Weighted Sparse Representation Using Collaborative Representation in Kernel Feature Space Based Classification

Kousuke Matsushima
Dept. of Control and Information Systems Engineering, National Institute of Technology, Kurume College, Fukuoka, Japan
Email: matsushima@kurume-nct.ac.jp

Mihoshi Matsusue
Graduate School of Information Sciences, Tohoku University, Sendai, Japan
Email: matsusue.mihoshi.p5@dc.tohoku.ac.jp

Kavin Ruengprateepsang
Faculty of Information Technology, King Mongkut's Institute of Technology Ladkrabang, Bangkok, Thailand
Email: 59070009@kmitl.ac.th

***Abstract*—In this paper, we proposed new method to deal non-linear information such as occlusion by using kernel feature space, which named Weighted Sparse Representation using collaborative representation in kernel feature space based classification (WSRCRKC). Kernel Sparse Representation-based Classification (KSRC) has shown good classification performance and robustness for the problem of nonlinear distribution of face images. To use locality information and eliminate the affection of luminance, Weighted Kernel Sparse Representation-based Classification (WKSRC) is proposed as an extension of KSRC by combining multiscale retinex algorithm. Furthermore, by making kernel Gram matrix sparse, we reduce the computation of face recognition. The experimental result shows that our proposal clearly improves the computational time while keeping accuracy high.**

***Index Terms*—face recognition, classification, kernel sparse representation, dimensionality reduction**

I. Introduction

Currently, image recognition has been used in many situations, such as automatic driving and unlocking smartphone instead of password. In Intelligent Transport System (ITS), face classification has been attracted too. Driver status estimation [1] and policing for over speed can be realized by using face classification [2], [3]. Therefore, we focused on face recognition in this paper.

Most recognition methods have the step of dimensionality reduction, which essentially has two important roles. First role is to reduce the computational time. Dimensionality Reduction (DR) method accelerate computational speed by reducing computational complexity. Second role to improve accuracy. Original information may include unnecessary features that are misleading. The DR method extracts essential information by removing extra elements. DR is indispensable step for those reasons. Famous DR methods are Principal Components Analysis (PCA) [4], [5] and Linear Discriminant Analysis (LDA) [6], [7]. These methods project high dimensional information into low dimension subspace. These methods are very simple and easy to use. However, these methods are too simple. Thus, when these are applied, essential information can be lost.

In recent years, Sparse Representation-based Classification (SRC) [8], [9] has been strongly attracted attentions in the region of machine learning. SRC is one of the classifiers and shows good performance under the situation that the face images are partly occluded. Sparsified Collaborative Representation-Based Local Discriminant Projection for feature extraction (SCRLDP) [10] is the method combined graph-based DR method with sparse representation, which indicates better results in accuracy and computing time than many conventional methods. Therefore, SCRLDP is used in this paper as a DR method.

Furthermore, nonlinear distribution of the image is the common problem. The problem arises with ornaments on the face, such as sunglasses and scarves, which obscure part of the face, causing the image to change discontinuously, making classification difficult, which makes the performance of SRC terrible. However, kernel based algorithm can solve nonlinear problem effectively. One of the most famous kernel based technique is SVMs, which combine the sparsity-induced technique and kernel trick. Kernel Principal Component Analysis (KPCA) [11], [12] is an algorithm applied kernel trick to linear method.

Manuscript received December 23, 2020; revised May 24, 2021.

doi: 10.12720/jait.12.3.220-225

The kernel trick project the original data into a high dimensional kernel feature map. In other words, a linear method can be converted into a nonlinear method by using kernel trick. Kernel Sparse Representation-based Classification (KSRC) [13], [14] combined SRC with kernel trick, which overcomes the problem occurred in SRC and has better performance.

To further improve the performance of KSRC, Weighted Kernel Sparse Representation-based Classification (WKSRC) [15] algorithm was proposed. WKSRC integrates locality structure of the image data into KSRC. In this paper, multiscale retinex algorithm [16] is used to preserve locality structure and overcome variation of illumination.

II. Backgrounds

A. Sparse Representation-Based Classification

SRC has shown its outstanding capability for face recognition. Given a set of training samples with c classes of subjects, $\boldsymbol{X} = [\boldsymbol{X}_1, \boldsymbol{X}_2, \dots, \boldsymbol{X}_c] \in \boldsymbol{R}^{m\times n}$, where m is a number of dimension of training samples and n is total number of training samples. Each $\boldsymbol{X}_i$ denotes the training samples from class i, each column of $\boldsymbol{X}_i$ is a sample. The sparse coefficient vector $\boldsymbol{\alpha}$ is represented as:

