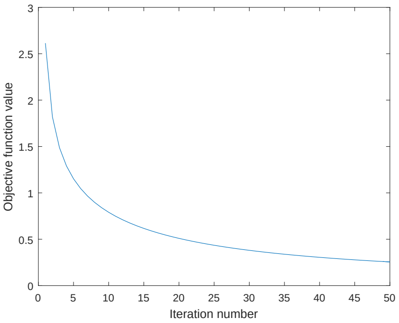

Fig. 4. The convergence of Algorithm.1.

5 Conclusion

This paper presented a novel sparse softmax vector coding based deep cascade model (SSVD). One important advantage of this model is using the class discrimination softmax vector. Besides, some sub-patterns show more discriminative than the whole image, which can amend the misclassified image by using max-polling or average-polling. We also explored the effectiveness of the concatenated softmax vector. The extensive experimental results clearly demonstrated that the proposed method outperforms significantly previous methods.

Acknowledgements. This work was supported by the National Science Fund of China under Grants (61771079, 61401048) and the Fundamental Research Funds for the Central Universities (No. 106112017CDJQJ168819).

References

1. I. Naseem, R. Togneri, and M. Bennamounand.: Linear regression for face recognition. IEEE Transactions on PAMI, vol. 32, no. 11, pp. 2106–2112 (2010)
2. J. Wright, A. Y. Yang, A. Ganesh, S. S. Sastry, and Y. Ma.:Robust face recognition via sparse representation. IEEE Transactions on PAMI, vol. 31, no. 2, pp. 210–227 (2009)
3. L. Zhang, M. Yang, and X. Feng.:Sparse representation or collaborative representation: Which helps face recognition?. in ICCV. IEEE, pp. 471–478 (2011).
4. M. Yang, D. Zhang, J. Yang, and D. Zhang.:Robust sparse coding for face recognition. in CVPR. IEEE, pp. 625–632 (2011)
5. R. He, W.-S. Zheng, and B.-G. Hu.:Maximum correntropy criterion for robust face recognition.IEEE Transactions on PAMI, vol. 33, no. 8, pp. 1561–1576 (2011)
6. R. He, W.-S. Zheng, T. Tan, and Z. Sun.:Half-quadratic-based iterative minimization for robust sparse representation. IEEE Transactions on PAMI, vol. 36, no. 2, pp. 261–275 (2014)
7. I. Naseem, R. Togneri, and M. Bennamoun.:Robust regression for face recognition. Pattern Recognition, vol. 45, no. 1, pp. 104–118 (2012)
8. Z. Zhou, A. Wagner, H. Mobahi, J. Wright, and Y. Ma.:Face recognition with contiguous occlusion using markov random fields. in ICCV. IEEE, pp. 1050–1057 (2009)
9. K. Jia, T.-H. Chan, and Y. Ma.:Robust and practical face recognition via structured sparsity. ECCV. Springer, pp. 331–344 (2009)
10. C. Ren, D. Dai, and H. Yan.:Coupled kernel embedding for low-resolution face image recognition. IEEE Transactions on Image Processing, vol. 21, no. 8, pp. 3770–3783 (2012)
11. Y. Xu, X. Li, J. Yang, Z. Lai, and D. Zhang.:Integrating conventional and inverse representation for face recognition. IEEE Transactions on Cybernetics, vol. 44, no. 10, pp. 1738–1746 (2014)
12. E. J. Cands, X. Li, Y. Ma, and J. Wright.:Robust principal component analysis?. J. ACM, vol. 58, no. 3, p. 11 (2011)
13. Y. Li, J. Liu, H. Lu, and S. Ma.:Learning robust face representation with classwise block-diagonal structure. IEEE Trans. Inf. Forensics Security, vol. 9, no. 12, pp. 2051–2062 (2014)
14. D. Gabay and B. Mercier.:A dual algorithm for the solution of nonlinear variational problems via finite element approximations. IEEE Transactions on Image Processing. , vol. 22, no. 1, pp. 17–40 (1976)

15. J. Yang and Y. Zhang.:Alternating direction algorithms for l_1-problems in compressive sensing. SIAM J. Sci. Comput., vol. 33, no. 1, pp. 250–278 (2011)
16. Z. Zhang, Z. Lai, Y. Xu, L. Shao, J. Xu, and G.-S. Xie.:Discriminative Elastic-Net Regularized Linear Regression. IEEE Transactions on Image Processing. , vol. 26, no. 3, pp. 1466–1481 (2016)
17. 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 Transactions on PAMI. , vol. 23, no. 6, pp. 643–660 (2001)
18. T. Sim, S. Baker, and M. Bsat.:The CMU pose, illumination, and expression (PIE) database in Proc. 5th IEEE Int. Conf. Autom. Face Gesture Recognit., pp. 46–51 (2002)
19. A. M. Martinez and R. Benavente.:The AR Face Database. CVC Tech. Rep. 24, Jun. (1998).
20. S. A. Nene, S. K. Nayar, and H. Murase.:Columbia Object Image Library (COIL-100). Tech. Rep. CUCS-006-96 (1996)
21. C.-C. Chang and C.-J. Lin.:LIBSVM: A library for support vector machines. ACM Trans. Intell. Syst. Technol., vol. 2, no. 3, pp. 27:1–27:27 (2011)
22. Z. Jiang, Z. Lin, and L. S. Davis.:Label consistent K-SVD: Learning a discriminative dictionary for recognition. IEEE Transactions on PAMI, vol. 35, no. 11, pp. 2651–2664 (2013)
23. Z.-H. Zhou, J. Feng.:Deep Forest: Toward An Alternative to Deep Neural Network. in IJCAI (2017)