This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/ACCESS.2020.3041154, IEEE Access

IEEE Access
Multidisciplinary | Rapid Review | Open Access Journal

Date of publication xxxx 00, 0000, date of current version xxxx 00, 0000.

Digital Object Identifier 10.1109/ACCESS.2017.DOI

Linear Representation-based Methods for Image Classification: A Survey

JIANHANG ZHOU[1], (Student Member, IEEE), SHAONING ZENG[1], (MEMBER, IEEE), AND BOB ZHANG[1], (Senior Member, IEEE)

[1]PAMI Research Group, Department of Computer and Information Science, Faculty of Science and Technology,University of Macau, Taipa, Macau (e-mail:yc07424@um.edu.mo, zns@outlook.com)

Corresponding author: Bob Zhang (e-mail: bobzhang@um.edu.mo).

ABSTRACT In recent years, linear representation-based methods have been widely researched and applied in the image classification field. Generally speaking, there are three steps within linear representation-based classification (LRC) algorithms. The first step is coding, which uses all training samples to represent the test sample in a linear combination. The second step is subspace approximation, where residuals between the test sample and the linear combination of each class are calculated. The third step is classification, which assigns the class label to the minimum class-specific residual. We classify the LRC methods into six categories: 1) linear representation-based classification methods with norm minimizations, 2) linear representation-based classification methods with constraints, 3) linear representation-based classification methods with feature spaces, 4) linear representation-based classification methods with structural information, 5) linear representation with subspace learning, and 6) linear representation in semi-supervised learning and unsupervised learning. The purpose of this paper is to: 1) make an accurate and clear definition of the linear representation-based method, 2) provide a categorization and a comprehensive survey of the existing linear representation-based classification methods for image classification, 3) Summarize the main applications of linear representation-based methods, 4) provide extensive classification results and a discussion of the linear representation-based methods. Furthermore, this paper summarizes specific applications of the linear representation-based methods. Particularly, we performed extensive experiments to compare twelve linear representation-based classification methods on seven image classification datasets.

INDEX TERMS Image Classification, Linear Representation, Optimization.

I. INTRODUCTION

IMAGE classification is a hot topic that has been extensively studied in recent years with the increasingly active developments of computer vision and pattern recognition. The problem of classification is identifying the category a instance belongs to, based on the given observations (training data), and category membership [1]. Visual applications like remote sensing [2], face recognition [3], [4], object recognition [5], biometrics [6]–[8] widely use the models and algorithms of image classification. The linear representation-based classification method is an active research area in the image classification field [9]–[11]. Up to now, extensive LRC methods have been proposed and developed for better and more robust image classification, such as Sparse representation-based classifier (SRC) [9], Collaborative representation-based classifier (CRC) [10], Non-negative representation-based classifier (NRC) [12], and so on. To the best of our knowledge, there is no related literature that provides a clear definition of the linear representation-based method. Accordingly, there is no article to comprehensively survey this group of methods. In this paper, we first make an accurate and clear definition of a linear representation-based method to establish the concept. Then, based on the definition, we proposed six categories to summarize various linear representation-based method into different perspectives to present both an overview and detailed interpretation. Afterwards, the main applications that widely apply linear representation-based methods were established. Lastly, we provided extensive experimental results and discussion.

The linear representation-based classification (LRC) method has a high correlation with the nearest subspace classification (NSC) [13] method by assigning the class associated with the optimal class subspace to the test sample. Like the NSC algorithm, in the LRC algorithm, all training

This work is licensed under a Creative Commons Attribution 4.0 License. For more information, see https://creativecommons.org/licenses/by/4.0/

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/ACCESS.2020.3041154, IEEE Access

samples represent the test sample in a linear combination. Therefore, the LRC performs classification in the linear system in order to organize the subspace for classification. Unlike NSC, LRC is composed of three main components: the reconstruction term, the regularization term, and the constraints. The reconstruction term ensures the training samples' linear combination will be as close as possible to the test sample. The regularization term imposes different assumptions on the coefficient in a linear combination to make a robust representation that overcomes variations existing within the data. For example, SRC for face recognition [14] utilizes the l_1 minimization to make the coefficient vector sparse by assuming the test sample can be represented by samples from the same class only. To reach the same purpose, TPTSR [15] perform the l_2 minimization twice to achieve the sparse coefficient vector. The CRC [10] argued that the collaboration representation organized by all training samples can perform robust classification. The NRC [12] considers the non-negative constraints on the coefficients will enhance the representation ability of the linear combination. Moreover, with different regularizations, LRC methods can be interpreted in different ways. For instance, both SRC and CRC have a geometric interpretation to prove the rationality of the proposed model and assumptions. The ProCRC [16] utilizes the theory of probability to interpret the newly-added term of enforcing the representation to be as close as possible to the class-specific subspaces. Generally speaking, there are three procedures in the LRC algorithm: coding, subspace approximation, and classification. The first procedure is coding, whereby we calculate the coefficients of the linear combination with the test sample and training samples. Next, the second procedure, called subspace approximation, obtains the subspace representation of each class using the already-calculated coefficients. The last procedure is classification, which computes the residuals between the test sample and each class before assigning the class label associated with the minimum residual to the test sample.

There are six categories of linear representation-based classifiers: 1) norm minimization, 2) constraints, 3) feature spaces, 4) structural information, 5) subspace learning, and 6) semi-supervised learning and unsupervised learning. This categorization covers all components and procedures involved in the linear representation-based method. The relationship between these six categories and the linear representation-based method are shown in Figure 2. As clearly shown in Figure 2, each category (Norm minimization, constraints, feature space, structural information, subspace learning, and semi-supervised/unsupervised learning) corresponds to a component/procedure of the linear representation-based method (more details about this figure can be found in section III). For the norm minimization, there are many LRC algorithms and its corresponding fusion extensions. The SRC [14] and CRC [10] are representative methods that use l_1 and l_2 minimization according to its assumptions of the data. To fully extract the properties of the data, many constraints were imposed on the basic LRC model to achieve a better recognition rate. For example, the sparsity [17], [18] and locality [19]–[21] are extracted to better represent the test sample in the linear representation. The non-negative constraint [12], [22] believes the non-negativity makes the representation more focused on the heterogeneous data and easier to interpret. Several works are devoted to creating an optimal projection or creating an ideal feature subspace for image classification [23]–[25]. In the new feature space, the data will be more discriminative than the original space. There are currently three methods to create a new feature space, the kernel-based representation [26], the mapping method, and the use of deep features. The structural information within the dataset is critical for image classification [27], [28]. The low-rank representation [27] attempts to recover the subspace structure by minimizing the rank of the coefficient matrix for robust image classification. The convolutional SRC [29], [30] used the dictionary filter to make convolutional operations on the coefficients, which involves the information of the entire image. Some LRC-based methods consider the neighbor information [31], [32] in the decision-making procedure. Besides fully supervised learning, there are existing methods proposed for semi-supervised learning and unsupervised learning [33]–[36], showing that the LRC-based method can be applied widely in the image classification domain.

The linear representation-based methods for image classification has numerous applications. Hyperspectral images are frequently used in the remote sensing scenario, which contains a wide range of electromagnetic spectrum information. The LRC algorithms are utilized to exploit the shared information among the spectral signals inside the image. Since the pixels in the hyperspectral image lie in the high dimensional space, it is helpful if an approximation can be provided from the low-dimensional subspace of the sample class [37]. The LRC methods are also widely applied in the tasks of medical biometrics and multimodal biometrics. For example, the ProCRC was applied for detecting diabetes mullitus and achieved a promising accuracy [38]. Group SRC was used to fuse the multimodal biometrics data. The other popular application is face recognition. There are many LRC methods specifically designed for face recognition [39]–[41]. For instance, the conventional SRC [14] was originally proposed for robust face recognition.

To this end, in this paper we conducted a survey on the linear representation-based methods for image classification. This paper makes contributions and points of inspiration for the readers in the following areas:

1) **Definition.** We present an accurate definition of the linear representation-based method for image classification by reviewing influential works in this research area. With this understanding, the readers are able to not confuse this kind of method with other methods easily.
2) **Categorization.** We categorized the LRC methods for image classification into six classes and discuss the different methods proposed within them. Readers can

This work is licensed under a Creative Commons Attribution 4.0 License. For more information, see https://creativecommons.org/licenses/by/4.0/

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/ACCESS.2020.3041154, IEEE Access

refer to the corresponding category for detail if they have interest in a certain area.

3) **Application.** We reviewed the applications of LRC methods in the four most frequently used areas: remote sensing, face recognition, medical biometrics, and multimodal biometrics. This allows readers to learn from the LRC methods in a specific application scenario. It is also beneficial to the readers who are seeking a solution in a related application.
4) **Experiments and discussion.** We performed extensive experiments on seven datasets to show the performances of the different LRC methods. Based on the experimental results, we provide a discussion of the LRC methods and point out the challenges and possible points of interest in linear representation for image classification. Readers can have an insight into the classification ability, properties, and the potential future directions of the linear representation-based methods.

The overview of the organization of this paper is shown in Figure 1. The remainder of this paper is organized as follows: In section II, we introduce the notations used in this paper, the preliminary background knowledge of the linear representation-based method, and some basic definitions of the linear representation-based method. In section III, six groups of linear representation-based classifiers are presented and discussed in detail. Different types of applications using linear representation-based methods will be reviewed in section IV. We performed extensive comparison experiments on seven image datasets and showed the results in section V. Finally, a discussion will be presented, and we will reach a conclusion to summarize this paper.

II. BACKGROUND AND DEFINITIONS

Throughout this section, we will first summarize the notations that are used in the later part of this paper. Afterwards, we will briefly talk about nearest subspace classification [13] and make a comparison with LRC since the two methods are highly correlated. Finally, an accurate definition of linear representation-based classification will be provided and discussed.

A. NOTATIONS

In this paper, we denote $y \in \mathbb{R}^m$ as the test sample which needs to be assigned a label. $X \in \mathbb{R}^{m\times n}$ represents n training samples, and $X_i \in \mathbb{R}^{m\times r}(r < n)$ means the training samples from the i^{th} class. The $\alpha \in \mathbb{R}^n$ is the coefficient vector of the linear combination in the LRC method obtained in the coding procedure. The α is the coefficient vector in the linear representation, and the α_i is the vector only contains the coefficients from the i^{th} class . $\lambda \in \mathbb{R}$ is the scaling factor. $I \in \mathbb{Z}^+$ is the output label of LRC method. The notations used in this paper are summarized in Table 1. k represents the number of classes in the dataset. H represents the number of training samples in the task. $\odot$ is the operator performing element-wise product between two vectors.

TABLE 1: Units for Magnetic Properties

Notation	Description
y	The test sample
X	The training samples
X_i	The training samples from class i
α	The coefficient of linear combination in LRC method
α_i	The coefficient of linear combination in LRC method from class i
λ	The scaling factor
I	The output label of LRC method
k	The number of classes
H	The number of training samples
$\odot$	The element-wise product operator

B. NEAREST SUBSPACE CLASSIFICATION

In the nearest subspace classifier [13], [42], there is an assumption proposed that is the foundation of classification. The assumption is that samples from the same class lie in the same subspace. We can describe the assumption as follows:

Assumption 1. *Given a collection of images* $C = \{C_1, C_2, C_3, ..., C_l\} \in \mathbb{R}^{m\times l}$*, the samples from the sample class* $X_i = \{I_p, I_{p+1}, ..., I_q\}$ *lies on the same linear subspace* $S_{X_i} \in \mathbb{R}^{k\times l_{X_i}}$.

According to the Assumption 1, the primary classification idea of NSC is to assign the class label associated with the closest class-specific subspace to the test sample, which is shown in the following formulation:

$$I = argmin_i dist(y, X_i\alpha_i) \tag{1}$$

The metric is usually l_2 norm distance:

$$I = argmin_i \|y - X_i\alpha_i\|_2^2 \tag{2}$$

C. LINEAR REPRESENTATION-BASED CLASSIFICATION

The linear representation-based classification method [10], [12], [14] inherits the assumption and theorem of NSC. However, compared with NSC, LRC emphasizes the characteristics of the coefficients in the linear combination, expecting to perform the robust image classification. Besides this, the LRC believes that each sample in the dataset can be represented in a linear combination composed of other samples, which can be described in the following Theorem 1:

Theorem 1. *Given a collection of images* $C = \{C_1, C_2, C_3, ..., C_l\} \in \mathbb{R}^{m\times l}$*, a sample* C_i *in* C *can be represented in a linear combination of other samples* $C_j \in \{C_j \subseteq C | j \neq i\}$.

Based on the Theorem 1, in the image collection C, a test sample y can be represented as follows:

$$y = x_1\alpha_1 + x_2\alpha_2 + ... + x_l\alpha_l \tag{3}$$

The above Eq. 3 can be rewritten as:

$$y = X\alpha \tag{4}$$

This work is licensed under a Creative Commons Attribution 4.0 License. For more information, see https://creativecommons.org/licenses/by/4.0/

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/ACCESS.2020.3041154, IEEE Access

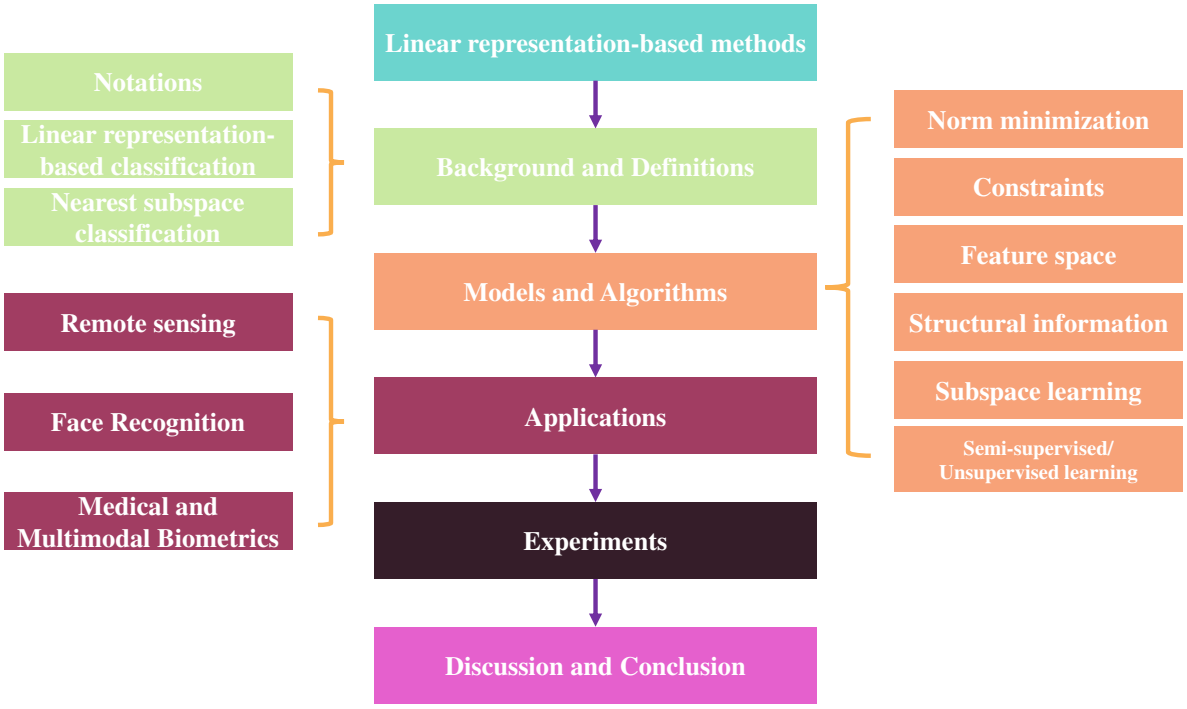


FIGURE 1: Overview of the organization of this survey.

Basically, the LRC method is to solve the following minimization problem:

$$\begin{aligned} &minimize \left\|y - X\alpha\right\|_2 + \lambda \left\|\alpha\right\|_p \\ &s.t. \begin{cases} constraint_1, \\ \vdots \\ constraint_n, \end{cases} \end{aligned} \tag{5}$$

where p represents the minimization norm depends on specific method (e.g., when $p = 1$, it becomes the sparse representation-based classification model [14]). λ is the regularization parameter. The first constraint represents a linear combination. Other constraints below are specified by different methods individually according to their assumption.

Generally speaking, the LRC method comprises of three procedures: coding, subspace approximation, and classification. The coding procedure seeks the optimal coefficients of the linear combination under different regularizations and constraints.

In the first step, The coding procedure can be considered to solve the following general problem:

$$\begin{aligned} \alpha = argmin_\alpha \left\|y - X\alpha\right\|_2 + \lambda \left\|\alpha\right\|_p \\ s.t.\ constraint_1, ... \end{aligned} \tag{6}$$

After the coefficient vector α is obtained using the above Eq.6, the second step is subspace approximation, which first organizes all class-specific subspaces then calculates the residuals between the test sample and all class-specific subspaces:

$$S = \{S_i | i = 2, ..., k\} = \{X_1\alpha_1, X_2\alpha_2, ..., X_k\alpha_k\} \tag{7}$$

$$R_i = \left\|y - S_i\alpha_i\right\|_2 \tag{8}$$

Finally, in the classification procedure, the class label with minimum residual is assigned to the test sample:

$$I = argmin_i\{R_i\} \tag{9}$$

As previously shown in section I, although there are plenty of algorithms based on LRC method, no uniform definition of LRC method is made, here, we made a definition as follows:

Definition 1. Linear representation-based classification: A method that classify a given test sample using all training samples by following three steps:

1) Coding using Eq.6.
2) Subspace approximation using Eq.7 and Eq.8.
3) Classification using Eq.9.

We summarized the LRC in the following Algorithm 1.

