

## Article

# Improved Multiple Vector Representations of Images and Robust Dictionary Learning

Chengchang Pan <sup>1</sup>, Yongjun Zhang <sup>1,\*</sup>, Zewei Wang <sup>1</sup> and Zhongwei Cui <sup>2,3</sup>

<sup>1</sup> Key Laboratory of Intelligent Medical Image Analysis and Precise Diagnosis of Guizhou Province, College of Computer Science and Technology, Guizhou University, Guiyang 550025, China; gzpan19@hotmail.com (C.P.); gs.wangzw19@gzu.edu.cn (Z.W.)

<sup>2</sup> School of Mathematics and Big Data, Guizhou Education University, Guiyang 550018, China; zhongweicui@gznc.edu.cn

<sup>3</sup> Big Data Science and Intelligent Engineering Research Institute, Guizhou Education University, Guiyang 550018, China

\* Correspondence: yjzhang1@gzu.edu.cn

**Abstract:** Each sparse representation classifier has different classification accuracy for different samples. It is difficult to achieve good performance with a single feature classification model. In order to balance the large-scale information and global features of images, a robust dictionary learning method based on image multi-vector representation is proposed in this paper. First, this proposed method generates a reasonable virtual image for the original image and obtains the multi-vector representation of all images. Second, the same dictionary learning algorithm is used for each vector representation to obtain multiple sets of image features. The proposed multi-vector representation can provide a good global understanding of the whole image contour and increase the content of dictionary learning. Last, the weighted fusion algorithm is used to classify the test samples. The introduction of influencing factors and the automatic adjustment of the weights of each classifier in the final decision results have a significant indigenous effect on better extracting image features. The study conducted experiments on the proposed algorithm on a number of widely used image databases. A large number of experimental results show that it effectively improves the accuracy of image classification. At the same time, to fully dig and exploit possible representation diversity might be a better way to lead to potential various appearances and high classification accuracy concerning the image.

**Keywords:** multiple vector representation; sparse representation; dictionary learning; image classification



check for updates

**Citation:** Pan, C.; Zhang, Y.; Wang, Z.; Cui, Z. Improved Multiple Vector Representations of Images and Robust Dictionary Learning. *Electronics* **2022**, *11*, 847. <https://doi.org/10.3390/electronics11060847>

Academic Editors: Andrea Prati, Luis Javier García Villalba and Vincent A. Cicirello

Received: 12 January 2022

Accepted: 5 March 2022

Published: 8 March 2022

**Publisher's Note:** MDPI stays neutral with regard to jurisdictional claims in published maps and institutional affiliations.



**Copyright:** © 2022 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<https://creativecommons.org/licenses/by/4.0/>).

## 1. Introduction

Image classification has always been one of the important research topics in the field of computer vision. It can effectively analyze the content of the image to obtain the key information in the image and give the correct judgment, which is of great significance to real work life and social development. Generally, image classification includes image preprocessing [1], image feature extraction [2,3], feature location, feature selection [4] and classifier design [5,6]. Image feature extraction is an important algorithm and faces many challenges in image classification, such as intra-class change, scale change, viewpoint change, illumination conditions and background interference. Therefore, how to effectively extract image features is an urgent problem to be solved.

In recent years, researchers have generally used sparse representation (SR) and dictionary learning (DL) to extract image features. These methods have a good learning ability for image representation and can show better recognition results in image classification tasks [7–9]. The original idea of sparse representation classification (SRC) is to assume that a given test sample can be linearly represented by all training samples and obtain the coefficients of linear representation. In classification, all training samples and

coefficients of each class are used to obtain the predicted value [10]. The predicted value of all classes is different from test samples, and the class with the smallest difference is the class of test samples. However, when the data volume increases, the optimization process may become very slow. In order to speed up the sparse decomposition process, there are generally two ways: One is to reduce the operation scale by selecting training samples with certain properties (such as neighbor samples) [11–13], and the other is to obtain a more compact dictionary containing a large amount of identification information through DL, such as the classical K-Singular Value Decomposition algorithm (KSVD) [7], Discriminative K-SVD (D-KSVD) [14], Label Consistent K-SVD (LC-KSVD) [15] and fisher discrimination dictionary learning (FDDL) [16]. It is worth noting that there is a novel and prospective research field-hybrid method between metaheuristics and machine learning with computer vision/image processing applications. The novel research field successfully combines machine learning and swarm intelligence approaches and proved to be able to obtain outstanding results in different areas [17,18].

The key of dictionary learning is to obtain a robust dictionary, which can reduce the difference between the same class of test samples and training samples. There are two ways to achieve robustness. One is to impose label consistency constraints on deep sparse features [19,20]. Another is the representation-based classification method (RBCM) [21–23]. When RBCM is applied to image data, the dictionary learning algorithm converts each image into a vector. References [24,25] also used the idea of converting each image into a vector when processing images. In the case of the fixed algorithm, fully exploiting the characteristics of the original sample can improve the classification effect [26–28]. Similarly, potential multiple representations directly from the original data can provide different observations of the samples. Combining the features of multiple representations not only makes the sample categories sparse but also makes the classification effect more stable. The information obtained by these methods is sparse and contains local sensitive information of the sample, which can make the feature robust to noise and outliers in the sample.

The main objective behind the approach proposed in this study is to further improve the RBCM, increase the classification performance of DL, and avoid the features missing by obtaining novel representations of images, from a practical scope. It shows that the diversity of vector representations of the image can be further consolidated by matrix decomposition in dictionary learning, so the resultant complementary information can be better exploited. The contribution of this research is threefold:

- (1) A robust dictionary learning method based on image multi-vector representation is proposed.
- (2) A novel representation of images is designed, which is represented by virtual samples and multi-vectors. Based on the original algorithm, four new vector representation methods are added to better mine the large-scale information and global features of images.
- (3) A reasonable weighted fusion image classification algorithm is proposed. The influence factor is introduced to automatically adjust the weight of each classifier in the final decision results, which has a significant effect on better extracting image features and can obtain very stable and accurate classification results.

The organizational structure of this paper is as follows. In Sections 2.1 and 2.2, how the new image representation method is generated is described in detail, and the process of weighted fusion algorithm based on SR is described. In Section 3, this paper analyzes the proposed algorithm and compares it with the original algorithm. In Section 4, comparative experiments are conducted with other algorithms on a large number of databases. The last part summarizes the findings of this research.

## 2. Proposed Method

### 2.1. To Obtain Novel Representations of Images

This paper proposes an improved image multi-vector representation algorithm based on [21] to represent the original image. Use the following steps to obtain a multi-vector representation of an image.

- (1) Generate a virtual image  $V$  from the original image  $I$ . The representation method of generating  $V$  is as follows:

$$V_{rc} = \sqrt[2]{I_{rc} \cdot (P_{\max} - I_{rc})} \tag{1}$$

where  $V_{rc}$  is the pixel value in the  $r$ -th row and  $c$ -th column in the virtual image,  $P_{\max}$  is the largest pixel value existing in the original image and  $I_{rc}$  is the pixel value in the  $r$ -th row and  $c$ -th column in the original image. The following conclusions can be easily drawn out by Formula (1).

**Proposition 1.** When  $I$  is a gray-scale image, the maximum value of  $V_{rc}$  is  $\sqrt[2]{127 * 128}$ .

**Proposition 2.** When  $I_{rc} = 0$  or  $I_{rc} = P_{\max}$ ,  $V_{rc} = 0$ .

**Proposition 3.** When  $P_{\max} = 2I_{rc}$ ,  $V_{rc} = I_{rc}$ .

The pixel value of the virtual image  $V$  generated in this way is symmetrical about the maximum value  $\sqrt[2]{127 * 128}$ , which increases the frequency of pixels value in the medium-intensity pixel and subtracts some other useless characteristic information. In other words, the generated virtual image not only has stronger discriminative information, but also maintains the local geometric features of the original image.

- (2) Convert  $I$  and  $V$  into an improved multi-vector representation of the image.

$I$  and  $V$  together constitute this experimental database.  $I$  is divided into training sample  $Y$  and test sample  $P$ .  $V$  is added to  $Y$  to enrich the training sample, and then all images are converted into multi-vector representations of images using the method in [21]. The study upgrades  $t = 4$  in the original algorithm to  $t = 8$ .

