Jointly Learning Non-negative Projection and Dictionary with Discriminative Graph Constraints for Classification

Weiyang Liu, Zhiding Yu, Yingzhen Yang, Meng Yang, and Thomas S. Huang

***Abstract*—Dictionary learning (DL) for sparse coding has shown impressive performance in classification tasks. But how to select a feature that can best work with the learned dictionary remains an open question. Current prevailing DL methods usually adopt existing well-performing features, ignoring the inner relationship between dictionaries and features. To address the problem, we propose a joint non-negative projection and dictionary learning (JNPDL) method. Non-negative projection learning and dictionary learning are complementary to each other, since the former leads to the intrinsic discriminative parts-based features for objects while the latter searches a suitable representation in the projected feature space. Specifically, discrimination of projection and dictionary is achieved by imposing to both projection and coding coefficients a graph constraint that maximizes the intra-class compactness and inter-class separability. Experimental results on both image classification and image set classification show the excellent performance of JNPDL by being comparable or outperforming many state-of-the-art approaches.**

***Index Terms*—Dictionary Learning, Non-negative Projection, Discriminative Graph Constraints**

I. Introduction

Over the past years, sparse coding has been widely applied in a variety of computer vision problems. Sparse coding seeks to represent a signal as a sparse linear combination of bases, i.e., a dictionary of atoms. The dictionary plays an important role as it is expected to faithfully and discriminatively represent the query signal. [40] proposed the sparse representation-based classification (SRC) which treats the entire training set as a structured dictionary. Methods taking off-the-shelf bases (e.g., wavelets) as the dictionary were also proposed [11]. But research also indicates that such strategy may not be optimal for classification. Current prevailing approaches mostly tend to learn the desired dictionary directly from training data, many of which have leaded to state-of-the-art results in image restoration [1], [24] face recognition [50], [13], [16], [44], [45], image classification [24], [52], [28], [8], [43]. These dictionary updating strategies are referred to as dictionary learning (DL). DL receives significant attention for its excellent representation power. Such advantage mainly comes from the fact that allowing the update of dictionary atoms often results in additional flexibility to discover better intrinsic signal patterns, therefore leading to more robust representations.

W. Liu is with the School of Electronic and Computer Engineering, Peking University. Z. Yu is with the Department of Electrical and Computer Engineering, Carnegie Mellon University. Y. Yang is with the Dept. of ECE, University of Illinois at Urbana-Champaign. M. Yang is the corresponding author (yang.meng@szu.edu.cn) and with the College of Computer Science and Software Engineering, Shenzhen University, Shenzhen, China.

Manuscript received xx, xxxx; revised August xx, xxxx.

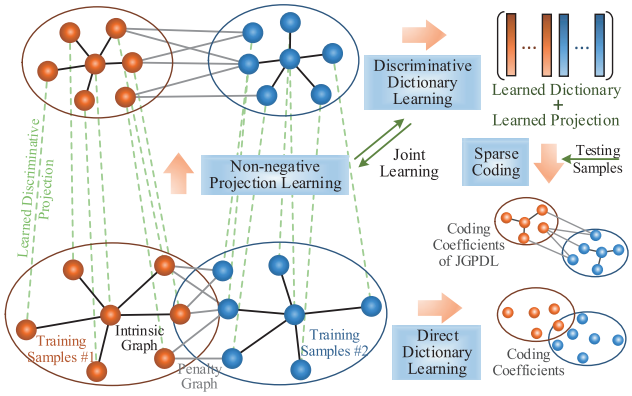

Fig. 1. An illustration of JNPDL. The dictionary and non-negative projection are jointly learned with discriminative graph constraints.

DL methods can be broadly categorized as unsupervised DL and supervised DL. A well-known unsupervised DL method is the K-SVD [1] which learns an over-complete dictionary to represent the query from unlabeled image patches. Recent studies on unsupervised DL includes image denoising [4], feature coding [42], etc. However, due to the lack of label information, unsupervised DL only guarantees to discover patterns that represent a signal sparsely, but not necessarily discriminatively.

Trained with labeled samples, the supervised DL exploits the discriminative information among classes and requires the learned dictionary not only need to well represent the images but also need to discriminatively represent them. [50] proposed a discriminative K-SVD algorithm that trains an optimal classifier with the dictionary and representation coefficients by simultaneously optimizing the classification error. To further enhance its discrimination, [13] strengthens the supervised label information by adding a label consistency term. A considerable number of regularization terms associated to the dictionary were introduced, including reducing the dictionary coherence [30], requiring a sub-dictionary to well represent certain classes but poorly for the others [2] and enhancing the discrimination via entropy-based criteria [29]. To exploit stronger discrimination, [44], [45] learn a discriminative dictionary via the Fisher discrimination criteria that enforces coding coefficients to have small within-class scatter but big between-class scatter. Very recently, [43] proposes a latent DL

method by jointly learning a latent matrix to adaptively build the relationship between dictionary atoms and class labels.

Different from the wide variety of conventional discriminative DL literatures, our work casts an alternative view on this problem. Instead of exploiting more discrimination in dictionary, we consider optimizing the input feature to further improve the learned dictionary. We believe such process can considerably influence the quality of learned dictionary, while a poor dictionary directly degrades subsequent classifications. Therefore, conducting DL on selected features without considering their inner relationship may be suboptimal. One of the major purposes of this paper is to jointly learn a feature projection that improves DL.

Inspired by the fact that only parts of features are discriminative for classification, we aim to learn a part-based projection that focuses on discrimination power. To achieve that, we propose to learn a non-negative projection, following the idea of non-negative matrix factorization (NMF) [17]. Non-negative property only allows additive, not subtractive, combinations, leading to parts-based representation. In the light of [47], [21], we formulate a non-negative parts-based projection model. Specifically, we constrain the projection matrix with non-negativity and jointly learn the projection and the non-negative basis. A novel NMF-like constraint that can be viewed as a tradeoff between NMF and feature learning [53] is proposed to learn the projection.

We propose the joint non-negative projection and dictionary learning (JNPDL) method with discriminative graph constraints. The basic idea of JNPDL is illustrated in Fig. 1. The graph constraints are constructed using the graph embedding framework [41]. An intrinsic graph is constructed to characterize the favorable relationship among training samples, and a penalty graph to characterize the unfavourable relationship. These two graphs are involved in the discriminative projection constraint, aiming to maximize the inter-class separability and the intra-class compactness. Similarly, we constrain the coding coefficients to be discriminative with a graph constraint that makes the coding coefficients within the same class to be similar and the coefficients among different classes to be dissimilar. These two graph constraints are essentially the same, but they are formulated in a different way for the convenience of optimization. Besides, we also adopt a discriminative reconstruction constraint so that the coding coefficients can only well represent samples of their own class but poorly for samples of the other classes. We test JNPDL in both image classification and image set classification for comprehensive evaluation. Extensive experimental results show that JNPDL is comparable or outperforms many state-of-the-art approaches.

