Journal of Electronic Imaging

JElectronicImaging.org

Class-specific residual constraint non-negative representation for pattern classification

He-Feng Yin
Xiao-Jun Wu

SPIE. IS&T imaging.org

He-Feng Yin, Xiao-Jun Wu, "Class-specific residual constraint non-negative representation for pattern classification," *J. Electron. Imaging* **29**(2), 023014 (2020), doi: 10.1117/1.JEI.29.2.023014

Class-specific residual constraint non-negative representation for pattern classification

He-Feng Yin[a,b] and Xiao-Jun Wu[a,b,*]
[a]Jiangnan University, School of Internet of Things Engineering, Lihu Avenue, Wuxi, China
[b]Jiangsu Provincial Laboratory of Pattern Recognition and Computational Intelligence, Lihu Avenue, Wuxi, China

Abstract. Representation-based classification methods (RBCM) remain one of the hottest topics in the community of pattern recognition, and the recently proposed non-negative representation-based classification (NRC) achieved impressive recognition results in various classification tasks. However, NRC ignores the relationship between the coding and classification stages. Moreover, there is no regularization term other than the reconstruction error term in the formulation of NRC, which may result in an unstable solution leading to misclassification. To overcome these drawbacks of NRC, we propose a class-specific residual constraint non-negative representation (CRNR) for pattern classification. CRNR introduces a class-specific residual constraint into the formulation of NRC, which encourages training samples from different classes to competitively represent the test sample. Based on the proposed CRNR, we develop a CRNR-based classifier (CRNRC) for pattern classification. Experimental results on several benchmark datasets demonstrate the superiority of CRNRC over conventional RBCM as well as the recently proposed NRC. Moreover, CRNRC works better than or comparable to some state-of-the-art deep approaches on diverse challenging pattern classification tasks. © *2020 SPIE and IS&T* [DOI: 10.1117/1.JEI.29.2.023014]

Keywords: pattern classification; representation-based classification; non-negative representation; class-specific residual constraint.

Paper 191001 received Oct. 30, 2019; accepted for publication Mar. 16, 2020; published online Mar. 31, 2020.

1 Introduction

During the past few years, representation-based classification methods (RBCM) have received considerable attention in various classification tasks, such as face classification[1] and hyperspectral image classification.[2] In the domain of face recognition, the popular work is the sparse representation-based classification (SRC).[3] SRC treats all training samples as a dictionary, and a test sample is sparsely coded over the dictionary with the ℓ_1-norm constraint; then the classification is accomplished by checking which class yields the least reconstruction error. SRC can achieve excellent recognition results even when the test samples are occluded or corrupted. Due to the principle of ℓ_1-minimization, SRC tends to lose locality information. However, Yu et al.[4] revealed that under certain assumptions locality is more essential than sparsity. To incorporate the locality structure of data into the sparse coding of test data, Wang et al.[5] developed a locality-constraint linear coding (LLC) scheme. Similarly, Lu et al.[6] proposed a weighted sparse representation-based classification (WSRC) method. Different from the ℓ_1-norm used in WSRC, LLC employs ℓ_2-norm by which the analytical solution can be derived. Zhong et al.[7] proposed an $\ell_{1/2}$ classification method for face recognition, which is inspired by the $\ell_{1/2}$ regularization.[8] Inspired by the biological discovery that dissimilar inputs have dissimilar codes, Liao and Gu[9] introduced a nonlocal constraint-based sparse representation classifier. When the size of the dictionary is huge, the ℓ_1-norm minimization problem in SRC is computationally expensive. To speed up the sparse coding process, Li et al.[10] presented a local sparse representation-based classification (LSRC) scheme that performs sparse decomposition in a local neighborhood. Ortiz and

*Address all correspondence to Xiao-Jun Wu, E-mail: wu_xiaojun@jiangnan.edu.cn

Becker[11] designed a linearly approximated sparse representation-based classification algorithm that employs linear regression to perform sample selection for ℓ_1-minimization. In SRC, the representation fidelity is measured by the ℓ_2-norm of the coding residual, which assumes that the coding residual follows Gaussian distribution. However, it may not be capable of describing the coding errors in practical applications. To tackle this problem, Yang et al.[12] proposed a robust sparse coding (RSC) model that seeks the maximum likelihood estimation solution to the sparse coding problem. Nevertheless, the iterative algorithm to solve RSC is computationally expensive. To improve the efficiency of the iterative algorithm and further increase the robustness of RSC, Zhong et al.[13] presented an improved robust sparse coding algorithm in which the dictionary is reduced by eliminating the objects with larger coding residuals.

Another prevailing approach of RBCM is collaborative representation-based classification (CRC). Zhang et al.[14] argued that it is the collaborative representation (CR) mechanism rather than the ℓ_1-norm sparsity that makes SRC powerful for classification. Compared with SRC, CRC can achieve comparable performance, but the computational complexity is significantly reduced. Similarly, Xu et al.[15] introduced a discriminative sparse representation method for robust face recognition via ℓ_2 regularization. Timofte and Van Gool[16] investigated the weighted variants of CR and proposed a weighted collaborative representation (WCR) algorithm. Li et al.[17] designed a sparsity augmented WCR-based classification approach for image recognition. To fully exploit the strengths of diverse RBCM, Chi and Porikli[18] presented a collaborative representation optimized classifier (CROC) that achieves a balance between the nearest subspace classifier (NSC)[19] and CRC. Although Zhang et al.[14] offered a geometric interpretation of the classification mechanism of CRC, understanding its intrinsic principle remains obscure. Afterward, Cai et al.[20] analyzed the classification mechanism of CRC from a probabilistic viewpoint and proposed a probabilistic collaborative representation-based classifier (ProCRC). Based on ProCRC, Gou et al.[21] presented a two-phase probabilistic collaborative representation-based classification that adopts the coarse-to-fine strategy.

Although SRC and CRC (and their extensions) achieve impressive recognition results in various classification tasks, they cannot avoid negative entries in their coding vectors for test samples. The negative coefficients indicate negative data correlations between the test and the training samples. Inspired by the principle of non-negative matrix factorization,[22] Xu et al.[23] proposed a non-negative representation-based classification (NRC) that introduces a non-negative constraint on the coding vector. Extensive experiments on diverse classification tasks demonstrate the superiority of NRC over many existing RBCM, including SRC, CRC, and ProCRC. NRC consists of two separate stages: first, employing all of the training samples to represent the test sample; second, classifying the test sample into the class that yields the least residual. Nevertheless, in addition to the reconstruction error term, there is no other regularization term in the formulation of NRC, which may produce an unstable solution and lead to misclassification. Moreover, NRC ignores the relationship between the first and second stages. To tackle these problems, we incorporate a class-specific residual constraint into the formulation of NRC and propose a class-specific residual constraint non-negative representation (CRNR) for pattern classification. Through this constraint, training samples across different classes are encouraged to competitively reconstruct the test sample. We find that minimizing this constraint is equivalent to the NSC model. Therefore, CRNR can be regarded as a combination of the CRC and NSC models with the non-negative constraint for obtaining the coding vector of the test sample.