For the  $m$ -th sample, this paper uses the vector  $y_m^i$  to represent the  $i$ -th training sample, and the vector  $p_m$  to represent the test sample. Both  $y_m^i$  and  $p_m$  are generated from the original image matrix  $Y_m^i$  and  $P_m$ . The structure of  $Y_m^i$  is shown in (2). There are many ways to generate  $y_m^i$  and  $p_m$ . There are eight ways in the proposed algorithm:

$$Y_m^i = \begin{pmatrix} V_{11} & V_{12} & \cdots & V_{1c} \\ V_{21} & V_{22} & \cdots & V_{2c} \\ \vdots & \vdots & & \vdots \\ V_{r1} & V_{r2} & \cdots & V_{rc} \end{pmatrix} \text{ or } \begin{pmatrix} I_{11} & I_{12} & \cdots & I_{1c} \\ I_{21} & I_{22} & \cdots & I_{2c} \\ \vdots & \vdots & & \vdots \\ I_{r1} & I_{r2} & \cdots & I_{rc} \end{pmatrix} \tag{2}$$

Concatenate the first to last rows in the sequence of image matrices  $Y_m^i$  and  $P_m$ , respectively, to obtain  $y_m^i$  and  $p_m$ . Concatenating the last to the first rows in the sequence of image matrices  $Y_m^i$  and  $P_m$ , respectively, is the second representation way. The third way concatenates the first to last columns in the sequence of image matrices  $Y_m^i$  and  $P_m$ , respectively. The fourth way, respectively, concatenate the last to first columns in the sequence of the same image matrices. The fifth way connects the elements in the same image matrix sequence from left to right according to the principal diagonal. The acquired  $y_m^i$  and  $p_m$  structures are shown in (3).

$$y_m^i = ( V_{r1} \quad V_{(r-1)1} \quad \cdots \quad V_{21} \quad V_{(r-1)(c-1)} \quad \cdots \quad V_{rc} )^T \tag{3}$$

Similarly, the sixth way connects the elements in the same image matrix sequence from right to left according to the principal diagonal to obtain  $y_m^i$  and  $p_m$ . The structure after image vector transformation is shown in (4).

$$y_m^i = ( V_{1c} \ V_{1(c-1)} \ V_{2c} \ \cdots \ V_{11} \ V_{22} \ \cdots \ V_{r1} )^T \tag{4}$$

The seventh way connects the elements in the same image matrix from right to left by counter-diagonal. The acquired  $y_m^i$  and  $p_m$  structures are shown in (5).

$$y_m^i = ( V_{11} \ V_{21} \ V_{12} \ \cdots \ V_{r1} \ V_{(r-1)2} \ \cdots \ V_{rc} )^T \tag{5}$$

The eighth way connects the elements in the same image matrix from left to right by counter-diagonal to obtain  $y_m^i$  and  $p_m$ .

Each image representation will form a new dataset and then be used in the dictionary learning image classification algorithm based on sparse representation. In addition to the eight vector representations described above, there are many possible vector representations. For instance, the image is divided into overlapped image blocks with  $\sqrt{n} \times \sqrt{n}$  as the size of the image block, and the image block is represented by the column vector according to the fixed step size (fixed pixels per interval). This point has been explained in [29,30].

### 2.2. The Weighted Fusion Classification Algorithm

This section explains how to do weighted fusion for the multi-vector representation of samples. After obtaining a variety of vector representations of the sample, it is applied to the dictionary learning image classification algorithm based on sparse representation. This paper constructs several sparse classifiers to verify the accuracy of classifiers generated by different features. Weights are allocated based on the accuracy of different classifiers in all feature cases. The weight proportions of different feature classifiers are updated continuously through iteration. Finally, the result is output by the final decision classifier. The specific process of the algorithm is as follows:

- (1) Suppose there are  $m$  classes of training samples.  $E_i = (e_{i1}, e_{i2}, \dots, e_{in})$  represents the set of training samples from the  $i$ -th class.  $e_{ij}$  represents the image vector representation of the  $j$ -th training sample of the  $c$  class and  $n$  is the number of training samples of the  $i$ -th class. The multi-vector representation method proposed in Section 2.2 is used to extract the feature of the whole training sample and obtain the characteristic matrix  $D$  of multiple sets of training samples.

$$D^k = (D_1, D_2, \dots, D_m) \in R^{d \times n} \tag{6}$$

where  $d$  is the feature dimensions and  $k$  is the  $k$ -th class feature.  $D_i^k$  is the feature vector of the  $i$ -th training sample of the  $k$ -th class sample.

- (2) Based on the methods in [21], corresponding sparse representation classifiers are generated for multiple feature vector matrices in (6). Multiple classifiers obtain the corresponding classification results  $Q_m$  of various sparse representations. Therefore, its residual is calculated via  $l_i(y) = \|y - D\delta_i(\hat{x}_1)\|_2$ , where  $i = 1, 2, \dots, m$ ,  $\delta_i(\hat{x}_1) = (0, \dots, 0, e_{i1}, e_{i2}, \dots, e_{in}, 0, 0, \dots, 0)^T$ . The degree of separation between different categories of the sparse classifier is obtained as follows:

$$S_i^k = \sum_{z=1}^m l_z(y) - l_i(y) \tag{7}$$

$S_i^k$  represents the degree of separation between the  $i$ -th sample of the  $k$ -th class classifier and the other classes in the whole training sample. The greater the separation degree, the more obvious the classification effect of the classifier on the training sample.

- (3) This study constructs and assigns initial weights for each sparse representation classifier. Using the degree of separation  $S$  of the first classifier output as the initial feature weight coefficient:

$$W_i = \frac{\max\{S_j^i\}}{\sum_k S^k} \tag{8}$$

$W_i$  represents the initialization weight ratio assigned to the  $i$ -th class sparse representation classification.

- (4) According to the weight coefficient of (8), the classification results are fused to determine the image category.

$$w_j = \frac{H_j}{\sum_{j=1}^C H_j} \tag{9}$$

$H_j = \sum_{k=1}^m W_i S_i^k, j = 1, 2, \dots, m$ .  $w_j$  is the influence proportion of the  $j$ -th classifier to all samples under the influence of multi-classification fusion decision. We combine the classification result  $Q$  and its proportion value  $w$  for linear weighting, and obtain the final classification result as follows:

$$class(y) = \operatorname{argmax} \sum_{i=1}^j w_i Q_i. \tag{10}$$

- (5) This study iterates and optimizes the  $W_i$  of each classifier based on the validation data of the test samples, and use factor  $\partial$  to adjust the weight ratio of each classifier. If all classifiers fail to detect rightly the vector representation of the sample is discarded. In the classification based on sparse representation, the weight coefficients are meaningful because they reflect the importance of each training sample.

The main steps of this algorithm are shown in Figure 1:

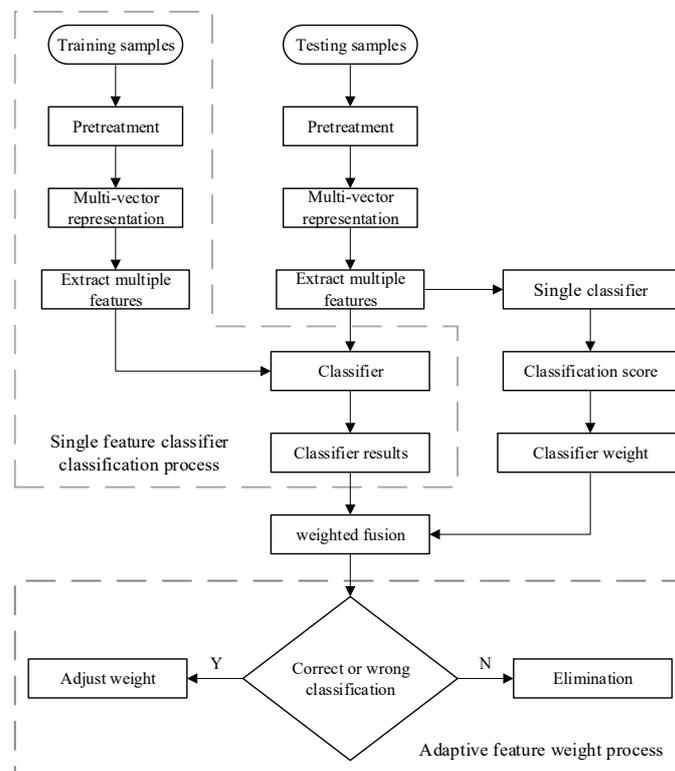


Figure 1. Flowchart of the proposed method.

Step 1. All the original samples are divided into training sample and testing sample and the corresponding preprocessing.

Step 2. The original sample generates a virtual sample by (1). The virtual sample and the original sample make up the experimental dataset.

Step 3. Obtain the multi-vector representation of all original samples using the methods in Section 2.1.

Step 4. The same dictionary learning algorithm is applied to each vector representation of the original image to obtain multiple sets of features.

Step 5. The  $k$ -th class features of the samples are classified using the same classification algorithm to obtain the classification results.

Step 6. Finally, the weighted fusion algorithm is used to classify the test samples.