II. Related Works

Only a few works [22], [5], [49] have discussed the idea of jointly learning the transformation of training samples and dictionary, but reported more competitive performance than conventional DL methods. [22] proposed a learning framework which simultaneously learns the feature and the dictionary for image set based face recognition. Their work focuses on learning a reconstructive feature (filterbank) using similar idea in [53], while ours is to learn a discriminative non-negative projection. [5] jointly learns a dimensionality reduction matrix and a discriminative dictionary for face recognition. Unlike JNPDL, their model focuses more on low-dimensional representation and therefore learns the dimensionality reduction matrix without discrimination constraints. The discrimination is enforced by a Fisher-like constraint on the coding coefficients. [49] presents a simultaneous projection and dictionary learning method using a carefully designed sigmoid reconstruction error (the ratio of intra-class error to inter-class error in a sigmoid function). Their method represents the input samples by the multiplication of projection matrix and dictionary, which is also different from our model.

III. The Proposed JNPDL Framework

A. The JNPDL Model

Let $\boldsymbol{Y}$ be a set of s-dimensional training samples, i.e., $\boldsymbol{Y} = \{\boldsymbol{Y}_1, \boldsymbol{Y}_2, \cdots, \boldsymbol{Y}_K\}$ where $\boldsymbol{Y}_i$ denotes the training samples from class i. The learned structured (class-specific) dictionary is denoted by $\boldsymbol{D} = \{\boldsymbol{D}_1, \boldsymbol{D}_2, \cdots, \boldsymbol{D}_K\}$ and the corresponding sparse representation of the training samples over dictionary $\boldsymbol{D}$ is defined as $\boldsymbol{X} = \{\boldsymbol{X}_1, \boldsymbol{X}_2, \cdots, \boldsymbol{X}_K\}$. $\boldsymbol{M}$ is defined as an intermediate non-negative basis matrix. The jointly learned projection matrix is written as $\boldsymbol{P} \in \mathbb{R}^{s_p \times s}$. In order to map the training samples into a discriminative space to avoid the high correlation among training sample from different classes and benefit the sparse coding stage thereafter, JNPDL jointly learns a discriminative non-negative projection and a discriminative structured dictionary. When a testing sample comes, we first use the learned projection matrix to project the sample to a discriminative space and then encode the sample over the learned discriminative dictionary. JNPDL model is formulated as

$$\begin{aligned}\langle \boldsymbol{D}, \boldsymbol{M}, \boldsymbol{P}, \boldsymbol{X} \rangle = \arg \min_{\boldsymbol{D},\boldsymbol{M}\geq\boldsymbol{0},\boldsymbol{P}\geq\boldsymbol{0},\boldsymbol{X}} \big\{ & R(\boldsymbol{D}, \boldsymbol{P}, \boldsymbol{X}) + \\ & \alpha_1 G_p(\boldsymbol{P}, \boldsymbol{M}) + \alpha_2 G_c(\boldsymbol{X}) + \alpha_3 \|\boldsymbol{X}\|_1 \big\}\end{aligned} \tag{1}$$

where $\alpha_1, \alpha_2, \alpha_3$ are scalar constants, $R(\boldsymbol{D}, \boldsymbol{P}, \boldsymbol{X})$ is the discriminative reconstruction error, $G_p(\boldsymbol{P}, \boldsymbol{M})$ is the graph-based projection term, $G_c(\boldsymbol{X})$ is the graph-based coding coefficients term and $\|\boldsymbol{X}\|_1$ denotes the l_1 sparsity penalty. We essentially adopt the idea of marginal Fisher graph embedding [41] to bring in discrimination.

B. Discriminative Reconstruction Error

We construct the discriminative reconstruction error from three objectives: minimizing the global reconstruction error, minimizing the local reconstruction error and maximizing the non-local reconstruction error. Local reconstruction stands for the reconstruction only with the atoms of the same class. The reconstruction error term is defined as

$$\begin{aligned}R(\boldsymbol{D}, \boldsymbol{P}, \boldsymbol{X}) = & \|\boldsymbol{P}\boldsymbol{Y} - \boldsymbol{D}\boldsymbol{X}\|_F^2 + \sum_{i=1}^{K} \|\boldsymbol{P}\boldsymbol{Y}_i - \boldsymbol{D}_i \boldsymbol{X}_i^i\|_F^2 \\ & + \sum_{i=1}^{K} \sum_{j=1, j\neq i}^{K} \|\boldsymbol{D}_j \boldsymbol{X}_i^j\|_F^2\end{aligned} \tag{2}$$

where $\boldsymbol{X}_i^j$ denotes the coding coefficients of samples $\boldsymbol{Y}_i$ associated with the sub-dictionary $\boldsymbol{D}_j$. $\|\boldsymbol{Y}-\boldsymbol{D}\boldsymbol{X}\|_F^2$ represents the global reconstruction error that implies the global dictionary $\boldsymbol{D}$ should well represent the input samples. $\sum_{i=1}^{K}\|\boldsymbol{Y}_i-\boldsymbol{D}_i\boldsymbol{X}_i^i\|_F^2$ denotes the local reconstruction error that requires the local sub-dictionaries $\boldsymbol{D}_i^i$ well represent samples from class i. The coding coefficients $\boldsymbol{X}_i^i$ should be large such that $\|\boldsymbol{Y}_i-\boldsymbol{D}_i\boldsymbol{X}_i^i\|_F^2$ can be small. $\boldsymbol{X}_i^j, i\neq j$ should have small energy such that $\|\boldsymbol{D}_j\boldsymbol{X}_i^j\|_F^2$ is small.

C. Graph-based Coding Coefficients Term

We constrain the coding coefficients from the same class to be similar and the coding coefficients from different classes to be significantly dissimilar. To achieve this, we first construct an intrinsic graph for intra-class compactness and a penalty graph for inter-class separability. After rewriting the sparse representation $\boldsymbol{X}$ as $\boldsymbol{X}=\{\boldsymbol{x}_1,\boldsymbol{x}_2,\cdots,\boldsymbol{x}_N\}$ where N is the number of training samples, the similarity matrix of the intrinsic graph is defined by

$$\{\boldsymbol{W}_c\}_{ij}=\begin{cases}1, & \text{if } i\in S_{k_1}^+(j) \text{ or } j\in S_{k_1}^+(i)\\ 0, & \text{otherwise}\end{cases} \tag{3}$$

where $S_{k_1}^+(i)$ indicates the index set of the k_1 nearest neighbors of the sample $\boldsymbol{x}_i$ in the same class. Similarly, the similarity matrix of the penalty graph is defined by

$$\{\boldsymbol{W}_c^p\}_{ij}=\begin{cases}1, & j\in S_{k_2}^-(i) \text{ or } i\in S_{k_2}^-(j)\\ 0, & \text{otherwise}\end{cases} \tag{4}$$

where $S_{k_2}^-(i)$ denotes the index set of the k_2 nearest neighbors of the sample $\boldsymbol{x}_i$ from the other classes (not the class that $\boldsymbol{x}_i$ belongs to). To maximize both the intra-class compactness and inter-class separability of coding coefficients, we construct the discriminative coefficients term as