To illustrate the mechanism of CRNRC, we carry out an experiment on the MNIST dataset. This dataset contains images for digits 0 to 9, and 50 images per class are selected to form the training set. The 500 images are arranged in an order of $[0,1,2,\ldots,9]$; thus the training data matrix is denoted by $\mathbf{X} = [\mathbf{X}_0, \mathbf{X}_1, \ldots, \mathbf{X}_9]$, and we choose a test sample from the 10th class (i.e., digit 9). The coding vector and residual obtained by NRC are shown in Figs. 1(a) and 1(b), respectively, while the coding vector and residual obtained by CRNRC are shown in Figs. 1(c) and 1(d), respectively. From Fig. 1(b), one can see that the fifth class has the least residual, i.e., the test sample is recognized as digit 4 even though the 10th class has dominant coefficients in Fig. 1(a). From Fig. 1(d), we can observe that the 10th class results in the minimum residual, i.e., the test sample is correctly recognized as digit 9. When we have a closer look at the coding vector of NRC and CRNRC, for the fifth class (indices 201 to 250), the number of nonzero entries of

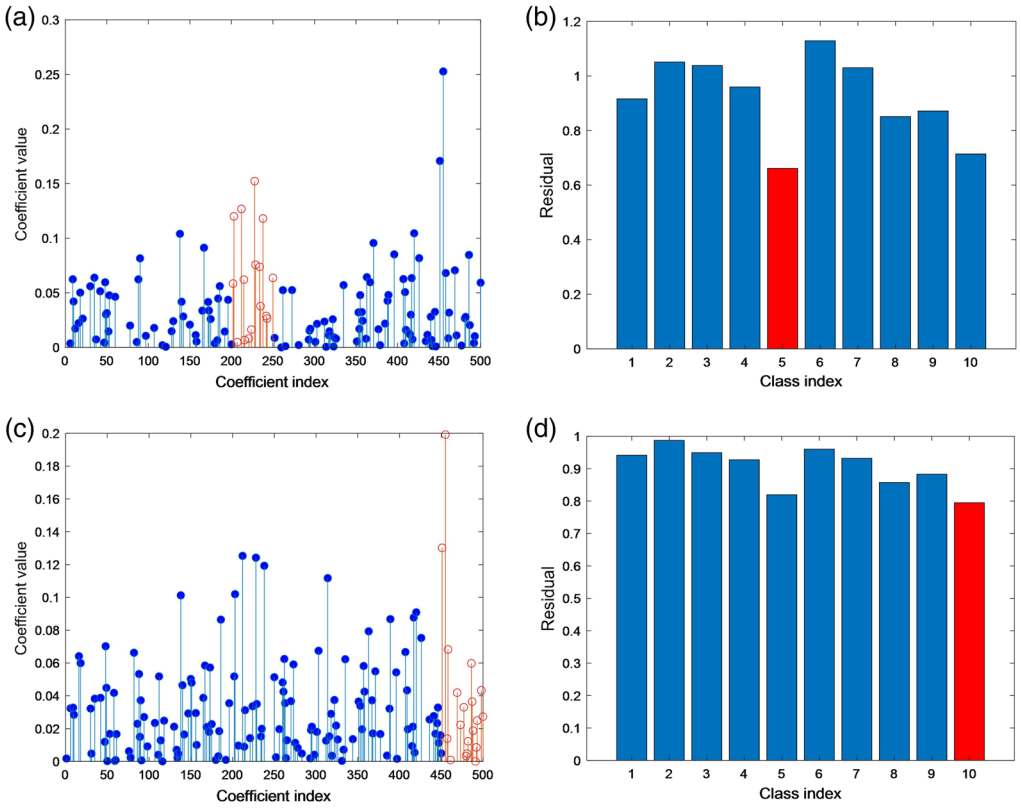

Fig. 1 Coding vector and residual obtained by NRC and CRNRC. The test sample belongs to the 10th class, i.e., digit 9. (a) Coding vector obtained by NRC; (b) class-specific residual of NRC, one can see that the fifth class yields the least residual, thus the test sample digit 9 is misclassified into digit 4; (c) coding vector obtained by our proposed method; (d) class-specific residual obtained by CRNRC, we can observe that the 10th class leads to the minimum residual. Therefore, the test sample is correctly classified into digit 9.

NRC and CRNRC is 16 and 14, respectively. While for the 10th class (indices 451 to 500), the number of nonzero entries of NRC and CRNRC is 15 and 19, respectively. By introducing the class-specific residual constraint into NRC, more coefficients are concentrated on the correct class; thus improved performance of our proposed CRNRC can be expected. The source code of our proposed CRNRC is accessible at https://github.com/yinhefeng/CRNRC.

Our main contributions are summarized as follows.

- A CRNR scheme is proposed. Moreover, a CRNR-based classifier (CRNRC) is designed for pattern classification.
- The resulting optimization problem of CRNR is elegantly solved under the framework of the alternating direction method of multipliers (ADMM).[24]
- The proposed CRNRC outperforms conventional RBCM and works better than or comparable to some state-of-the-art (SOTA) deep approaches on various standard datasets.

The rest of the paper is structured as follows. Section 2 reviews the related work. Section 3 presents the CRNRC approach. Experimental evaluation on several benchmark databases is presented in Sec. 4. Finally, Sec. 5 concludes this paper.

2 Related Work

In this section, we will briefly review some related work, including SRC,[3] CRC,[14] and NRC.[23] To begin with, we present some notations used throughout this paper. Suppose we have n training samples from K classes; then the training data matrix is denoted by $\mathbf{X} = [\mathbf{X}_1, \mathbf{X}_2, \ldots, \mathbf{X}_K] =$

$[\boldsymbol{x}_1, \boldsymbol{x}_2, \ldots, \boldsymbol{x}_n] \in \mathbb{R}^{d \times n}$, where $\mathbf{X}_i$ is the data matrix of the i'th class. The i'th class has n_i training samples and $\sum_{i=1}^{K} n_i = n$ $(i = 1,2,\ldots,K)$; d is the dimensionality of vectorized samples.

2.1 *Sparse Representation-Based Classification*

In SRC, a test sample $\boldsymbol{y} \in \mathbb{R}^d$ is first expressed as a sparse linear superposition of all of the training data; then the classification is done by checking which class yields the minimum reconstruction error, and the objective function of SRC is formulated as

$$\min_{\boldsymbol{c}} \|\boldsymbol{y} - \mathbf{X}\boldsymbol{c}\|_2^2 + \lambda\|\boldsymbol{c}\|_1, \tag{1}$$

where $\lambda > 0$ is a balancing parameter. Once we obtain the coefficient vector $\boldsymbol{c}$ of $\boldsymbol{y}$; the test sample $\boldsymbol{y}$ is classified according to the following equation:

$$\text{identity}(\boldsymbol{y}) = \arg\min_i \|\boldsymbol{y} - \mathbf{X}_i\boldsymbol{c}_i\|_2, \tag{2}$$

where $\boldsymbol{c}_i$ is the coefficient vector that corresponds to the i'th class.