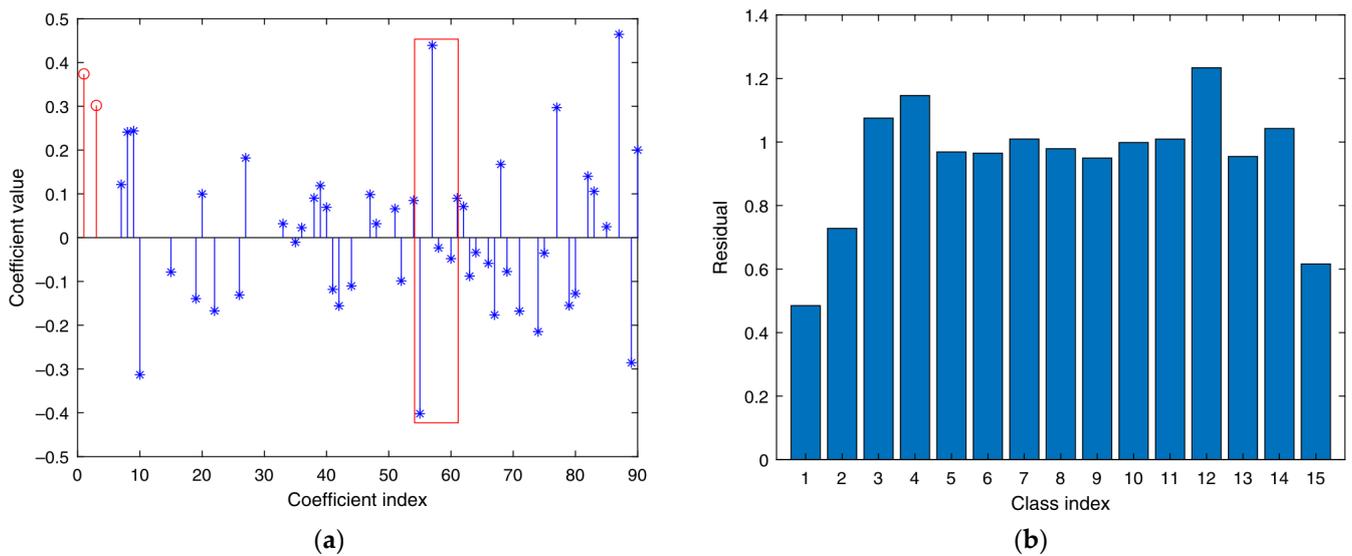
### 3. The Analysis of the Proposed Algorithm

This section mainly analyses the differences between the original algorithm [21] and the improved algorithm.

The method in Section 2.2 is used to generate multiple representations in vector form directly from the original image. Each representation is generated by a specific transformation scheme. If the original image has  $t$  vector representations, the dataset will increase at least  $t$  times. It adds four representation methods to the original algorithm in this study. That is to increase the content of dictionary learning and enrich the features of the sample. The multi-vector representation of this study is reasonable because it reveals more possible changes in the original sample. In other words, multiple representations of an image can be understood as different observations of the same object. Using these multiple representations together can better extract information from objects.

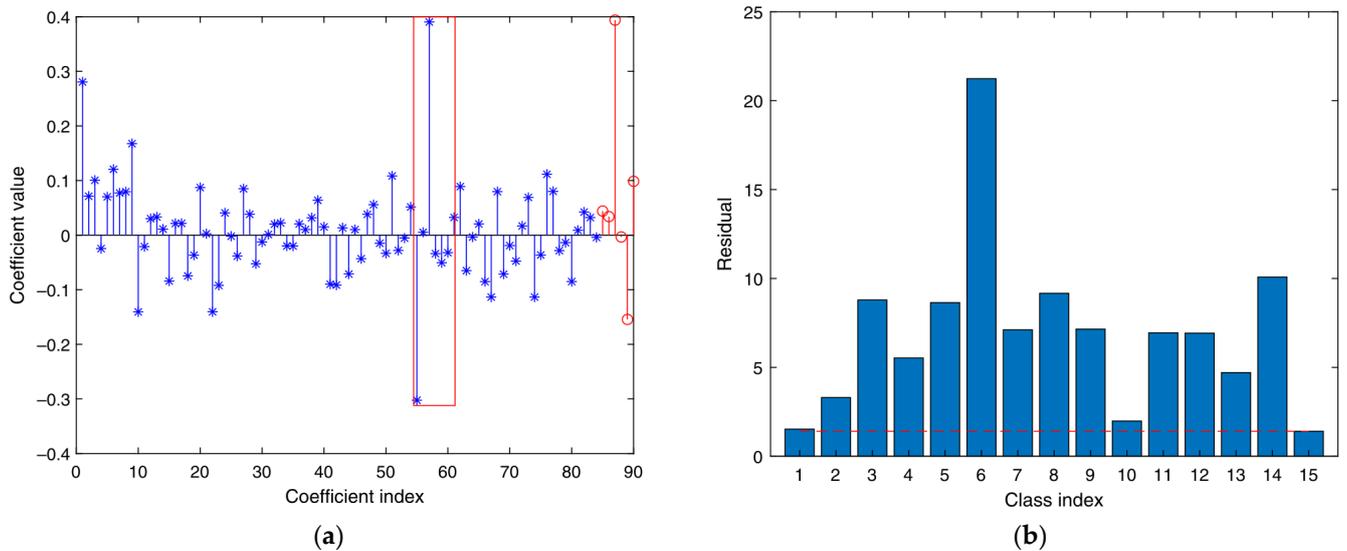
The fusion scheme of the original algorithm has been improved. The influence factor is introduced to adaptively adjust the weight ratio of each classifier in the final decision results because each sparse representation classifier has different classification accuracy for different samples. Therefore, factor  $\partial$  reasonably adjusts each adaptive classifier, and the value of  $\partial$  also has a certain influence on the classification effect. The optimal multi-classifier fusion decision model is obtained by iterative and adaptive updating parameters. It is difficult to achieve good performance with a single feature classification model. However, the multi-vector representation can provide a good global understanding of the overall image contour. For the classification task, it is not ideal to classify images only by global features. The proposed weighted fusion classification algorithm focuses on the extraction of image contour and direction gradient features, and balances global features and local features. Under the multi-classifier fusion decision, it can deal with the image classification problem well.

In order to intuitively illustrate the performance of improved algorithms in classification, the following will be analyzed in detail through the Yale face database experiments. The algorithm has more advantages than the original one. Choose the first face image of the first person as the test sample. Without multi-vector representation, through the SRC method, the sparse coefficient and residual value of each training sample corresponding to this test sample are shown in Figure 2. Each of the six index coefficients corresponds to a category, and the red box corresponds to the 10th class. It can be seen from Figure 2a that none of the representation coefficients of the test sample corresponding to the first type of training samples are less than 0. The 15th training samples also have relatively large positive coefficients. In addition, it is clear from Figure 2b that the residual values calculated from the first training sample do not differ much from those of the fifteenth training sample. This shows that, although the SRC method can correctly classify the test sample, the effect is not obvious.



**Figure 2.** (a) Coefficients obtained by SRC method; (b) residuals obtained by SRC method.

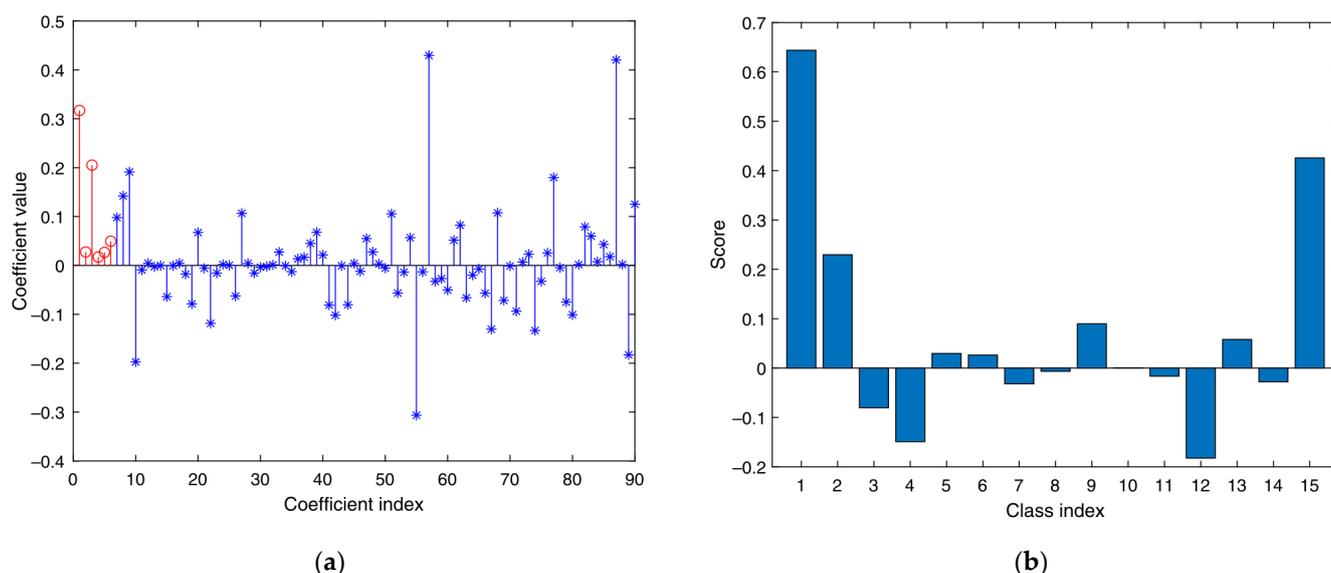
Figure 3a shows the coefficient values obtained by the original algorithm for this test sample, and the coefficient values of the fifteenth class are more significant. In Figure 3b, the residual value of the 15th class is smaller than the first class. Therefore, the test sample is misclassified into the 15th class.