$$\begin{aligned}G_c(\boldsymbol{X})=&\sum_{i=1}^{N}\sum_{i\in S_{k_1}^+(j)\ or\ j\in S_{k_1}^+(i)}\frac{1}{2}\|\boldsymbol{x}_i-\boldsymbol{x}_j\|_2^2-\\ &\sum_{i=1}^{N}\sum_{i\in S_{k_2}^-(j)\ or\ j\in S_{k_2}^-(i)}\frac{1}{2}\|\boldsymbol{x}_i-\boldsymbol{x}_j\|_2^2\\ =&\mathrm{Tr}(\boldsymbol{X}^T\boldsymbol{L}_c\boldsymbol{X})-\mathrm{Tr}(\boldsymbol{X}^T\boldsymbol{L}_c^p\boldsymbol{X})\end{aligned} \tag{5}$$

where $\boldsymbol{L}_c=\boldsymbol{B}_c-\boldsymbol{W}_c$ in which $\boldsymbol{B}_c=\sum_{j\neq i}\{\boldsymbol{W}_c\}_{ij}$ and $\boldsymbol{L}_c^p=\boldsymbol{B}_c^p-\boldsymbol{W}_c^p$ in which $\boldsymbol{B}_c^p=\sum_{j\neq i}\{\boldsymbol{W}_c^p\}_{ij}$. Imposing the graph-based discrimination makes the coding coefficients more discriminative. Interestingly, most existing discriminative coding coefficients term, such as in [45], [22], are special cases of the graph-based discrimination constraint.

D. Graph-based Non-negative Projection Term

We aim to learn a non-negative projection that can preserves the useful information and map the training samples to a discriminative space where training samples from different classes have low correlation and become more discriminative. Inspired by projective non-negative graph embedding [21], we design a structured projection matrix by dividing the projection matrix $\boldsymbol{P}$ into two parts:

$$\boldsymbol{Y}^{proj}=\begin{bmatrix}\hat{\boldsymbol{Y}}^{proj}\\ \tilde{\boldsymbol{Y}}^{proj}\end{bmatrix}=\boldsymbol{P}\boldsymbol{Y}=\begin{bmatrix}\hat{\boldsymbol{P}}\\ \tilde{\boldsymbol{P}}\end{bmatrix}\boldsymbol{Y} \tag{6}$$

where $\hat{\boldsymbol{Y}}^{proj}=\{\hat{\boldsymbol{y}}_1^{proj},\cdots,\hat{\boldsymbol{y}}_N^{proj}\}\in\mathbb{R}^{q\times N}$ that serves for the certain purpose of graph embedding, and $\tilde{\boldsymbol{Y}}^{proj}=\{\tilde{\boldsymbol{y}}_1^{proj},\cdots,\tilde{\boldsymbol{y}}_N^{proj}\}\in\mathbb{R}^{(s_p-q)\times N}$ that contains the additional information for data reconstruction ($\hat{\boldsymbol{Y}}^{proj}$ is a relaxed matrix that compensates the information loss in $\hat{\boldsymbol{Y}}^{proj}$). Note that $\hat{\boldsymbol{Y}}^{proj}$ preserves the discriminative graph properties while the whole $\boldsymbol{Y}^{proj}$ is used for data reconstruction purpose. Therefore the purposes of data reconstruction and graph embedding coexist harmoniously and do not mutually compromise like conventional formulations with multiple objectives. The basis matrix $\boldsymbol{M}$ is correspondingly divided into two parts $\boldsymbol{M}=\{\hat{\boldsymbol{M}},\tilde{\boldsymbol{M}}\}$ in which $\hat{\boldsymbol{M}}\in\mathbb{R}^{s\times q}$ and $\tilde{\boldsymbol{M}}\in\mathbb{R}^{s\times(s_p-q)}$. $\tilde{\boldsymbol{M}}$ can be considered as the complementary space of $\hat{\boldsymbol{M}}$.

We first define $\boldsymbol{y}_j^{proj}$ as the jth column vector of $\boldsymbol{Y}^{proj}$ and then construct the intrinsic graph and the penalty graph using the same procedure as graph-based coding coefficients term. The construction of the similarity matrix $\boldsymbol{W}_p$ and $\boldsymbol{W}_p^p$ for intrinsic graph and the penalty graph is identical to $\boldsymbol{W}_c$ and $\boldsymbol{W}_c^p$, except that $\boldsymbol{W}_p$ and $\boldsymbol{W}_p^p$ measure the similarities among features and adopt different parameters. Differently, $\boldsymbol{W}_c$ and $\boldsymbol{W}_c^p$ measure similarities among coding coefficients. As [41] suggests, we have two objectives to preserve graph properties and enhance discrimination:

$$\begin{cases}\max_{\hat{\boldsymbol{Y}}^{proj}}\sum_{i\neq j}\|\hat{\boldsymbol{y}}_i^{proj}-\hat{\boldsymbol{y}}_j^{proj}\|_2^2\{\boldsymbol{W}_p^p\}_{ij}\\ \min_{\hat{\boldsymbol{Y}}^{proj}}\sum_{i\neq j}\|\hat{\boldsymbol{y}}_i^{proj}-\hat{\boldsymbol{y}}_j^{proj}\|_2^2\{\boldsymbol{W}_p\}_{ij}\end{cases}. \tag{7}$$

Because $\sum_{i\neq j}\|\boldsymbol{y}_i^{proj}-\boldsymbol{y}_j^{proj}\|_2^2\{\boldsymbol{W}_p\}_{ij}=\sum_{i\neq j}\|\hat{\boldsymbol{y}}_i^{proj}-\hat{\boldsymbol{y}}_j^{proj}\|_2^2\{\boldsymbol{W}_p\}_{ij}+\sum_{i\neq j}\|\tilde{\boldsymbol{y}}_i^{proj}-\tilde{\boldsymbol{y}}_j^{proj}\|_2^2\{\boldsymbol{W}_p\}_{ij}$, for a specific $\boldsymbol{Y}^{proj}$, minimizing the objective function with respect to $\tilde{\boldsymbol{Y}}^{proj}$ is equivalent to maximizing the objective function associated with the complementary part, i.e., $\hat{\boldsymbol{Y}}^{proj}$. Thus we constrain the projection matrix with the following equivalent objective:

$$\begin{cases}\min_{\hat{\boldsymbol{P}}}\mathrm{Tr}(\hat{\boldsymbol{P}}\boldsymbol{Y}\boldsymbol{L}_p\boldsymbol{Y}^T\hat{\boldsymbol{P}}^T)\\ \min_{\tilde{\boldsymbol{P}}}\mathrm{Tr}(\tilde{\boldsymbol{P}}\boldsymbol{Y}\boldsymbol{L}_p^p\boldsymbol{Y}^T\tilde{\boldsymbol{P}}^T)\end{cases} \tag{8}$$