III. MODELS AND ALGORITHMS

Based on the basic LRC method mentioned in section II, various extended classifiers are proposed. In this section, we categorize them into six directions and separately introduce

This work is licensed under a Creative Commons Attribution 4.0 License. For more information, see https://creativecommons.org/licenses/by/4.0/

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/ACCESS.2020.3041154, IEEE Access

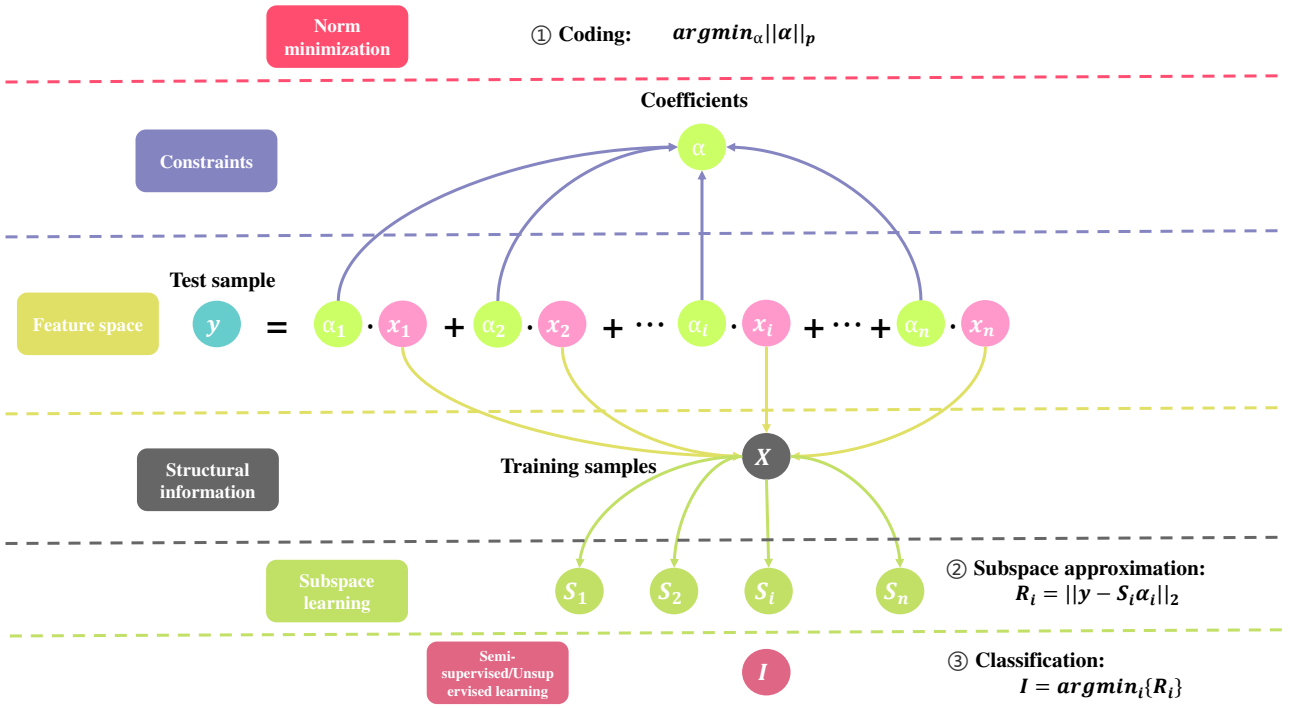


FIGURE 2: Relationships between six categories and the linear representation-based method.

Algorithm 1 Linear representation-based classification

Require: Test sample y, Training set X, Number of class k, Training Label set X^L.
Coding:
Solve the problem in Eq.6 to get α;
Subspace approximation:
Construct the subspace set S using Eq.7;
Calculate the residuals R_i using Eq.8;
Classification:
Assign the class label I to sample y using Eq.9;
return class label I;

them in detail. Figure 2 illustrates the relationship between the six categories and the linear representation-based method: the norm-minimization category focuses on the coding procedure; the constraints category concentrates on the property of the coefficients; the feature space category contains methods that perform linear representation in different feature spaces; the structural information category exploits the structural information in the training samples; the subspace learning category devotes to learning the best subspace for the subspace approximation; the semi-supervise learning and unsupervised learning category consists of methods that perform classification in a semi-supervised or unsupervised way.

A. LINEAR REPRESENTATION-BASED CLASSIFICATION WITH NORM MINIMIZATION

1) l_1 minimization

By using l_1 minimization in Eq.5, the coefficient of linear combination will be "sparse". The term "sparse" in the linear representation means a majority of elements in the coefficient vector are zero [43]. Therefore, the linear representation-based classification with l_1 minimization is called "Sparse Representation". Wright et al. proposed the sparse representation-based classification (SRC) method [14] to perform robust face recognition. Based on Assumption 1, the SRC intends to represent the test sample by utilizing only the training sample from the same class as the test sample. In order to make the coefficient sparse, the coding procedure can be formulated as a l_0 minimization problem:

$$\begin{aligned} \alpha &= argmin \|\alpha\|_0 \\ s.t.\ y &= X\alpha \end{aligned} \tag{10}$$

$$\alpha = argmin_\alpha \frac{1}{2}\|y - X\alpha\|_2^2 + \lambda\|\alpha\|_0 \tag{11}$$

However, in the above Eq.10, the l_0 is a NP-hard problem, which means there is no existing algorithm to solve this. Therefore, when the coefficient α is sparse enough, the Eq. 10 can be replaced by l_1 minimization:

$$\begin{aligned} \alpha &= argmin \|\alpha\|_1 \\ s.t.\ y &= X\alpha \end{aligned} \tag{12}$$

$$\alpha = argmin_\alpha \frac{1}{2}\|y - X\alpha\|_2^2 + \lambda\|\alpha\|_1 \tag{13}$$

This work is licensed under a Creative Commons Attribution 4.0 License. For more information, see https://creativecommons.org/licenses/by/4.0/

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/ACCESS.2020.3041154, IEEE Access

There are four groups optimization method to solve the problem showing in Eq. 12: 1) greedy strategy approximation, 2) constrained optimization, 3) proximity algorithm-based optimization, and 4) homotopy algorithm-based sparse representation [44]. Here we introduce two most frequently used algorithms to solve the l_1 minimization problem in Eq. 10: The orthogonal matching pursuit (OMP) [45] algorithm and fast iterative shrinkage thresholding (FISTA) [46] algorithm.

The OMP algorithm belongs to the greedy strategy approximation method group, which achieves the local minima in each step and obtains the global minima in the final step. The core idea of OMP is to select the most contributing training sample each time to approximate the test sample. The algorithm will stop when the reconstruction representation $\Lambda\alpha$ is close enough to the test sample. In each iteration, the contribution of the i^{th} training sample x_i is evaluated by the inner product between the training sample x_i and the residual vector r_{d-1} of the last step:

$$\omega_d = argmax_{i \notin \Omega_{d-1}} | \langle x_i, r_{d-1} \rangle | \tag{14}$$

The residual vector of the last step r_{d-1} is orthogonal with the selected samples in reconstruction matrix Λ_{d-1}. Then, the index of most contributing sample will be added to the index set Ω and a reconstruction matrix Λ

$$\Omega_d = \Omega_{d-1} \cup \omega_d \tag{15}$$

$$\Lambda_d = \Lambda_{d-1} \cup x_{\omega_d} \tag{16}$$

Next, the coefficient vector α is updated according to the reconstruction matrix Λ:

$$\alpha = \|y - \Lambda_d \alpha\|_2^2 \tag{17}$$

As the final step of an iteration, the residual vector is calculated:

$$r_d = y - \Lambda_d \alpha \tag{18}$$

To control the algorithm to stop when the reconstruction representation is close enough to the test sample, a criteria is set:

$$|r_d| > \gamma \tag{19}$$

The fast iterative shrinkage thresholding (FISTA) algorithm utilizes the proximity algorithm to solve the Eq.13. Here we reformulate the Eq.13 as follows:

$$\alpha = argmin_\alpha u(\alpha) + v(\alpha), \tag{20}$$

where $u(\alpha) = \frac{1}{2}\|X\alpha - y\|_2^2$ and $v(\alpha) = \|\alpha\|_1$.

In each iteration of FISTA algorithm, the update formulation is shown as follows:

$$\begin{aligned} \alpha_d = argmin_\alpha \{ & u(\alpha_d) + \langle \alpha_d - \alpha_{d-1}, \bigtriangledown u(\alpha) \rangle \\ & + \frac{L}{2}\|\alpha_d - \alpha_{d-1}\|_2^2 + v(\alpha) \}, \end{aligned} \tag{21}$$

$$t_{k+1} = \frac{1 + \sqrt{1 + 4t_k^2}}{2}, \tag{22}$$

$$\alpha_{d-1} = \alpha_d + (\frac{t_k - 1}{t_{k+1}})(\alpha_d - \alpha_{d-1}), \tag{23}$$

where L is the Lipschitz constant [47].

There were weighted LRC methods with l_1 minimization proposed to enhance the stability of the original classifier [48], [49]. The weight can be imposed on two places in LRC: the training samples and the coefficients. To impose weight on the training samples, the Eq.13 can be extended as follows:

$$\alpha = argmin_\alpha \frac{1}{2}\|y - XW\alpha\|_2^2 + \lambda \|\alpha\|_1 \tag{24}$$

In [49], Fan et al. evaluate the weights of training samples based on the Gaussian kernel distances between test sample and training samples $dist(x_i, y) = exp(-\|x - y\|^2 / 2\sigma^2)$. Therefore, the weighted matrix $diag(W) = [dist(x_1, y), [dist(x_2, y), ..., [dist(x_n, y)]$. The Gaussian kernel distance can not only measure the similarity of samples, but capture the nonlinear information in the dataset. The other way of weighting is imposing weights on the coefficients, which can be described as the extended Eq. 12:

$$\begin{aligned} & \alpha = argmin \left\|W\alpha^T\right\|_1 \\ & s.t.\ y = X\alpha \end{aligned} \tag{25}$$

In [48], the Euclid distance between test sample and each training sample $dist(x_i, y) = \|y - x_i\|_2$ is applied to be the weight of each training sample. The weighted matrix $diag(W) = [dist(x_1, y), [dist(x_2, y), ..., [dist(x_n, y)]$. This weighting strategy considers the similarity between test sample and its neighbor into account when represent the test sample.

The SRC (Linear representation-based classification with l_1 minimization) algorithm is summarized in the following Algorithm 2.

Algorithm 2 Sparse representation-based classification

Require: Test sample y, Training set X, Number of class k, Training Label set X^L.

Coding:

Solve the l_1-minimization problem to get coefficient vector α:

$$\alpha = argmin_\alpha \|\alpha\|_1 \quad s.t. \quad y = X\alpha$$

Subspace approximation:

Construct the subspace set S using Eq.7;

Calculate the residuals R_i using Eq.8;

Classification:

Assign the class label I to sample y using Eq.9;

return class label I;

This work is licensed under a Creative Commons Attribution 4.0 License. For more information, see https://creativecommons.org/licenses/by/4.0/

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/ACCESS.2020.3041154, IEEE Access

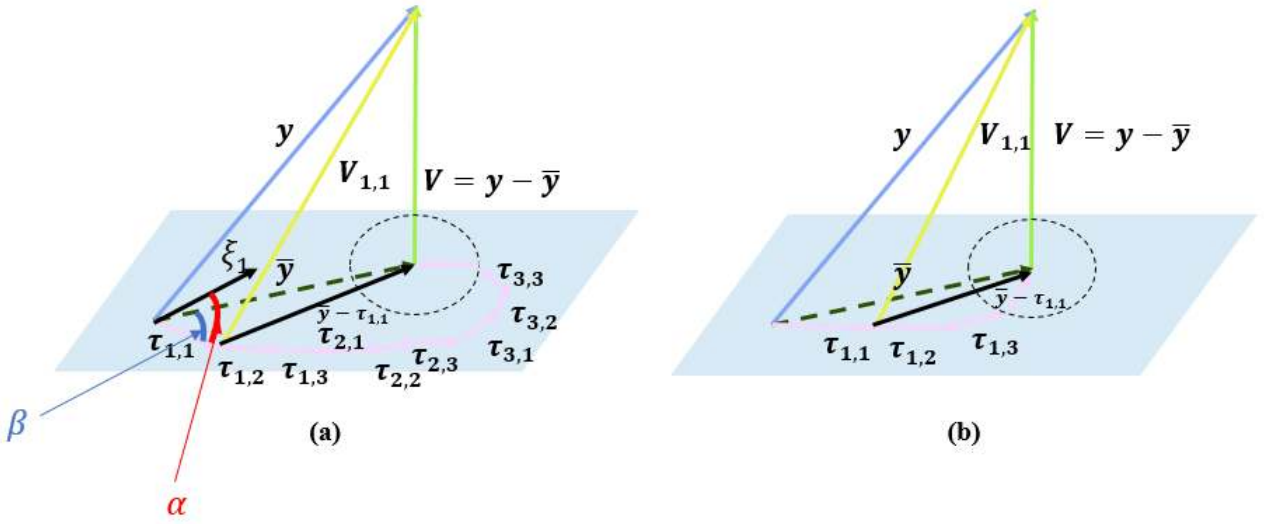


FIGURE 3: The geometric interpretation of CRC and SRC. (a) CRC, (b) SRC. One blue plate is the space spanned by training set X, $\tau_{i,j}$ is the j^{th} representation component from the i^{th} class on the space, $\bar{y}$ is the projection of y on the space. $V_{i,j}$ is the residual between $\bar{y}$ and $\tau_{i,j}$. V is the residual between the $\bar{y}$ and the representation organized by $\tau_{i,j}$. $\xi_{i,j} = \sum_{i,j \neq 1} \tau_{i,j}$. α is the angle between ξ and $\tau_{1,1}$. β is the angle between $\bar{y}$ and $\tau_{1,1}$.

2) l_2 minimization

The linear representation-based classification with l_2 minimization uses all training samples collaboratively to represent the test sample. The problem is formulated as follows:

$$\begin{aligned} &\alpha = argmin \left\| \alpha \right\|_2 \\ &s.t.\ y = X\alpha \end{aligned} \tag{26}$$

$$\alpha = argmin_\alpha \frac{1}{2} \left\| y - X\alpha \right\|_2^2 + \lambda \left\| \alpha \right\|_2 \tag{27}$$

Unlike l_1 minimization in SRC, the l_2 minimization makes the coefficient "dense" which means the majority of elements in the coefficient vector are non-zero. This type of representation makes all training samples from the dataset to collaborate to represent the test sample, rather than only using the training samples from the same class as the test sample. Zhang et al. used the l_2 minimization based on the LRC method to perform image classification, and called it collaborative representation-based classification (CRC) [10] method. CRC believes that the samples from different classes share similarities so that all training samples will better represent the test sample. The CRC is able to achieve competitive classification result compared with SRC, and moreover, it costs less computation time than SRC since there exists a closed-form solution to solve the l_2 minimization:

$$\alpha = (X^T X - \lambda I)^{-1} X y \tag{28}$$

The CRC has an elegant geometric interpretation to support its rationality, which is shown in the following Figure 3 (a).We can observe that the CRC involves all training class samples to represent the test sample, and that SRC only utilizes the samples from a single class. We take the representation $\tau_{1,1}$ as the example to explain the robustness brought by the CRC. Since the representation $\xi_{1,1}$ is parallel to $\bar{y} - \tau_{1,1}$, according to the law of sines, we have the following Eq.

$$\frac{\left\| \bar{y} \right\|_2}{sin(\alpha)} = \frac{\left\| \bar{y} - \tau_1 \right\|_2}{sin(\beta)} \tag{29}$$

The $\left\| \bar{y} - \tau_1 \right\|_2$ is the residual e_1 between test sample and the representation τ_1, therefore, the Eq.29 can be rewritten as:

$$e_1 = \frac{\left\| \bar{y} \right\|_2 sin(\beta)}{sin(\alpha)} \tag{30}$$

According to Eq.30, the CRC not only enforce the test sample to be as close as possible to the representation τ_1, but also consider make $\tau_{1,1}$ be as far as possible to the representation composed of other samples $\xi_1 = \sum_{i,j \neq 1} \tau_{i,j}$. This is called "double check" mechanism [10] ensuring the robust classification.

The weighted version of LRC with l_2 minimization (termed as weighted CRC) is proposed in [50] to improve the classification performance. The formulation of weighted CRC is shown in the following equation:

$$\alpha = argmin_\alpha \frac{1}{2} (y - X\alpha) \Theta^{-1} (y - X\alpha) + \lambda \left\| \Upsilon \alpha \right\|^2, \tag{31}$$

where the Θ is a diagonal matrix whose non-zero elements are estimated by the squared residuals between the test sample and representation obtained from the train samples. Υ is the Tikhonov matrix which is usually $\Upsilon = \varepsilon I$ to avoid the ill-posed problems. In [51], the adaptive WCRC (AWCRC) was proposed, where Υ is replaced

This work is licensed under a Creative Commons Attribution 4.0 License. For more information, see https://creativecommons.org/licenses/by/4.0/

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/ACCESS.2020.3041154, IEEE Access

by a diagonal matrix whose non-zero elements are distance between training samples and the test sample $\Upsilon = diag(\|x_1 - y\|, \|x_2 - y\|, ..., \|x_H - y\|)$.

In [16], Cai et al. proposed the Probabilistic collaborative representation-based classification (ProCRC) by maximizing the test sample's class-specific likelihood. It assumes the coefficient vector in the linear representation determines the confidence of a test sample belongs to each class. The probability of a test sample belongs to a specific class can be described as follows:

$$\begin{aligned} &P(\delta(y) = k) \\ &= P(\delta(y) \in \delta_X) \cdot P(\delta(x) = k | \delta(x) \in \delta_X) \\ &\propto exp(-(\|y - X\alpha\|_2^2 + \lambda \|\alpha\|_2^2 + \eta \|X\alpha - X_k\alpha_k\|_2^2)), \end{aligned} \tag{32}$$

where δ_X is the label set in the class of training set X, $\delta(y)$ is the label of test sample y. The ProCRC tries to construct a probabilistic collaborative subspace that ensure the class-specific likelihood is maximum. Therefore, the objective function of ProCRC is maximizing the joint probability of the test sample:

$$\begin{aligned} &maxP(\delta(y) = 1, ..., \delta(y) = k) \\ &= maxexp(-(\|y - X\alpha\|_2^2 + \lambda \|\alpha\|_2^2 + \frac{\eta}{k} \sum_{i=1}^{k} \|X\alpha - X_i\alpha_i\|_2^2)) \end{aligned} \tag{33}$$

Based on the Eq. 33, we can obtain the coefficient by solving the following optimization problem:

$$\alpha = argmin_\alpha(\|y - X\alpha\|_2^2 + \lambda \|\alpha\|_2^2 + \frac{\eta}{k} \sum_{i=1}^{k} \|X\alpha - X_i\alpha_i\|_2^2), \tag{34}$$

where the term $\|X\alpha - X_i\alpha_i\|_2^2$ enforces the representation to be as closed as possible to the class-specific collaborative subspace. The Eq. 34 has closed-form solution [16].

The collaborative representation-based classification method and probabilistic collaborative representation-based classification method are summarized in the Algorithm 3 and Algorithm 4.

3) $l_{2,1}$ minimization

The LRC with $l_{2,1}$minimization assumes that the sparsity should be imposed on the training samples from the incorrect class instead of an individual sample. In other words, only the training samples from the correct class will have non-zero coefficients in the linear representation. The basic formulation of LRC with $l_{2,1}$ minimization is showing as follows:

$$argmin_\alpha \|y - X\alpha\|_2 + \lambda \|\alpha\|_{2,1} \tag{35}$$

This LRC with $l_{2,1}$ minimization is called group sparse classification (GSC) [52]. In the GSC, the group sparsity is imposed on the training samples. The illustration of group sparsity and sparsity is shown in the following Figure 4. The problem in the above Eq. 35 can be solved by SPGL1 algorithm [53].