**Figure 3.** (a) Coefficients obtained by the original algorithm; (b) residuals obtained by the original algorithm.

Figure 4 shows the coefficient values and classification scores obtained by the proposed algorithm. The first class has two coefficients greater than 0.2 and both are greater than 0. Figure 4b shows that the classification score of the test sample calculated by our algorithm is the highest, which shows that the test sample has achieved correct classification through the proposed algorithm.

According to the above experimental analysis, it can be found that the “dense” vector representation in the original method will produce multiple representation coefficients, which may lead to classification errors. The weighted fusion algorithm can be used to balance the coefficients, which can avoid classification errors in some cases. This also proves the effectiveness of the proposed algorithm in classification.



**Figure 4.** (a) Coefficients obtained by the proposed algorithm; (b) classification score values obtained by the proposed algorithm.

#### 4. Experiments and Results

This section evaluates the proposed algorithm experimentally on four publicly used databases. These databases include three face databases and one object database, specifically the Yale B face database [31], the PIE face database [32], the AR face database [33] and the COIL-20 database [34]. Statistical tests and comparisons with deep learning methods are also conducted.

In order to illustrate the performance of the proposed algorithm in classification, this paper compares it with the classical dictionary learning algorithm and the original algorithm [21]. The original algorithm is represented as improvement to K-SVD (IKSVD), improvement to D-KSVD (IDKSVD), improvement to LC-KSVD (ILCKSVD), improvement to FDDL (IFDDL) and improvement to RDCDL [27] (IRDCDL). The experimental process is as follows. Firstly, sample sets are divided into training samples and test samples. The original sample generates a virtual sample by (1). Virtual samples and original samples constitute the sample set. The multi-vector representation of all original samples is obtained by using the method in Section 2.2. Then, the multi-vector representation is applied to the same dictionary learning algorithm to obtain multiple sets of features. Finally, the weighted fusion algorithm is used to classify the test samples, and the classification results are obtained. Each dictionary learning algorithm performs 20 times and calculates its average accuracy. The results show that the accuracy of the proposed algorithm has been greatly improved by comparing with the original algorithm and other similar algorithms. The following is the specific experimental analysis of various datasets.

##### 4.1. Experiments on the Extended Yale B Face Database

The Extended Yale B face database contains 2414 faces of 38 persons. Each person has nine different postures and 64 face images with different illuminations. According to the light intensity, it is divided into five subsets, namely subsets I–V. The light intensity of these five subsets is increasing, with a total of 2414 images. Some experimental samples are shown in Figure 5. All images are cut to  $32 \times 32$  pixels. In the experiment, a total of 1170 face images were selected as the test set in subsets I–III, and their images were used as the training set. Figure 6 and Table 1 show the average recognition rate of different algorithms and improved algorithms under different atomic number when the number of training samples is extended concerning the Yale B face database.



Figure 5. Examples of the Yale B's extended face database.

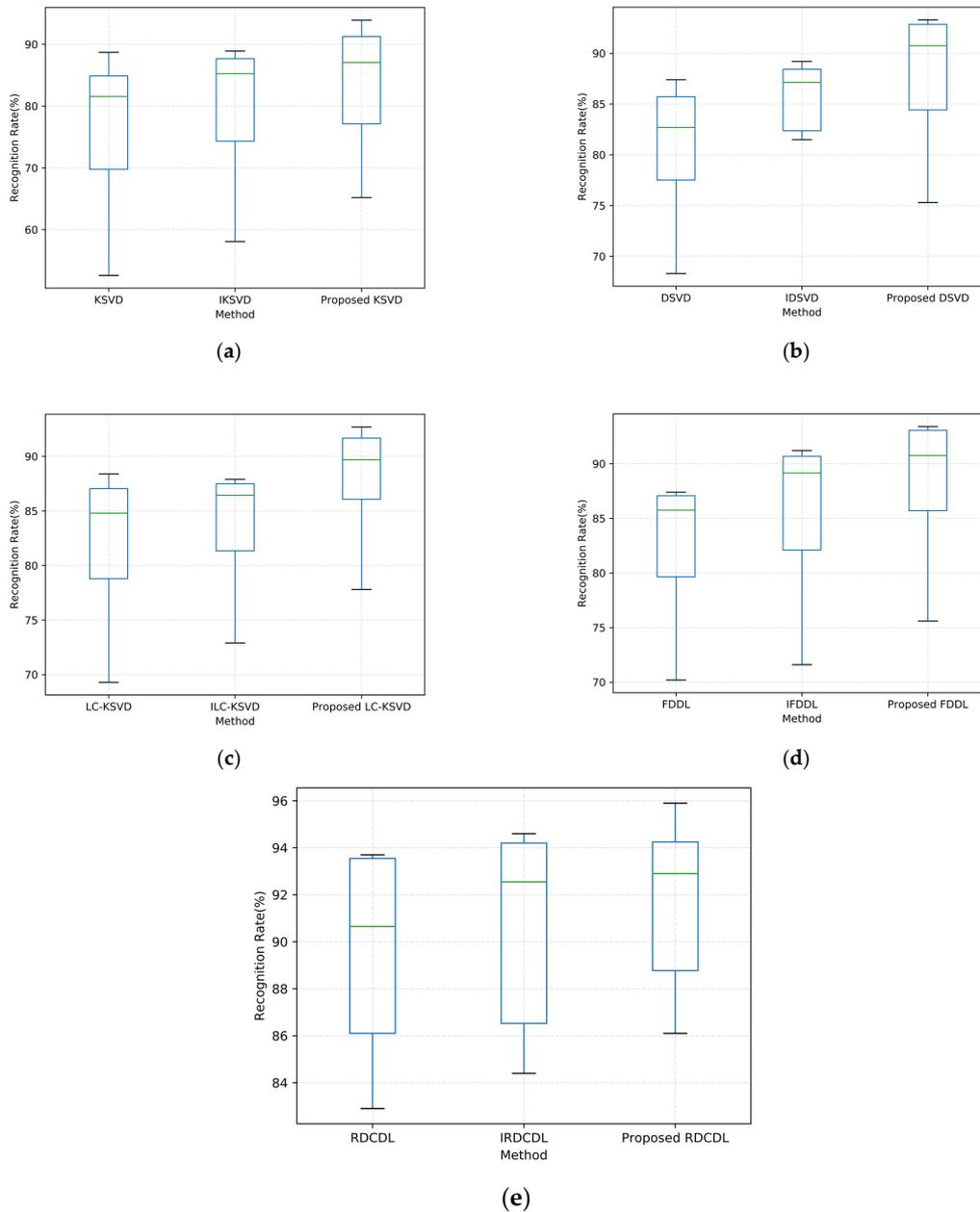


Figure 6. (a) Comparison of different KSVD algorithms; (b) Comparison of different DSVD algorithms. (c) Comparison of different LC-KSVD algorithms; (d) Comparison of different FDDL algorithms; (e) Comparison of different RDCDL algorithms. The accuracy of image classification increases steadily with the increase of atomic number. Compared with other algorithms, the proposed algorithm generally has advantages.

**Table 1.** Average accuracy (%) on the Yale B face database.

The Number of Atoms	38	76	114	152	190	228	266	304	342	380
KSVD	55.2	68.0	75.1	80.2	82.9	85.3	86.2	83.6	52.6	88.7
IKSVD	58.1	72.4	80.1	84.0	86.5	87.9	88.3	87.0	63.8	88.9
Proposed KSVD	65.2	75.1	83.2	86.5	87.6	90.2	92.2	91.6	66.6	93.9
DSVD	52.4	68.3	76.4	80.9	84.1	85.8	87.4	87.1	85.5	81.3
IDSVD	54.1	72.9	81.5	85.0	87.2	88.5	89.2	88.9	88.2	87.1
Proposed DSVD	60.2	75.3	83.5	87.2	90.3	91.2	92.9	93.3	93.2	92.7
LC-KSVD	55.3	69.3	77.9	81.5	84.0	85.6	86.6	87.9	88.4	87.2
ILC-KSVD	55.4	72.9	80.5	83.9	85.9	87.0	87.5	87.9	87.9	87.5
Proposed LC-KSVD	60.4	77.8	85.2	88.7	89.4	90	90.7	92	92.7	92.3
FDDL	45.7	70.2	78.5	83.1	85.2	86.3	87.0	87.1	87.4	87.4
IFDDL	50.0	71.6	80.9	85.7	88.4	89.9	90.6	91.1	91.2	90.7
Proposed FDDL	60.1	75.6	84.4	89.6	90.3	91.2	92.9	93.3	93.4	93.1
RDCDL	71.9	82.9	85.3	88.5	90.1	91.2	93.4	93.6	93.6	93.7
IRDCDL	73.6	84.4	85.4	89.9	91.9	93.2	93.9	94.3	94.6	94.6
Proposed RDCDL	76.2	86.1	88.2	90.5	92.6	93.2	93.2	94.6	95.6	95.9

From Table 1, it can be seen that, in most cases, the improved algorithm obtains a higher average recognition rate than the original algorithm. When the atom is 342, our FDDL algorithm achieves 93.4% accuracy, which is 2.2% higher than that of IFDDL and 6% higher than that of the classical FDDL algorithm. The most obvious improvement is when the atom is 38; our improved FDDL algorithm has accuracy more than 10% higher than IFDDL, and nearly 15% higher than the common FDDL algorithm. Our KSVD is 3.51% higher than IKSVD on average. Our DSVD is 3.72% higher than IDSVD on average. For FDDL, our FDDL is 3.38% higher than IFDDL on average. With the increase of atomic number ( $K = 38, 76, \dots, 342, 380$ ), our LC-KSVD increases by 4.28% on average compared with the original algorithm. By comparing the accuracy of the above algorithm, our improved algorithm has a certain degree of improvement, and the smaller the number of atoms, the more obvious, which verifies that our algorithm involves different observations of the same object. The improved algorithm has better classification accuracy and stability.