2.2 *Collaborative Representation-Based Classification*

Zhang et al.[14] presented a CRC algorithm that replaces the ℓ_1-norm in SRC with the ℓ_2-norm constraint. The objective function of CRC is formulated as follows:

$$\min_{\boldsymbol{c}} \|\boldsymbol{y} - \mathbf{X}\boldsymbol{c}\|_2^2 + \lambda\|\boldsymbol{c}\|_2^2. \tag{3}$$

CRC has the following closed-form solution:

$$\boldsymbol{c} = (\mathbf{X}^T\mathbf{X} + \lambda\mathbf{I})^{-1}\mathbf{X}^T\boldsymbol{y}, \tag{4}$$

where $\mathbf{I}$ is an identity matrix. Let $\mathbf{P} = (\mathbf{X}^T\mathbf{X} + \lambda\mathbf{I})^{-1}\mathbf{X}^T$; we can see that $\mathbf{P}$ is only determined by the training data matrix $\mathbf{X}$. Therefore, when given all of the training data, $\mathbf{P}$ can be precomputed, which makes CRC very efficient. CRC employs the following regularized residual for classification:

$$\text{identity}(\boldsymbol{y}) = \arg\min_i \frac{\|\boldsymbol{y} - \mathbf{X}_i\boldsymbol{c}_i\|_2}{\|\boldsymbol{c}_i\|_2}. \tag{5}$$

2.3 *Non-Negative Representation-Based Classification*

SRC and CRC have become two representative approaches in RBCM. However, the coding vector of conventional RBCM contains negative entries. The test sample should be better expressed by homogeneous samples with non-negative representation coefficients. Moreover, Lee and Seung[22] pointed out that it is unsuitable to approximate the test sample by allowing the training samples to "cancel each other out" with complex additions and subtractions. Therefore, Xu et al.[23] proposed the following non-negative representation model by imposing the non-negative constraint on the coding vector:

$$\min_{\boldsymbol{c}} \|\boldsymbol{y} - \mathbf{X}\boldsymbol{c}\|_2^2, \quad \text{s.t. } \boldsymbol{c} \geq 0. \tag{6}$$

Similar to SRC, NRC employs the class-specific residual to classify the test sample, i.e., $\text{identity}(\boldsymbol{y}) = \arg\min_i \|\boldsymbol{y} - \mathbf{X}_i\boldsymbol{c}_i\|_2$.

3 Class-Specific Residual Constraint Non-Negative Representation

In this section, first we present the model of our proposed CRNR; then we design a CRNRC for pattern classification.

3.1 *Proposed Model*

From Eq. (6), we can see that, apart from the reconstruction error term, there is no other regularized term in the objective function of NRC. As demonstrated by the illustration in Sec. 1, due to lack of regularization, NRC would result in misclassification. Furthermore, NRC ignores the relationship between the coding and classification stages. To alleviate these problems, we incorporate a class-specific residual constraint into the formulation of NRC, and the objective function of our CRNR is formulated as follows:

$$\min_{\boldsymbol{c}} \|\boldsymbol{y}-\mathbf{X}\boldsymbol{c}\|_2^2+\lambda\sum_{i=1}^{K}\|\boldsymbol{y}-\mathbf{X}_i\boldsymbol{c}_i\|_2^2, \quad \text{s.t. } \boldsymbol{c}\geq 0, \tag{7}$$

where $\lambda>0$ is a balancing parameter. The first term in Eq. (7) is the CR term, and the second term is the class-specific residual constraint. For a specific class, the second term becomes $\|\boldsymbol{y}-\mathbf{X}_i\boldsymbol{c}_i\|_2^2$, which is the formulation of NSC.[19] Therefore, our proposed CRNR can be viewed as an integration of CRC and NSC with the non-negative constraint to compute the representation. One can see that, when $\lambda=0$, CRNR is degenerated to NRC. Therefore, NRC can be viewed as a special version of CRNRC.

3.2 *Optimization*

We adopt an alternative strategy to solve the CRNR model. By introducing an auxiliary variable z, Eq. (7) is rewritten as

$$\min_{\boldsymbol{c},z} \|\boldsymbol{y}-\mathbf{X}\boldsymbol{c}\|_2^2+\lambda\sum_{i=1}^{K}\|\boldsymbol{y}-\mathbf{X}_i\boldsymbol{c}_i\|_2^2, \quad \text{s.t. } \boldsymbol{c}=z,\ z\geq 0. \tag{8}$$

Equation (8) can be solved by the ADMM[24] technique, and the Lagrangian function of Eq. (8) is

$$\mathcal{L}(\boldsymbol{c},z,\boldsymbol{\delta},\mu)=\|\boldsymbol{y}-\mathbf{X}\boldsymbol{c}\|_2^2+\lambda\sum_{i=1}^{K}\|\boldsymbol{y}-\mathbf{X}_i\boldsymbol{c}_i\|_2^2+\langle\boldsymbol{\delta},z-\boldsymbol{c}\rangle+\frac{\mu}{2}\|z-\boldsymbol{c}\|_2^2, \tag{9}$$

where $\boldsymbol{\delta}$ is the Lagrange multiplier and $\mu>0$ is a penalty parameter. The optimization of Eq. (9) can be solved iteratively by updating $\boldsymbol{c}$ and z one at a time. The detailed updating procedures are presented as follows.

Update $\boldsymbol{c}$. Fix the other variables and update $\boldsymbol{c}$ by solving the following problem:

$$\min_{\boldsymbol{c}} \|\mathbf{y}-\mathbf{X}\boldsymbol{c}\|_2^2+\lambda\sum_{i=1}^{K}\|\boldsymbol{y}-\mathbf{X}_i\boldsymbol{c}_i\|_2^2+\frac{\mu}{2}\left\|z_t-\boldsymbol{c}+\frac{\delta_t}{\mu}\right\|_2^2. \tag{10}$$

Suppose $\mathbf{X}_i'$ is a matrix that has the same size as $\mathbf{X}$ and $\mathbf{X}_i'$ only consists of samples from the i'th class, i.e., $\mathbf{X}_i'=[\mathbf{0},\ldots,\mathbf{X}_i,\ldots,\mathbf{0}]$; Eq. (10) is reformulated as

$$\min_{\boldsymbol{c}}\|\mathbf{y}-\mathbf{X}\boldsymbol{c}\|_2^2+\lambda\sum_{i=1}^{K}\|\boldsymbol{y}-\mathbf{X}_i'\boldsymbol{c}\|_2^2+\frac{\mu}{2}\left\|z_t-\boldsymbol{c}+\frac{\delta_t}{\mu}\right\|_2^2. \tag{11}$$

Setting the partial derivative of Eq. (11) with respect to $\boldsymbol{c}$ to zero, we obtain the following closed-form solution:

$$\boldsymbol{c}_{t+1}=\left(\mathbf{X}^T\mathbf{X}+\lambda\sum_{i=1}^{K}(\mathbf{X}_i')^T(\mathbf{X}_i')+\frac{\mu}{2}\mathbf{I}\right)^{-1}\left[(1+\lambda)\mathbf{X}^T\boldsymbol{y}+\frac{\mu z_t+\delta_t}{2}\right]. \tag{12}$$

Update z. To update z, we fix variables other than z and solve the following problem accordingly:

Algorithm 1 Solve Eq. (7) via ADMM.