$$\boldsymbol{y} = \boldsymbol{X\alpha} \tag{1}$$

where $\boldsymbol{y}$ denotes testing sample and $\boldsymbol{\alpha}_i$ is the sub-vector associated with $\boldsymbol{X}_i$. When the testing sample is from class i, $\boldsymbol{\alpha}$ would be sparse and most items of $\boldsymbol{\alpha}$ will be zero except some of those associated with class i. The vector $\boldsymbol{\alpha}$ will be sparse enough as $\boldsymbol{\alpha} = [\boldsymbol{0}^T, \dots, \boldsymbol{\alpha}_i^T, \dots, \boldsymbol{0}^T]$, where $\boldsymbol{\alpha}^T$ is the transposition $\boldsymbol{\alpha}_i$ which is a nonzero item. Furthermore, (1) is modified as following to be robust to noises:

$$\boldsymbol{y} = \boldsymbol{X\alpha} + \varepsilon \tag{2}$$

where ε is noisy data.

In SRC, the sparse regularization technique has important role in achieving the sparse coefficient vector. Sparse regularization methods such as L0-norm, L1-norm and L2-norm regularization are used to recover to the sparse signal. However, in L0-norm regularization, it is difficult to optimize the objective function and L2-norm has insufficient sparseness. Therefore, L1-norm regularization is widely used in sparse signal reconstruction which can also induce sparseness and solve the problem of discontinuousness as an extension of L0-norm regularization.

For each class i SRC uses (3) to reconstruct testing sample $\hat{\boldsymbol{y}}_i$ which could be used to compare with y. Reconstructed $\hat{\boldsymbol{y}}_i$ which has the minimum residual could be associated with the class label.

$$\hat{\boldsymbol{y}}_i = \boldsymbol{X}_i \hat{\alpha}_i \tag{3}$$

where $\hat{\alpha}_i$ is the sparse coefficient associated with class i. The procedure of SRC is summarized as following.

Input: Matrix of training samples $\boldsymbol{X}$**,** testing sample y.

First step: Normalize the columns of $\boldsymbol{X}$ to have unit L1-norm.

Second step: Solve the L1-minimaization problem to obtain the coefficient vector

$$\hat{\alpha} = \underset{i}{\operatorname{argmin}} \|\boldsymbol{\alpha}\|_1, s.t. \|\boldsymbol{y} - \boldsymbol{X\alpha}\|_2 \le \varepsilon \tag{4}$$

Third step: Calculate the residuals

$$r_i = \|\boldsymbol{y} - \boldsymbol{X}\hat{\alpha}_i\|_2 \tag{5}$$

Output:

$$\text{identity}(\boldsymbol{y}) = \arg\min_i r_i \tag{6}$$

B. Kernel Sparse Representation-Based Classification

In kernel based algorithm, the original data is projected into the kernel space. At this time, mapping function is unnecessary. The most popular technique is kernel trick, which could map the original data into a high dimensionality kernel space. Generally, Mercer kernel $k(\cdot,\cdot)$ is the suitable one, which is used to calculate the kernel Gram matrix $\boldsymbol{K} \in \boldsymbol{R}^{n\times n}$ and vector $\boldsymbol{k}(\cdot, \boldsymbol{y}) \in \boldsymbol{R}^n$ as shown in (7) and (8).

$$\boldsymbol{K}_{i,j} = \left[k(\boldsymbol{x}_i, \boldsymbol{x}_j)\right]_{1\le i,j \le n} \tag{7}$$

$$\boldsymbol{k}(\cdot, \boldsymbol{y}) = [k(\boldsymbol{x}_1, \boldsymbol{y}), \dots, k(\boldsymbol{x}_n, \boldsymbol{y})]^T \tag{8}$$

where $\boldsymbol{x}_i$ and $\boldsymbol{x}_j$ denote the training samples. When the number of training samples is large, computational complexity will be very high. Therefore, KPCA and KLDA were applied to $\boldsymbol{X}$ and $\boldsymbol{y}$ to build the transformation matrix $\boldsymbol{B} \in \boldsymbol{R}^{n\times d}$, where d is the dimension of feature space after dimensionality reduction. The coefficient vector could be computed in the low kernel space as follow:

$$\|\boldsymbol{B}^T \boldsymbol{k}(\cdot, \boldsymbol{y}) - \boldsymbol{B}^T \boldsymbol{K\alpha}\| \le \varepsilon \tag{9}$$

To classify the testing sample, the last step of KSRC compute the minimized residual between $\boldsymbol{B}^T \boldsymbol{k}(\cdot, \boldsymbol{y})$ and the reconstructed sample $\boldsymbol{B}^T \boldsymbol{K}\hat{\boldsymbol{\alpha}}_i$ as shown by (10) for class i.

$$r_i = \|\boldsymbol{B}^T \boldsymbol{k}(\cdot, \boldsymbol{y}) - \boldsymbol{B}^T \boldsymbol{K}\hat{\boldsymbol{\alpha}}_i\|_2 \tag{10}$$

The procedure of KSRC is given as following.