Finally, to establish a visual difference between methods included in the comparison, results over 20 runs for some benchmark instances and better performing methods using box and whiskers diagrams are shown in Figure 6.

#### 4.2. Experiments on the PIE Face Database

This PIE face database contains 41,368 frontal images of 68 people. Each person collects facial images with 13 different poses, 43 different lighting conditions, and four different expressions. Figure 7 shows several example images from the PIE face database. We normalized the size of each image to  $32 \times 32$ , then randomly selected ten pictures of each person as training samples, and the remaining samples as test samples.

**Figure 7.** Examples of the PIE face database.

Table 2 shows the average recognition rate of different algorithms in the PIE face database. It can be seen from Table 2 that the performance of our algorithm is significantly better than that of other algorithms. For example, when the atom is 68, the accuracy of our FDDL algorithm is 68.4%, which is 11% higher than IFDDL. Atoms rose by more than 5.12% overall from 68 to 340. For the LC-KSVD algorithm, the average recognition rate of our improved algorithm is 81.36%, and the average recognition rate of the original algorithm is 75.38%, which is 5.98% higher than the original algorithm. Compared with other common algorithms, the improved algorithm has higher accuracy.

**Table 2.** Average accuracy (%) on the PIE face database.

The Number of Atoms	68	136	204	272	340
KSVD	63.7	68.1	71.4	74.9	77.2
IKSVD	68.4	75.4	78.6	81.2	82.4
Proposed KSVD	72.5	77.4	82.3	85.4	88.1
DSVD	63.8	68.3	73.1	75.3	77.7
IDSVD	69.2	77.7	81.1	82.5	83.4
Proposed DSVD	73.4	83.2	85.1	88.7	89.4
LC-KSVD	62.8	73.0	72.6	74.3	76.4
ILC-KSVD	67.9	79.3	77.7	76.5	75.5
Proposed LC-KSVD	72.2	83.4	84.7	83.1	83.4
FDDL	47.6	69.9	75.5	78.5	80.1
IFDDL	57.2	76.5	81.4	82.9	83.8
Proposed FDDL	68.4	80.4	84.5	86.7	87.4
RDCDL	75.7	81.1	83.4	82.9	84.2
IRDCDL	78.2	85.6	85.7	84.1	85.6
Proposed RDCDL	79.1	86.8	87.5	87.4	88.5

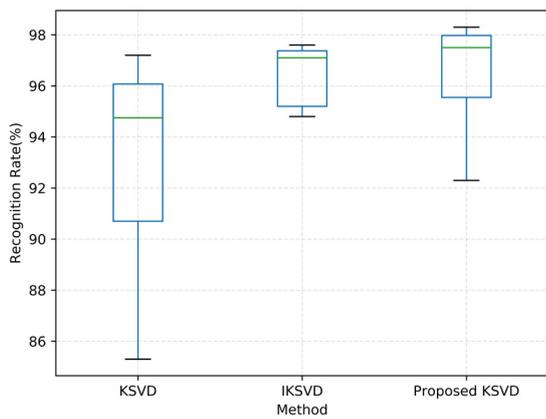
#### 4.3. Experiments on the AR Face Database

In the AR face database, 50 male and 50 female faces were selected to form a subset of the database, and 14 facial images with changes in lighting and expressions of each person were extracted, of which seven were used for training, and the remaining seven photos were used as the test set. The image size was uniformly adjusted to  $40 \times 50$ .

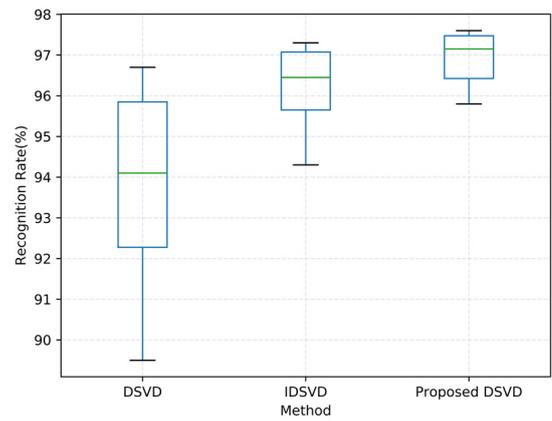
Table 3 shows the comparison of different algorithms in the AR face database. It can be seen from Table 3 that when the atomic number takes different values, our algorithm has higher recognition rate than other algorithms. The most significant improvement is that when the atom is 1080, the average accuracy of our KSVD algorithm reaches 94.5%, which is 14.4% higher than that of IKSVD. In KSVD, the overall average recognition rate of our algorithm is also the highest, reaching 1.97%. For the FDDL algorithm, although our improved algorithm has little improvement, the overall basic accuracy is as high as 88.7% (atoms from 120 to 1200), and the average accuracy is 95.72%. Figure 8b shows the average recognition rates of DSVD, IDSVD and our DSVD algorithms with different atomic numbers. When the atomic number increases, the average recognition rates of DSVD, IDSVD and our DSVD algorithms also gradually increase, but the performance of K-SVD and IKSVD is not stable. Results over 20 runs for some benchmark instances, and better performing methods using box and whiskers diagrams, are shown in Figure 8.

**Table 3.** Average accuracy (%) on the AR face database.

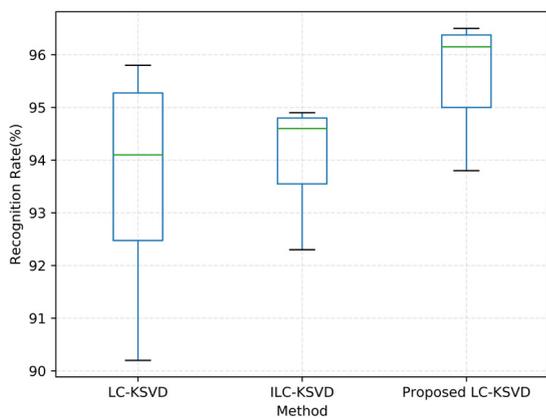
The Number of Atoms	120	240	360	480	600	720	840	960	1080	1200
KSVD	85.3	89.9	93.1	94.3	95.2	96.0	96.4	96.1	79.8	97.2
IKSVD	91.0	94.8	96.4	97.0	97.3	97.6	97.6	97.4	80.1	97.2
Proposed KSVD	92.3	95.1	96.9	97.4	97.9	98.0	98.1	98.3	94.5	97.6
DSVD	82.8	89.5	92.0	93.6	94.6	95.4	96.2	96.0	93.1	96.7
IDSVD	90.2	94.3	95.5	96.3	96.6	97.0	97.3	97.2	96.1	97.1
Proposed DSVD	91.0	95.8	96.4	97.0	97.5	97.6	97.6	97.4	96.5	97.3
LC-KSVD	86.8	90.2	92.7	93.7	94.5	94.9	95.4	95.4	95.8	92.4
ILC-KSVD	88.7	92.3	94.0	94.5	94.7	94.8	94.9	94.9	94.8	93.4
Proposed LC-KSVD	90.3	93.8	96.1	96.2	96.3	96.4	96.4	96.5	95.9	94.7
FDDL	83.4	91.4	92.2	93.1	93.5	93.9	93.9	93.9	93.9	93.9
IFDDL	86.6	94.1	95.4	95.9	96.0	96.2	96.4	96.4	96.5	96.1
Proposed FDDL	88.7	95.3	95.9	96.3	96.4	96.7	97	97	97.1	96.8
RDCDL	81.2	85.2	84.1	84.7	83.8	81.7	86.8	87.5	87.2	88.4
IRDCDL	82.8	86.1	85.7	85.2	86.1	82.6	86.5	87.5	87.3	88.4
Proposed RDCDL	84.2	86.5	88.5	88.7	89.1	90.2	90.8	90.8	90.8	90.8