Input: Test sample $\boldsymbol{y}$, training data matrix $\mathbf{X}$, balancing parameter λ, tol > 0, $\mu > 0$ and the maximum iteration number T.

1: Initialize $\boldsymbol{z}_0 = \boldsymbol{c}_0 = \boldsymbol{\delta}_0 = \mathbf{0}$;

2: **while** not converged **do**

3: Update $\boldsymbol{c}$ by Eq. (12);

4: Update $\boldsymbol{z}$ by Eq. (14);

5: Update δ by Eq. (15);

6: **end while**

Output: Coding vectors $\boldsymbol{z}$ and $\boldsymbol{c}$.

$$\min_{z} \left\| z - \left(\boldsymbol{c}_{t+1} - \frac{\boldsymbol{\delta}_t}{\mu} \right) \right\|_2^2, \quad \text{s.t. } z \geq 0. \tag{13}$$

The solution to z is given as

$$z_{t+1} = \max\left(0, \boldsymbol{c}_{t+1} - \frac{\boldsymbol{\delta}_t}{\mu}\right), \tag{14}$$

where the "max" operator performs element by element.

Update $\boldsymbol{\delta}$. The Lagrange multiplier $\boldsymbol{\delta}$ is updated according to the following formulation:

$$\boldsymbol{\delta}_{t+1} = \boldsymbol{\delta}_t + \mu(z_{t+1} - \boldsymbol{c}_{t+1}). \tag{15}$$

The detailed procedures of solving Eq. (7) are described in Algorithm 1.

3.3 *Classification*

For the test sample $\boldsymbol{y} \in \mathbb{R}^d$, first we obtain its coding vector $\boldsymbol{c}$ over the entire training data $\mathbf{X}$ by solving Eq. (7); then the test sample is designated into the class that yields the least residual, i.e., identity$(\boldsymbol{y}) = \arg\min_i \|\boldsymbol{y} - \mathbf{X}_i \boldsymbol{c}_i\|_2$, where $\boldsymbol{c}_i$ is the coding vector that belongs to the i'th class. The complete process of our proposed CRNRC is summarized in Algorithm 2.

4 Experiments

In this section, we evaluate the classification performance of CRNRC on diverse benchmark datsets: three face databases including AR,[25] Extended Yale B,[26] and Georgia Tech (GT)[27],

Algorithm 2 Our proposed CRNRC algorithm.

Input: Training data matrix $\mathbf{X} = [\mathbf{X}_1, \mathbf{X}_2, \ldots, \mathbf{X}_K] \in \mathbb{R}^{d \times n}$, test data $\boldsymbol{y} \in \mathbb{R}^d$ and balancing parameter λ.

1: Normalize the columns of $\mathbf{X}$ and $\boldsymbol{y}$ to have unit ℓ_2 norm;

2: Obtain the coding vector $\boldsymbol{c}$ of $\boldsymbol{y}$ on $\mathbf{X}$ by solving the CRNR model in Eq. (7);

3: Compute the class-specific residuals $\boldsymbol{r}_i = \|\boldsymbol{y} - \mathbf{X}_i \boldsymbol{c}_i\|_2$;

Output: label$(\boldsymbol{y}) = \arg\min_i(\boldsymbol{r}_i)$.

two handwritten digit datasets including USPS[28] and MNIST,[29] and four large-scale datasets including the Stanford 40 dataset,[30] the Oxford 102 Flowers dataset,[31] the Aircraft dataset,[32] and the Cars dataset.[33] We compare the classification accuracy of CRNRC with NSC,[19] linear SVM, SRC,[3] CRC,[14] CROC,[18] ProCRC,[20] and NRC,[23] respectively. In addition, on the GT database, we compare CRNRC with several inception deep architectures, i.e., GoogLeNet,[34] Inception-v3,[35] Xception,[36] and Inception-ResNet-v2.[37] On the four large-scale datasets, we compare CRNRC with the SOTA methods.

4.1 *Face Recognition*

4.1.1 *Experiments on the AR database*

The AR database[25] consists of more than 4000 color images of 126 subjects (70 men and 56 women); these images have variations in facial expressions, illumination conditions, and occlusions. Example images from this database are shown in Fig. 2. Following the experimental settings in Ref. 23, in our experiments, we use a subset with only illumination and expression changes that contains 50 male subjects and 50 female subjects from the AR database. For each individual, 7 images from session 1 are used as training samples, and the other 7 images from session 2 as test samples. All of the images are first cropped to 60×43 pixels and projected to a subspace of dimensions 54, 120, and 300 by principal component analysis (PCA). Experimental results are summarized in Table 1; the balancing parameter λ of CRNRC under dimensions 54, 120, and 300 are set to 0.001, 0.01, and 0.001, respectively. One can see that our proposed CRNRC achieves the highest recognition accuracy under all of the three reduced dimensions.

4.1.2 *Experiments on the Extended Yale B database*

The Extended Yale B database[26] contains 2414 face images from 38 individuals, each having 59 to 64 images. These images have illumination variations; example images from this database are shown in Fig. 3. The original images are of 192×168 pixels. In our experiments, all of the images are resized to 54×48 pixels. We randomly selected 32 images per subject for training and the remaining for testing. The resized images are projected to a subspace of dimensions 84, 150, and 300 by PCA. Table 2 lists the classification accuracy of the competing approaches; the balancing parameter λ of CRNRC under dimensions 84, 150, and 300 is set to 0.001. It is seen that CRNRC consistently outperforms its competing approaches in all cases.

Fig. 2 Example images from the AR database.

Table 1 Recognition accuracy (%) of competing approaches on the AR database.

	Dimensions		
	54	120	300
NSC[19]	70.7	75.5	76.1
SVM	81.6	89.3	91.6
SRC[3]	82.1	88.3	90.3
CRC[14]	80.3	90.1	93.8
CROC[18]	82.0	90.8	93.7
ProCRC[20]	81.4	90.7	93.7
NRC[23]	85.2	**91.3**	93.3
CRNRC	**85.7**	**91.3**	**94.0**

Bold values signify the best classification result.

Fig. 3 Example images from the Extended Yale B database.

Table 2 Recognition accuracy (%) of competing approaches on the Extended Yale B database.

	Dimensions		
	84	150	300
NSC[19]	91.2	95.3	96.6
SVM	93.4	95.8	96.9
SRC[3]	95.5	96.9	97.7
CRC[14]	95.0	96.3	97.8
CROC[18]	95.5	97.1	98.2
ProCRC[20]	93.4	95.3	96.2
NRC[23]	96.7	97.2	**98.4**
CRNRC	**96.8**	**97.3**	**98.4**

Bold values signify the best classification result.

Fig. 4 Example images from the GT database.