where $\boldsymbol{L}_p=\boldsymbol{B}_p-\boldsymbol{W}_p$ in which $\boldsymbol{B}_p=\sum_{j\neq i}\{\boldsymbol{W}_p\}_{ij}$ and $\boldsymbol{L}_p^p=\boldsymbol{B}_p^p-\boldsymbol{W}_p^p$ in which $\boldsymbol{B}_p^p=\sum_{j\neq i}\{\boldsymbol{W}_p^p\}_{ij}$. we formulate the graph-based projection term as follows:

$$\begin{aligned}G_p(\boldsymbol{P},\boldsymbol{M})=&\|\boldsymbol{Y}-\boldsymbol{M}\boldsymbol{P}\boldsymbol{Y}\|_F^2+\beta\cdot\mathrm{Tr}(\hat{\boldsymbol{P}}\boldsymbol{Y}\boldsymbol{L}_p\boldsymbol{Y}^T\hat{\boldsymbol{P}}^T)\\ &+\beta\cdot\mathrm{Tr}(\tilde{\boldsymbol{P}}\boldsymbol{Y}\boldsymbol{L}_p^p\boldsymbol{Y}^T\tilde{\boldsymbol{P}}^T)+\|\boldsymbol{M}-\boldsymbol{P}^T\|_F^2\end{aligned} \tag{9}$$

in which β is a scaling constant and each column of $\boldsymbol{M}$ is normalized to unit l_2 norm. We use $\|\boldsymbol{Y}-\boldsymbol{M}\boldsymbol{P}\boldsymbol{Y}\|_F^2+\|\boldsymbol{M}-\boldsymbol{P}^T\|_F^2$ to incorporate the projection matrix into a NMF-like framework, and these two terms can further ensure the reconstruction ability of the projection $\boldsymbol{P}$ and avoid the trivial solutions of $\boldsymbol{P}$. they serve similar role to the auto-encoder style reconstruction penalty term in [54], [33], [53]. The other terms in $G_p(\boldsymbol{P})$ preserve the graph properties and enhance discrimination.

IV. Optimization

We adopt a standard iterative learning framework to jointly learn the sparse representation $\boldsymbol{X}$, the non-negative projection matrix $\boldsymbol{P}$, the intermediate non-negative basis matrix $\boldsymbol{M}$ and the dictionary $\boldsymbol{D}$. The proposed algorithm is shown in Algorithm 1. In Fig. 2, its convergence evaluation on extended YaleB shows JNPDL converges efficiently and the non-negative projection learning also converges as we prove in **Appendix A**.

A. Non-negative projection learning

To learn the non-negative projection, we optimize $\boldsymbol{P}, \boldsymbol{M}$ with $\boldsymbol{D}, \boldsymbol{X}$ fixed. Thus Eq. (1) is rewritten as

$$\begin{aligned}\min_{\boldsymbol{P}\geq\boldsymbol{0},\boldsymbol{M}\geq\boldsymbol{0}} \ \big\{&\|\boldsymbol{PY}-\boldsymbol{DX}\|_F^2+\sum_{i=1}^{K}\|\boldsymbol{PY}_i-\boldsymbol{D}_i\boldsymbol{X}_i^i\|_F^2\\ &+\alpha_1\|\boldsymbol{Y}-\boldsymbol{MPY}\|_F^2+\alpha_1\beta\cdot\mathrm{Tr}(\hat{\boldsymbol{P}}\boldsymbol{Y}\boldsymbol{L}_p\boldsymbol{Y}^T\hat{\boldsymbol{P}}^T)\\ &+\alpha_1\beta\cdot\mathrm{Tr}(\tilde{\boldsymbol{P}}\boldsymbol{Y}\boldsymbol{L}_p^p\boldsymbol{Y}^T\tilde{\boldsymbol{P}}^T)+\alpha_1\|\boldsymbol{M}-\boldsymbol{P}^T\|_F^2\big\}\end{aligned} \tag{10}$$

which is essentially a projective non-negative matrix factorization problem [47], [21]. We use the multiplicative iterative solution [47], [21], [34] to solve Eq. (10). Specifically, we transform it into tractable sub-problems and optimize $\boldsymbol{M}$ and $\boldsymbol{P}$ by a multiplicative non-negative iterative procedure.

Because $\boldsymbol{M}$ is the basis matrix, it is necessary to require each column $\boldsymbol{m}_i$ to have unit l_2 norm, i.e., $\|\boldsymbol{m}_i\|=1$. This extra constraint makes the optimization more complicated, so we compensate the norms of the basis matrix into the coefficient matrix as in [34] and replace $\alpha_1\beta\mathrm{Tr}(\hat{\boldsymbol{P}}\boldsymbol{Y}\boldsymbol{L}_p\boldsymbol{Y}^T\hat{\boldsymbol{P}}^T)+\alpha_1\beta\mathrm{Tr}(\tilde{\boldsymbol{P}}\boldsymbol{Y}\boldsymbol{L}_p^p\boldsymbol{Y}^T\tilde{\boldsymbol{P}}^T)$ with

$$\alpha_1\beta\cdot\big(\mathrm{Tr}(\hat{\boldsymbol{Q}}\hat{\boldsymbol{P}}\boldsymbol{Y}\boldsymbol{L}_p\boldsymbol{Y}^T\hat{\boldsymbol{P}}^T\hat{\boldsymbol{Q}}^T)+\mathrm{Tr}(\tilde{\boldsymbol{Q}}\tilde{\boldsymbol{P}}\boldsymbol{Y}\boldsymbol{L}_p^p\boldsymbol{Y}^T\tilde{\boldsymbol{P}}^T\tilde{\boldsymbol{Q}}^T)\big) \tag{11}$$

where $\hat{\boldsymbol{Q}}$ equals $\mathrm{diag}\{\|\boldsymbol{m}_1\|,\cdots,\|\boldsymbol{m}_q\|\}$ and $\tilde{\boldsymbol{Q}}$ equals $\mathrm{diag}\{\|\boldsymbol{m}_{q+1}\|,\cdots,\|\boldsymbol{m}_s\|\}$.

On optimizing $\boldsymbol{M}$ with $\boldsymbol{P}, \boldsymbol{D}, \boldsymbol{X}$ fixed. We can further rewrite Eq. (30) as $\mathrm{Tr}(\boldsymbol{M}\boldsymbol{G}_m\boldsymbol{M}^T)$ where

$$\begin{aligned}\boldsymbol{G}_m&=\boldsymbol{G}_{m+}-\boldsymbol{G}_{m-}\\ &=\begin{bmatrix}\hat{\boldsymbol{P}}\boldsymbol{Y}(\alpha_1\beta\boldsymbol{B}_p)\boldsymbol{Y}^T\hat{\boldsymbol{P}}^T & 0\\ 0 & \tilde{\boldsymbol{P}}\boldsymbol{Y}(\alpha_1\beta\boldsymbol{B}_p^p)\boldsymbol{Y}^T\tilde{\boldsymbol{L}}^T\end{bmatrix}\odot\boldsymbol{I}\\ &-\begin{bmatrix}\hat{\boldsymbol{P}}\boldsymbol{Y}(\alpha_1\beta\boldsymbol{W}_p)\boldsymbol{Y}^T\hat{\boldsymbol{P}}^T & 0\\ 0 & \tilde{\boldsymbol{P}}\boldsymbol{Y}(\alpha_1\beta\boldsymbol{W}_p^p)\boldsymbol{Y}^T\tilde{\boldsymbol{L}}^T\end{bmatrix}\odot\boldsymbol{I}\end{aligned} \tag{12}$$