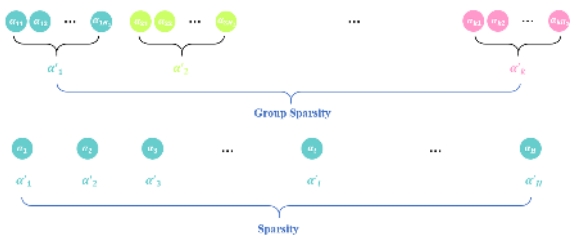


FIGURE 4: The illustration of group sparsity and sparsity.

Algorithm 3 Collaborative representation-based classification

Require: Test sample y, Training set X, Number of class k, Training Label set X^L.

Coding:

Solve the l_2-minimization problem to get coefficient vector α:

$$\alpha = (X^T X + \lambda I)^{-1} y$$

Subspace approximation:

Construct the subspace set S using Eq.7;

Calculate the residuals R_i using Eq.8;

Classification:

Assign the class label I to sample y using Eq.9;

return class label I;

Algorithm 4 Probabilistic Collaborative representation-based classification.

Require: Test sample y, Training set X, Number of class k, Training Label set X^L.

Coding:

Solve the problem described in Eq. 34 to get coefficient vector α:

$$\alpha = (X^T X + \frac{\eta}{k} \sum_{i=1}^{k} (X - X_k)^T (X - X_k) + \lambda I)^{-1} X^T y$$

Subspace approximation:

Construct the subspace set S using Eq.7;

Calculate the residuals R_i using Eq.8;

Classification:

Assign the class label I to sample y using Eq.9;

return class label I;

B. LINEAR REPRESENTATION-BASED CLASSIFICATION WITH CONSTRAINTS

The constraints imposed in the linear representation will enforce the coefficient vector in this combination to be discriminative, which boosts LRC's classification ability. Several properties of the coefficients have been exploit, such as non-negativity [12], sparsity [17], locality [20].

This work is licensed under a Creative Commons Attribution 4.0 License. For more information, see https://creativecommons.org/licenses/by/4.0/

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/ACCESS.2020.3041154, IEEE Access

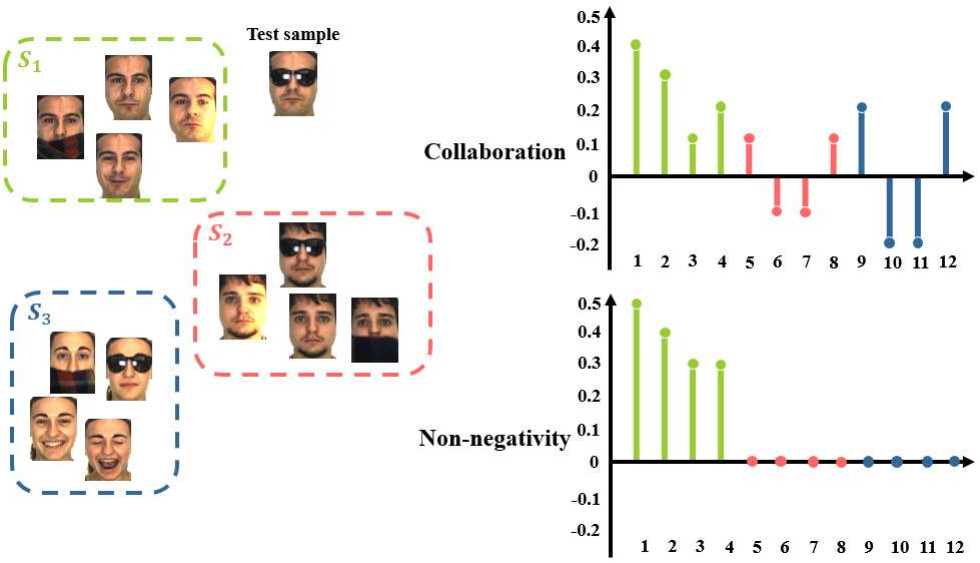


FIGURE 5: The illustration of nonnegativity in NRC.

1) Non-negative representation-based classification method

The non-negativity in the linear representation means all elements in the coefficient vector are non-negative. Since only the additive linear combination has a clear visual intuition meaning [54], the non-negative representation is suitable to perform the image classification in the real-world application with both physical interpretation and mathematical feasibility. Figure 5 shows the non-negativity of coefficients in linear representation.

The non-negative representation-based classification (NRC) [15] is an extension of the LRC method, which imposes a non-negativity constraint on the coefficient vector to enforce all elements in the vector to be non-negative. Using the non-negative representation, samples from the homogeneous class will be enhanced in the linear representation; meanwhile, samples from the heterogeneous class will be suppressed. The NRC can be described in the following Eq.36.

$$\begin{aligned} &min_\alpha \|y - X\alpha\|_2^2 \\ &s.t. \quad \alpha = \rho, \rho \geq 0, \end{aligned} \tag{36}$$

where ρ is an auxiliary variable in a linear equality-constraints problem.

The above Eq.36 is the no-negativity constrained least square model [55], which can be solved using ADMM method [56]. The augmented lagrange function can be written as follows:

$$L(\alpha, \rho, \eta, q) = \|y - X\alpha\|_2^2 + \langle \eta_p, \rho_p - \alpha \rangle + \frac{q}{2} \|\rho_p - \alpha\|_2^2 \tag{37}$$

The ADMM method is applied to alternatively optimize one variable when fixing others until the stop requirement is met. The update of each variable can be described as follows:

$$\begin{aligned} \alpha_{p+1} &= argmin_\alpha \|y - X\alpha\|_2^2 + \frac{q}{2} \left\|\alpha - (\rho_p + q^{-1}\eta_p)\right\|_2^2 \\ &= (X^T X + \frac{q}{2} I)^{-1} (X^T y + \frac{q}{2}\rho_p + \frac{1}{2}\eta_p), \end{aligned} \tag{38}$$

$$\begin{aligned} \rho_{p+1} &= argmin_\rho \frac{q}{2} \left\|\alpha_{p+1} - (\rho + q^{-1}\eta_p)\right\|_2^2 \quad s.t. \rho \geq 0 \\ &= \begin{cases} 0, & \alpha_{p+1} - q^{-1}\eta_p < 0 \\ \alpha_{p+1} - q^{-1}\eta_p, & \alpha_{p+1} - q^{-1}\eta_p \geq 0, \end{cases} \end{aligned} \tag{39}$$

$$\eta_{p+1} = \eta_p + q(\rho_{p+1} - \alpha_{p+1}) \tag{40}$$

The algorithm of NRC is summarized in the Algorithm 5.

2) Locality constraint

The locality means the information in data brought by the neighbors of the sample. Usually, in the classification scenario, the farther a sample is from the test sample, the less likely it belongs to the same class. Therefore, the locality is beneficial for the classification by referring information from the neighbors. Figure 6 shows the locality in the LRC.

The locality constraint in the linear representation will enforce the coefficient to consider the neighbors' sample. By considering information from the neighbors, the LRC will reconstruct the test sample and generate a discriminative representation for classification. In addition, by applying the locality constraint, the coefficient vector will naturally become sparse [21], since the sample's neighbors are a portion of the whole data. In [21], the locality constraint is first implemented for feature coding. The basic form of locality constraints in the LRC method can be described as follows:

This work is licensed under a Creative Commons Attribution 4.0 License. For more information, see https://creativecommons.org/licenses/by/4.0/

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/ACCESS.2020.3041154, IEEE Access

Algorithm 5 Non-negative representation-based classification

Require: Test sample y, Training set X, Number of class k, Training label set X^L, Number of iteration T, Tolerabce ϵ.

Coding:

Solve no-negativity constrained least square problem to get coefficient vector α:

$\alpha_1 = \rho_1 = \eta_1 = 0, o = 0;$

for o=1:T

Update α_{o+1} using Eq.38;

Update ρ_{o+1} using Eq.39;

Update η_{o+1} using Eq.40;

if $\|\alpha_{p+1} - \alpha_p\|_2 < \epsilon$ && $\|\alpha_p - \rho_p\|_2 < \epsilon$ && $\|\rho_{p+1} - \rho_p\|_2 < \epsilon$:

break;

end if o=1:T **do**

end for

Subspace approximation:

Construct the subspace set S using Eq.7;

Calculate the residuals R_i using Eq.8;

Classification:

Assign the class label I to sample y using Eq.9;

return class label I;

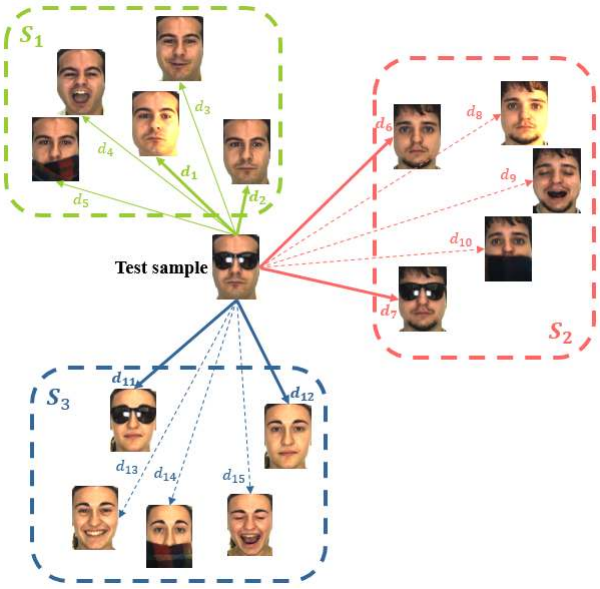


FIGURE 6: The locality in LRC.The bold lines indicate high weight between test sample and training sample, and the dash line indicate low weight between test sample and training sample.

$$\begin{aligned} \alpha &= \sum_{i=1}^{N} \|y_i - X\alpha_i\|_2^2 + q\,\|dist_i \odot \alpha_i\|_2^2 \\ s.t. \quad & 1^T\alpha_i = 1, \end{aligned} \tag{41}$$

where $dist_i$ represents the distance vector whose elements are the distances between the i^{th} test sample and all training samples, $\odot$ represents the element-wise produce between two vectors. The constraint $1^T\alpha = 1$ is a shift-invariant constraint. Here, the locality information of a test sample is quantified by measuring similarities with all training samples. For the coefficients with low value, there is a threshold set to reset them as zero. As shown in Figure 6, the closed samples with the test sample have larger weights (bold lines), and the samples far from the test sample have low weights (dash line).

Inspired by the locality constraint in LLC, the locality-sensitive dictionary learning (LCDL) [19] is proposed, which utilizes the locality constraint to construct a representative dictionary. When constructing the dictionary, all training samples are involved, and the distances between each test sample and all training samples are considered the locality information. Rather than using the distances with all training samples, the locality-constrained collaborative representation (LCCR) [20] used the sum of distances between each test sample and its neighbors as the locality constraint. The WSRC [48], [49] takes different similarity measurement (e.g. Gaussian kernel distance) as the locality information.

3) Sparsity constraint

The sparsity means some coefficients in the linear representation are zero. The sparse coefficient tends to produce a representation using samples from the correct class [57]. Therefore, the sparsity constraint will lead to a better class-specific residual error for classification. The sparse representation (SR) [14] fully exploits the sparsity in LRC, which only uses the training sample from the correct class to represent the test sample. Unlike the SR, some LRC extensions impose the sparsity constraint to enhance the performance, rather than use the sparsity for classification.

The general formulation of sparsity constraint on the linear representation can be described as follows:

$$\begin{aligned} \alpha &= argmin_\alpha \|y - X\alpha\|_2^2 \\ s.t. \quad & \|\alpha\|_0 \le \iota, \end{aligned} \tag{42}$$

where ι is the number of training samples used in the linear representation. In Eq. 42, the number of training samples is restricted in a small number ι, which enforces the coefficients to be sparse.

The sparsity augmented collaborative representation (SACR) is proposed by imposing the sparsity constraint in the above Eq. 42 on the LRC with l_2 minimization (dense representation). In this method, the coefficient vector of representation is the fusion of two components: 1) the coefficient vector of LRC with sparsity, and 2) the coefficient vector of dense representation. The coefficient vector of SACR can be described as follows:

$$\acute{\alpha} = \frac{\tilde{\alpha} + \hat{\alpha}}{\|\tilde{\alpha} + \hat{\alpha}\|_2}, \tag{43}$$

where $\tilde{\alpha}$ is the coefficient vector of dense representation, and $\tilde{\alpha}$ is the coefficient vector of linear representation with a

This work is licensed under a Creative Commons Attribution 4.0 License. For more information, see https://creativecommons.org/licenses/by/4.0/

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/ACCESS.2020.3041154, IEEE Access

sparsity constraint. As the name of the representations says, the coefficients under sparsity augment the coefficients of dense representation by enlarging the value of the coefficients from the correct class. This augmentation operation makes the correct class's coefficients discriminative since it enlarges the gaps of the coefficients' values between the correct class and the other classes, which is proven to be the advantage brought by the sparsity [58], [59] since the gap enlargment can be viewed as the sparsity enhancement. Similarly, to enhance the sparsity, Tian et al. proposed the FFT Consolidated Sparse and Collaborative Representation [18], making representation fusion between SRC and CRC in the frequency domain by using FFT. This method shows a more robust performance than FFT and CRC.

As one way to implement the sparsity constraint, a representation strategy named sample removing. In [15], the two-phase test sample representation (TPTSR) is proposed, where the test sample will be represented in two phrases. For the first phrase, the representation of the test sample is produced. Then, c training samples will be removed from the representation. The strategy of removing is removing c training samples with the smallest residuals of the representation $\{r_j = \|y - X_j\alpha_j\| \,|j = 1, 2, ..., H\}$. In the second phase, a new representation organized by the remaining $H-c$ training samples is produced for the classification. Figure 7 shows the coefficient distribution of two phases in TPTSR. This representation strategy follows the idea of sparsity constraint described in Eq. 43. The removing operation in the first phase does the same thing as the sparsity constraint. The removal of samples is equal to setting coefficients of these samples to zero making the coefficient vector sparse. Similarly, based on this strategy, samples can be removed in a heuristic way [40]. In this method, the removing operation is performed repeatedly until meeting the stop criteria. The strategy of removing is removing the training samples with minimum absolute value in each iteration.

Besides the sparsity for each element in linear representation, the group sparsity is proposed, which imposes the sparsity on class rather than the element in the coefficient vector. The group sparsity can also be realized using the sample removing strategy. The coarse-to-fine face recognition (CFFR) [60] method is proposed by removing all samples of the c classes with minimum residuals between the test sample and class-wise representation. In [17], the class-wise sparse representation (CSR) is proposed which focuses on the sparsity between classes. The problem of CSR can be described as follows:

$$\begin{aligned} \alpha &= \frac{1}{2}\|y - X\alpha\|_2^2 + g\|\rho\|_0 \\ \rho_i &= \|\alpha_i\|_2 \\ where \quad \rho &= [\rho_1, \rho_2, ..., \rho_k], \end{aligned} \tag{44}$$

where α_i represents the coefficient vector of i^{th} class samples. The second term is the class-wise sparsity measurement, g is the scaling factor. The constraint below assigns the l_2 norm of samples from each class to the variable ρ, which enforces the l_0 regularizer to focus on the sparsity between classes.

C. LINEAR REPRESENTATION-BASED CLASSIFICATION WITH FEATURE SPACES

The data in different feature spaces will have different discriminative properties. The kernel methods [61] map the data used in linear representation to the kernel feature space, enabling the LRC to perform classification on the non-linear separable data. Different mapping methods [25] aim to map the data to distinct feature spaces that fit the classification mechanism of the LRC method. The deep features extracted from deep learning architecture [62] form the deep feature space, which is able to improve the performance of LRC methods for image classification.

1) Kernel-based representation

The kernel trick was extensively applied in SVMs [63] in the very beginning. As a linear classifier, the SVMs perform well in the data, which is linear-separable. When processing the data, which is not linear-separable, the original SVMs will be extended to the non-linear classifiers by using the kernel trick. The kernel trick will map the low-dimensional data to kernel space, making it linear separable in this high-dimensional space. The kernel methods usually use the mercer's kernel, which can be described as follows:

$$k(x_1, x_1) = \phi(x_1)^T\phi(x_2), \tag{45}$$

where x_1 and x_2 are two data sample in the space, $k(\cdot,\cdot)$ is the kernel function, $\phi(\cdot)$ is the mapping function. Since the distribution varies when the data changes, the mapping function ϕ is undetermined in different scenarios. Three types of mapping functions are frequently used based on three different data assumption. They are linear kernel, polynomial kernel, and Gaussian radial basis function kernel:

$$k(x_1, x_2) = x_1^T x \tag{46}$$

$$k_u(x_1, x_2) = (\lambda x_1^T x_2 + \zeta)^u \tag{47}$$

$$k(x_1, x_2) = exp(-\lambda\|x_1 - x_2\|_2^2) \tag{48}$$

When the kernel trick is applied in the LRC methods, it will extend it to nonlinear classifiers. However, they perform the linear representation and classification as linear representation-based methods in the kernel feature space. All three kernel-based LRC methods meet the definition of the linear representation-based classification. The Kernel sparse representation-based classifier (KSRC) [64], [65] was proposed to kernelize the SRC method, which can be described as follows:

$$\begin{aligned} \alpha &= argmin_\alpha \|\alpha\|_1 \\ s.t. \quad \phi(y) &= \phi\alpha, \end{aligned} \tag{49}$$

This work is licensed under a Creative Commons Attribution 4.0 License. For more information, see https://creativecommons.org/licenses/by/4.0/

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/ACCESS.2020.3041154, IEEE Access

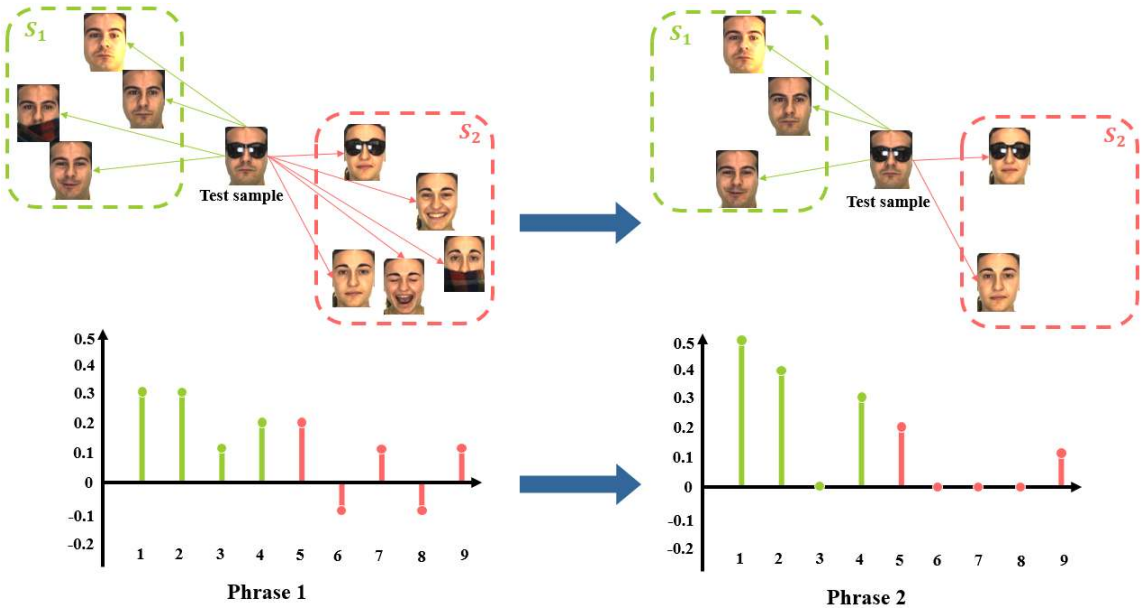


FIGURE 7: The coefficient distribution of TPTSR.

where $\phi = [\phi(x_1), \phi(x_2), ..., \phi(x_H)]$ is the mapped training samples, $\phi(y)$ is represented by a linear combination of mapped training samples in the kernel feature space.