Input: Matrix of training samples $\boldsymbol{X}$**,** testing sample $\boldsymbol{y}$.

First step: Select a Mercer kernel $k(\cdot,\cdot)$ and its parameters. And then build the kernel Gram matrix $\boldsymbol{K}$ and vector $\boldsymbol{k}(\cdot,\cdot)$.

Second step: Make the transformation matrix $\boldsymbol{B}$.

Third step: Normalize the columns of $\boldsymbol{B}^T\boldsymbol{K}$ and $\boldsymbol{B}^T \boldsymbol{k}(\cdot, \boldsymbol{y})$ to have unit L2-norm.

Fourth step: Compute the L1-minimization problem to get the coefficient vector $\hat{\boldsymbol{\alpha}}$.

Fifth step: Compute the residuals r_i.

Output: $\text{identity}(\boldsymbol{y}) = \arg\min_i r_i$.

C. Multiscale Retinex Algorithm

In real situation, the facial image has various luminance which could hinder the performance of face recognition.

Multiscale Retinex Algorithm (MSR) is an important processing algorithm to deal with the variation of luminance, which has been successfully used in face recognition. It removes low frequency information by dividing the smooth components from the original data. It has good performance on gray images. Therefore, it is widely used in face recognition under difficult lightning conditions. The MSR integrates weighted outputs of several different Single Scale Retinex (SSR). The SSR is defined as:

$$R_i(\boldsymbol{x},\boldsymbol{y}) = \log\left[\frac{I_i(\boldsymbol{x},\boldsymbol{y})}{F(\boldsymbol{x},\boldsymbol{y}) * I_i(\boldsymbol{x},\boldsymbol{y})}\right] \tag{11}$$

where $I_i(\boldsymbol{x},\boldsymbol{y})$ denotes the image distribution in the i-th spectral band. It presents gray-scale image when i equals 1. $F(\boldsymbol{x},\boldsymbol{y})$ is the surround function. The MSR is defines as

$$R_{MSR_i} = \sum_{s=1}^{S} w_s R_{si} \tag{12}$$

where S denotes the number of scales, R_{si} denotes the i-th component of the s-th scale, and w_s is the weight associated with the s scale. Fig. 1 shows original face images and processed images by MSR. It obviously shows that the MSR algorithm is effective for verification.

Figure 1. Example of MSR result when applied to face images with various illumination.

D. Weighted Kernel Sparse Representation-Based Classification

WKSRC is an extension of KSRC. In WKSRC, MSR algorithm is used to reduce the impact of illumination and effectively extract the locality information by penalizing the distance between the testing and each training sample. Furthermore, the L1-minimization problem is transformed into the Weighted L1-minimization problem in kernel feature space by combining the local information and kernel feature space.

To extract locality information by MSR algorithm, we use (12) shown as

$$R(\boldsymbol{x},\boldsymbol{y}) = \sum_{s=1}^{S}\left(\log\left[\frac{I(\boldsymbol{x},\boldsymbol{y})}{I(\boldsymbol{x},\boldsymbol{y}) * G_s(\boldsymbol{x},\boldsymbol{y})}\right]\right) \tag{13}$$

where $I(x,y)$ denotes the original image and $G_s(x,y)$ denotes the surround function of size s which is a free parameter.

In order to eliminate the impact of luminance and obtain more discriminating information, each training sample $\boldsymbol{x}_i$ in $\boldsymbol{X}$ and the testing sample y are processed by (14).

$$\begin{cases} R(\boldsymbol{x}_i) = \sum_{s=1}^{S}\left(\log\left[\frac{\boldsymbol{x}_i}{\boldsymbol{x}_i * G_s}\right]\right) \\ R(\boldsymbol{y}) = \sum_{s=1}^{S}\left(\log\left[\frac{\boldsymbol{y}}{\boldsymbol{y} * G_s}\right]\right) \end{cases} \tag{14}$$

The processed samples are robust to illumination variation. Furthermore, to obtain the locality adaptor, we compute the distance between the testing sample and each training sample by (15).

$$dist\big(R(\boldsymbol{y}), R(\boldsymbol{x}_i)\big) = \|R(\boldsymbol{y}) - R(\boldsymbol{x}_i)\|^t \tag{15}$$

where t denote the locality adaptor parameter. In general, the distance indicates the similarity between the testing sample and each training sample. The larger $dist\big(R(\boldsymbol{y}), R(\boldsymbol{x}_i)\big)$ is, the further distance between $\boldsymbol{y}$ and $\boldsymbol{x}_i$ is. Especially, the similarity removed the impact of luminance is characterized. With (18), the locality adaptor matrix $\boldsymbol{W}$ can be expressed by (16).