where $\odot$ denotes the element-wise matrix multiplication, and $\boldsymbol{I}$ is an identity matrix. Then we put the non-negative constraints into the objective function with respect to $\boldsymbol{M}$, and define $\boldsymbol{\psi}_{ij}$ as the Lagrange multiplier for $\boldsymbol{M}\geq\boldsymbol{0}$. With $\boldsymbol{\Psi}=[\psi_{ij}]$, the Lagrange $\mathcal{L}(\boldsymbol{W})$ is defined as

$$\begin{aligned}\mathcal{L}(\boldsymbol{M})=&\|\boldsymbol{PY}-\boldsymbol{DX}\|_F^2+\sum\|\boldsymbol{PY}_i-\boldsymbol{D}_i\boldsymbol{X}_i^i\|_F^2+\\ &\alpha_1\|\boldsymbol{Y}-\boldsymbol{MPY}\|_F^2+\mathrm{Tr}(\boldsymbol{M}\boldsymbol{G}_m\boldsymbol{M}^T)\\ &+\alpha_1\|\boldsymbol{M}-\boldsymbol{P}^T\|_F^2+\mathrm{Tr}(\boldsymbol{\Psi}\boldsymbol{M}^T)\end{aligned}. \tag{13}$$

Thus the partial derivative of $\mathcal{L}$ with respect to $\boldsymbol{M}$ is

$$\begin{aligned}\frac{\partial\mathcal{L}}{\partial\boldsymbol{M}}=&-2\alpha_1\boldsymbol{Y}\boldsymbol{Y}^T\boldsymbol{P}^T+2\alpha_1\boldsymbol{M}\boldsymbol{P}\boldsymbol{Y}\boldsymbol{Y}^T\boldsymbol{P}^T\\ &+2\boldsymbol{M}\boldsymbol{G}_m+2\alpha_1\boldsymbol{M}-2\alpha_1\boldsymbol{P}^T+\boldsymbol{\Psi}\end{aligned}. \tag{14}$$

According to the Karush-Kuhn-Tucker (KKT) condition ($\psi_{ij}\boldsymbol{M}_{ij}=0$) and $\frac{\partial\mathcal{L}}{\partial\boldsymbol{M}}=0$, we can obtain the update rule:

$$\boldsymbol{M}_{ij}^{(t+1)}=\boldsymbol{M}_{ij}^{(t)}\frac{(\alpha_1\boldsymbol{Y}\boldsymbol{Y}^T\boldsymbol{P}^T+\boldsymbol{M}\boldsymbol{G}_{m-}+\alpha_1\boldsymbol{P}^T)_{ij}}{(\alpha_1\boldsymbol{M}\boldsymbol{P}\boldsymbol{Y}\boldsymbol{Y}^T\boldsymbol{P}^T+\boldsymbol{M}\boldsymbol{G}_{m+}+\alpha_1\boldsymbol{M})_{ij}}. \tag{15}$$

On optimizing $\boldsymbol{P}$ with $\boldsymbol{M}, \boldsymbol{D}, \boldsymbol{X}$ fixed. After updating $\boldsymbol{M}$, we normalize the column vectors of $\boldsymbol{M}$ and multiply the norm to the projective matrix $\boldsymbol{P}$, namely $\boldsymbol{P}_i\leftarrow\boldsymbol{P}_i\times\|\boldsymbol{m}_i\|_2$, $\boldsymbol{m}_i\leftarrow\boldsymbol{m}_i\|\boldsymbol{m}_i\|_2,\forall i$. By using the normalized $\boldsymbol{M}$, we simplify Eq. (30) as $\alpha_1\beta\mathrm{Tr}(\hat{\boldsymbol{P}}\boldsymbol{Y}\boldsymbol{L}_p\boldsymbol{Y}^T\hat{\boldsymbol{P}}^T)+\alpha_1\beta\mathrm{Tr}(\tilde{\boldsymbol{P}}\boldsymbol{Y}\boldsymbol{L}_p^p\boldsymbol{Y}^T\tilde{\boldsymbol{P}}^T)$. Thus the Lagrange $\mathcal{L}(\boldsymbol{P})$ is

$$\begin{aligned}\mathcal{L}(\boldsymbol{P})=&\|\boldsymbol{PY}-\boldsymbol{DX}\|_F^2+\sum\|\boldsymbol{PY}_i-\boldsymbol{D}_i\boldsymbol{X}_i^i\|_F^2+\\ &\alpha_1\beta\mathrm{Tr}(\hat{\boldsymbol{P}}\boldsymbol{Y}\boldsymbol{L}_p\boldsymbol{Y}^T\hat{\boldsymbol{P}}^T)+\alpha_1\beta\mathrm{Tr}(\tilde{\boldsymbol{P}}\boldsymbol{Y}\boldsymbol{L}_p^p\boldsymbol{Y}^T\tilde{\boldsymbol{P}}^T)+\\ &\alpha_1\|\boldsymbol{Y}-\boldsymbol{MPY}\|_F^2+\alpha_1\|\boldsymbol{M}-\boldsymbol{P}^T\|_F^2+\mathrm{Tr}(\boldsymbol{\Phi}\boldsymbol{P}^T)\end{aligned} \tag{16}$$

where ϕ is the Lagrange multiplier for constraint $\boldsymbol{P}_{ij}\geq 0$ and $\boldsymbol{\Phi}=[\phi_{ij}]$. After setting $\frac{\partial\mathcal{L}}{\partial\boldsymbol{P}}=\boldsymbol{0}$ and applying KKT condition ($\phi_{ij}\boldsymbol{P}_{ij}=0$), we obtain the update rule for $\boldsymbol{P}$:

$$\boldsymbol{P}_{ij}^{(t+1)}=\boldsymbol{P}_{ij}^{(t)}\frac{\begin{pmatrix}\boldsymbol{D}\boldsymbol{X}\boldsymbol{Y}^T+\sum\boldsymbol{D}_i\boldsymbol{X}_i^i\boldsymbol{Y}_i^T+\alpha_1\boldsymbol{M}^T\boldsymbol{Y}\boldsymbol{Y}^T\\ +\alpha_1\boldsymbol{M}^T+\alpha_1\beta\begin{bmatrix}\hat{\boldsymbol{P}}^t\boldsymbol{Y}\boldsymbol{W}_p\boldsymbol{Y}^T\\ \tilde{\boldsymbol{P}}^t\boldsymbol{Y}\boldsymbol{W}_p^p\boldsymbol{Y}^T\end{bmatrix}\end{pmatrix}_{ij}}{\begin{pmatrix}\boldsymbol{P}^t\boldsymbol{Y}\boldsymbol{Y}^T+\sum\boldsymbol{P}^t\boldsymbol{Y}_i\boldsymbol{Y}_i^T+\alpha_1\boldsymbol{P}^t+\\ \alpha_1\boldsymbol{M}^T\boldsymbol{M}\boldsymbol{P}^t\boldsymbol{Y}\boldsymbol{Y}^T+\alpha_1\beta\begin{bmatrix}\hat{\boldsymbol{P}}^t\boldsymbol{Y}\boldsymbol{B}_p\boldsymbol{Y}^T\\ \tilde{\boldsymbol{P}}^t\boldsymbol{Y}\boldsymbol{B}_p^p\boldsymbol{Y}^T\end{bmatrix}\end{pmatrix}_{ij}}. \tag{17}$$

Now both Eq. (15) and Eq. (53) are non-negative update. We prove that the convergence of the updating rule for $\boldsymbol{P}$ and $\boldsymbol{M}$ can be guaranteed. Detailed proof can refer to the supplementary material (Appendix A).