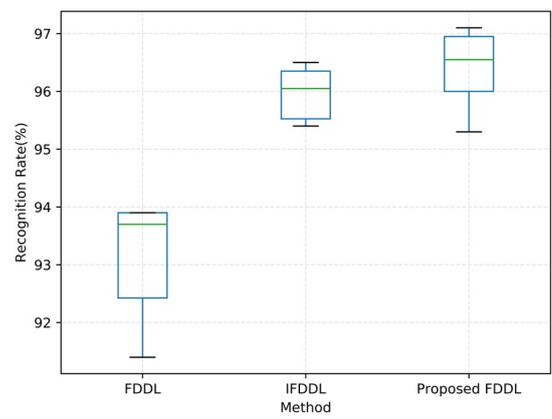
(a)



(b)

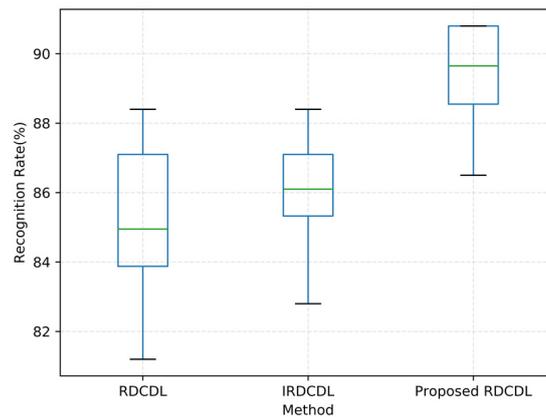


(c)



(d)

**Figure 8.** Cont.



(e)

**Figure 8.** (a) Comparison of different KSVD algorithms; (b) Comparison of different DSVD algorithms. (c) Comparison of different LC-KSVD algorithms; (d) Comparison of different FDDL algorithms; (e) Comparison of different RDCDL algorithms. The accuracy of image classification increases steadily with the increase of atomic number. Compared with other algorithms, the proposed algorithm generally has advantages.

4.4. Experiments on the COIL-20 Database

The COIL-20 dataset is a collection of grayscale pictures, including 20 objects taken from different angles; one image is taken every five degrees, and 72 images are taken for each object. The dataset contains two subsets. The first group contains a total of 720 unprocessed images of 10 objects. After processing 20 things, the second group includes a total of 1440 images. The image size is uniformly adjusted to  $32 \times 32$ . Figure 9 shows several example images in the COIL-20 database. Randomly select ten images as training samples and the remaining images as test samples.



**Figure 9.** Examples of the COIL-20 database.

The comparison results of classification accuracy of different algorithms in the COIL20 database are shown in Table 4. With the increasing number of atoms ( $K = 40, 60, \dots, 180, 200$ ), our KSVD algorithm increases by 2.74% on average compared with IKSVD, and the overall average accuracy is 91.19%. When the atoms are 60 and 180, the improvement effect is the smallest, and our FDDL is only 0.4% higher than IFDDL. When the atoms are 120 and 140, the improvement effect is the largest, and our LSVD is 3.8% higher than IKSVD. For non-face database, the proposed algorithm can also effectively improve the accuracy of image classification.

**Table 4.** Average accuracy (%) on the COIL-20 database.

The Number of Atoms	40	60	80	100	120	140	160	180	200
KSVD	82.5	85.5	87.2	88.0	88.4	89.0	89.5	84.7	89.0
IKSVD	84.0	86.8	88.2	89.1	89.3	89.5	89.8	90.3	89.0
Proposed KSVD	86.3	89.7	90.3	91.9	93.1	93.3	93.2	92.6	90.3
DSVD	84.3	85.6	86.9	87.5	88.5	89.1	88.7	87.0	84.1
IDSVD	84.6	86.8	88.0	89.0	89.6	90.0	90.0	89.9	89.0
Proposed DSVD	85.7	88.8	90.7	90.9	91.2	93.1	93	92.7	92.6
LC-KSVD	84.7	87.8	88.3	88.7	89.2	89.1	89.1	88.9	89.4
ILC-KSVD	85.3	88.9	89.7	90.2	90.5	90.6	90.5	90.8	90.5
Proposed LC-KSVD	86.3	89.9	90.4	92.3	93.2	93.3	93.4	93.4	93.2
FDDL	84.5	87.8	88.2	89.2	89.9	89.9	90.6	90.6	91.1
IFDDL	80.5	86.5	89.1	89.1	91.1	91.3	91.7	91.9	91.8
Proposed FDDL	84.0	86.9	91.2	91.2	92	92	92.2	92.3	92.5
RDCDL	84.8	87.5	89.4	88.5	90.2	91.4	92.1	92.1	92.9
IRDCDL	85.2	87.3	89.3	89.5	90.7	91.6	92.9	92.3	93.0
Proposed RDCDL	86.9	89.1	90.1	91.9	93.5	93.6	94.1	94.3	94.1

#### 4.5. Statistical Tests

In this paper, the Friedman test [35] and the Nemenyi test [36] are applied to show whether there are statistical differences between the performance of classification algorithms. The experiment first assumes that  $H_0$  represents no significant difference between all algorithms (all classification algorithms perform equally),  $H_1$  represents that there are significant differences between algorithms. This is assumed certain and evidence is searched for in the data to reject it.

This paper compares all the algorithms on the Yale B, PIE, AR and COIL-20 datasets. Each algorithm runs 20 times and takes the average as the test result. Then, each dataset is sorted according to the test performance from good to bad, and the order values are 1, 2, 3, ..., 15. If the test performance of the algorithm is the same, the order value is divided equally, as shown in Table 5. Then use the Friedman test to determine whether these algorithms perform the same. Friedman statistics can be calculated as follows:

$$F_F = \frac{(N-1)\chi_F^2}{N(k-1) - \chi_F^2} \quad (11)$$

which is compared to an  $F$ -distribution with  $k-1$  and  $(k-1)(n-1)$  degrees of freedom, where  $n$  denotes the total number of experiments,  $k$  is the amount of classification models and  $\chi_F^2$  is the value of the Chi-square distribution with  $F$  degrees of freedom. The variable  $F_F = 4.593$  is calculated through (11). When the number of datasets is 4 and the number of algorithms is 15, the critical value of the  $F$  test is 1.935, so  $H_0$  is rejected and  $H_1$  is accepted. Nemenyi's test is continued, and the critical value domain  $CD$  of the average ordinal value is calculated according to (12).

$$CD = q_\alpha \sqrt{\frac{k(k+1)}{6N}} \quad (12)$$

The CD computed through Nemenyi's test ( $CD = 3.049$ ) strongly suggests the existence of significant differences among the algorithms considered. The proposed DSVD and RDCDL algorithms obtained good rankings.

**Table 5.** Ranks achieved by the Friedman and Nemenyi tests in the study. Our DSVD and our RDCDL algorithms achieve the best rank in the procedures. The statistics computed and related p-values are also shown.

Algorithms	Friedman
KSVD	13.5
IKSVD	10.25
Proposed KSVD	4.5
DSVD	13
IDSVD	7.75
Proposed DSVD	3.25
LC-KSVD	11.25
ILC-KSVD	9.25
Proposed LC-KSVD	4.75
FDDL	12.25
IFDDL	8.25
Proposed FDDL	4.25
RDCDL	7.5
IRDCDL	6.25
Proposed RDCDL	4
Statistic	33.875
$F_f$	4.593
CD	3.049
p-value	0.002150951