Since the mapping function $\phi(\cdot)$ is unknown, the Eq.49 above should be rewritten in the following form:

$$\begin{aligned} &\alpha = argmin_\alpha \|\alpha\|_1 \\ &s.t. \quad \|K(\cdot, y) - K\alpha\|_2 \leq \epsilon, \end{aligned} \tag{50}$$

where $K = \phi^T\phi \in \mathbb{R}^{H\times H}$ is the Gram matrix. After the coefficient α is obtained, the class-specific residuals are calculated:

$$r_i = \|K(\cdot, y) - K\alpha_i\|_2 , \tag{51}$$

where the α_i represents the coefficient vector of the i^{th} class. Like the conventional LRC method, the final classification output is the index of class associated with the minimum class-specific residual.

Rather than using the l_1 minimization, the kernelized LRC with l_2 minimization is proposed in [66], called Kernel collaborative representation-based classification (KCRC). The formulation of KCRC is showing as follows:

$$\begin{aligned} &\alpha = argmin_\alpha \|\alpha\|_2 \\ &s.t. \quad \phi(y) = \phi\alpha \end{aligned} \tag{52}$$

The solution of the above Eq. 52 is:

$$\alpha = (K + \lambda I)^{-1}K(\cdot, y) \tag{53}$$

The work in [26] proposed a weighted kernel representation-based method (WKRBM) to impose a weight on the kernel-based representation of LRC for better performance.

2) Deep learning features

The deep learning architectures [62] show a powerful capability to learn the discriminative feature representation for the image classification. Several convolutional neural networks proposed recently have achieved state-of-the-art performance in the visual classification task [67]–[69]. Based on the deep neural networks already trained on the large-scale image dataset, the concept of transfer learning [70] is proposed, which means mapping the original data from the raw feature space to the deep feature space where data representation is discriminative by using the pre-trained deep learning architecture.

Trials of implementing deep learning features to the LRC method have achieved success [10], [71]–[73], which obtained the competitive performance. Some LRC methods directly use the data from deep feature spaces and improve the performance smoothly compared with the results based on the raw feature space [10], [12], [73]. In [71], a test sample is represented in parallel by two groups of linear representation with l_2 minimization: a linear representation with data from the raw feature space and a linear representation with data from deep feature space. Thus, two coefficients are obtained simultaneously:

$$\begin{aligned} &\alpha = argmin_\alpha \|y - X\alpha\|_2 + \lambda \|\alpha\|_2 \\ &\alpha_{deep} = argmin_\alpha \|y - X_{deep}\alpha_{deep}\|_2 + \lambda \|\alpha_{deep}\|_2 , \end{aligned} \tag{54}$$

where X_{deep} represents the training samples from the deep feature space, α_{deep} is the coefficient obtained based on the X_{deep}.

Then, two groups of class-specific residuals are calculated and fused in a element-wise multiplication manner:

This work is licensed under a Creative Commons Attribution 4.0 License. For more information, see https://creativecommons.org/licenses/by/4.0/

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/ACCESS.2020.3041154, IEEE Access

$$r_i = \|y - X\alpha\|_2 \odot \|y - X_{deep}\alpha_{deep}\|_2 \qquad (55)$$

Since the class-specific residuals indicate the probability of the test sample belonging to a specific class, the smaller the residual of this class is, the higher probability the test sample belongs to this class. Therefore, the fusion between two class-specific residuals can be viewed as the 'weighting' operation that imposing weights obtained from deep feature spaces to the residuals of the raw feature spaces. Figure 8 shows the effect of 'weighting' operation.

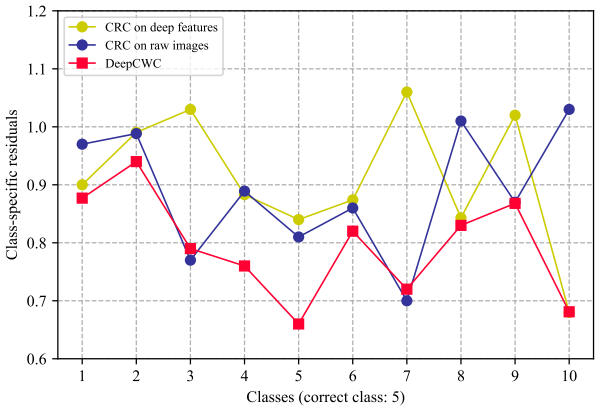


FIGURE 8: The class-specific residuals of CRC on raw image, CRC on deep features, and DeepCWC. (From Zeng et al. [71])

Besides implementing the deep learning feature to the residual in LRC method, cheng et al. [72] used the well-trained deep learning feature to achieve the state-of-the-art performance in face recognition:

$$\alpha = argmin_\alpha \|f(y) - f(X)\alpha\|_2 + \lambda \|\alpha\|_2 , \qquad (56)$$

where $f(\cdot)$ represents the deep feature space mapping. In this method, each image is fed into a specially-designed 5-layers CNN for deep feature extraction. Then, the SRC is applied to data from deep feature space to output the classification result.

3) Other feature spaces

There are other methods that map the data to different spaces and then apply the LRC methods to perform classification. In [25], the Euler sparse representation-based classification (Euler SRC) maps the data samples to an Euler space before inputting them to SRC in order to boost the robustness of the classifier. Correspondingly, in the complex space, the conventional l_2-norm distance metric used for calculating residual in SRC is replaced by the cosine distance [74]. With the cosine distance, the margin between data samples of two classes will be larger than using the Euclidean distance. In [24], the data samples are transformed to a latent space where samples from the same class can be represented in one point to overcome the pose variant in the face identification problem. Then, the SRC is applied to classify the data in the latent space. In order to map the data to a space that makes it discriminative and fits the SRC method, the SRC-DP [75] is proposed. In the SRC-DP, a projection matrix is proposed to map the data samples to space where the between-class residual is maximized, and the within-class residual is minimized. In [76], a log-euclidean space is learning for the SRC method, where the data from the same class will lie on a subspace with discriminative structure. In [77], the proposed projection representation-based classification (PRC) method constructed an ideal representation that maps the data samples from each class to a hyperplane with the nearest projected test sample. The [23] proposed an algorithm to generate approximately symmetrical images to recover flaws existing in the raw images in data. All data processed by this algorithm is fed to an SRC classifier to predict the label. This method can be viewed as the refinement of feature space, which, to some extent, improves the classification performance of the SRC method. In [78], Liu et al. proposed a discriminative sparse embedding (DSE) that projects data from the high-dimensional space to a low-dimensional feature space for classification by integrating SRC and a graph-based method to capture the local information of the noised data. In [79], a discriminative feature extraction based on sparse and low-rank representation (DFE) was proposed to map the data to the feature space that was embedded with both local information and global information from the raw feature space.

D. LINEAR REPRESENTATION-BASED CLASSIFICATION WITH STRUCTURAL INFORMATION

The conventional LRC methods usually consider each sample separately in the representation, which ignores the structural information inside the data. The structural information is the relationship among samples in the dataset or relationships among pixels of a sample. For example, the SRC performs the l_1 minimization on the image level, and no emphasis is imposed on the correlation in the pixel level. For dealing with this problem, the structural information is modeled in different LRC methods. In [80], Wang et al. proposed the adaptive sparse representation-based classification (ASRC) by considering the correlation structure in the SRC model. Since the correlation structure describes the relationship between the samples, it can be a variable to control the attention of the LRC classifier. For example, if the samples are highly correlated, the classifier will pay more attention to the representation correlation. When the samples have a low correlation, the classifier focuses more on the sparsity. The LRC can be adaptive between l_1 and l_2 norm minimization by using the trace norm, which also captures the correlation in the data. The objective function of the ASRC is showing as follows:

This work is licensed under a Creative Commons Attribution 4.0 License. For more information, see https://creativecommons.org/licenses/by/4.0/

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/ACCESS.2020.3041154, IEEE Access

$$\begin{aligned} &\alpha = argmin_{\alpha} \left\| Xdiag(\alpha)\alpha \right\|_* \\ &s.t. \quad y = X\alpha \end{aligned} \tag{57}$$

where the $\|\cdot\|_*$ is the nuclear norm. The term $\|Xdiag(\alpha)\alpha\|_*$ is called the correlation regularizer, which involves the training samples X into consideration in order to exploit the correlation structure inside the training samples. The correlation regularizer can be decomposed as the following forms when $X^TX = 1$:

$$\begin{aligned} &\left\| Xdiag(\alpha)\alpha \right\|_* \\ &= Tr((Xdiag(\alpha))^T(Xdiag(\alpha))) \\ &= Tr(diag(\alpha)^T diag(\alpha)) \\ &= \left\| \alpha \right\|_1, \end{aligned} \tag{58}$$

when $X = X_1 1^T$:

$$\begin{aligned} \left\| Xdiag(\alpha)\alpha \right\|_* &= \left\| X_1 1^T \alpha \right\|_* \\ &= \left\| X_1 \alpha^T \right\|_* \\ &= \left\| x_1 \right\|_2 \left\| \alpha \right\|_2 \\ &= \left\| \alpha \right\|_2, \end{aligned} \tag{59}$$

The $X^TX = 1$ means each training sample is orthogonal from each other, indicating a low correlation in the data. The $X = x_1$ means each training sample x_i is the same as x_1, indicating a high correlation in the data. In the two extreme cases above, we can observe that the trace norm takes both sparsity and correlation into account when representing the test sample.

Besides the correlation structure information in the data in [81], the structural error is also considered in the LRC model. The proposed matrix-based representation in [81] is described as follows:

$$\begin{aligned} &(E, \Psi) = argmin_{E,\Psi} \left\| E \right\|_* + \lambda \left\| \Psi \right\|_1 \\ &s.t. \quad E = Y - X\Psi, \end{aligned} \tag{60}$$

where $\Psi \in \mathbb{R}^{n_{test} \times n_{training}}$ is the coefficient matrix composed of coefficient vector corresponding to each sample. E is the structural error matrix of all test samples. In Eq. 60, representation coefficient Ψ and structural error E are optimized simultaneously. The first term ensure the structural error in the representation is minimized, and the second makes the representation of each test sample sparse. Since the structural error is modeled, the method becomes more robust.

The patch-based representation [41], [82], [83] represents the test sample using small patches from the whole image. The patch is a fixed-scale small partition of a whole image. By using the patches to represent the patch, the local structure in the dataset is fully utilized. The patch-based collaborative representation-based classification (PCRC) [41] method applied N_p CRC classifiers to separately classify N_p patches. For each image in the dataset, N_p patches are cropped with a fixed size and same location. When representing the test sample, there are N_p coding procedures performed in parallel. The objective function of the PCRC is showing as follows:

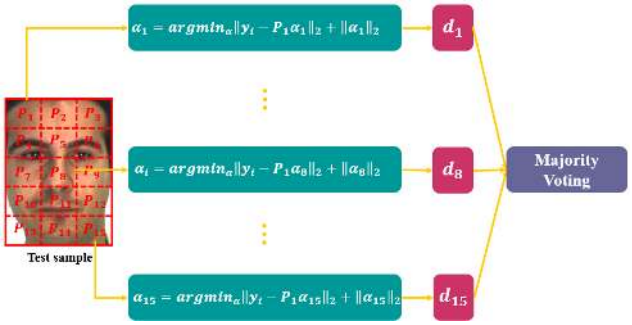


FIGURE 9: The pipeline of PCRC [41] method.

$$\alpha_i = argmin_{\alpha} \left\| y_i - P_i\alpha_i \right\|_2 + \left\| \alpha_i \right\|_2, \tag{61}$$

where α_i represents the coefficient vector of the i^{th} patch, y_i represents the i^{th} patch of the test sample, $P_i = \left\{P_i^1, P_i^2, ..., P_i^H\right\}$ represents the image set containing the i^{th} patch of all training samples. After coefficient of each patch α_i is obtained, the following subspace approximation and classification procedures are performed as normal. Finally, this algorithm will produce N_p classification outputs. To ensemble these N_p classification outputs, a class-specific weights $\hat{w}_i$ is calculated using the constrained l_1-regularized optimization:

$$\begin{aligned} &\hat{w}_i = argmin_w \left\| e - Dw \right\|_2 + \left\| w \right\|_1 \\ &s.t. \quad \sum_{i=1}^{k} w_i = 1, w_i > 0, i = 1, 2, ..., k, \end{aligned} \tag{62}$$

where e is the vector only contains elements of 1, D is the decision matrix whose the element d_{ij} corresponding to the j^{th} patch of the i^{th} image. If the classification result of a certain patch is equal to the label of its image, $d_{ij} = 1$. Otherwise, $d_{ij} = 0$. After the class-specific weight vector $\hat{w}$ is calculated, weights from all patches will be summed up to generate an overall class-specific weight vector. The class label with the highest weight will be selected as the output. The pipeline of PCRC is illustrated in Figure 9. Besides using the ensemble learning technique to decide the outputs from different patches. Gao et al. [83] proposed regularized patch-based representation (RPR), which established a uniform model to classify the patches. This model can be described as follows:

$$\begin{aligned} &min_{\alpha,\beta,E} \left\| E \right\|_F + \lambda \left\| \beta \right\|_1 + \mu \left\| \alpha \right\|_{2,1} \\ &s.t. \quad Y_i = X_i\alpha_i + D_i\beta_i + E_i, \end{aligned} \tag{63}$$

where $E = \left[E_1, E_2, ..., E_{N_p}\right]$ represents the error of the i^{th} patch, $\alpha_i = \left[\alpha_1, \alpha_2, ..., \alpha_{N_p}\right]$ represents the sparse coefficient vector of the i_{th} patch, D_i represents the intra-class variance matrix proposed in the ESRC [34] of the i^{th} patch. $\beta_i = \left[\beta_1, \beta_2, ..., \beta_{N_p}\right]$ represents the intra-class variance coefficient vector of the i_{th} patch. This method imposes group sparsity ($l_{2,1}$ minimization) to the representation

This work is licensed under a Creative Commons Attribution 4.0 License. For more information, see https://creativecommons.org/licenses/by/4.0/

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/ACCESS.2020.3041154, IEEE Access

of intra-class variance and sparsity to the representation of training sample. The sparsity ensures the correct samples are selected for representation and the group sparsity ensures the correct class variance is selected for representation. The intra-class variance describes the degree of difference among the samples of the same class, which belongs to the structural information between samples.

E. LINEAR REPRESENTATION-BASED CLASSIFICATION WITH SUBSPACE LEARNING

In the subspace approximation procedure of the LRC methods, the composition of each class-specific subspace is critical to the final accuracy. In [84], the collaborative representation optimized classifier (CROC) is proposed by seeking for the trade-off between the nearest subspace classifier (NSC) and collaborative representation-based classifier (CRC). The strategy of CROC can be described as follows:

$$\begin{aligned} \alpha^{CRC} &= argmin_{\alpha} \left\| y - X\alpha \right\|_2 + \lambda \left\| \alpha \right\|_2, \\ \beta^{NSC} &= argmin_{\beta} \left\| y - X\beta \right\|_2, \\ r_i(\mu) &= \mu \left\| y - X_i\alpha_i \right\|_2 + (1-\mu) \left\| y - X_i\beta_i \right\|_2, \end{aligned} \tag{64}$$

where r_i represents the final residual of the i^{th} class, α^{CRC} and α^{NR} are the representation coefficient of CRC and NSC, respectively. μ is the weight to balance the significance between CRC and NSC, which can be determined by performing the cross-validation [85].

In [86], the PLRC is proposed to classify a set of images by calculating two coefficient vectors. The figure illustrates two subspaces construction strategy and the calculation of two coefficient vectors. One coefficient vector contains the joint coefficients between related subspace and test space; the other coefficient vector contains the joint coefficients between the test sample and unrelated subspace. These two types of the coefficient vector construct a pair of metrics related to metrics and unrelated metrics. The two metrics are combined as follows:

$$dist_f^c = dist_r^c / dist_r^u, \tag{65}$$

where $dist_f$ represents the combined metric, $dist_r$ represents the related metric, and $dist_u$ represents the unrelated metric. This combination strategy maximizes the related metric meanwhile minimizes the unrelated metric.

In [31], k subspaces of each class are constructed for further representation in LRC methods, which is called two-stage LSCL. In the first stage of LSCL, nearest c samples with the test sample from each class are selected to form a subspace of the i^{th} class, then the average sample of each subspace is calculated:

$$\bar{X}_i = \frac{1}{k} \sum_{i=1}^{k} x_{ij}^k, \tag{66}$$

where $\bar{X}_i$ is the average sample of the i^{th} class, x_{ij}^k is the j^{th} sample of the i^{th} class. In the second stage, the subspaces are fed into the LSRC classifier [87]:

$$argmin_{\alpha} = \left\| y - X\alpha \right\|_2 + \lambda \left\| W\alpha \right\|_1, \tag{67}$$

where W is the weighted diagonal matrix whose elements on the main diagonal are the distance between the test sample and samples in each subspace.