B. Discriminative dictionary learning

On optimizing $\boldsymbol{X}$ with $\boldsymbol{D}, \boldsymbol{P}, \boldsymbol{M}$ fixed. With $\boldsymbol{D}, \boldsymbol{P}$ fixed, the optimization of Eq. (1) becomes

$$\begin{aligned}\min_{\boldsymbol{X}}\big\{&\|\boldsymbol{PY}-\boldsymbol{DX}\|_F^2+\sum_{i=1}^{K}\|\boldsymbol{PY}_i-\boldsymbol{D}_i\boldsymbol{X}_i^i\|_F^2+\alpha_3\|\boldsymbol{X}\|_1\\ &+\sum_{i=1}^{K}\sum_{j=1,j\neq i}^{K}\|\boldsymbol{D}_j\boldsymbol{X}_i^j\|_F^2+\alpha_2\mathrm{Tr}(\boldsymbol{X}^T(\boldsymbol{L}')\boldsymbol{X})\big\}\end{aligned} \tag{18}$$

where $\boldsymbol{L}'=\boldsymbol{L}_c-\boldsymbol{L}_c^p$. Eq. (18) can be solved using feature sign search algorithm [18] after certain formulation based on [51], [22]. We optimize $\boldsymbol{X}$ class by class. Following [18], we update $\boldsymbol{X}_i$ one by one in the ith class. We define $\boldsymbol{x}_{i,j}$ as the coding coefficients of the jth sample in the ith class and reformulated the problem as

$$\begin{aligned}\min_{\boldsymbol{X}}\big\{&\|\boldsymbol{PY}_i-\boldsymbol{D}\boldsymbol{x}_{i,j}\|_F^2+\|\boldsymbol{PY}_i-\boldsymbol{D}_i\boldsymbol{x}_{i,j}^i\|_F^2\\ &+\sum_{n=1,n\neq i}^{K}\|\boldsymbol{D}_n\boldsymbol{x}_{i,j}^n\|_F^2+\alpha_2Q(\boldsymbol{x}_{i,j})+\alpha_3\|\boldsymbol{x}_{i,j}\|_1\end{aligned} \tag{19}$$

Algorithm 1 Training Procedure of JNPDL

Input: Training samples $\boldsymbol{Y} = \|\boldsymbol{Y}_1, \cdots, \boldsymbol{Y}_N\|$, intrinsic graph $\boldsymbol{W}_c, \boldsymbol{W}_p$, penalty graph $\boldsymbol{W}_c^p, \boldsymbol{W}_p^p$, parameters $\alpha_1, \alpha_2, \alpha_3, \beta$, iteration number T.

Output: Non-negative projection matrix $\boldsymbol{P}$, dictionary $\boldsymbol{D}$, coding coefficient matrix $\boldsymbol{X}$.

Step1: Initialization

$t = 1$.

Randomly initialize columns in $\boldsymbol{D}^0, \boldsymbol{M}^0$ with unit l_2 norm.

Initialize $\boldsymbol{x}_{i,1\leq i\leq N}$ with $((\boldsymbol{D}^0)^T(\boldsymbol{D}^0)+\lambda_2\boldsymbol{I})^{-1}(\boldsymbol{D}^0)^T\boldsymbol{y}_i$ where $\boldsymbol{y}_i$ is the ith training sample (regardless of label).

Step2: Search local optima

while not convergence or $t < T$ **do**

  Solve $\boldsymbol{P}^t, \boldsymbol{M}^t$ iteratively with fixed $\boldsymbol{D}^{t-1}$ and $\boldsymbol{X}^{t-1}$ via Eq. (10).

  Solve $\boldsymbol{X}^t$ with fixed $\boldsymbol{M}^t$, $\boldsymbol{D}^{t-1}$ and $\boldsymbol{P}^t$ via Eq. (19).

  Solve $\boldsymbol{D}^t$ with fixed $\boldsymbol{M}^t$, $\boldsymbol{P}^t$ and $\boldsymbol{X}^t$ via Eq. (21).

  $t \leftarrow t + 1$.

end while

Step3: Output

Output $\boldsymbol{P} = \boldsymbol{P}^t$, $\boldsymbol{D} = \boldsymbol{D}^t$ and $\boldsymbol{X} = \boldsymbol{X}^t$.

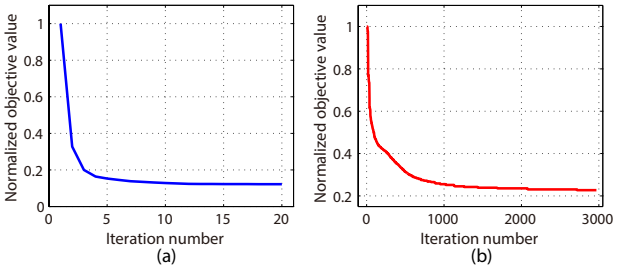

Fig. 2. An example of convergence in extended Yale B dataset. (a) iteratively solve Eq. (1). (b) iteratively solve Eq. (10)

where $Q(\boldsymbol{x}_{i,j}) = \alpha_2(\boldsymbol{x}_{i,j}^T\boldsymbol{X}_i\boldsymbol{L}'_j + (\boldsymbol{X}_i\boldsymbol{L}'_j)^T\boldsymbol{x}_{i,j} - \boldsymbol{x}_{i,j}^T\boldsymbol{L}'_{jj})$ in which $\boldsymbol{L}'_i$ is the ith column of $\boldsymbol{L}$, and $\boldsymbol{L}'_{ii}$ is the entry in the ith row and ith column of $\boldsymbol{L}$. $\boldsymbol{x}_{i,j}$ can be solved via feature sign search algorithm as in [51], [22].