$$\boldsymbol{W} = \left[dist\big(R(\boldsymbol{y}), R(\boldsymbol{x}_1)\big), \dots, dist\big(R(\boldsymbol{y}), R(\boldsymbol{x}_N)\big)\right]^T \tag{16}$$

where N is the total number of training samples. This locality information can be used to generate sparser and discriminative coefficient vector in

$$\hat{\alpha} = \underset{i}{\operatorname{argmin}} \|\boldsymbol{W\alpha}\|_1, s.t. \|\boldsymbol{y} - \boldsymbol{X\alpha}\|_2 \leq \varepsilon \tag{17}$$

The same can be performed in kernel space as in

$$\widehat{\boldsymbol{\alpha}} = \underset{i}{\operatorname{argmin}} \|\boldsymbol{W\alpha}\|_1, s.t. \|\boldsymbol{B}^T\boldsymbol{k}(\cdot,\boldsymbol{y}) - \boldsymbol{B}^T\boldsymbol{K\alpha}\| \leq \varepsilon \tag{18}$$

Then, more robust coefficient vector can be obtained.

After getting coefficient vector $\widehat{\boldsymbol{\alpha}}$ from (18), The residual with c classes training samples will be calculated to judge the label of the testing sample. Similar to SRC and KSRC, the coefficient vector would be sparse that most items of $\widehat{\boldsymbol{\alpha}}$ are zero except some of those related with the class the testing sample belongs. To obtain the residual of class i, the new coefficient vector $\boldsymbol{\xi}$ which will represent the testing sample can be denoted as $\boldsymbol{\xi} = [\boldsymbol{0}^T, \dots, \widehat{\boldsymbol{\alpha}}_i^T, \dots, \boldsymbol{0}^T]$, where $\widehat{\boldsymbol{\alpha}}_i^T$ is the coefficient items which are associated with class i in $\widehat{\boldsymbol{\alpha}}$. Therefore, the residual of class i can be computed as

$$r_i = \|\boldsymbol{B}^T\boldsymbol{k}(\cdot,\boldsymbol{y}) - \boldsymbol{B}^T\boldsymbol{K\xi}\| \tag{19}$$

Because c classes samples are selected as training sample set, the label of testing sample can be computed easily by minimizing the c residuals.

We show the algorithm of WKSRC as below:

Input: Training image vectors $\boldsymbol{X} = [\boldsymbol{x_1}, \boldsymbol{x_2}, \dots, \boldsymbol{x_n}] \in \boldsymbol{R^{m\times n}}$, the testing sample $\boldsymbol{y}$, and an optional error tolerance $\varepsilon > 0$.

First step: Select a Mercer kernel $k(\cdot,\cdot)$ and its parameters. And then construct the kernel Gram matrix $\boldsymbol{K}$ and vector $\boldsymbol{k}(\cdot,\boldsymbol{y})$ by using Merce kernel.

Second step: Obtain the transformation matrix $\boldsymbol{B}$ by applying DR method with $\boldsymbol{K}$.
Third step: Normalize the columns of $\boldsymbol{B}^T\boldsymbol{K}$ and $\boldsymbol{B}^T\boldsymbol{k}(\cdot,\boldsymbol{y})$ to have unit L2-norm.
Fourth step: Process the samples by MSR algorithm.
Fifth step: Obtain the locality adaptor matrix $\boldsymbol{W}$ with (16).
Sixth step: Compute the weighted L1-minimization problem or the robust version considering occlusion in kernel feature space to get the coefficient vector.
Output: identity$(\boldsymbol{y}) = \arg\min_i\{r_i\}$.

E. Sparsified Collaborative Representation-Based Local Discriminant Projection for Feature Extraction

We will briefly explain SCRLDP. SCRLDP construct two graphs, within-class scatter matrix $\boldsymbol{S_w}$ and between-class scatter matrix $\boldsymbol{S_b}$. Two graphs are used to solve objective function defined as

$$J(p) = \arg\max_{\boldsymbol{P}} \frac{\boldsymbol{P}^T\tilde{\boldsymbol{S}}_{\boldsymbol{b}}\boldsymbol{P}}{\boldsymbol{P}^T\tilde{\boldsymbol{S}}_{\boldsymbol{w}}\boldsymbol{P}} \tag{20}$$

where $\boldsymbol{P}$ is projection matrix, and output matrix. $\boldsymbol{L} = [l_1, l_2, \dots, l_n]$ is label of each samples. Collaborative representation vector $\boldsymbol{v}_i$ is obtained by solving follow:

$$\boldsymbol{v_i} = argmin\{\|\boldsymbol{x_i} - \boldsymbol{X}\boldsymbol{v_i}\|_2^2| + \lambda||\boldsymbol{v_i}||_2^2\} \tag{21}$$