F. LINEAR REPRESENTATION-BASED CLASSIFICATION IN SEMI-SUPERVISED LEARNING AND UNSUPERVISED LEARNING

Semi-supervised learning means the machine learns from on the dataset containing both labeled data and unlabeled data [88]. When there are few data labeled, the variations in each class are hard to capture. Likewise, in the LRC method, the lack of labeled training samples will heavily influence the performance [89]–[91]. The S^3RC was proposed in [33] to perform semi-supervised classification with the LRC method to address this problem. In S^3RC, the linear variations are firstly eliminated from the raw samples, and all samples are normalized to fit the zero-mean Gaussian distribution. Since the dataset contains both labeled and unlabeled data, the Gaussian Mixture Model (GMM) [92] is applied to estimate the prototype sample of each class. Next, to estimate the parameters in the GMM, S^3RC utilizes the EM algorithm. Finally, the prototypes of all classes are organized to construct a new training set for further classification using ESRC [34] method. The basic model of S^3RC can be described as follows:

$$\begin{bmatrix} \alpha \\ \beta \end{bmatrix} = argmin_{\alpha,\beta} \left\| y - [X, \nu] \begin{bmatrix} \alpha \\ \beta \end{bmatrix} \right\|_2^2 + \lambda \left\| \begin{bmatrix} \alpha \\ \beta \end{bmatrix} \right\|_1, \tag{68}$$

where $\nu = [X_1 - \varrho_1 1^T, X_1 - \varrho_2 1^T, ..., X_1 - \varrho_k 1^T]$ is the variation dictionary whose each column is the subtraction between training sample and the extended prototype ϱ_i of each class, α, β are the sparse coefficient of linear representation of training samples and atoms in the variation dictionary. The prototype ϱ_i will be estimated using the GMM model:

$$\begin{aligned} \vartheta &= argmax_{\varrho_i, \Sigma_i, \pi_i} log p\left(X^{norm} | \varrho_i, \Sigma_i, \pi_i\right) \\ &= \sum_{i=1}^{k} log \pi_{l_i} N\left(\hat{y}_i^{norm} | \varrho_{l_i}, \Sigma_{l_i}\right) \\ &+ \sum_{i=h_l+1}^{H} \sum_{j=1}^{k} u_{ij} log \pi_{l_i} N\left(\hat{y}_i^{norm} | \varrho_i, \Sigma_i\right), \end{aligned} \tag{69}$$

where π represents the prior probability of the i^{th} class, $X^{norm} = \left[x_{l_1}^{norm}, x_{l_2}^{norm}, ..., x_{l_n}^{norm}, x_{ul_1}^{norm}, x_{ul_2}^{norm}, ..., x_{ul_n}^{norm}\right]$ represents the training set whose each element is normalized and variation eliminated to fit the non-zero gaussian distribution, Σ_i represents the covariance matrix, $\hat{y} =$ represents the image set contains both labeled normalized samples after variation elimination and unlabeled normalized samples after variation elimination, u is the label of unlabeled samples. The parameters in Eq. 69 can be estimated by EM algorithm.

This work is licensed under a Creative Commons Attribution 4.0 License. For more information, see https://creativecommons.org/licenses/by/4.0/

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/ACCESS.2020.3041154, IEEE Access

Finally, the output label is decided by ESRC [34] method, which the formulation of decision is:

$$label(y) = argmin_i \left\| y - [\varrho * \quad \nu] \begin{bmatrix} \alpha *_i \\ \beta * \end{bmatrix} \right\|_F, \tag{70}$$

where $\varrho *$ is the new estimated training set, $\alpha *_i$ and $\beta *$ is the newly calculated sparse coefficient vectors using ESRC.

In [93], an active learning paradigms [94] is imposed to the TPTSR [15] which uses two-phase coding to represent the test sample. There are two groups of samples separately represent the test sample in each phrase: the group of labeled data and the group of the labeled data and unlabeled data. Based on the sample removing strategy introduced in section III-B3, the residual to decide the final result in the second phrase is showing as follows:

$$\begin{aligned} label(y) = argmin_i \lambda \left\| y - \sum_{i=1}^{H_l} x_{l_i} \alpha_{l_i} \right\|_2 \\ + (1-\lambda) \left\| y - \sum_{i=1}^{H} x_i \alpha_i \right\|_2, \end{aligned} \tag{71}$$

where H_l is the number of samples of the labeled sample, $x_{l_i}\alpha_{l_i}$ is the i^{th} representation of labeled samples. This decision function seeks for the trade-off between labeled samples and all samples to perform semi-supervised classification.

In unsupervised learning for image classification, an optimal projection is first learned using a linear representation method. Then, the clustered data is fed to a LRC method for classification. The adaptive weighted nonnegative low-rank representation (AWNLRR) [35] is a typical method that performs image classification with unsupervised learning. Firstly, a low rank projection is learned:

$$\begin{aligned} & min_{P,Q} ||P^{1/2} \odot (X - XQ)||_F^2 + \frac{\lambda_2}{2} ||P||_F + \lambda_2 ||Q||_* \\ & + \lambda_3 tr(B^T P) \\ & s.t. P \geq 0, S^T 1 = 1, Z \geq 0, \end{aligned} \tag{72}$$

where P is the weighted matrix, Q is the affinity graph to capture the intrinsic feature from the data, B is a matrix, where each element B_{ij} is the distance between the i^{th}sample and the j^{th}sample, $||\cdot||_*$ the nuclear norm, and $||\cdot||_F$ denotes the Frobenius norm. The first term is the reconstruction term, the second term and constraint $S^T 1 = 1$ ensure the weighted matrix is in a reasonable range, the third term makes the matrix Q be low-rank such that the global structure is preserved, and the last term enforces the affinity graph Q to learn the local information in the data. The non-negative constraints $P \geq 0$,$Q \geq 0$ ensures the learned weighted and projection matrix have good interpretability. After the affinity graph Q is learned, the data is first clustered by the Normalized cut (Ncut) algorithm. Then, CRC performs classification on the data with clustered labels.

Besides AWNLRR, the low-rank preserving projection via graph regularized reconstruction (LRPP_GRR) [36] constructs the graph in the reconstruction term before classification. The Double Low-Rank Representation (DLRR) [95] learns two low-rank matrices simultaneously capture global intrinsic information in the row space and column space, and the LatLRR [96] learns a pair of low-rank matrices to capture the intrinsic and salient features for image classification.

IV. APPLICATIONS

A. REMOTE SENSING

In the remote sensing research area, the hyperspectral imagery (HSI) [97], [98] are widely used for different applications [99]–[101]. In HSI, each pixel on the image contains information on a wide range of wavelength channels, which makes the whole image informative and high-dimensional. Hyperspectral image classification aims to assign the class label to each pixel of the image. Since pixels in the same class lie in the same subspace, the LRC methods classify the hyperspectral test pixel with a pixel subspace constructed using a linear combination of the hyperspectral pixels in the training set, which makes full use of the informative high-dimensional image in the representation and alleviates the computational cost in the classification. Figure 10 shows the sample from the Indian Pine Site 3 AVIRIS hyperspectral dataset [102].

(a) (b) (c)

FIGURE 10: The sample of hyperspectral image (Indian Pine Site 3 AVIRIS hyperspectral dataset). (a) overview image, (b) cross section, (c) ground truth.

In [103], Chen et al. proposed two strategies using the sparse representation of the pixels in the training set to represent the given test pixel. The first strategy considered the contextual information when performing classification, where four neighbor pixels in the spatial domain are utilized to represent the test pixel sparsely. The sparse coefficient vector was obtained from the linear combination composed of these four neighbors. The second strategy takes the inter-pixel correlation into account during classification. The joint sparsity model is implemented here to calculate the shared sparse coefficients among the N neighbors of the test pixel. The classification output was determined by the residual between the class-specific representation and test sample, which can be viewed as the SRC scheme. Based on the above second strategy, the joint sparsity model was extended to a kernel version in [104] to improve the classification performance. The joint sparsity model can also be extended

This work is licensed under a Creative Commons Attribution 4.0 License. For more information, see https://creativecommons.org/licenses/by/4.0/

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/ACCESS.2020.3041154, IEEE Access

to a multi-task model, which is the multi-task joint sparse representation (MJSR) proposed in [105]. In MJSR, image sets of different bands were clustered into B band sets, with t tasks established by selecting one band in each set. Next, a coefficient matrix whose row is the sparse coefficient vector of a task was learned for classification. The MJSR calculated the joint sparse coefficient while persevering the correlations in the spectral field. In [106], an adaptive neighborhood system was constructed by introducing self-paced learning (SPL) [107]. The proposed self-paced joint sparse representation (SPJSR) learned the weight of each neighbor approximation and sparse coefficient in a self-paced scheme.

In [108], the multiscale adaptive sparse representation (MASR) was proposed to exploit the spatial information using multiscale test pixels. The different scales of the test sample provided different spatial structures and properties. This method jointly optimized different scale-level representations to produce a shared sparse coefficient vector across multiple scales. As for the output, the class label was determined by the residuals similar to SRC. MASR showed better classification performance than in the above-mentioned joint sparsity model using a single scale [103].

For dealing with the unstableness brought by the sparse representation, the manifold-based sparse representation algorithm was proposed in [109]. Two regularization terms were imposed in the conventional SRC method. The first regularization term was the locally linear embedding regularization so that the extension of SRC regularized by this term is called LLESR. In order to enhance the robustness of the sparse representation, the LLESR considers the local structure by minimizing the distance between the test sample and the representation composed of its neighbors. The second regularization term is the laplacian eigenmap regularization. Therefore the extension of SRC with this regularization term is called LESRC. The LESRC considers the local structure by minimizing the overall distances between each pair of neighbors from the test sample. Since the correlation between some classes is high in the HSI classification, using conventional sparse representation which imposes sparsity individually on each sample, is not suitable. The class-dependent sparse representation classifier (cdSRC) was proposed to make the SRC robust in HSI classification. The cdSRC combines the SRC and KNN [110] algorithm together in the coding procedure and subspace approximation procedure, respectively. The class-dependent sparse representation imposes sparsity on a class rather than a sample to make the class-specific residual of the test sample more discriminative as samples across the class will not represent the test sample. The class-dependent KNN produced a class-specific distance between the test sample with the average samples in the neighborhood of samples from each class, preserving the locality information of the test sample.

In [111], collaborative representation (CR) was applied in hyperspectral imagery classification. Based on the idea of CR, the joint collaborative representation (JCR) model was built to perform classification in a competitive and efficient way. The JCR calculates the shared coefficient vector of the test pixel and its neighbors using the training sample. Furthermore, a nonlocal joint-signal matrix is constructed by the top n_{corr} neighbors according to the degree of correlation to filter the pixels that are not similar to the test pixel. Similarly, using the collaborative representation, Jia et al. [112] applied the 3-D Gabor feature to the collaborative representation, termed as 3GCR, to boost the robustness of the classification performance on hyperspectral images. The 3-D Gabor feature provides an informative feature space that contains a large number of feature dimensions to ensure the robustness of the CR. The CR is an efficient representation method since it has a closed-form solution. Combining them together will produce a classification method, that is both effective and efficient. Inherited from the idea of fusing the Gabor feature with the CR, the Gabor cube selection based multitask joint sparse representation-based classification (GS-MTJSRC) was proposed in [113]. The 3-D Gabor transformation was applied to the data sample to generate the Gabor cubes based on three directions. Then, a filter removed the Gabor cubes with a low Fisher discriminative score for representation. Next, the multitask sparse representation represents the test pixel using the filtered Gabor cubes in the training set and finally outputs the classification result. This method outperformed the aforementioned 3GCR. Since the Gabor feature is beneficial to HSI classification, in [114], a multi-feature learning strategy that utilizes CR processing four types of features (global feature, local feature, shape feature, and spectral feature) was proposed. In this strategy, the CR will generate the coefficient vector for each feature, where an overall coefficient vector is obtained by summing up the subtraction between the coefficient vector and the mean overall coefficient vector.

B. FACE RECOGNITION

LRC methods have been extensively applied in face recognition applications due to the critical assumption that: the face images captured under different conditions (e.g., lighting, corruption, expression) lie on a low-dimensional subspace. According to Assumption 1, LRC methods are able to construct the face subspace for each class of face images. Therefore, by using LRC methods, there exists an effective and robust classification ability to perform face recognition. The SRC [14] method was originally designed for robust face recognition, which assumes the test sample can be represented by only the training samples from the same class. Therefore, the subspace constructed by the sparse representation is the face subspace of the test sample. Although the coefficients are sparsely distributed, the coefficients of the correct class are densely distributed. This phenomenon explains why the SRC is able to perform robust face recognition: the subspace built by samples from the correct class is more discriminative than the subspace constructed by the other samples. In [10], the CRC performed competitive and efficient face recognition by using collaborative representation, which represents the test sample using samples across

This work is licensed under a Creative Commons Attribution 4.0 License. For more information, see https://creativecommons.org/licenses/by/4.0/

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/ACCESS.2020.3041154, IEEE Access

different classes. The collaborative representation is closer to the test sample since all possible training samples are utilized, where the constructed class-specific subspace is nearer to the test sample as much as possible. It is argued that the locality brought by the CRC is more critical than the sparsity brought by the SRC. To produce the sparsity based on the collaborative representation, [15], [40], [60], [73] used a two-stage strategy and achieved acceptable classification performances. The patch-based representation [17], [41] divided the face image into several non-overlapped patches and integrated their outputs to make the final decision. For large-scale face recognition, the two-stage non-negative representation sparse representation [115] was proposed by reduce the scale of the dataset in the first stage and perform efficient non-negative sparse representation in the second stage. In [80], an adaptive SRC was proposed based on the trace norm, which maintained a balance between l_1 and l_2 minimization according to the data's correlation. For multiview face recognition, the joint sparse representation-based classification (JSRC) [116] was proposed, where it constructed a shared sparse coefficient for different views of an individual's face image. To overcome the pose variation of the face, synthesized face images were generated [23], [24] to produce the coefficient vector that was invariant to pose.

C. MEDICAL AND MULTIMODAL BIOMETRICS

1) Medical biometrics

Medical biometrics is a research field that monitors an individual's health condition based on the characteristics of a certain disease [117]. Specifically, LRC has also been frequently used in this domain to detect disease individual according to the appearance of body surface features (e.g., regions of the face and tongue). In [118], [119], the SRC was used for microaneurysm detection by performing classification on the extracted retinal blood vessel image, which is a binary image containing the outline of the retinal blood vessels. In [8], the SRC was applied for diabetes mellitus detection based on the color features extracted from human facial block images. The detection result was promising, reaching 97.54% in accuracy. Based on the color features of the different combinations from the facial blocks, Shu et al. utilized the probabilistic CRC [16] to perform disease detection and achieved an impressive accuracy of 99.88% [38]. Using the same strategy in [120], a high accuracy was achieved for heart disease detection utilizing ProCRC based on the facial blocks again feature. Besides the facial images, tongue images were also involved, which is regarded as another view in medical biometrics. In [121], a joint discriminative collaborative representation (JCDR) method was proposed as a multimodal method to simultaneously process the facial blocks and tongue blocks as multiple views and color with texture as multiple features for detecting liver disease.

2) Multimodal biometrics

Usually, in the biometrics field, there are different biometric information sources that require multimodal techniques to fuse the information for better results. As a LRC method, the group sparse representation based classification (GSRC) [7] method (as an extension of the SRC method) considered multimodal information when representing the test sample. In this method, the test sample was the concatenation of a N_m modal sample, where N_m sparse coefficient vectors corresponding to the modals were concatenated to construct a sparse coefficient matrix. All coefficient vectors were learned simultaneously. For processing the multimodal data on a high-dimensional space, KGSRC [122] was proposed as an extension of the GSRC method. In [123], the joint deep convolutional feature representation (JDFR) was proposed to perform hyperspectral palmprint recognition. In JDFR, a specially-designed CNN with 16 layers was used to extract each band's deep feature. Therefore, to use the information from all bands in the hyperspectral palmprint dataset, a CNN stack whose basic element is a 16-layer CNN was constructed. Followed by the CNN stack feature extraction, CRC was applied to perform classification on the concatenation of the deep features from the CNN corresponding to each band. The JDFR-CRC architecture outperformed other state-of-the-art methods in hyperspectral palmprint recognition.

V. EXPERIMENTS

In this section, we performed extensive experiments using 13 different LRC methods to show its performances on 7 image datasets by taking the face and object images as examples of image classification. First, we will briefly introduce the datasets used in the experiments: GT [124], AR [125], ORL [126], COIL20 [127], FEI [128], Yale B [129], and Flavia [130]. Then, we will introduce the experiment settings. Following this, we will show the experimental results of 13 LRC methods: SRC [14], CRC [10], NRC [12], ProCRC [16], SCRC [131], SARC [57], KSRC [64], KCRC [66], AWCRC [51], KWCRC [51], TPTSR [15], CFFR [60], and AWNLRR [35]. The KSRC, KCRC, and KWCRC are nonlinear extensions of the LRC method. Finally, a discussion based on the experimental results is given.

A. DATASET DESCRIPTION

In this subsection, we make a briefing of each dataset used in the experiments:

GT. The Geogia Tech face database contains 750 face images from 50 people. Each image is JPEG image in the size of 150 $\times$ 150 on average. There are several forms of variation in each subject, such as different facial expression and lighting conditions. In the experiment, We resize each image to 40 $\times$ 30 pixels. Figure 11 (a) shows the samples in the GT face database.

ORL. The ORL database of faces contains 400 images from 40 classes. Each image is PGM image in the size of 92 $\times$ 112 pixels. Since the images are taken in different times, some images has different lighting condition, facial expression and details. In the experiment, we resize each image to 32 $\times$ 32 pixels. Figure 11 (b) shows the samples in the ORL face database.

This work is licensed under a Creative Commons Attribution 4.0 License. For more information, see https://creativecommons.org/licenses/by/4.0/

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/ACCESS.2020.3041154, IEEE Access

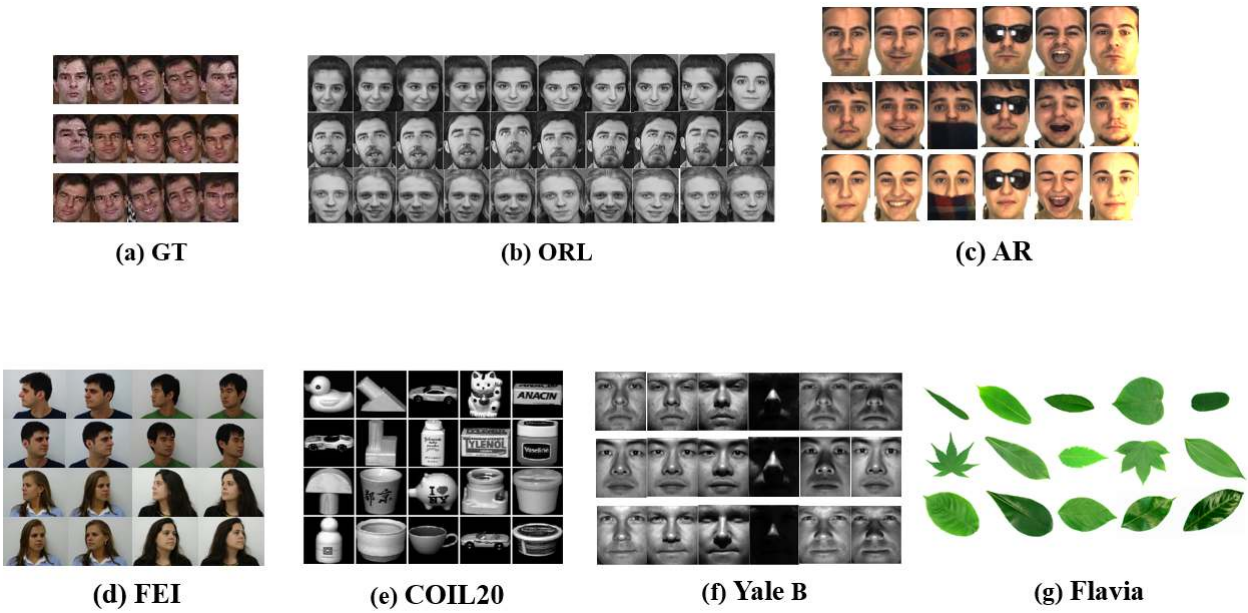


FIGURE 11: Samples in the dataset. (a) GT, (b) ORL, (c) AR, (d) COIL20, (e) FEI, (f) Yale B, (g) Flavia

AR. The AR face database contains 4000 color images from 126 people. Each image is a RGB RAW file in the size of 768 $\times$ 576 pixels. For each subject(person), there are two sessions which taken from 2 different days. Therefore, each subject contains images of different lighting condition, facial expression, and other intra-class variant. In the experiment, we resize each image to 40 $\times$ 32 pixels. Figure 11 (c) shows the samples in the AR face database.