On optimizing $\boldsymbol{D}$ with $\boldsymbol{P}, \boldsymbol{X}, \boldsymbol{M}$ fixed. By fixing $\boldsymbol{P}$ and $\boldsymbol{X}$, Eq. (1) is rewritten as

$$\min_{\boldsymbol{D}}\{\|\boldsymbol{PY} - \boldsymbol{DX}\|_F^2 + \sum_{i=1}^{K}\|\boldsymbol{PY}_i - \boldsymbol{D}_i\boldsymbol{X}_i^i\|_F^2 + \sum_{i=1}^{K}\sum_{j=1,j\neq i}^{K}\|\boldsymbol{D}_j\boldsymbol{X}_i^j\|_F^2\} \tag{20}$$

for which we update $\boldsymbol{D}$ class by class sequentially. When we update $\boldsymbol{D}_v$, the sub-dictionaries $\boldsymbol{D}_i, i \neq v$ associated to the other classes will be fixed. Thus Eq. (20) can be further rewritten to

$$\min_{\boldsymbol{D}_i, i\in\{1,2,\cdots,K\}}\{\|\boldsymbol{PY}_i - \boldsymbol{D}_i\boldsymbol{X}_i\|_F^2 + \|\boldsymbol{PY}_i - \boldsymbol{D}_i\boldsymbol{X}_i^i\|_F^2 + \sum_{j=1,j\neq i}^{K}\|\boldsymbol{D}_j\boldsymbol{X}_i^j\|_F^2\} \tag{21}$$

which is essentially a quadratic programming problem and can be directly solved by the algorithm presented in [46] (update $\boldsymbol{D}_i$ atom by atom). Note that each atom in the dictionary should have unit l_2 norm.

V. Classification Strategy

When the projection and the dictionary have been learned, we need to project the testing image via learned projection, code the projected sample over the learned dictionary and eventually obtain its coding coefficients which is expected to be discriminative. We first project the testing sample and then code it over the learning dictionary via

$$\hat{\boldsymbol{x}} = \arg\min_{\boldsymbol{x}}\{\|\boldsymbol{Py} - \boldsymbol{Dx}\|_2^2 + \lambda_1\|\boldsymbol{x}\|_1\} \tag{22}$$

where λ is a constant. After obtaining the coding coefficients $\hat{\boldsymbol{x}}$, we classify the testing sample via

$$label(\boldsymbol{y}) = \arg\min_i\left\{\|\boldsymbol{Py} - \boldsymbol{D}_i\delta_i(\boldsymbol{x})\|_2^2 + \sigma\|\boldsymbol{x} - \boldsymbol{m}_i\|_2^2\right\} \tag{23}$$

where σ is a weight to balance these two terms, and $\delta_i(\cdot)$ is the characteristic function that selects the coefficients associated with ith class. $\boldsymbol{m}_i$ is the mean vector of the learned coding coefficient matrix of class i, i.e., $\boldsymbol{X}_i$. Incorporating the term $\|\boldsymbol{x} - \boldsymbol{m}_i\|_2^2$ is to make the best of the discrimination within the dictionary, because the dictionary is learned to make coding coefficients similar from the same class and dissimilar among different classes.

VI. Experiments

A. Implementation Details

JNPDL is evaluated by two typical applications including image classification and image set classification. We construct the intrinsic graph $\boldsymbol{W}_p$ and penalty graph $\boldsymbol{W}_p^p$ with correlation similarity, and set the number of nearest neighbors of each sample as $\min(n_c - 1, 5)$ where n_c is the training sample number in class c. The number of shortest pairs from different classes is 20 for each class. For $\boldsymbol{W}_c$ and $\boldsymbol{W}_c^p$, we first remove the graph-based coding coefficients term in Eq. (1) and solve the optimization. The Euclidean distances among the coding coefficients of training samples are used as initial neighbor metric. We set $k_1 = \min(n_c - 1, 5), k_2 = 30$. For all experiments, we fix $\alpha_1 = 1, \alpha_2 = 1$ and $\alpha_3 = 0.05$. The other parameters for JNPDL are obtained via 5-fold cross validation to avoid over-fitting. Specifically, we use $\beta = 0.7, \lambda_1 = 5 \times 10^{-6}$ and $\sigma = 0.05$ for image classification. For image set based face recognition, we set $\beta = 0.8, \lambda_2 = 0.001 \times N/700$ where N is the number of training samples. For all the baseline approaches, we usually use their original settings, and for those that do not have open source, we carefully implement them following the paper. More detailed experiment setup can be found in the corresponding subsections. All results are the average value of 10 times independent experiments.

B. Application to Image Classification

1) Face recognition: We evaluate JNPDL on the extended Yale B [19] and AR [26] face datasets. The extended Yale B dataset consists of 2,414 frontal face images from 38 individuals captured under various laboratory-controlled lighting conditions. All images in extended Yale B are normalized to 54×48. The AR dataset consists of over 4,000 frontal images for 126 individuals. For each individual, 26 pictures were taken in two separated sessions. All images in AR are normalized

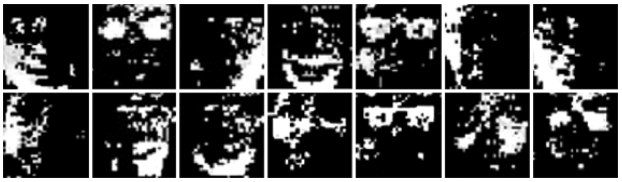

Fig. 3. Visualization of some learned projection bases in JNPDL.

to 60×43. Following the same experimental settings in [44], we randomly select 20 images per subject for training, and the rest for testing in extended Yale B dataset. For AR dataset, a subset consisting of 50 male subjects and 50 female subjects is used. We randomly select 7 images with illumination and expression changes from Session 1 and use them for training. Another 7 images with the same condition from Session 2 are randomly selected for testing. The dictionary size is set as the half of the training samples. SRC [40] uses all the training samples as the dictionary. Fig. 3 shows a visualization of some projection bases in JNPDL, which gives direct intuition of the part-based projection.

Comparison with state-of-the-art approaches. We compare JNPDL with the current state-of-the-art dictionary learning approaches including D-KSVD [50], LC-KSVD [13] and FDDL [43]. Two other approaches (DSRC [49], JDDRDL [5]) that shares similar philosophy with ours are also compared in experiments. JNPDL, DSRC [49] and JDDRDL [5] uses the original images for training and set the feature dimension after projection as 300. All the other methods use the 300-dimensional Eigenface feature. SRC and Linear SVM are used as baselines. Results are shown in Table I. One can see that JNPDL achieves promising recognition accuracy on extended Yale B and AR dataset. It achieves at least 2.2% and 2.6% improvement over the second best approach on extended Yale B and AR.

TABLE I. Recognition accuracy (%) on extended Yale B and AR.

Dataset	Extended Yale B	AR Dataset
SVM	88.8	87.1
SRC [40]	91.0	88.5
D-KSVD [50]	75.3	85.4
LC-KSVD [13]	86.8	90.2
FDDL [45]	91.9	92.1
DSRC [49]	89.6	88.2
JDDRDL [5]	90.1	90.9
JNPDL	**94.1**	**94.7**

Accuracy vs. Feature dimensionality. We vary the feature dimension after projection to evaluate the performance of JNPDL on AR dataset. For SRC and FDDL, the dimensionality reduction is performed by Eigenface. From results in Table II, it can be learned that the jointly learned projection can preserve discriminative information with low feature dimension, leading to competitive accuracy of JNPDL.