#### 4.6. Comparison with Deep Learning Methods

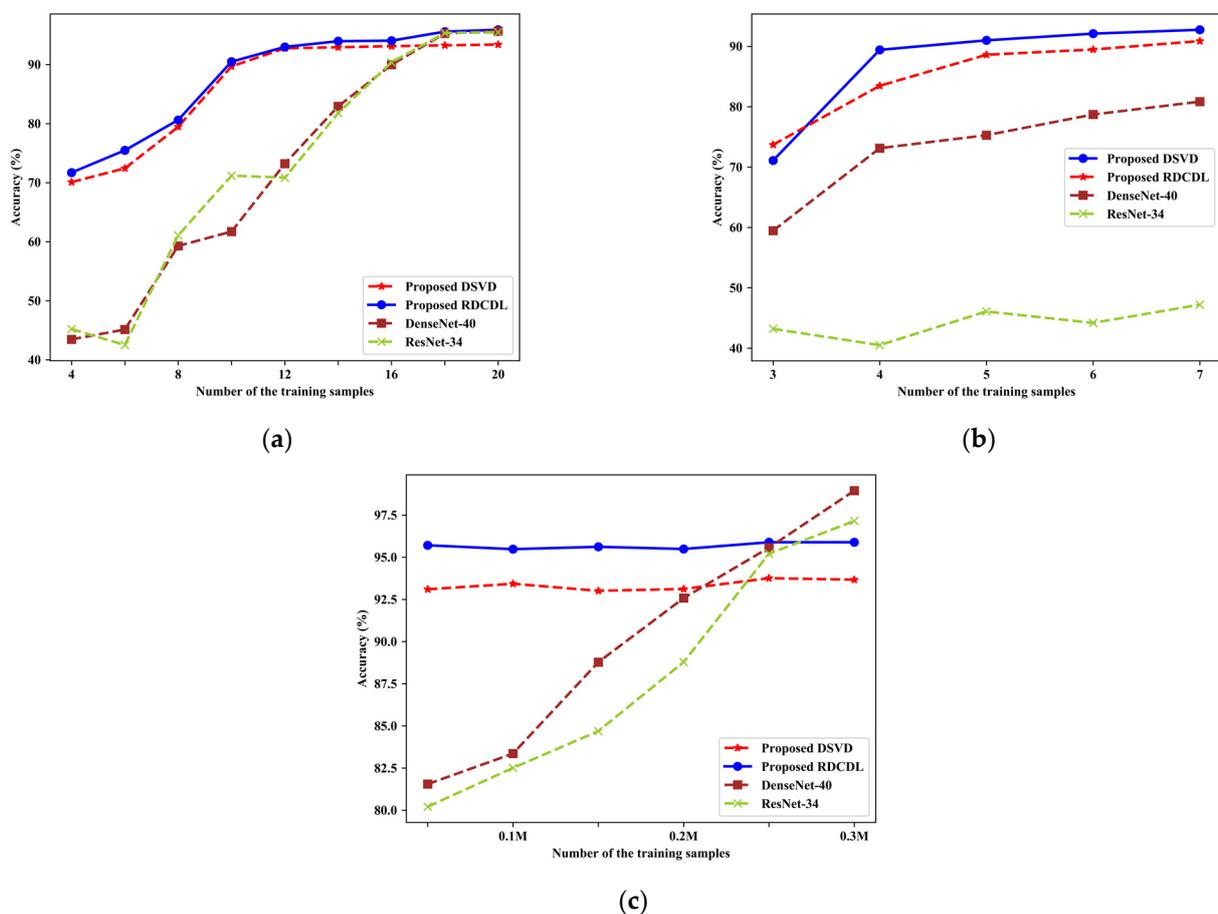
In order to comprehensively evaluate the proposed methods, some methods based on deep learning are compared. For example, ResNet-34 [37] and DenseNet-40 [38]. We select the two methods (proposed DSVD and RDCDL) that perform best in statistical tests to compare with them. In this experiment, the operating system is Ubuntu18.04, the CPU is Intel(R) Core (TM) i7-8700 CPU 3.20 GHz and the GPU is NVIDIA GeForce GTX1080Ti 8 GB. Training and testing are carried out in Python language under the PyTorch deep learning framework. The batch-size in the training is set to 64, the initial learning rate is set to 0.05, and the dictionary size is set to 380 (10 atoms per category) in the Yale B database and 600 in AR database. All the methods are repeated 20 times, and their average accuracy is shown in Table 6.

**Table 6.** Average accuracy (%) of various methods on the AR, Yale B database and CASIA-WebFace dataset.

Methods	Ext. Yale B	AR	CASIA-WebFace
Proposed DSVD	93.39	92.76	93.44
Proposed RDCDL	95.90	90.89	95.37
ResNet-34	95.47	47.21	97.15
DenseNet-40	95.63	80.85	98.95

In comparison with methods based on deep learning, the performance of the proposed DSVD and RDCDL is better than that of ResNet-34 and DenseNet-40 on the Extended Yale B and AR database. However, methods based on deep learning are better on the CASIA-WebFace dataset [39]. In order to further evaluate the performance of dictionary learning methods and deep learning methods based on these three databases, we investigated the

accuracy of the proposed DSVD, the proposed RDCDL, ResNet-34 and DenseNet-40 under different training samples, as shown in Figure 10. According to Figure 10a, when the number of training samples is less than or equal to 16, the proposed DSVD and RDCDL has obvious advantages over ResNet-34 and DenseNet-40. Combined with Figure 10c, it is not difficult to find that ResNet-34 and DenseNet-40 are more suitable for large-scale training sets. When the dataset is small, a large number of effective features cannot be extracted. It can also be found from the figure that when the number of training samples is less than 16, the accuracy of ResNet-34 is very unstable. Because the discriminant information in the small-scale training data is insufficient to update the parameters in ResNet-34 to obtain stable accuracy, the accuracy is often determined by the randomly selected initial parameters. This indicates that the proposed method has a prominent superiority over deep learning based methods when provided with only a few training samples on a small-size dataset.



**Figure 10.** (a) The average performance of different training samples on the Extended Yale B database; (b) The average performance of different training samples on the AR database; (c) The average performance of different training samples on the CASIA-WebFace dataset.

This paper also conducted time cost experiments on the CASIA-WebFace dataset. Table 7 shows the average training and testing time of different methods on the CASIA-WebFace dataset. The time required for dictionary learning to extract features is very short, and their fastest time is less than 70 s, which is very obvious compared with ResNet-34 and DenseNet-40. Because methods based on deep learning have many parameters and the gradient descent method is used to iteratively train the convolution filter, it requires not only a large number of training samples, but also a large amount of calculation.

The experimental results show that deep learning and traditional methods have their advantages and disadvantages. Methods based on deep learning require a large number of

training samples, consume a lot of computing time and the model is complex and difficult to deploy. However, proposed models do not require much iterative training and the average test time is only 0.85 s.

**Table 7.** Average training time and testing time on CASIA-WebFace dataset.

Methods	Average Training Time	Average Testing Time
Proposed DSVD	67.53 s	0.85 s
Proposed RDCDL	73.55 s	0.87 s
ResNet-34	2.5 h	2.16 s
DenseNet-40	3.2 h	2.04 s

## 5. Conclusions

In this paper, a new image classification algorithm is proposed, which uses different observations of the original sample to generate multiple vector representations, thereby enhancing the potential correlation of the original sample. This paper combines the multi-vector representation of images with robust dictionary learning, and designs a weighted fusion algorithm so that dictionary learning can extract higher-dimensional and abstract features from data, and effectively use the sample information between multi-vector representations to improve the discriminant ability. After comparative experiments, the classification performance of the proposed algorithm is better than that of the original algorithm before improvement. In addition, this algorithm effectively improves the classification accuracy, and has the advantages of simple implementation and high degree of automation. This paper uses the original sample and multi-vector representation to learn the dictionary for the same object. This operation makes the proposed algorithm more general. Research shows that more diversified and deeper features can be obtained through deep mining of the original samples.

A little flaw of the proposed method is that it has a higher computational cost than that of the conventional dictionary learning algorithm. Because there are kinds of representations, the next research direction is to explore more effective representation and fusion strategies, so as to improve classification performance and reduce computation.

**Author Contributions:** In order to balance the large-scale information and global features of images, a robust dictionary learning method based on image multi-vector representation is proposed in this paper. C.P. and Y.Z. conceptualized the methodology and all the research and wrote large sections of the paper. Z.W., Y.Z. and Z.C. were involved in the interpretation of the results. C.P. was responsible for the visualization and presentation of the mobility indicators. All authors have read and agreed to the published version of the manuscript.

**Funding:** This work was supported by the Research Foundation for Advanced Talents of Guizhou University under Grant: (2016) No. 49, Key Disciplines of Guizhou Province Computer Science and Technology (ZDXK[2018]007), Research Projects of Innovation Group of Education (QianJiao-HeKY[2021]022), Project supported by the Guizhou Province Graduate Research Fund (YJSCXJH[2020]53, YJSCXJH[2020]189), and supported by the National Natural Science Foundation of China (62062023).

**Data Availability Statement:** The Extended Yale B face database: <http://vision.ucsd.edu/~leekc/ExtYaleDatabase/download.html> (accessed on 10 December 2021); The PIE face database: [http://web.mit.edu/emeyers/www/face\\_databases.html](http://web.mit.edu/emeyers/www/face_databases.html) (accessed on 8 December 2021); The AR face database: [http://web.mit.edu/emeyers/www/face\\_databases.html](http://web.mit.edu/emeyers/www/face_databases.html) (accessed on 8 December 2021); The COIL-20 database: <https://www.cs.columbia.edu/CAVE/software/softlib/coil-20.php> (accessed on 8 December 2021); The CASIA-WebFace dataset: <https://paperswithcode.com/dataset/casia-webface> (accessed on 7 February 2022).