4.1.3 *Experiments on the GT database*

The GT database[27] contains 750 face images of 50 individuals, and each of them has 15 images taken at resolution of 480×640 pixels. These images have variations in pose, expression, cluttered background, and illumination. Some example images are depicted in Fig. 4. Apart from the comparison methods used on the AR and Extended Yale B databases, here we compare CRNRC with several prevailing deep architectures, i.e., GoogLeNet,[34] Inception-v3,[35] Xception,[36] and Inception-ResNet-v2.[37] According to the experimental settings in Ref. 38, we employ the pretrained deep models to extract features; then these features are classified by the nearest-neighbors with a cosine metric. The layers of GoogLeNet, Inception-v3, Xception, and Inception-ResNet-v2 for feature extraction are loss3-classifier, avg_pool, avg_pool, and avg_pool, respectively. The first eight images per subject are used as training samples with the remaining as test samples. Experimental results are shown in Table 3, and the balancing parameter λ of CRNRC is set as 0.001. We can observe that our proposed CRNRC outperforms the conventional RBCM as well as the deep architectures. Compared with Inception-ResNet-v2, CRNRC achieves a modest 0.3% improvement.

4.2 *Handwritten Digit Classification*

4.2.1 *Experiments on the USPS dataset*

The USPS dataset[28] is composed of 9298 images for digit numbers of 10 classes, i.e., from 0 to 9. The training and test sets contain 7291 and 2007 images, respectively. All of the images are resized into 16×16 pixels. We randomly selected N ($N = 50$, 100, 200, and 300) images per class from the training set for training and all of the images in the test set for testing. Experiments are repeated 10 times, and the average results are recorded. Experimental results are shown in Table 4; the balancing parameter λ of CRNRC is set to 0.1 in all cases. We observe that CRNRC achieves the best recognition result under all scenarios. With the increase of the number of training images, recognition accuracy of all competing approaches improves steadily.

4.2.2 *Experiments on the MNIST dataset*

The MNIST dataset[29] has 60,000 training samples and 10,000 testing samples for digit numbers from 0 to 9. The original images are of size 28×28. We randomly selected N ($N = 50$, 100, 300,

Table 3 Recognition accuracy (%) of competing approaches on the GT database.

Methods	NSC[19]	SVM	SRC[3]	CRC[14]	CROC[18]	ProCRC[20]
Accuracy	90.2	89.7	90.5	91.0	92.0	91.7
Methods	NRC[23]	GoogLeNet[34]	Inception-v3[35]	Xception[36]	Inception-ResNet-v2[37]	CRNRC
Accuracy	92.0	90.8	91.7	90.3	92.0	**92.3**

Bold values signify the best classification result.

Table 4 Recognition accuracy (%) of competing approaches on the USPS database.

	N			
	50	100	200	300
NSC[19]	91.2	92.2	92.8	92.8
SVM	91.6	92.5	93.1	93.2
SRC[3]	89.1	91.2	92.9	93.8
CRC[14]	89.8	90.8	91.5	91.5
CROC[18]	91.9	91.3	91.7	91.8
ProCRC[20]	90.9	91.9	92.2	92.2
NRC[23]	90.3	91.6	92.7	93.0
CRNRC	**92.3**	**93.6**	**94.6**	**94.6**

Bold values signify the best classification result.

Table 5 Recognition accuracy (%) of competing approaches on the MNIST database.

	N			
	50	100	300	500
NSC[19]	**91.6**	**92.7**	84.8	71.3
SVM	86.6	88.5	90.8	91.1
SRC[3]	82.4	86.7	91.2	92.7
CRC[14]	86.3	88.2	89.2	89.4
CROC[18]	91.1	92.5	88.7	87.2
ProCRC[20]	86.6	89.5	92.5	93.5
NRC[23]	86.1	88.5	90.7	91.7
CRNRC	89.2	92.1	**94.4**	**95.1**

Bold values signify the best classification result.

and 500) samples per class from the training set for training and utilize all of the samples in the test set for testing. Experiments are repeated 10 times, and the average results are reported. Recognition accuracy of competing approaches is presented in Table 5; the balancing parameter λ of CRNRC is set to 0.1 in all cases. One can see that when the number of training images per class is 50 and 100, CRNRC underperforms NSC. Unfortunately, with increasing numbers of training images, the classification performance of NSC dramatically decreases. The reason for phenomenon is that NSC employs the class-specific training samples to represent the test sample, and the test sample can be well expressed by the training data when the number of training images per class is 50 and 100. Nevertheless, the training data matrix will be singular when the number of training images per class is 300 and 500, which will deteriorate the performance of NSC.

In addition to the recognition accuracy, here we present the $F1$ score of different methods on the MNIST dataset. The $F1$ score is the harmonic mean of precision and recall and, therefore, provides a single metric that summarizes the classifier performance in terms of both recall and precision. Table 6 summarizes the $F1$ score of competing approaches. With the increasing number of training samples per class, the $F1$ score of our proposed CRNRC is higher than that of the other approaches, which further demonstrates the effectiveness of our method.

Table 6 $F1$ score (%) of competing approaches on the MNIST database.

	N			
	50	100	300	500
NSC[19]	**91.5**	**92.7**	84.9	71.4
SVM	86.4	88.3	90.6	90.9
SRC[3]	82.0	86.5	91.1	92.6
CRC[14]	86.0	88.0	89.0	89.2
CROC[18]	91.0	92.4	88.5	87.1
ProCRC[20]	86.3	89.3	92.4	93.4
NRC[23]	85.8	88.2	90.6	91.5
CRNRC	89.0	92.0	**94.3**	**95.0**

Bold values signify the best classification result.

4.3 *Large-Scale Pattern Classification*

To fully evaluate the performance of CRNRC, in this section, we compare it with conventional RBCM and SOTA approaches on four large-scale datasets, i.e., the Stanford 40 Actions dataset,[30] the Oxford 102 Flowers dataset,[31] the Aircraft dataset,[32] and the Cars dataset.[33] First, we give a description of these datasets; then we evaluate our proposed CRNRC and its competing approaches on these datasets.

The Stanford 40 Actions dataset[30] is composed of 9532 images of humans performing 40 actions, such as reading a book, throwing a frisbee, and brushing teeth. Example images from this dataset are shown in Fig. 5(a). Each action class has 180 to 300 images. Following the common training–testing split settings presented in Ref. 23, 100 images per class are randomly chosen for training and the remaining for testing.

The Oxford 102 Flowers dataset[31] contains 8189 images from 102 flower classes; example images from this dataset are shown in Fig. 5(b). The flowers chosen are commonly occurring in the United Kingdom. Each class has 40 to 258 images, in which the images have large-scale, pose, and light variations.

The Aircraft dataset[32] includes 10,000 images of aircraft spanning 100 aircraft models. The models appear at different scales, design structures, and appearances, making this dataset challenging for the visual classification task. Example images from this dataset are shown in Fig. 5(c). We use the same experimental settings as in Ref. 23 to conduct our experiments.

The Cars dataset[33] has 16,185 images of 196 classes of cars; example images from this dataset are shown in Fig. 5(d). According to the standard split scheme, 8144 images are used as the training samples and the other 8041 images as the testing samples.