COIL20. The columbia object image library contains two sets of images. The first set has 720 raw images of 10 objects, and the second set has 1440 images from 20 objects. In the experiment, we select the second set. Each image in the dataset is a PGM image in the size of 128 $\times$ 128 pixels. For each subject (object), there are 72 images captured by a CCD with 360 degree rotating around it. Each image has 5 degree angle changing compared with the previous one or the next one. We resize the each image to 32 $\times$ 32 pixels for the experiment. Figure 11 (d) shows the samples in the COIL dataset.

FEI. The FEI face database contains 2800 images from 200 people. Each image is a JPEG image in the size of 640 $\times$ 480 pixels. In the experiment, we resize each image to 24 $\times$ 96 pixels. For each subject (person), there are 14 images with different angle of face (ranging from 0 degree to 180 degree) and lighting conditions. Figure 11 (e) shows the samples in the FEI dataset.

Yale B. The Yale face database B contains 5760 images from 10 person. Each image is a PMG image in the size of 640 $\times$ 480 pixels. In the experiment, we resize each image to 320 $\times$ 240 pixels. For each subject (person), there are 576 images with 9 different poses and 64 lighting conditions. Figure 11 (f) shows the samples in the YaleB database.

Flavia. The Flavia is a leaf recognition system [130]. Here we call the leaf image dataset used for the system as the Flavia image dataset. The Flavia image dataset contains 1907 images from 32 species. Each image is a JPEG image in the size of 1600 $\times$ 1200 pixels. In the experiment, we resize each image to 30 $\times$ 40 pixels. The samples of the Flavia dataset are shown in Figure 11 (g).

B. EXPERIMENT SETTING

In the experiments, we applied the 13 LRC methods on 7 image datasets. We ran all the experiments on a PC with an Intel Core i7-6700 CPU and 16GB RAM. The software platform was Matlab 2018a. There are two parts to the experiments: parameter analysis and results. We will first perform parameter analysis to ensure the results shown in section V-C2 are the optimal for each classifier. Then, for each dataset, we show the results of different methods by increasing the number of training samples from each class. For each result in section V-C2, we ran it 10 times to calculate the standard deviation.

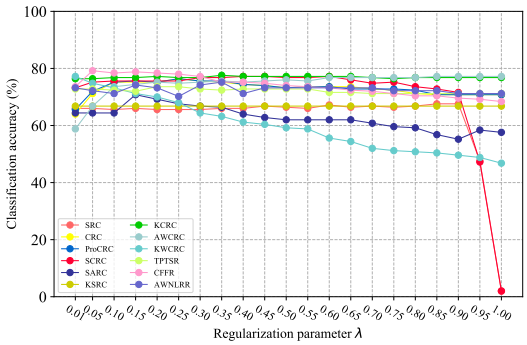
C. EXPERIMENTAL RESULTS

1) Parameter analysis

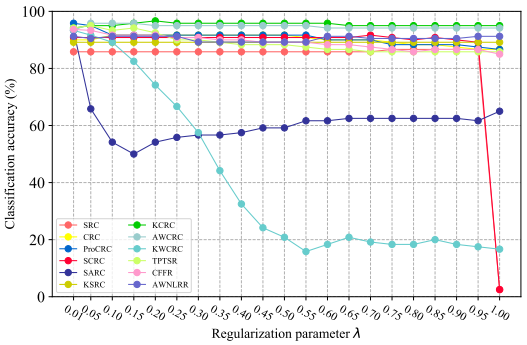
We first adjusted the regularization parameter λ used in the coding procedure of the LRC methods (refer to Eq. 6). The parameter analysis is performed on seven datasets: GT, ORL, AR, FEI, COIL20, Yale B, and Flavia. We set the number of training samples in each class as 10 (GT), 7 (ORL), 16 (AR), 8 (FEI), 20 (COIL20), 30 (Yale B), and 40 (Flavia) for the seven datasets. The parameter λ is selected within the list: $[0.01, 0.05, 0.1, 0.1, 0.2, 0.25, 0.3, 0.35, 0.4, 0.45, 0.5, 0.55, 0.6, 0.65, 0.7, 0.75, 0.8, 0.85, 0.9, 0.95, 1]$. Figure 12 shows the performances of each classifier under the different reg-

This work is licensed under a Creative Commons Attribution 4.0 License. For more information, see https://creativecommons.org/licenses/by/4.0/

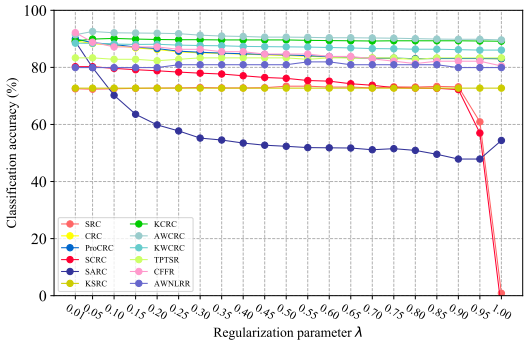
This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/ACCESS.2020.3041154, IEEE Access



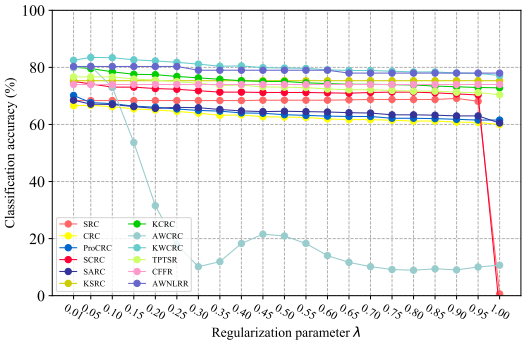
(a) GT.



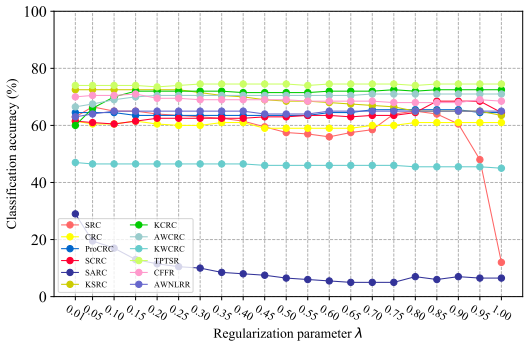
(b) ORL.



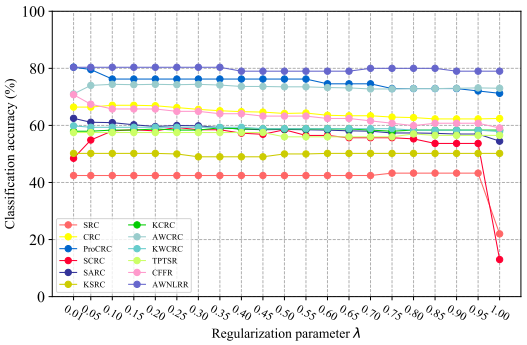
(c) AR.



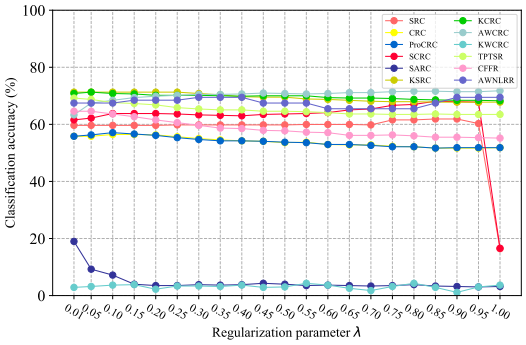
(d) FEI.



(e) COIL20.



(f) Yale B.



(g) Flavia.

FIGURE 12: Regularization parameter λ analysis on the different datasets.

This work is licensed under a Creative Commons Attribution 4.0 License. For more information, see https://creativecommons.org/licenses/by/4.0/

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/ACCESS.2020.3041154, IEEE Access

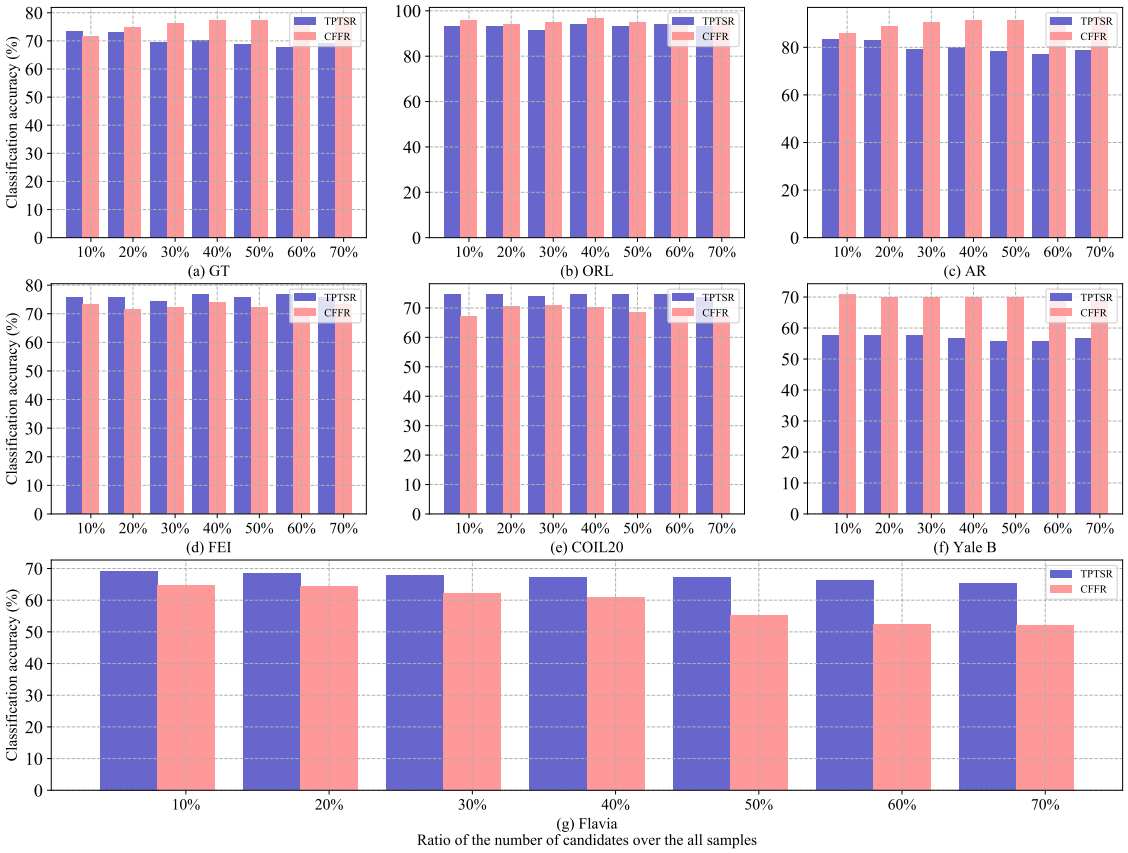


FIGURE 13: Parameter analysis of number of the candidates on the different datasets.

ularization parameter λ settings. The NRC is not in the analysis since its λ is set to zero (see Eq. 36). It is clearly seen that SRC will have sharp drops when the value of λ approaches 1. Some classifiers may not be stable when testing on different datasets. For example, the accuracy of KWCRC decreased rapidly on the ORL dataset and FEI dataset (see Figure 12 (b) and (d)), and SARC achieved a low accuracy in the COIL20 dataset. For the face database, the best accuracies obtained by the different classifiers are relatively close to each other. The standard deviation of the best accuracies for the GT and ORL datasets are 4.08% and 3.07%, respectively. For the object dataset, the best accuracies attained by different classifiers are relatively large. The standard deviations of the best accuracies in the COIL20 and Flavia dataset are 13.62% and 6.23%, respectively.

Besides the regularization parameter λ, we adjusted the number of candidates in the whole training set for TPTSR and CFFR methods. For TPTSR, the candidates are the samples and for the CFFR, the candidates are the classes. We changed the ratio of candidates over the whole training samples to obtain the optimal performance of these two methods. We selected the optimal λ based on the results of Figure 12. The performances of these two methods under the different candidates' ratios are shown in Figure 13. We can observe that for different datasets, better performances are achieved by different methods. TPTSR showed a stable performance on all seven datasets. CFFR had larger fluctuations on the GT and Flavia datasets (see Figure 13 (a), (g). For TPTSR, the best ratio of candidates was within 10% to 20%, while for CFFR, the best ratio of candidates is within 40% to 70%.

2) Results

Now that the optimal parameters have been selected for the LRC methods, we next show its recognition rates on 7 datasets with an increasing number of training samples. Table 2 illustrates the experimental results, with Figure 14 the standard deviations of the LRC methods on different datasets. We can observe that the LRC methods showed diverse standard deviations for these datasets, indicating the stability of a certain method depends on the data being

This work is licensed under a Creative Commons Attribution 4.0 License. For more information, see https://creativecommons.org/licenses/by/4.0/

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/ACCESS.2020.3041154, IEEE Access

Dataset	SRC	CRC	NRC	ProCRC	SCRC	SARC	TPTSR	CFFR	KSRC	KCRC	AWCRC	KWCRC	AWNLRR
GT (4)	48.91	54.55	48.18	**54.45**	56.73	49.09	54.55	61.45	48.91	58.00	57.64	56.36	60.34
GT (6)	60.44	61.78	56.44	61.87	67.56	56.67	63.33	68.22	59.33	**70.22**	68.44	65.56	61.78
GT (8)	67.43	68.29	64.29	69.21	72.57	60.57	68.00	73.14	65.43	75.71	**74.86**	73.14	69.43
GT(10)	67.64	75.22	77.60	75.63	76.88	64.47	72.84	**77.61**	66.83	77.60	76.00	77.21	56.36
AR (4)	57.08	67.35	54.80	66.97	64.39	67.50	56.36	67.20	57.05	65.53	**69.39**	75.20	68.93
AR (8)	57.69	69.49	54.25	71.39	64.31	71.11	58.19	69.91	58.19	69.54	**71.57**	68.75	69.30
AR (12)	63.75	77.56	59.41	76.55	70.83	75.36	65.18	78.21	64.11	75.60	**77.86**	73.15	78.27
AR (16)	72.67	89.42	65.91	91.92	80.25	90.42	83.33	**92.17**	71.25	90.00	91.08	89.08	81.93
AR (20)	71.53	91.39	68.44	**91.94**	76.67	91.11	75.00	91.94	67.92	88.33	90.56	89.44	88.35
ORL (5)	80.50	88.50	92.00	**92.50**	86.00	91.50	86.00	89.00	85.50	91.00	91.5	90.00	87.50
ORL (6)	81.88	91.88	**94.37**	93.13	88.13	91.88	86.88	90.00	86.25	90.63	91.88	91.88	89.00
ORL (7)	86.67	91.67	**95.83**	95.83	90.83	93.33	93.33	93.33	89.17	**95.83**	95.00	93.33	91.25
ORL (8)	82.50	90.00	95.00	**96.25**	88.75	88.75	90.00	92.50	87.50	**96.25**	92.50	92.50	91.67
COIL20 (5)	48.80	56.40	54.80	55.20	55.20	34.20	54.20	55.40	53.20	54.40	**57.00**	47.80	55.60
COIL20 (10)	50.25	52.75	54.25	54.25	54.25	25.00	**56.75**	54.00	55.25	53.75	56.50	42.25	54.00
COIL20 (15)	55.67	57.33	58.00	58.33	60.33	27.33	**65.67**	61.67	61.00	60.00	62.67	43.00	58.00
COIL20 (20)	64.00	61.50	62.50	64.50	68.50	29.00	**74.00**	70.50	72.50	72.00	70.50	47.00	65.00
FEI (5)	46.94	39.83	43.78	41.11	50.06	40.11	50.89	46.22	52.39	**54.50**	52.33	54.00	41.22
FEI (6)	63.44	50.13	66.70	52.13	65.56	51.06	67.13	59.75	66.81	69.06	67.75	**72.63**	64.73
FEI (7)	69.07	61.29	77.63	64.79	72.86	63.71	76.21	70.29	73.79	78.21	77.50	**81.57**	76.55
FEI (8)	69.33	66.58	79.35	70.17	75.08	68.50	76.67	74.00	75.33	80.08	80.42	**83.50**	80.32
YALE B (15)	45.68	63.72	**76.32**	74.50	60.15	69.36	56.39	66.42	53.20	63.72	68.55	62.16	75.42
YALE B (20)	42.11	62.66	**76.46**	74.96	57.97	69.42	56.40	65.22	50.36	63.30	67.57	67.67	76.40
YALE B (25)	43.75	64.80	**79.11**	75.58	56.25	67.60	61.02	67.27	49.01	60.53	69.33	59.95	78.22
YALE B (30)	43.27	67.06	**81.34**	80.41	52.48	62.48	57.50	70.76	50.19	59.16	74.27	59.84	80.37
YALE B (35)	37.20	67.99	**81.77**	79.31	58.75	65.55	61.48	70.57	44.02	58.01	75.24	60.29	81.32
Flavia (20)	52.33	50.99	54.61	51.22	53.20	22.81	60.46	54.22	60.30	61.01	**61.09**	54.62	52.96
Flavia (25)	55.47	49.41	56.83	52.12	54.20	22.31	**64.77**	52.94	**64.32**	62.51	63.41	51.58	65.42
Flavia (30)	43.27	67.06	81.77	80.41	58.48	62.48	57.50	70.76	65.36	**65.79**	65.47	50.58	54.62
Flavia (35)	59.34	52.73	65.55	52.86	59.72	24.52	70.39	55.86	**71.41**	70.27	70.65	50.49	69.47

TABLE 2: The performances of different LRC methods on different datasets.