**Conflicts of Interest:** The authors declare no conflict of interest.

## References

1. Zhang, J.; Liu, W.; Bo, L.; Zhang, H.; Li, H.; Xu, S. Joint Reflectance Field Estimation and Sparse Representation for Face Image Illumination Preprocessing and Recognition. *Neural Process. Lett.* **2020**, 1–14. [[CrossRef](#)]
2. Wang, L.; Li, T. Research on Image Feature Extraction Method Fusing HOG and Canny Algorithm. In Proceedings of the 2021 4th International Conference on Data Science and Information Technology, Shanghai, China, 23–25 July 2021; pp. 208–211.
3. Lacombe, T.; Favreliere, H.; Pillet, M. Modal features for image texture classification. *Pattern Recognit. Lett.* **2020**, *135*, 249–255. [[CrossRef](#)]
4. Li, Y.; Liu, J.; Tang, Z.; Lei, B. Deep spatial-temporal feature fusion from adaptive dynamic functional connectivity for MCI identification. *IEEE Trans. Med. Imaging* **2020**, *39*, 2818–2830. [[CrossRef](#)] [[PubMed](#)]
5. Zhu, X.; Li, X.; Zhang, S. Block-row sparse multiview multilabel learning for image classification. *IEEE Trans. Cybern.* **2015**, *46*, 450–461. [[CrossRef](#)]
6. Luo, Y.; Liu, T.; Tao, D.; Xu, C. Multiview matrix completion for multilabel image classification. *IEEE Trans. Image Process.* **2015**, *24*, 2355–2368. [[CrossRef](#)]
7. Aharon, M.; Elad, M.; Bruckstein, A. K-SVD: An algorithm for designing overcomplete dictionaries for sparse representation. *IEEE Trans. Signal Process.* **2006**, *54*, 4311–4322. [[CrossRef](#)]
8. Xu, Y.; Zhong, Z.; Yang, J.; You, J.; Zhang, D. A new discriminative sparse representation method for robust face recognition via  $l_2$  regularization. *IEEE Trans. Neural Netw. Learn. Syst.* **2016**, *28*, 2233–2242. [[CrossRef](#)]
9. Li, Z.; Lai, Z.; Xu, Y.; Yang, J.; Zhang, D. A locality-constrained and label embedding dictionary learning algorithm for image classification. *IEEE Trans. Neural Netw. Learn. Syst.* **2015**, *28*, 278–293. [[CrossRef](#)]
10. Zhang, H.; Wang, F.; Chen, Y.; Zhang, W.; Wang, K.; Liu, J. Sample pair based sparse representation classification for face recognition. *Expert Syst. Appl.* **2016**, *45*, 352–358. [[CrossRef](#)]
11. Xian, Y.; Akata, Z.; Sharma, G.; Nguyen, Q.; Hein, M.; Schiele, B. Latent embeddings for zero-shot classification. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Las Vegas, NV, USA, 27–30 June 2016; pp. 69–77.
12. Gao, J.; Xu, L. A novel spatial analysis method for remote sensing image classification. *Neural Process. Lett.* **2016**, *43*, 805–821. [[CrossRef](#)]
13. Albukhanajer, W.A.; Jin, Y.; Briffa, J.A. Classifier ensembles for image identification using multi-objective Pareto features. *Neurocomputing* **2017**, *238*, 316–327. [[CrossRef](#)]
14. Zhang, Q.; Li, B. Discriminative K-SVD for dictionary learning in face recognition. In Proceedings of the 2010 IEEE Computer Society Conference on Computer Vision and Pattern Recognition, San Francisco, CA, USA, 13–18 June 2010; pp. 2691–2698.
15. Jiang, Z.; Lin, Z.; Davis, L.S. Label consistent K-SVD: Learning a discriminative dictionary for recognition. *IEEE Trans. Pattern Anal. Mach. Intell.* **2013**, *35*, 2651–2664. [[CrossRef](#)] [[PubMed](#)]
16. Yang, M.; Zhang, L.; Feng, X.; Zhang, D. Fisher discrimination dictionary learning for sparse representation. In Proceedings of the 2011 International Conference on Computer Vision, Barcelona, Spain, 6–13 November 2011; pp. 543–550.
17. Bacanin, N.; Stoean, R.; Zivkovic, M.; Petrovic, A.; Rashid, T.A.; Bezdán, T. Performance of a novel chaotic firefly algorithm with enhanced exploration for tackling global optimization problems: Application for dropout regularization. *Mathematics* **2021**, *9*, 2705. [[CrossRef](#)]
18. Malakar, S.; Ghosh, M.; Bhowmik, S.; Sarkar, R.; Nasipuri, M. A GA based hierarchical feature selection approach for handwritten word recognition. *Neural Comput. Appl.* **2020**, *32*, 2533–2552. [[CrossRef](#)]
19. Hu, C.; Wu, X.J.; Shu, Z.Q. Discriminative feature learning via sparse autoencoders with label consistency constraints. *Neural Process. Lett.* **2019**, *50*, 1079–1091. [[CrossRef](#)]
20. Huang, Y.; Quan, Y.; Liu, T.; Xu, Y. Exploiting label consistency in structured sparse representation for classification. *Neural Comput. Appl.* **2019**, *31*, 6509–6520. [[CrossRef](#)]
21. Xu, Y.; Li, Z.; Tian, C.; Yang, J. Multiple vector representations of images and robust dictionary learning. *Pattern Recognit. Lett.* **2019**, *128*, 131–136. [[CrossRef](#)]
22. Liu, Z.; Wu, X.J.; Shu, Z. Multi-resolution dictionary collaborative representation for face recognition. *Pattern Anal. Appl.* **2021**, *24*, 1793–1803. [[CrossRef](#)]
23. Zheng, S.; Zhang, Y.; Liu, W.; Zou, Y.; Zhang, X. A dictionary learning algorithm based on dictionary reconstruction and its application in face recognition. *Math. Probl. Eng.* **2020**, *2020*, 8964321. [[CrossRef](#)]
24. Liu, Z.; Wu, X.J.; Yin, H.; Xu, T.; Shu, Z. Locality-Constrained Collaborative Representation with Multi-resolution Dictionary for Face Recognition. In Proceedings of the Chinese Conference on Pattern Recognition and Computer Vision (PRCV), Beijing, China, 29 October–1 November 2021; Springer: Berlin/Heidelberg, Germany, 2021; pp. 55–66.
25. Fan, J.; Yang, C.; Udell, M. Robust Non-Linear Matrix Factorization for Dictionary Learning, Denoising, and Clustering. *IEEE Trans. Signal Process.* **2021**, *69*, 1755–1770. [[CrossRef](#)]
26. Xu, Y.; Zhang, B.; Zhong, Z. Multiple representations and sparse representation for image classification. *Pattern Recognit. Lett.* **2015**, *68*, 9–14. [[CrossRef](#)]
27. Lin, G.; Yang, M.; Yang, J.; Shen, L.; Xie, W. Robust, discriminative and comprehensive dictionary learning for face recognition. *Pattern Recognit.* **2018**, *81*, 341–356. [[CrossRef](#)]
28. Li, L.; Peng, Y.; Qiu, G.; Sun, Z.; Liu, S. A survey of virtual sample generation technology for face recognition. *Artif. Intell. Rev.* **2018**, *50*, 1–20. [[CrossRef](#)]

29. Li, H.; He, X.; Tao, D.; Tang, Y.; Wang, R. Joint medical image fusion, denoising and enhancement via discriminative low-rank sparse dictionaries learning. *Pattern Recognit.* **2018**, *79*, 130–146. [[CrossRef](#)]
30. Zhang, Y.; Wang, Q.; Xiao, L.; Cui, Z. An improved two-step face recognition algorithm based on sparse representation. *IEEE Access* **2019**, *7*, 131830–131838. [[CrossRef](#)]
31. Georghiadis, A.S.; Belhumeur, P.N. Illumination cone models for faces recognition under variable lighting. In Proceedings of the CVPR98, Santa Barbara, CA, USA, 23–25 June 1998.
32. Sim, T.; Baker, S.; Bsat, M. The CMU pose, illumination, and expression (PIE) database. In Proceedings of the Fifth IEEE International Conference on Automatic Face Gesture Recognition, Washington, DC, USA, 21–21 May 2002; pp. 53–58.
33. Martinez, A.; Benavente, R. *The AR Face Database: CVC Technical Report, 24*; Autonomous University of Barcelona: Barcelona, Spain, 1998.
34. Geusebroek, J.M.; Burghouts, G.J.; Smeulders, A.W. The Amsterdam library of object images. *Int. J. Comput. Vis.* **2005**, *61*, 103–112. [[CrossRef](#)]
35. Iman, R.L.; Davenport, J.M. Approximations of the critical region of the fbietkan statistic. *Commun. Stat. Theory Methods* **1980**, *9*, 571–595. [[CrossRef](#)]
36. Derrac, J.; García, S.; Molina, D.; Herrera, F. A practical tutorial on the use of nonparametric statistical tests as a methodology for comparing evolutionary and swarm intelligence algorithms. *Swarm Evol. Comput.* **2011**, *1*, 3–18. [[CrossRef](#)]
37. He, K.; Zhang, X.; Ren, S.; Sun, J. Identity mappings in deep residual networks. In Proceedings of the European Conference on Computer Vision, Amsterdam, The Netherlands, 8–16 October 2016; Springer: Berlin/Heidelberg, Germany, 2016; pp. 630–645.
38. Huang, G.; Liu, Z.; Van Der Maaten, L.; Weinberger, K.Q. Densely connected convolutional networks. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Honolulu, HI, USA, 21–26 July 2017; pp. 4700–4708.
39. Yi, D.; Lei, Z.; Liao, S.; Li, S.Z. Learning face representation from scratch. *arXiv* **2014**, arXiv:1411.7923.