4.3.1 *Evaluation with deep features*

Following the experimental setting in Ref. 23, on the Stanford 40 Actions dataset and the Oxford 102 Flowers dataset, VGG-19[39] is employed to extract convolutional neural network (CNN) features (referred to as VGG19 features), and the final feature dimensionality of each image is 4096 for these two datasets. For the Aircraft and Cars datasets, a VGG-16 network[39] is used to extract features; then these deep features are fed into CRNRC and its competing approaches. Recognition accuracy of competing approaches on the above four datasets are presented in Table 7; the balancing parameter λ of CRNRC on the Stanford 14 dataset, Oxford 102 dataset, Aircraft dataset, and Cars dataset is set to 1×10^{-3}, 1×10^{-5}, 0.1, and 0.1, respectively. As seen from Table 7, CRNRC achieves the best result on the four datasets. Specifically, CRNRC makes an improvement of 0.2%, 0.4%, and 0.6% over NRC on the Stanford 40, Aircraft, and Cars datasets, respectively.

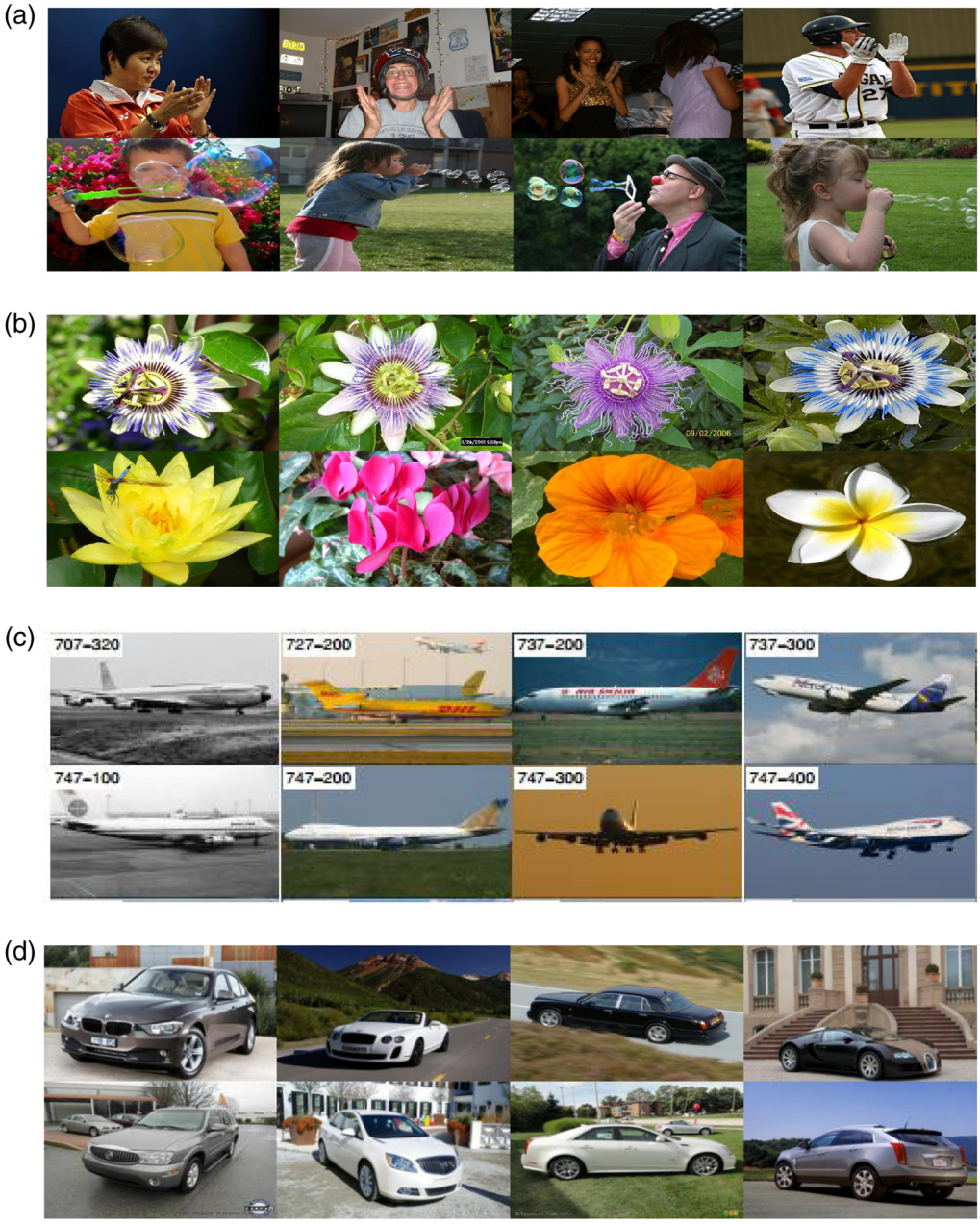

Fig. 5 Example images from four large-scale datasets. Example images from (a) the Stanford 40 dataset, (b) the Flower 102 dataset, (c) the Aircraft dataset, and (d) the Cars dataset.

4.3.2 *Comparison with SOTA methods*

In this section, we evaluate CRNRC with SOTA methods. As mentioned earlier, on the Stanford 40 Actions and Oxford 102 Flowers datasets, VGG19 features are used in NRC and CRNRC, while on the Aircraft and Cars datasets, VGG16 features are employed by NRC and CRNRC. It should be noted that the compared CNN-based approaches exploit more sophisticated network architectures or features than our utilized VGG19 features.

On the Stanford 40 Actions dataset, AlexNet network,[40] VGG19 network,[39] EPM,[41] and ASPD[42] are used for comparison. EPM and ASPD are two leading approaches on action recognition for still images. The classification accuracy is shown in Table 8. One can see that CRNRC achieves the highest classification accuracy and outperforms NRC by 0.2%.

On the Flower 102 dataset, AlexNet network,[40] VGG19 network,[39] GMP,[43] OverFeat,[44] and NAC[45] are employed for comparison. Classification results are presented in Table 9. We can

Table 7 Recognition accuracy (%) of competing approaches on the four fine-grained datasets.

Methods	Stanford 40	Flower 102	Aircraft	Cars
Softmax	77.2	87.3	85.6	88.7
NSC[19]	74.7	90.1	85.5	88.3
SRC[3]	78.7	93.2	86.1	89.2
CRC[14]	78.2	93.0	86.7	90.0
CROC[18]	79.2	93.1	86.9	90.3
ProCRC[20]	80.9	94.8	86.8	90.1
NRC[23]	81.1	**95.3**	87.3	90.7
CRNRC	**81.3**	**95.3**	**87.7**	**91.1**

Bold values signify the best classification result.

Table 8 Recognition accuracy (%) of competing approaches on the Stanford 40 actions dataset.

	Methods					
	AlexNet[40]	EPM[41]	ASPD[42]	VGG19[39]	NRC[23]	CRNRC
Accuracy	68.6	72.3	75.4	77.2	81.1	**81.3**

Bold values signify the best classification result.

Table 9 Recognition accuracy (%) of competing approaches on the Flower 102 dataset.