The collaborative representation vector $\boldsymbol{v}_i$ denotes similarity between $\boldsymbol{x}_i$ and all training samples. To enhance the connection between same label, the within-class weight matrix $\boldsymbol{H}$ is defined as follow:

$$\boldsymbol{H}_{ij} = \begin{cases} v_{ij}, & \text{if } l_i = l_j \\ 0, & \text{otherwise} \end{cases} \tag{22}$$

To use the separability information between different class, SCRLDP uses standard deviation and average of similarity matrix. Therefore, between-class weight matrix $\boldsymbol{H}'$ is defined as:

$$\boldsymbol{H}'_{ij} = \begin{cases} v_{ij}, & \text{if } l_i \neq l_j \text{ and } v_{ij} > \min\{\boldsymbol{v}'_i \geq \mu + \tau\sigma\} \\ 0, & \text{otherwise} \end{cases} \tag{23}$$

$\boldsymbol{v}'_i$ is defined by:

$$\boldsymbol{v'_i} = \begin{cases} v_{ij}, & \text{if } l_i = l_j \\ 0, & \text{otherwise} \end{cases} \tag{24}$$

where μ, σ and τ denote average of $\boldsymbol{v'_i}$, standard deviation of $\boldsymbol{v'_i}$ and hyperparameter respectively.

These weight matrices contribute what decreasing the total information value. Furthermore, SRCLDP constructs a scatter matrix effectively by using weight matrices. A within-class scatter matrix $\tilde{\boldsymbol{S}}_{\boldsymbol{w}}$ is defined as:

$$\begin{aligned}\tilde{\boldsymbol{S}}_{\boldsymbol{w}} &= \mathbf{X}[(\boldsymbol{D}^c + \boldsymbol{D}^r) - (\boldsymbol{H} + \boldsymbol{H}^T)]\boldsymbol{X}^T \\ &= \mathbf{X}(\widetilde{\boldsymbol{D}} - \widetilde{\boldsymbol{H}})\boldsymbol{X}^T\end{aligned} \tag{25}$$

where $\boldsymbol{D}^c$ and $\boldsymbol{D}^r$ are diagonal matrices, $\boldsymbol{D}_{ii}^c = \sum_{j=1}^n H_{ij}$ and $\boldsymbol{D}_{jj}^r = \sum_{i=1}^n H_{ij}$ respectively.

Likewise, $\tilde{\boldsymbol{S}}_{\boldsymbol{b}}$ is defined as follow:

$$\begin{aligned}\tilde{\boldsymbol{S}}_{\boldsymbol{b}} &= \boldsymbol{X}[(\boldsymbol{D}^{c\prime} + \boldsymbol{D}^{r\prime}) - (\boldsymbol{H}' + \boldsymbol{H}'^T)]\boldsymbol{X}^T \\ &= \boldsymbol{X}(\widetilde{\boldsymbol{D}'} - \widetilde{\boldsymbol{H}'})\boldsymbol{X}^T\end{aligned} \tag{26}$$

where $\boldsymbol{D}^{c\prime}$and $\boldsymbol{D}^{r\prime}$are diagonal matrices, $\boldsymbol{D}_{ii}^{c\prime} = \sum_{j=1}^n H_{ij}{}'$ and $\boldsymbol{D}_{jj}^{r\prime} = \sum_{i=1}^n H_{ij}{}'$ respectively.

By using above equations, target projection matrix $\boldsymbol{P}$ is calculated as

$$\tilde{\boldsymbol{S}}_{\boldsymbol{b}}\boldsymbol{P} = \boldsymbol{\lambda}\tilde{\boldsymbol{S}}_{\boldsymbol{w}}\boldsymbol{P} \tag{27}$$

We show the SCRLDP algorithm as below:
Input: Training image vectors $\boldsymbol{X} = [\boldsymbol{x_1}, \boldsymbol{x_2}, \dots, \boldsymbol{x_n}] \in \boldsymbol{R^{m\times n}}$, the label of each samples l_i $(i = 1,2,\dots,n)$
First step: Calculate the collaborative representation vector $\boldsymbol{v}_i$ by solving (21).
Second step: Create the within-class graph, between-class graph, and determine the weight matrices on basis of (22), (23) and (24).
Third step: Calculate $\tilde{\boldsymbol{S}}_{\boldsymbol{w}}$ and $\tilde{\boldsymbol{S}}_{\boldsymbol{b}}$ using (25) and (26).
Fourth step: Compute the eigenvectors associated with the first d largest eigenvalues of (27). where $\boldsymbol{p}_1, \boldsymbol{p}_2, \dots, \boldsymbol{p}_d$ are the calculated eigenvectors matching to the first *d* largest eigenvalues.
Output: The projection matrix: $\boldsymbol{P} = [\boldsymbol{p}_1, \boldsymbol{p}_2, \dots, \boldsymbol{p}_d]$.