FIGURE 14: The standard deviations of LRC methods in different datasets. (a) GT, (b) AR, (c) ORL, (d) COIL, (e) FEI, (f) YALE B, (g) Flavia.

This work is licensed under a Creative Commons Attribution 4.0 License. For more information, see https://creativecommons.org/licenses/by/4.0/

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/ACCESS.2020.3041154, IEEE Access

classified. The highest value of each dataset using a certain number of training samples in each class is marked in bold font. Generally speaking, there is no classifier that achieved the highest accuracy on all datasets. For the GT database, CFFR achieved the best accuracy of 77.61% by using 10 training samples. For the AR database, ProCRC is the best classifier with an accuracy of 91.94% using 20 samples. In the ORL database, both ProCRC and KCRC were the best classifiers with the same highest accuracy of 96.25% using 8 samples per class. However, the standard deviation of KCRC (0.02) is lower than ProCRC (0.86), which is considered as a better classifier due to its stronger classification stability. For the COIL20 dataset, TPTSR achieved the highest accuracy of 74% when using 20 training samples per class. In the the FEI dataset, KWCRC was the best classifier with an accuracy of 83.5% when using 8 training samples per class. As for the Yale B database, the highest accuracy was obtained by NRC with an accuracy of 81.77% using 30 samples. KSRC produced the highest accuracy of 71.41% by using 35 samples per class on the Flavia database. It should be pointed out that some classifiers were able to achieve the highest accuracy on one dataset using any number of the training samples, showing its superiority over other classifiers. For example, according to Table 2, NRC achieved the highest accuracies on the YaleB database using the number of training samples per class ranging from 15 samples to 35 samples per class. KWCRC obtained the highest accuracies on the FEI dataset using the number of training samples per class ranging from 6 to 8, according to Table 2. TPTSR achieved the highest accuracies on the COIL20 dataset using the number of training samples ranging from 10 samples to 20 samples per class. Among all classifiers, NRC had the 7 highest accuracies on the different datasets. Also, we can observe that the weighted strategy effectively enhances the classifier's performance. For example, AWCRC held the 6 best accuracies compared with others. Similarly, the kernel version of the LRC methods also showed its effectiveness in the enhancement. KSRC and KCRC achieved 2 and 5 of the highest accuracies in the comparison. For the object recognition task (COIL20 and Flavia), the gap between the highest accuracy and the lowest accuracy was relatively larger. The standard deviation of accuracies with different classifiers in COIL20 and Flavia were 12.98% and 13.27%, respectively.

D. DISCUSSION

We can discuss the following items in terms of the experiments from the previous sections:

The regularization parameter of the LRC methods will influence the accuracy of image classification. A change in accuracy ranging from 3.07%-4.08% was caused on the face dataset, and a change of 6.23%-13.62% in accuracy occurred on the object dataset. Besides, different classifiers will be influenced by different extents. SRC will have its accuracy rapidly decreasing when the regularization parameter is approaching 1. SARC will be unstable when the regularization parameter changes. However, for the other classifiers, the influence caused by the parameter affects the performs less.

There is no one LRC method dominating over the other methods in image classification. The highest accuracies achieved in Table 2 are distributed in different cases. The LRC method with the highest number of accuracies is NRC. Furthermore, there are few LRC methods that achieved the best accuracy among all training samples per class (except for NRC in the Yale B database), indicating some methods cannot ensure the best classification ability when given insufficient information/samples even though it achieved the highest accuracy in this dataset. Since different classifiers have different image space assumptions in the image space, its recognition abilities will be more apparent when a specific assumption is met in the real classification scenario.

The weighting strategy and kernel extension of the LRC methods can truly enhance the performance. AWCRC, as an extension of the CRC using a weighting strategy, outperforms CRC on almost all the cases (except for the AR dataset using 20 training samples per class). Moreover, the AWCRC shows its superiority in classifying using insufficient information/samples on GT, AR, and Flavia datasets. The additional attention to the critical samples in the representation brings the enhancement of the weighting strategy. For the kernel extension, both KSRC and KCRC achieved a better classification performance than its original version on the majority of cases. This is because the kernel extension has the ability to process in the non-linear space. However, KWCRC, which is the kernel extension of Weighted CRC, does not outperform KCRC in most cases, indicating there is no accumulated enhancement when imposing both kernel extension and a weighting strategy on the LRC methods.

The sample removing strategy also shows its effectiveness according to the experiments. As typical LRC methods that are based on the sample removing strategy, TPTSR and CFFR showed better performance in some cases. TPTSR had a better classification result on object recognition, which achieved 4 of the highest accuracies (COIL20 (15, 20, 25), Flavia (25)). For CFFR, it had a better performance on the face recognition task, where 2 of the highest accuracies (GT (10) and AR (16)) were achieved. However, both of these methods are influenced heavily by the number of candidates. According to Figure 13, in the Flavia dataset using 40 training samples per class, the accuracy of CFFR changed from 52.15% (70% of all classes) to 64.59% (10% of all classes). Similarly, in TPTSR, it had the largest accuracy gap between the highest accuracy and the lowest accuracy of 6% on the GT database using 7 training samples per class. These phenomena were related to the sparsity in the representation: the number of candidate settings should ensure sufficient sparsity in the representation to guarantee its classification ability.

LRC shows different properties in different image classification tasks. For the face recognition tasks, different methods achieved relatively similar performances with lower standard deviations. In contrast, in the object recognition task, the performance gap is larger. This maybe because one object

This work is licensed under a Creative Commons Attribution 4.0 License. For more information, see https://creativecommons.org/licenses/by/4.0/

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/ACCESS.2020.3041154, IEEE Access

in the image set usually has many views, which makes the object subspace more complicated than the face subspace in face recognition.

The LRC methods still suffer from a lack of sufficient information/samples when performing image classification. By observing Table 2, there are still large gaps between the most training samples per class used and the least training samples per class used, implying that the number of training samples per class used significantly impacts the performance of LRC methods. For example, in the FEI dataset, SRC achieved an accuracy of 64% when using 8 training samples per class, while it only achieved 46.94% accuracy when using 5 training samples per class. The difference of 3 samples per class brought a 17.06% reduction in accuracy. The corresponding interpretation for it can be: insufficient samples have a higher probability of failing to construct a fine class-specific subspace.

Based on previous works and the above discussion, here, we point out the challenges and potential future research points of linear representation for image classification:

1) The LRC methods still require sufficient data in each class when performing classification.
2) The LRC methods will achieve a poor performance when the dataset is highly imbalanced.
3) The parameters in the LRC methods still have a large impact on its performance.
4) It is necessary to develop an efficient algorithm for LRC methods to process high-dimensional data.
5) t is necessary to investigate the fusion among properties of the coefficients, such as sparsity, collaboration, and non-negativity.

VI. CONCLUSION

This survey reviewed the linear representation-based classification methods for image classification, termed as LRC methods. We provided a clear definition of the LRC methods and summarized them in a specific algorithm. The various LRC methods can be categorized into 6 classes: 1) linear representation-based classification methods with norm minimizations, 2) linear representation-based classification methods with constraints, 3) linear representation-based classification methods with feature spaces, 4) linear representation-based classification methods with structural information, 5) linear representation with subspace learning, and 6) linear representation for semi-supervised learning. Moreover, we discussed three application areas in image classification that extensively apply the LRC methods. Finally, we performed comprehensive experiments on 7 image datasets to analyze and show the performances of different LRC methods.

ACKNOWLEDGMENT

This document is the result of a research project funded by the University of Macau (MYRG2019-00006-FST).

REFERENCES

[1] E. Alpaydin, *Introduction to machine learning*. MIT press, 2020.
[2] P. M. Mather and M. Koch, *Computer processing of remotely-sensed images: an introduction*. John Wiley & Sons, 2011.
[3] K. A. Gates, *Our biometric future: Facial recognition technology and the culture of surveillance*. NYU Press, 2011, vol. 2.
[4] A. K. Jain and S. Z. Li, *Handbook of face recognition*. Springer, 2011, vol. 1.
[5] Y. LeCun, L. Bottou, Y. Bengio, and P. Haffner, "Gradient-based learning applied to document recognition," *Proceedings of the IEEE*, vol. 86, no. 11, pp. 2278–2324, 1998.
[6] S. Shekhar, V. M. Patel, N. M. Nasrabadi, and R. Chellappa, "Joint sparse representation for robust multimodal biometrics recognition," *IEEE Transactions on pattern analysis and machine intelligence*, vol. 36, no. 1, pp. 113–126, 2013.
[7] G. Goswami, P. Mittal, A. Majumdar, M. Vatsa, and R. Singh, "Group sparse representation based classification for multi-feature multimodal biometrics," *Information Fusion*, vol. 32, pp. 3–12, 2016.
[8] B. Zhang, D. Zhang *et al.*, "Noninvasive diabetes mellitus detection using facial block color with a sparse representation classifier," *IEEE transactions on biomedical engineering*, vol. 61, no. 4, pp. 1027–1033, 2013.
[9] J. Wright, Y. Ma, J. Mairal, G. Sapiro, T. S. Huang, and S. Yan, "Sparse representation for computer vision and pattern recognition," *Proceedings of the IEEE*, vol. 98, no. 6, pp. 1031–1044, 2010.
[10] L. Zhang, M. Yang, and X. Feng, "Sparse representation or collaborative representation: Which helps face recognition?" in *2011 International conference on computer vision*. IEEE, 2011, pp. 471–478.
[11] T. Shu, B. Zhang, and Y. Y. Tang, "Sparse supervised representation-based classifier for uncontrolled and imbalanced classification," *IEEE transactions on neural networks and learning systems*, 2018.
[12] J. Xu, W. An, L. Zhang, and D. Zhang, "Sparse, collaborative, or nonnegative representation: which helps pattern classification?" *Pattern Recognition*, vol. 88, pp. 679–688, 2019.
[13] K.-C. Lee, J. Ho, and D. J. Kriegman, "Acquiring linear subspaces for face recognition under variable lighting," *IEEE Transactions on pattern analysis and machine intelligence*, vol. 27, no. 5, pp. 684–698, 2005.
[14] J. Wright, A. Y. Yang, A. Ganesh, S. S. Sastry, and Y. Ma, "Robust face recognition via sparse representation," *IEEE transactions on pattern analysis and machine intelligence*, vol. 31, no. 2, pp. 210–227, 2008.
[15] Y. Xu, D. Zhang, J. Yang, and J.-Y. Yang, "A two-phase test sample sparse representation method for use with face recognition," *IEEE Transactions on circuits and systems for video technology*, vol. 21, no. 9, pp. 1255–1262, 2011.
[16] S. Cai, L. Zhang, W. Zuo, and X. Feng, "A probabilistic collaborative representation based approach for pattern classification," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2016, pp. 2950–2959.
[17] J. Lai and X. Jiang, "Classwise sparse and collaborative patch representation for face recognition," *IEEE Transactions on Image Processing*, vol. 25, no. 7, pp. 3261–3272, 2016.
[18] C. Tian, Q. Zhang, G. Sun, Z. Song, and S. Li, "Fft consolidated sparse and collaborative representation for image classification," *Arabian Journal for Science and Engineering*, vol. 43, no. 2, pp. 741–758, 2018.
[19] C.-P. Wei, Y.-W. Chao, Y.-R. Yeh, and Y.-C. F. Wang, "Locality-sensitive dictionary learning for sparse representation based classification," *Pattern Recognition*, vol. 46, no. 5, pp. 1277–1287, 2013.
[20] X. Peng, L. Zhang, Z. Yi, and K. K. Tan, "Learning locality-constrained collaborative representation for robust face recognition," *Pattern Recognition*, vol. 47, no. 9, pp. 2794–2806, 2014.
[21] J. Wang, J. Yang, K. Yu, F. Lv, T. Huang, and Y. Gong, "Locality-constrained linear coding for image classification," in *2010 IEEE computer society conference on computer vision and pattern recognition*. IEEE, 2010, pp. 3360–3367.
[22] Y. Liu, F. Wu, Z. Zhang, Y. Zhuang, and S. Yan, "Sparse representation using nonnegative curds and whey," in *2010 IEEE Computer Society Conference on Computer Vision and Pattern Recognition*. IEEE, 2010, pp. 3578–3585.
[23] Y. Xu, Z. Zhang, G. Lu, and J. Yang, "Approximately symmetrical face images for image preprocessing in face recognition and sparse representation based classification," *Pattern Recognition*, vol. 54, pp. 68–82, 2016.
[24] H. Zhang, Y. Zhang, and T. S. Huang, "Pose-robust face recognition via sparse representation," *Pattern Recognition*, vol. 46, no. 5, pp. 1511–1521, 2013.

This work is licensed under a Creative Commons Attribution 4.0 License. For more information, see https://creativecommons.org/licenses/by/4.0/

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/ACCESS.2020.3041154, IEEE Access

[25] Y. Liu, Q. Gao, J. Han, and S. Wang, "Euler sparse representation for image classification," in *AAAI*, 2018.

[26] Y. Qin and C. Tian, "Weighted feature space representation with kernel for image classification," *Arabian Journal for Science and Engineering*, vol. 43, no. 12, pp. 7113–7125, 2018.

[27] G. Liu, Z. Lin, S. Yan, J. Sun, Y. Yu, and Y. Ma, "Robust recovery of subspace structures by low-rank representation," *IEEE transactions on pattern analysis and machine intelligence*, vol. 35, no. 1, pp. 171–184, 2012.

[28] Y. Zhang, Z. Jiang, and L. S. Davis, "Learning structured low-rank representations for image classification," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2013, pp. 676–683.

[29] B. Wohlberg, "Efficient algorithms for convolutional sparse representations," *IEEE Transactions on Image Processing*, vol. 25, no. 1, pp. 301–315, 2015.

[30] G.-J. Peng, "Joint and direct optimization for dictionary learning in convolutional sparse representation," *IEEE Transactions on Neural Networks and Learning Systems*, vol. 31, no. 2, pp. 559–573, 2019.

[31] S. Zhang, H. Wang, and W. Huang, "Two-stage plant species recognition by local mean clustering and weighted sparse representation classification," *Cluster computing*, vol. 20, no. 2, pp. 1517–1525, 2017.

[32] X. Liu, A. Srivastava, and K. Gallivan, "Optimal linear representations of images for object recognition," in *2003 IEEE Computer Society Conference on Computer Vision and Pattern Recognition, 2003. Proceedings.*, vol. 1. IEEE, 2003, pp. I–I.

[33] Y. Gao, J. Ma, and A. L. Yuille, "Semi-supervised sparse representation based classification for face recognition with insufficient labeled samples," *IEEE Transactions on Image Processing*, vol. 26, no. 5, pp. 2545–2560, 2017.

[34] W. Deng, J. Hu, and J. Guo, "Extended src: Undersampled face recognition via intraclass variant dictionary," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 34, no. 9, pp. 1864–1870, 2012.

[35] J. Wen, B. Zhang, Y. Xu, J. Yang, and N. Han, "Adaptive weighted nonnegative low-rank representation," *Pattern Recognition*, vol. 81, pp. 326–340, 2018.

[36] J. Wen, N. Han, X. Fang, L. Fei, K. Yan, and S. Zhan, "Low-rank preserving projection via graph regularized reconstruction," *IEEE Transactions on Cybernetics*, vol. 49, no. 4, pp. 1279–1291, 2018.

[37] H. Liang and Q. Li, "Hyperspectral imagery classification using sparse representations of convolutional neural network features," *Remote Sensing*, vol. 8, no. 2, p. 99, 2016.

[38] T. Shu, B. Zhang, and Y. Y. Tang, "An improved noninvasive method to detect diabetes mellitus using the probabilistic collaborative representation based classifier," *Information Sciences*, vol. 467, pp. 477–488, 2018.

[39] T. V. Pham and A. W. Smeulders, "Sparse representation for coarse and fine object recognition," *IEEE transactions on pattern analysis and machine intelligence*, vol. 28, no. 4, pp. 555–567, 2006.

[40] Y. Xu, W. Zuo, and Z. Fan, "Supervised sparse representation method with a heuristic strategy and face recognition experiments," *Neurocomputing*, vol. 79, pp. 125–131, 2012.

[41] P. Zhu, L. Zhang, Q. Hu, and S. C. Shiu, "Multi-scale patch based collaborative representation for face recognition with margin distribution optimization," in *European Conference on Computer Vision*. Springer, 2012, pp. 822–835.

[42] Y. Chi and F. Porikli, "Connecting the dots in multi-class classification: From nearest subspace to collaborative representation," in *2012 IEEE Conference on Computer Vision and Pattern Recognition*. IEEE, 2012, pp. 3602–3609.

[43] D. L. Donoho, "For most large underdetermined systems of linear equations the minimal l1-norm solution is also the sparsest solution," *Communications on Pure and Applied Mathematics: A Journal Issued by the Courant Institute of Mathematical Sciences*, vol. 59, no. 6, pp. 797–829, 2006.

[44] Z. Zhang, Y. Xu, J. Yang, X. Li, and D. Zhang, "A survey of sparse representation: algorithms and applications," *IEEE access*, vol. 3, pp. 490–530, 2015.

[45] Y. C. Pati, R. Rezaiifar, and P. S. Krishnaprasad, "Orthogonal matching pursuit: Recursive function approximation with applications to wavelet decomposition," in *Proceedings of 27th Asilomar conference on signals, systems and computers*. IEEE, 1993, pp. 40–44.

[46] A. Beck and M. Teboulle, "A fast iterative shrinkage-thresholding algorithm for linear inverse problems," *SIAM journal on imaging sciences*, vol. 2, no. 1, pp. 183–202, 2009.

[47] J. Heinonen, *Lectures on Lipschitz analysis*. University of Jyväskylä, 2005, no. 100.

[48] C.-Y. Lu, H. Min, J. Gui, L. Zhu, and Y.-K. Lei, "Face recognition via weighted sparse representation," *Journal of Visual Communication and Image Representation*, vol. 24, no. 2, pp. 111–116, 2013.

[49] Z. Fan, M. Ni, Q. Zhu, and E. Liu, "Weighted sparse representation for face recognition," *Neurocomputing*, vol. 151, pp. 304–309, 2015.

[50] R. Timofte and L. Van Gool, "Weighted collaborative representation and classification of images," in *Proceedings of the 21st International Conference on Pattern Recognition (ICPR2012)*. IEEE, 2012, pp. 1606–1610.

[51] R. Timofte and L. Van Gool, "Adaptive and weighted collaborative representations for image classification," *Pattern Recognition Letters*, vol. 43, pp. 127–135, 2014.