	Methods						
	GMP[43]	OverFeat[44]	AlexNet[40]	VGG19[39]	NAC[45]	NRC[23]	CRNRC
Accuracy	84.6	86.8	90.4	93.1	**95.3**	**95.3**	**95.3**

Bold values signify the best classification result.

Table 10 Recognition accuracy (%) of competing approaches on the Aircraft and Cars datasets.

Datasets	VGG16[39]	Symbiotic[46]	FV-FGC[47]	B-CNN[48]	NRC[23]	CRNRC
Aircraft	85.6	72.5	80.7	84.1	87.3	**87.7**
Cars	88.7	78.0	82.7	90.6	90.7	**91.1**

Bold values signify the best classification result.

observe that CRNRC has the same classification accuracy as NRC and NAC, and CRNRC achieves performance gains of 2.2% over VGG19.

For the Aircraft and Cars datasets, VGG16 network,[39] Symbiotic,[46] FV-FGC,[47] and B-CNN method[48] are used for comparison. Classification accuracy of competing approaches is summarized in Table 10. Once again, CRNRC exhibits the best recognition performance, and it makes an improvement of 0.4% and 0.4% over NRC on the Aircraft dataset and Cars dataset, respectively.

4.4 *Parameter Sensitiveness Analysis*

To examine how the balancing parameter λ influences the performance of CRNRC, we conduct experiments on the AR database. The experimental setting is the same as in Sec. 4.1.1 and the

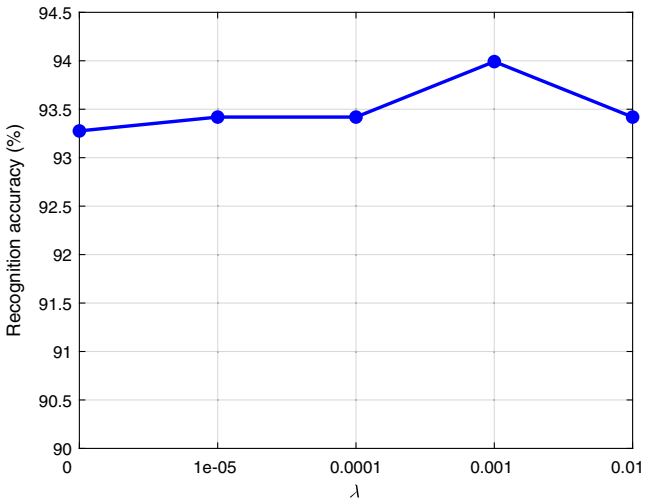

Fig. 6 Classification accuracy (%) of CRNRC with varying parameter λ on the AR database.

dimension of reduced samples is 300. Figure 6 shows the recognition accuracy with varying λ. We can see that in quite a wide range of λ, CRNRC performance is stable. Specifically, when λ increases from 0 to 0.001, the classification accuracy of CRNRC also increases steadily. When λ increases from 0.001 to 0.01, the performance of CRNRC drops a little. Larger values of λ mean that CRNRC will emphasize the class-specific residual constraint, which would undermine the collaborative mechanism of all training samples in representing the test sample. Therefore, we set a relatively small value for λ in our experiments.

5 Conclusions

In this paper, we presented a CRNR for pattern classification. Through this class-specific residual constraint, training samples from different classes are encouraged to competitively express the test sample. The proposed NRCR is solved by the ADMM technique, and each subproblem has a closed-form solution. Based on CRNR, we develop a CRNRC. Experimental results on face databases, handwritten digit datasets, and large-scale datasets validate that our proposed CRNRC outperforms NRC and several conventional RBCM, such as SRC, CRC, CROC, and ProCRC. In this paper, we did not explicitly consider the situation that both the training and test samples are contaminated due to occlusion or corruption; thus in the future, we will extend CRNRC to tackle the above scenarios.

Acknowledgments

This work was supported by the National Key Research and Development Program of China (Grant No. 2017YFC1601800), the National Natural Science Foundation of China (Grant Nos. 61672265 and U1836218), the 111 Project of Ministry of Education of China (Grant No. B12018), the Postgraduate Research and Practice Innovation Program of Jiangsu Province (Grant No. KYLX_1123), the Overseas Studies Program for Postgraduates of Jiangnan University, and the China Scholarship Council (Grant No. 201706790096).

References

1. X. Song et al., "Collaborative representation based face classification exploiting block weighted LBP and analysis dictionary learning," *Pattern Recognit.* **88**, 127–138 (2019).
2. S. Jia et al., "Collaborative representation-based multiscale superpixel fusion for hyperspectral image classification," *IEEE Trans. Geosci. Remote Sens.* **57**(10), 7770–7784 (2019).
3. J. Wright et al., "Robust face recognition via sparse representation," *IEEE Trans. Pattern Anal. Mach. Intell.* **31**(2), 210–227 (2009).