III. Proposed Method

We propose the new method named weighted sparse representation using collaborative representation in kernel feature space based classification (WSRCRKC). In WKSRC, sparse representation is calculated by using kernel Gram matrix and dimension reduced kernel Gram matrix. Then, computational speed can be accelerated by making kernel Gram matrix sparse. Thus, we focused on SCRLDP which can make original matrix sparse with keeping relationship between same label. The objective of this method is to combine WKSRC with SCRLDP, which results in higher speed and accuracy.

$\boldsymbol{X}$ is defined as sample matrix $\boldsymbol{X} = [\boldsymbol{x}_1, \boldsymbol{x}_2, \dots, \boldsymbol{x}_n] \in \boldsymbol{R}^{n\times m}$ and sorted by class likes $\{\boldsymbol{x}_1, \boldsymbol{x}_2, \boldsymbol{x}_3\} \in$ class 1, $\{\boldsymbol{x}_4, \boldsymbol{x}_5, \boldsymbol{x}_6\} \in$ class 2, …, $\{\boldsymbol{x}_{n-2}, \boldsymbol{x}_{n-1}, \boldsymbol{x}_n\} \in$ class c. Let c is the number of class. $\boldsymbol{L} = [l_1, l_2, \dots, l_n]$ is label of each samples. Kernel Gram matrix $\boldsymbol{K}$ is generated by using kernel function like (7).

To apply SCRLDP for kernel Gram matrix, we modified some equations of SCRLDP. Firstly, we modify the collaborative representation vector. In SCRLDP, $\boldsymbol{v}_i$ is computed by (21). However, it cannot deal with kernel Gram matrix. Therefore, (21) is modified as:

$$\boldsymbol{v_i} = \operatorname{argmin}\{\|\boldsymbol{K_i} - \boldsymbol{K}\boldsymbol{v_i}\|_2^2 + \lambda\|\boldsymbol{K_i}\|_2^2\} \tag{28}$$

where $\boldsymbol{K}_i$ denotes i-th column (or row) of $\boldsymbol{K}$ and $\boldsymbol{v}_i = [v_{i,1}, \dots, v_{i,i-1}, 0, v_{i,i+1}, \dots, v_{i,n}]$.

The element of collaborative representation vector $\boldsymbol{v_i}$ shows similarity between two sample in SCRLDP. In kernel feature space, the relationships between same label samples and different label samples are not changed from original. Therefore, we can apply (22), (23), (24), (25), (26) and (27) similarly. According to above mentioned, our algorithm of proposed method is determined and shown as:

Input: Training image vectors $\boldsymbol{X} = [\boldsymbol{x_1}, \boldsymbol{x_2}, .., \boldsymbol{x_n},] \in \boldsymbol{R}^{m \times n}$, the testing sample $\boldsymbol{y}$, and an optional error tolerance $\varepsilon > 0$.

First step: Select a Mercer kernel $k(\cdot,\cdot)$ and its parameters. And then construct the kernel Gram matrix $\boldsymbol{K}$ and vector $\boldsymbol{k}(\cdot, \boldsymbol{y})$ by using Merce kernel.

Second step: Obtain the transformation matrix $\boldsymbol{B}$ by applying SCRLDP with $\boldsymbol{K}$.

Third step: Normalize the columns of $\boldsymbol{B}^T\boldsymbol{K}$ and $\boldsymbol{B}^T\boldsymbol{k}(\cdot, \boldsymbol{y})$ to have unit L2-norm.

Fourth step: Process the samples by MSR algorithm.

Fifth step: Obtain the locality adaptor matrix $\boldsymbol{W}$ with (16).

Sixth step: Compute the weighted L1-minimization problem or the robust version considering occlusion in kernel feature space to get the coefficient vector.

Output: $\text{identity}(\boldsymbol{y}) = \arg\min_i\{r_i\}$.

IV. Experiments

We conduct experiments to evaluate our method on Extend Yale B database [17], which has images in various luminance situations. The experiments were programmed in Python (3.7.3). In the experiments, we used the Sparse Modeling Software (SPAMS) package [18], [19] to solve the weighted L1-minimization problem in kernel feature space. SPAMS is an open-source optimization toolbox for solving machine learning and signal processing problems.

From here, we will write experimental environment, result and consideration. The database includes 28 different classes under 9 poses and 64 illumination. We randomly sample one illumination from each pose on each class, which results in 252 samples. In this experiment, all images were resized into 40×40 density. Then we equally split samples into training and testing data.

To compare our proposed method with WKSRC, we choose PCA and LDA as DR method, which are widely used in pattern recognition. When the experiment is conducted, there is necessary to choose Mercer kernel. Widely used Merce kernels are the linear kernel and the Gaussian Radial Basis Function (RBF). To reduce the impact of hyper parameter, we focus only on linear kernel.