[52] A. Majumdar and R. K. Ward, "Classification via group sparsity promoting regularization," in *2009 IEEE International Conference on Acoustics, Speech and Signal Processing*. IEEE, 2009, pp. 861–864.

[53] E. van den Berg and M. P. Friedlander, "Spgl1: A solver for large-scale sparse reconstruction," 2007.

[54] D. D. Lee and H. S. Seung, "Learning the parts of objects by non-negative matrix factorization," *Nature*, vol. 401, no. 6755, pp. 788–791, 1999.

[55] R. Bro and S. De Jong, "A fast non-negativity-constrained least squares algorithm," *Journal of Chemometrics: A Journal of the Chemometrics Society*, vol. 11, no. 5, pp. 393–401, 1997.

[56] S. Boyd, N. Parikh, and E. Chu, *Distributed optimization and statistical learning via the alternating direction method of multipliers*. Now Publishers Inc, 2011.

[57] N. Akhtar, F. Shafait, and A. Mian, "Efficient classification with sparsity augmented collaborative representation," *Pattern Recognition*, vol. 65, pp. 136–145, 2017.

[58] C. Tian, Y. Xu, Z. Li, W. Zuo, L. Fei, and H. Liu, "Attention-guided cnn for image denoising," *Neural Networks*, vol. 124, pp. 117–129, 2020.

[59] C. Tian, L. Fei, W. Zheng, Y. Xu, W. Zuo, and C.-W. Lin, "Deep learning on image denoising: An overview," *Neural Networks*, 2020.

[60] Y. Xu, Q. Zhu, Z. Fan, D. Zhang, J. Mi, and Z. Lai, "Using the idea of the sparse representation to perform coarse-to-fine face recognition," *Information sciences*, vol. 238, pp. 138–148, 2013.

[61] H. Sun and Q. Wu, "Sparse representation in kernel machines," *IEEE Transactions on Neural Networks and Learning Systems*, vol. 26, no. 10, pp. 2576–2582, 2015.

[62] I. Goodfellow, Y. Bengio, A. Courville, and Y. Bengio, *Deep learning*. MIT press Cambridge, 2016, vol. 1.

[63] C. Cortes and V. Vapnik, "Support-vector networks," *Machine learning*, vol. 20, no. 3, pp. 273–297, 1995.

[64] L. Zhang, W.-D. Zhou, P.-C. Chang, J. Liu, Z. Yan, T. Wang, and F.-Z. Li, "Kernel sparse representation-based classifier," *IEEE Transactions on Signal Processing*, vol. 60, no. 4, pp. 1684–1695, 2011.

[65] J. Yin, Z. Liu, Z. Jin, and W. Yang, "Kernel sparse representation based classification," *Neurocomputing*, vol. 77, no. 1, pp. 120–128, 2012.

[66] B. Wang, W. Li, N. Poh, and Q. Liao, "Kernel collaborative representation-based classifier for face recognition," in *2013 IEEE International Conference on Acoustics, Speech and Signal Processing*. IEEE, 2013, pp. 2877–2881.

[67] K. Simonyan and A. Zisserman, "Very deep convolutional networks for large-scale image recognition," *arXiv preprint arXiv:1409.1556*, 2014.

[68] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2016, pp. 770–778.

[69] C. Szegedy, W. Liu, Y. Jia, P. Sermanet, S. Reed, D. Anguelov, D. Erhan, V. Vanhoucke, and A. Rabinovich, "Going deeper with convolutions," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2015, pp. 1–9.

[70] S. J. Pan and Q. Yang, "A survey on transfer learning," *IEEE Transactions on knowledge and data engineering*, vol. 22, no. 10, pp. 1345–1359, 2009.

[71] S. Zeng, B. Zhang, Y. Zhang, and J. Gou, "Collaboratively weighting deep and classic representation via l2 regularization for image classification," *arXiv preprint arXiv:1802.07589*, 2018.

[72] E.-J. Cheng, K.-P. Chou, S. Rajora, B.-H. Jin, M. Tanveer, C.-T. Lin, K.-Y. Young, W.-C. Lin, and M. Prasad, "Deep sparse representation classifier for facial recognition and detection system," *Pattern Recognition Letters*, vol. 125, pp. 71–77, 2019.

This work is licensed under a Creative Commons Attribution 4.0 License. For more information, see https://creativecommons.org/licenses/by/4.0/

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/ACCESS.2020.3041154, IEEE Access

[73] J. Zhou, S. Zeng, and B. Zhang, "Two-stage knowledge transfer framework for image classification," *Pattern Recognition*, vol. 107, p. 107529, 2020.

[74] A. J. Fitch, A. Kadyrov, W. J. Christmas, and J. Kittler, "Fast robust correlation," *IEEE Transactions on Image Processing*, vol. 14, no. 8, pp. 1063–1073, 2005.

[75] J. Yang, D. Chu, L. Zhang, Y. Xu, and J. Yang, "Sparse representation classifier steered discriminative projection with applications to face recognition," *IEEE transactions on neural networks and learning systems*, vol. 24, no. 7, pp. 1023–1035, 2013.

[76] M. E. Fathy, A. Alavi, and R. Chellappa, "Discriminative log-euclidean feature learning for sparse representation-based recognition of faces from videos." in *IJCAI*, 2016, pp. 3359–3367.

[77] Q. Feng and Y. Zhou, "Discriminant projection representation-based classification for vision recognition," *arXiv preprint arXiv:1712.01643*, 2017.

[78] Z. Liu, K. Shi, K. Zhang, W. Ou, and L. Wang, "Discriminative sparse embedding based on adaptive graph for dimension reduction," *Engineering Applications of Artificial Intelligence*, vol. 94, p. 103758, 2020.

[79] Z. Liu, W. Ou, W. Lu, and L. Wang, "Discriminative feature extraction based on sparse and low-rank representation," *Neurocomputing*, vol. 362, pp. 129–138, 2019.

[80] J. Wang, C. Lu, M. Wang, P. Li, S. Yan, and X. Hu, "Robust face recognition via adaptive sparse representation," *IEEE transactions on cybernetics*, vol. 44, no. 12, pp. 2368–2378, 2014.

[81] J. Chen, J. Yang, L. Luo, J. Qian, and W. Xu, "Matrix variate distribution-induced sparse representation for robust image classification," *IEEE transactions on neural networks and learning systems*, vol. 26, no. 10, pp. 2291–2300, 2015.

[82] Z. Xu, Y. Liu, M. Ye, L. Huang, H. Yu, and X. Chen, "Patch based collaborative representation with gabor feature and measurement matrix for face recognition," *Mathematical Problems in Engineering*, vol. 2018, 2018.

[83] S. Gao, K. Jia, L. Zhuang, and Y. Ma, "Neither global nor local: Regularized patch-based representation for single sample per person face recognition," *International Journal of Computer Vision*, vol. 111, no. 3, pp. 365–383, 2015.

[84] Y. Chi and F. Porikli, "Classification and boosting with multiple collaborative representations," *IEEE transactions on pattern analysis and machine intelligence*, vol. 36, no. 8, pp. 1519–1531, 2013.

[85] R. Kohavi *et al.*, "A study of cross-validation and bootstrap for accuracy estimation and model selection," in *Ijcai*, vol. 14, no. 2. Montreal, Canada, 1995, pp. 1137–1145.

[86] Q. Feng, Y. Zhou, and R. Lan, "Pairwise linear regression classification for image set retrieval," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2016, pp. 4865–4872.

[87] C.-G. Li, J. Guo, and H.-G. Zhang, "Local sparse representation based classification," in *2010 20th International Conference on Pattern Recognition*. IEEE, 2010, pp. 649–652.

[88] X. Zhu and A. B. Goldberg, "Introduction to semi-supervised learning," *Synthesis lectures on artificial intelligence and machine learning*, vol. 3, no. 1, pp. 1–130, 2009.

[89] S. Zeng, B. Zhang, Y. Zhang, and J. Gou, "Dual sparse learning via data augmentation for robust facial image classification," *International Journal of Machine Learning and Cybernetics*, pp. 1–18, 2020.

[90] S. Zeng, B. Zhang, and J. Gou, "Learning double weights via data augmentation for robust sparse and collaborative representation-based classification," *Multimedia Tools and Applications*, pp. 1–22, 2020.

[91] J. Gou, L. Wang, Z. Yi, J. Lv, Q. Mao, and Y.-H. Yuan, "A new discriminative collaborative neighbor representation method for robust face recognition," *IEEE Access*, vol. 6, pp. 74 713–74 727, 2018.

[92] D. A. Reynolds, "Gaussian mixture models." *Encyclopedia of biometrics*, vol. 741, 2009.

[93] F. Dornaika, "Active two phase collaborative representation classifier," *ACM Transactions on Knowledge Discovery from Data (TKDD)*, vol. 13, no. 4, pp. 1–10, 2019.

[94] C. C. Bonwell and J. A. Eison, *Active Learning: Creating Excitement in the Classroom. 1991 ASHE-ERIC Higher Education Reports.* ERIC, 1991.

[95] M. Yin, S. Cai, and J. Gao, "Robust face recognition via double low-rank matrix recovery for feature extraction," in *2013 IEEE International Conference on Image Processing*. IEEE, 2013, pp. 3770–3774.

[96] G. Liu and S. Yan, "Latent low-rank representation for subspace segmentation and feature extraction," in *2011 international conference on computer vision*. IEEE, 2011, pp. 1615–1622.

[97] C.-I. Chang, *Hyperspectral imaging: techniques for spectral detection and classification*. Springer Science & Business Media, 2003, vol. 1.

[98] H. Grahn and P. Geladi, *Techniques and applications of hyperspectral image analysis*. John Wiley & Sons, 2007.

[99] R. L. Lawrence, S. D. Wood, and R. L. Sheley, "Mapping invasive plants using hyperspectral imagery and breiman cutler classifications (randomforest)," *Remote Sensing of Environment*, vol. 100, no. 3, pp. 356–362, 2006.

[100] J. G. Ferwerda, "Charting the quality of forage: measuring and mapping the variation of chemical components in foliage with hyperspectral remote sensing." ITC, 2005.

[101] D. Manolakis, D. Marden, G. A. Shaw *et al.*, "Hyperspectral image processing for automatic target detection applications," *Lincoln laboratory journal*, vol. 14, no. 1, pp. 79–116, 2003.

[102] M. F. Baumgardner, L. L. Biehl, and D. A. Landgrebe, "220 band aviris hyperspectral image data set: June 12, 1992 indian pine test site 3," Sep 2015. [Online]. Available: https://purr.purdue.edu/publications/1947/1

[103] Y. Chen, N. M. Nasrabadi, and T. D. Tran, "Hyperspectral image classification using dictionary-based sparse representation," *IEEE transactions on geoscience and remote sensing*, vol. 49, no. 10, pp. 3973–3985, 2011.

[104] Y. Chen, N. M. Nasrabadi, and T. D. Tran, "Hyperspectral image classification via kernel sparse representation," *IEEE Transactions on Geoscience and Remote sensing*, vol. 51, no. 1, pp. 217–231, 2012.

[105] Y. Yuan, J. Lin, and Q. Wang, "Hyperspectral image classification via multitask joint sparse representation and stepwise mrf optimization," *IEEE transactions on cybernetics*, vol. 46, no. 12, pp. 2966–2977, 2015.

[106] J. Peng, W. Sun, and Q. Du, "Self-paced joint sparse representation for the classification of hyperspectral images," *IEEE Transactions on Geoscience and Remote Sensing*, vol. 57, no. 2, pp. 1183–1194, 2018.

[107] L. Jiang, D. Meng, Q. Zhao, S. Shan, and A. G. Hauptmann, "Self-paced curriculum learning." in *AAAI*, vol. 2, no. 5.4, 2015, p. 6.

[108] L. Fang, S. Li, X. Kang, and J. A. Benediktsson, "Spectral–spatial hyperspectral image classification via multiscale adaptive sparse representation," *IEEE Transactions on Geoscience and Remote Sensing*, vol. 52, no. 12, pp. 7738–7749, 2014.

[109] Y. Y. Tang, H. Yuan, and L. Li, "Manifold-based sparse representation for hyperspectral image classification," *IEEE Transactions on Geoscience and Remote Sensing*, vol. 52, no. 12, pp. 7606–7618, 2014.

[110] N. S. Altman, "An introduction to kernel and nearest-neighbor nonparametric regression," *The American Statistician*, vol. 46, no. 3, pp. 175–185, 1992.

[111] J. Li, H. Zhang, Y. Huang, and L. Zhang, "Hyperspectral image classification by nonlocal joint collaborative representation with a locally adaptive dictionary," *IEEE Transactions on Geoscience and Remote Sensing*, vol. 52, no. 6, pp. 3707–3719, 2013.

[112] S. Jia, L. Shen, and Q. Li, "Gabor feature-based collaborative representation for hyperspectral imagery classification," *IEEE Transactions on Geoscience and Remote Sensing*, vol. 53, no. 2, pp. 1118–1129, 2014.

[113] S. Jia, J. Hu, Y. Xie, L. Shen, X. Jia, and Q. Li, "Gabor cube selection based multitask joint sparse representation for hyperspectral image classification," *IEEE Transactions on Geoscience and Remote Sensing*, vol. 54, no. 6, pp. 3174–3187, 2016.

[114] H. Su, B. Zhao, Q. Du, P. Du, and Z. Xue, "Multifeature dictionary learning for collaborative representation classification of hyperspectral imagery," *IEEE Transactions on Geoscience and Remote Sensing*, vol. 56, no. 4, pp. 2467–2484, 2018.

[115] R. He, W.-S. Zheng, B.-G. Hu, and X.-W. Kong, "Two-stage nonnegative sparse representation for large-scale face recognition," *IEEE transactions on neural networks and learning systems*, vol. 24, no. 1, pp. 35–46, 2012.

[116] H. Zhang, N. M. Nasrabadi, Y. Zhang, and T. S. Huang, "Joint dynamic sparse representation for multi-view face recognition," *Pattern Recognition*, vol. 45, no. 4, pp. 1290–1298, 2012.

[117] D. Zhang, W. Zuo, and N. Li, *Medical biometrics: Computerized TCM data analysis*. World Scientific, 2016.

[118] B. Zhang, F. Karray, Q. Li, and L. Zhang, "Sparse representation classifier for microaneurysm detection and retinal blood vessel extraction," *Information Sciences*, vol. 200, pp. 78–90, 2012.

[119] B. Zhang, L. Zhang, J. You, and F. Karray, "Microaneurysm (ma) detection via sparse representation classifier with ma and non-ma dictionary learning," in *2010 20th International Conference on Pattern Recognition*. IEEE, 2010, pp. 277–280.

This work is licensed under a Creative Commons Attribution 4.0 License. For more information, see https://creativecommons.org/licenses/by/4.0/

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/ACCESS.2020.3041154, IEEE Access

[120] T. Shu, B. Zhang, and Y. Y. Tang, "Effective heart disease detection based on quantitative computerized traditional chinese medicine using representation based classifiers," *Evidence-Based Complementary and Alternative Medicine*, vol. 2017, 2017.
[121] J. Li, B. Zhang, and D. Zhang, "Joint discriminative and collaborative representation for fatty liver disease diagnosis," *Expert Systems with Applications*, vol. 89, pp. 31–40, 2017.
[122] G. Goswami, R. Singh, M. Vatsa, and A. Majumdar, "Kernel group sparse representation based classifier for multimodal biometrics," in *2017 International Joint Conference on Neural Networks (IJCNN)*. IEEE, 2017, pp. 2894–2901.
[123] S. Zhao, B. Zhang, and C. P. Chen, "Joint deep convolutional feature representation for hyperspectral palmprint recognition," *Information Sciences*, vol. 489, pp. 167–181, 2019.
[124] L. Chen, H. Man, and A. V. Nefian, "Face recognition based on multi-class mapping of fisher scores," *Pattern Recognition*, vol. 38, no. 6, pp. 799–811, 2005.
[125] A. M. Martinez, "The ar face database," *CVC Technical Report24*, 1998.
[126] F. S. Samaria and A. C. Harter, "Parameterisation of a stochastic model for human face identification," in *Proceedings of 1994 IEEE workshop on applications of computer vision*. IEEE, 1994, pp. 138–142.
[127] S. Nane, S. Nayar, and H. Murase, "Columbia object image library: Coil-20," *Dept. Comp. Sci., Columbia University, New York, Tech. Rep*, 1996.
[128] C. E. Thomaz and G. A. Giraldi, "A new ranking method for principal components analysis and its application to face image analysis," *Image and vision computing*, vol. 28, no. 6, pp. 902–913, 2010.
[129] 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 transactions on pattern analysis and machine intelligence*, vol. 23, no. 6, pp. 643–660, 2001.
[130] S. G. Wu, F. S. Bao, E. Y. Xu, Y.-X. Wang, Y.-F. Chang, and Q.-L. Xiang, "A leaf recognition algorithm for plant classification using probabilistic neural network," in *2007 IEEE international symposium on signal processing and information technology*. IEEE, 2007, pp. 11–16.
[131] S. Zeng, X. Yang, and J. Gou, "Multiplication fusion of sparse and collaborative representation for robust face recognition," *Multimedia Tools and Applications*, vol. 76, no. 20, pp. 20 889–20 907, 2017.

JIANHANG ZHOU (S'18) received the B.S. degree from Nanjing Forestry University (NJFU), Nanjing, China, in 2018, a M.S. degree in Computer Science from University of Macau in 2020. He is currently pursuing the Ph.D. degree in computer science in the Department of Computer and Information Science, Faculty of Science and Technology at the University of Macau.

His research interest includes image classification, pattern recognition, and medical biometrics.

SHAONING ZENG (M'18) received the B.S. degree and M.S. degree from Beihang University (BUAA), Beijing, China, in 2004 and 2007, respectively. He is currently pursuing the Ph.D. degree in computer science in the Department of Computer and Information Science, Faculty of Science and Technology at the University of Macau.

From 2009 to now, he is a Researcher and Lecturer in the School of Information Science and Technology at Huizhou University, China. His research interest includes computer vision, pattern recognition, machine learning and deep learning for multimedia and image processing applications. He has published over 10 scientific publications in these areas.

BOB ZHANG (M'11) received his B.A. in Computer Science from York University in 2006, a M.A.Sc. in Information Systems Security from Concordia University in 2007, and a Ph.D. in Electrical and Computer University from the University of Waterloo in 2011.

After graduating from Waterloo he remained with the Center for Pattern Recognition and Machine Intelligence, and later worked as a Post-Doctoral Researcher in the Department of Electrical and Computer Engineering at Carnegie Mellon University. Currently, he is an Associate Professor in the Department of Computer and Information Science at the University of Macau, and is leading the team of Pattern Analysis and Machine Intelligence Group. His research interests focus on biometrics, pattern recognition, and image processing.

This work is licensed under a Creative Commons Attribution 4.0 License. For more information, see https://creativecommons.org/licenses/by/4.0/