4. K. Yu, T. Zhang, and Y. Gong, "Nonlinear learning using local coordinate coding," in *Adv. Neural Inf. Process. Syst.*, pp. 2223–2231 (2009).
5. J. Wang et al., "Locality-constrained linear coding for image classification," in *IEEE Comput. Soc. Conf. Comput. Vision and Pattern Recognit.*, pp. 3360–3367 (2010).
6. C.-Y. Lu et al., "Face recognition via weighted sparse representation," *J. Visual Commun. Image Represent.* **24**(2), 111–116 (2013).
7. D. Zhong et al., "Loose L 1/2 regularised sparse representation for face recognition," *IET Comput. Vision* **9**(2), 251–258 (2015).
8. Z. Xu et al., "L 1/2 regularization: a thresholding representation theory and a fast solver," *IEEE Trans. Neural Networks Learn. Syst.* **23**(7), 1013–1027 (2012).
9. M. Liao and X. Gu, "Face recognition using feature coding and nonlocal constraint-based sparse representation classifier," *J. Electron. Imaging* **27**(5), 053050 (2018).
10. C.-G. Li, J. Guo, and H.-G. Zhang, "Local sparse representation based classification," in *20th Int. Conf. Pattern Recognit.*, IEEE, pp. 649–652 (2010).
11. E. G. Ortiz and B. C. Becker, "Face recognition for web-scale datasets," *Comput. Vision Image Understanding* **118**, 153–170 (2014).
12. M. Yang et al., "Robust sparse coding for face recognition," in *CVPR*, IEEE, pp. 625–632 (2011).
13. D. Zhong et al., "An improved robust sparse coding for face recognition with disguise," *Int. J. Adv. Rob. Syst.* **9**(4), 126 (2012).
14. L. Zhang, M. Yang, and X. Feng, "Sparse representation or collaborative representation: Which helps face recognition?" in *Int. Conf. Comput. Vision*, IEEE, pp. 471–478 (2011).
15. Y. Xu et al., "A new discriminative sparse representation method for robust face recognition via l_2 regularization," *IEEE Trans. Neural Networks Learn. Syst.* **28**(10), 2233–2242 (2017).
16. R. Timofte and L. Van Gool, "Weighted collaborative representation and classification of images," in *Proc. 21st Int. Conf. Pattern Recognit.*, IEEE, pp. 1606–1610 (2012).
17. Z.-Q. Li et al., "Sparsity augmented weighted collaborative representation for image classification," *J. Electron. Imaging* **28**(5), 053032 (2019).
18. Y. Chi and F. Porikli, "Classification and boosting with multiple collaborative representations," *IEEE Trans. Pattern Anal. Mach. Intell.* **36**(8), 1519–1531 (2014).
19. K.-C. Lee, J. Ho, and D. J. Kriegman, "Acquiring linear subspaces for face recognition under variable lighting," *IEEE Trans. Pattern Anal. Mach. Intell.* **27**(5), 684–698 (2005).
20. S. Cai et al., "A probabilistic collaborative representation based approach for pattern classification," in *Proc. IEEE Conf. Comput. Vision and Pattern Recognit.*, pp. 2950–2959 (2016).
21. J. Gou et al., "Two-phase probabilistic collaborative representation-based classification," *Expert Syst. Appl.* **133**, 9–20 (2019).
22. D. D. Lee and H. S. Seung, "Learning the parts of objects by non-negative matrix factorization," *Nature* **401**(6755), 788–791 (1999).
23. J. Xu et al., "Sparse, collaborative, or nonnegative representation: which helps pattern classification?" *Pattern Recognit.* **88**, 679–688 (2019).
24. S. Boyd et al., "Distributed optimization and statistical learning via the alternating direction method of multipliers," *Found. Trends Mach. Learn.* **3**(1), 1–122 (2011).
25. A. M. Martinez, "The AR face database," CVC Technical Report 24 (1998).
26. A. S. Georghiades, P. N. Belhumeur, and D. J. Kriegman, "From few to many: illumination cone models for face recognition under variable lighting and pose," *IEEE Trans. Pattern Anal. Mach. Intell.* **23**(6), 643–660 (2001).
27. Q. Feng et al., "Superimposed sparse parameter classifiers for face recognition," *IEEE Trans. Cybern.* **47**(2), 378–390 (2017).
28. J. J. Hull, "A database for handwritten text recognition research," *IEEE Trans. Pattern Anal. Mach. Intell.* **16**(5), 550–554 (1994).
29. Y. LeCun et al., "Gradient-based learning applied to document recognition," *Proc. IEEE* **86**(11), 2278–2324 (1998).
30. B. Yao et al., "Human action recognition by learning bases of action attributes and parts," in *Int. Conf. Comput. Vision*, IEEE, pp. 1331–1338 (2011).

31. M.-E. Nilsback and A. Zisserman, "Automated flower classification over a large number of classes," in *Sixth Indian Conf. Comput. Vision, Graphics and Image Process.*, IEEE, pp. 722–729 (2008).
32. S. Maji et al., "Understanding objects in detail with fine-grained attributes," in *Proc. IEEE Conf. Comput. Vision and Pattern Recognit.*, pp. 3622–3629 (2014).
33. J. Krause et al., "3D object representations for fine-grained categorization," in *Proc. IEEE Int. Conf. Comput. Vision Workshops*, pp. 554–561 (2013).
34. C. Szegedy et al., "Going deeper with convolutions," in *Proc. IEEE Conf. Comput. Vision and Pattern Recognit.*, pp. 1–9 (2015).
35. C. Szegedy et al., "Rethinking the inception architecture for computer vision," in *Proc. IEEE Conf. Comput. Vision and Pattern Recognit.*, pp. 2818–2826 (2016).
36. F. Chollet, "Xception: deep learning with depthwise separable convolutions," in *Proc. IEEE Conf. Comput. Vision and Pattern Recognit.*, pp. 1251–1258 (2017).
37. C. Szegedy et al., "Inception-v4, inception-ResNet and the impact of residual connections on learning," in *Thirty-First AAAI Conf. Artif. Intell.* (2017).
38. M. M. Ghazi and H. K. Ekenel, "A comprehensive analysis of deep learning based representation for face recognition," in *Proc. IEEE Conf. Comput. Vision and Pattern Recognit. Workshops*, pp. 34–41 (2016).
39. K. Simonyan and A. Zisserman, "Very deep convolutional networks for large-scale image recognition," in *Proc. Int. Conf. Learn. Represent.*, pp. 1–14 (2015).
40. A. Krizhevsky, I. Sutskever, and G. E. Hinton, "Imagenet classification with deep convolutional neural networks," in *Adv. Neural Inf. Process. Syst.*, pp. 1097–1105 (2012).
41. G. Sharma, F. Jurie, and C. Schmid, "Expanded parts model for human attribute and action recognition in still images," in *Proc. IEEE Conf. Comput. Vision and Pattern Recognit.*, pp. 652–659 (2013).
42. F. S. Khan et al., "Recognizing actions through action-specific person detection," *IEEE Trans. Image Process.* **24**(11), 4422–4432 (2015).
43. N. Murray and F. Perronnin, "Generalized max pooling," in *Proc. IEEE Conf. Comput. Vision and Pattern Recognit.*, pp. 2473–2480 (2014).
44. A. S. Razavian et al., "CNN features off-the-shelf: an astounding baseline for recognition," in *Proc. IEEE Conf. Comput. Vision and Pattern Recognit. Workshops*, pp. 806–813 (2014).
45. M. Simon and E. Rodner, "Neural activation constellations: unsupervised part model discovery with convolutional networks," in *Proc. IEEE Int. Conf. Comput. Vision*, pp. 1143–1151 (2015).
46. Y. Chai, V. Lempitsky, and A. Zisserman, "Symbiotic segmentation and part localization for fine-grained categorization," in *Proc. IEEE Int. Conf. Comput. Vision*, pp. 321–328 (2013).
47. P.-H. Gosselin et al., "Revisiting the Fisher vector for fine-grained classification," *Pattern Recognit. Lett.* **49**, 92–98 (2014).
48. T.-Y. Lin, A. RoyChowdhury, and S. Maji, "Bilinear CNN models for fine-grained visual recognition," in *Proc. IEEE Int. Conf. Comput. Vision*, pp. 1449–1457 (2015).

He-Feng Yin received his BS degree from the School of Computer Science and Technology from Xuchang University, Xuchang, China, in 2011. Currently, he is a PhD candidate in the School of IoT Engineering, Jiangnan University, Wuxi, China. He was a visiting PhD student at the Centre for Vision, Speech and Signal Processing (CVSSP), University of Surrey, under the supervision of Professor Josef Kittler. His research interests include representation-based classification methods, dictionary learning, and low-rank representation.

Xiao-Jun Wu received his BS degree in mathematics from Nanjing Normal University, Nanjing, China, in 1991. He received his MS degree in 1996 and his PhD in pattern recognition and intelligent systems in 2002, both from Nanjing University of Science and Technology, Nanjing, China. He joined Jiangnan University in 2006, where he is currently a professor. He has published more than 300 papers in his fields of research. He was a visiting researcher at the CVSSP, University of Surrey, United Kingdom, from 2003 to 2004. His current research interests include pattern recognition, computer vision, fuzzy systems, neural networks, and intelligent systems.