Mercer kernel $k(\boldsymbol{x}, \boldsymbol{y})$ is the linear kernel in this experiment, which defined as (29):

$$k(\boldsymbol{x}, \boldsymbol{y}) = \boldsymbol{x}^{\mathrm{T}}\boldsymbol{y} \tag{29}$$

TABLE I. Comparison of Average Accuracy

Number of dimensions	Methods		
	WKSRC with PCA	WKSRC with LDA	WSRCRKC
25	0.8150	0.8293	0.7875
50	0.8614	0.8711	0.8789
75	0.8786	0.8964	0.8850
100	0.8925	0.8436	0.8975

TABLE II. Comparison of Computational Time

Number of dimensions	Methods		
	WKSRC with PCA	WKSRC with LDA	WSRCRKC
25	51.2443	61.2360	36.8070
50	53.8723	69.5778	39.8374
75	63.9410	77.2630	40.1678
100	74.4228	81.4671	40.6945

In all methods, we set $\varepsilon = 0.0005$ as the given tolerance. In the procedure of locality information extraction, we set $t = 1.5$. When conducting MSR algorithm, three scales are enough for most images according to [16]. Thus, we set $S = 3$ in weighted methods. We ran the all methods three times and calculated the average of accuracy and computational time to obtain the not biased result. We show the comparison about computational time and accuracy between our proposed method and other existed methods in Table I and Table II. Fig. 2 and Fig. 3 indicate what the performance how to change depend on dimension number.

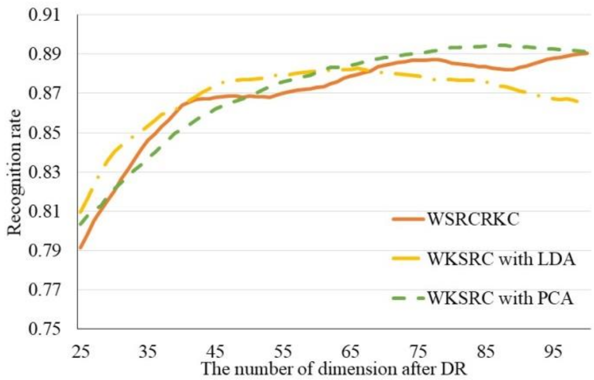

Figure 2. The comparison of accuracy.

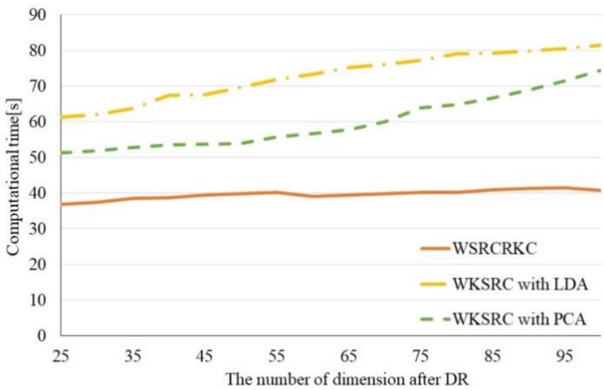

Figure 3. The comparison of computational time.

From Table I and Table II, our proposed method has lower computational time than PCA and LDA though keeping the high accuracy. According to Fig. 2, accuracies of PCA and proposed method is stable around 75.

In addition, computational time of proposed method is stably faster than other methods over 10. After integration above, our proposed method shows good performance.

V. Conclusion

We proposed the method named Weighted Sparse Representation using Collaborative Representation in Kernel feature space based Classification (WSRCRKC) in this paper.

According to the experiment on Extend Yale B database, our proposed method shows better performance on the computational time while keeping high accuracy for images affected by various luminance. As a reason, collaborative representation makes matrix using similarity between different sample. However, proposed method increases 0 element of matrix by removing low similarity. Therefore, misleading elements are eliminated. In addition, if the ratio of 0 components was large, the calculation cost is reduced. Therefore, computational time can be decreased.

In the future, we will focus on modifying by extracting essential information more effectively and efficiently.

Conflict of Interest

The authors declare no conflict of interest.

Author Contributions

Matsushima planned and implemented the idea for the whole study; Matsusue and Ruengprateepsang experimented; Matsushima and Matsusue analyzed the data and wrote a paper; all authors had approved the final version.

References

[1] S. Reiß, A. Roitberg, M. Haurilet, and R. Stiefelhagen, "Deep classification-driven domain adaptation for cross-modal driver behavior recognition," in *Proc. IEEE Intelligent Vehicles Symposium*, Oct. 2020.

[2] Y. Wang, X. Ban, H. Wang, D. Wu, H. Wang, S. Yang, S. Liu, and J. Lai, "Detection and classification of moving vehicle from video using multiple spatio-temporal features," *IEEE Access*, vol. 7, pp. 80287-80299, Jun. 2019.

[3] E. Derman and A. A. Salah, "Continuous real-time vehicle driver authentication using convolutional neural network based face recognition," in *Proc. IEEE International Conference on Automatic Face and Gesture Recognition*, May 2018, pp. 577-584.

[4] I. T. Jolliffe, *Principal Component Analysis*, Springer Verlag, 1986.

[5] D. García -Gil, S. Ramírez -Gallegoa, S. Garcíaa, and F. Herrera, "Principal components analysis random discretization ensemble for big data," *Knowledge-Based Systems*, vol. 150, pp. 166-174, Jun. 2018.

[6] P. N. Belhumeur, J. P. Hespanha, and D. J. Kriegman, "Eigenfaces vs. fisherfaces: Recognition using class specific linear projection," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 19, no. 7, pp. 711-720, Jul. 1997.

[7] F. Nie, Z. Wang, R. Wang, Z. Wang, and X. Li, "Adaptive local linear discriminant analysis," *ACM Transactions on Knowledge Discovery from Data*, vol. 14, no. 1, pp. 1-19, Feb. 2020.

[8] J. Wright, A. Y. Yang, A. Ganesh, 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, Feb. 2008.

[9] M. Abavisani and V. M. Pate, "Deep sparse representation-based classification," *IEEE Signal Processing Letters*, vol. 26, no. 6, pp. 948-952, Apr. 2019.

[10] M. Matsusue, K Sae-pae, and K. Matsushima, "Sparsified collaborative representation based local discriminant projection for feature extraction," in *Proc. International Conference on Green and Human Information Technology*, Feb. 2019, part 1, pp. 187-189.

[11] B. Schölkopf, A. J. Smola, and K. R. Müller, "Nonlinear component analysis as a kernel eigenvalue problem," *Neural Computation*, vol. 10, no. 5, pp. 1299-1319, Jul. 1998.

[12] T. Zhou and Y. Peng, "Kernel principal component analysis-based gaussian process regression modelling for high-dimensional reliability analysis," *Computers and Structures*, vol. 241, pp. 1-22, Dec. 2020.

[13] J. Yin, X. Liu, Z. Jin, and W. Yang, "Kernel sparse representation based classification," *Neurocomputing*, vol. 77, pp. 120-128, Feb. 2012.

[14] D. Li, Q. Wang, and F. Kong, "Adaptive kernel sparse representation based on multiple feature learning for hyperspectral image classification," *Neurocomputing*, vol. 440, pp. 97-112, Aug. 2020.

[15] X. Liu, L. Lu, Z. Shen, and K. Lu, "A novel face recognition algorithm via weighted kernel sparse representation," *Future Generation Computer Systems*, vol. 80, pp. 653-663, Mar. 2018.

[16] D. Jobson, Z. Rahman, and G. Woodell, "A multiscale retinex for bridging the gap between color images and the human observation of scenes," *IEEE Transactions on Image Processing*, vol. 6, no. 7, pp. 965-976, Jul. 1997.

[17] 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 Transaction on Pattern Analysis and Machine Intelligence*, vol. 23, no. 6, pp. 643-660, Jun. 2001.

[18] J. Mairal, F. Bach, J. Ponce, and G. Sapiro, "Online dictionary learning for sparse coding," in *Proc. the 26th Annual International Conference on Machine Learning*, Jun. 2009, pp. 689-696.

[19] J. Mairal, F. Bach, J. Ponce, and G. Sapiro, "Online learning for matrix factorization and sparse coding," *Journal of Machine Learning Research*, vol. 11, pp. 19-60, Mar. 2010.

Kousuke Matsushima received his M.S and Ph.D. degree from Kumamoto University in 2002 and 2008, respectively. In 2002, he joined Oki Electric Industry Co., Ltd. In 2008, he joined National Institute of Technology, Kurume College, where he is currently an associate professor. He was a Visiting Researcher at McMaster University (Canada) in 2008. His research interests are image processing and recognition, computer vision, and other fields. He is a member of the IEICE in Japan.

Mihoshi Matsusue graduated Advanced Engineering Course, National Institute of Technology, Kurume College, Fukuoka, Japan in 2020. He is a student of Graduate School of Information Sciences, Tohoku University, Sendai, Japan. His research interests are computer vision and deep learning. Currently, he is studying character recognition using Convolutional Neural Network.

Kavin Ruengprateepsang graduated Bachelor of Science from Faculty of Information Technology, King Mongkut's Institute of Technology Ladkrabang, Bangkok, Thailand in 2020. He is pursuing a M.S. degree at King Mongkut's Institute of Technology Ladkrabang. His research interests include image processing, video processing and biometrics. Currently, he is studying biometric authentication using lip movement.