TABLE II. Accuracy (%) vs. feature dimension on AR dataset.

Dimension	100	200	300	500
SRC [40]	84.0	87.3	88.5	89.7
FDDL [45]	85.7	88.5	92.0	92.2
DSRC [49]	84.8	86.9	88.2	89.1
JDDRDL [5]	82.5	87.7	90.9	91.6
JNPDL	**88.3**	**92.4**	**94.7**	**95.1**

Joint projection learning vs. Separate projection. The projection and the dictionary can be learned separately. We compare the separate learning and the joint learning and validate the superiority of JNPDL. We also remove the projection learning part of JNPDL and use Eigenface to extract features from faces. The feature dimension of all strategies is set as 300. Results are shown in Table III. Results show that JNPDL that jointly learns non-negative projection and dictionary achieves the best accuracy.

TABLE III. Accuracy (%) of different projection.

Dataset	Extended Yale B	AR dataset
JNPDL (with Eigenface)	91.8	92.2
JNPDL (Separate Learning)	92.1	92.5
JNPDL (Joint Learning)	**94.1**	**94.7**

Face recognition in the wild. In order to further evaluate JNPDL, we apply it in a more challenging face recognition task. LFW [10] is a large-scale dataset containing variations of pose, illumination, expression, misalignment and occlusion. LWFa dataset [39] is an aligned version of LFW. We use 143 subject with no less than 11 samples per subject in LFWa dataset (4174 images in total) to perform the experiment. The first 10 samples are selected as the training samples and the rest is for testing. Following [43], histogram of uniform-LBP is extracted by partitioning a face image into 10×8 patches. PCA is used to reduce the dimension to 1000. Results are shown in Table IV. One can see JNPDL achieves 78.1% on LWFa and has approximately 1% improvement compared to the second best approach.

Fig. 4. Sample from LFW and LFWa Datasets. Left: Original samples from LFW. Right: The corresponding aligned samples from LFWa.

TABLE IV. Recognition Accuracy (%) on LFWa Dataset.

Method	Accuracy	Method	Accuracy
SVM	63.0	COPAR [16]	72.6
SRC [40]	72.7	FDDL [45]	74.8
D-KSVD [50]	65.9	LDL [43]	77.2
LC-KSVD [13]	66.0	JNPDL	**78.1**

Control Experiment. In order to evaluate the performance gain achieved by individual components, we remove some of the proposed constraints in the model and compare the recognition accuracy using the same experimental setup as in the LFWa dataset. GNP, SRE, DRE and GCC denotes the graph-based non-negative projection term, standard reconstruction error term (i.e. $\|\boldsymbol{Y} - \boldsymbol{DX}\|_2^2$), discriminative reconstruction error term and graph-based coding coefficients term, respectively. Results are shown in Table V. Note that, DRE+GNP+GCC is in fact the proposed JNPDL. One can

observe that the gain achieved by each constraint is significant and the proposed JNPDL obtains the best accuracy.

TABLE V. Accuracy (%) of Different Constraint Combinations on LFWa.

Method	Accuracy	Method	Accuracy
SRE	62.3	DRE	70.7
SRE+GNP	68.2	DRE+GNP	73.9
SRE+GCC	72.0	DRE+GCC	75.0
SRE+DRE+GCC	75.8	DRE+GNP+GCC	**78.1**

Running Time Comparison. We compare the training time and testing time of JNPDL in the previous dataset. The experimental settings are the same as the corresponding section. We iterate both the JNPDL model and SFDL model for 5 times. The iteration number of FDDL is 15. The running time results in Table VI. One can observe that the price that JNPDL has to pay for stronger discrimination power is higher training complexity. Fortunately, the testing speed is nearly as fast as conventional DL.

TABLE VI. Running Time (s) Comparison on Different Datasets.

Method	Extended Yale B		AR dataset	
	Training	Testing	Training	Testing
FDDL [45]	639.6	0.705	2657	0.564
SFDL [22]	2688	0.755	7921	0.669
JNPDL	4303	0.714	9594	0.622

2) Digit recognition: We perform the handwritten digit recognition on the USPS dataset [12] that contains 7,291 training images and 2,007 testing images. JNPDL is compared to some advanced dictionary learning methods such as FDDL, D-KSVD, LC-KSVD, SDL [25], bag-of-words method like LLC [36] as well as the projection and dictionary learning methods like JDDRDL and DSRC. Additionally, the standard Euclidean KNN, Gaussian SVM and SRC are used as baselines. SVM, KNN, DLSI, FDDL follows the same parameter settings as in [44]. The 16×16 original images are directly used as the features. The dictionary size is set as 900, each class with 90 atoms. The accuracy for JDDRDL, DSRC and JNPDL use the best result with projected dimension (s_p) equal to 30, 50, 100, 150 and 200. Table VII shows JNPDL outperforms all the other competitive approaches including approaches of the same category and state-of-the-art DL, achieving 97.23% accuracy.

TABLE VII. Recognition Accuracy (%) on USPS Dataset.

Method	Accuracy	Method	Accuracy
SVM	95.80	KNN	94.80
SRC [40]	96.10	DLSI [30]	96.02
LLC (20 bases) [36]	96.38	SDL [25]	96.46
JDDRDL [50]	96.05	FDDL [45]	96.31
DSRC [49]	95.94	JNPDL	**97.23**

3) Object categorization: We perform the object categorization experiment on 17 Oxford Flower dataset [27] which consists of 80 images per class. We use the default experiment setup as in [45]. Note that the features are extracted from the flower regions which are well cropped by segmentation. We compare JNPDL with MTJSRC [48], COPAR, JDDRDL, DSRC, FDDL, SDL, LLC and two baseline: SRC, SVM. For fair comparison, we use the Frequent Local Histogram (FLH) feature [6] to generate a kernel-based feature descriptor that is the same as in [45]. Experimental results in Table VIII show that JNPDL achieve 91.6% accuracy which is slightly worse than FDDL but better than most competitive approaches. We believe that it is due to the kernel features are already extremely discriminative, so performing a linear non-negative projection will not help much. Thus the accuracy gain mainly comes from the discriminative reconstruction error term and the graph-based coefficients term.

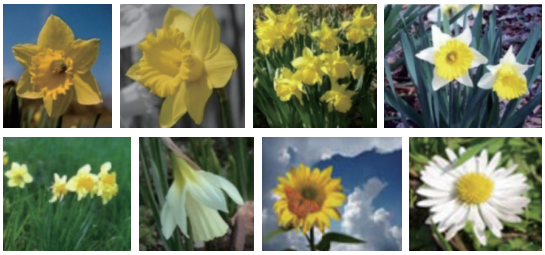

Fig. 5. Samples from 17 Oxford flower dataset.

TABLE VIII. Recognition Accuracy (%) on 17 Oxford Flower Dataset.

Method	Accuracy	Method	Accuracy
SVM	88.6	MTJSRC [48]	88.4
SRC [40]	88.4	COPAR [16]	88.6
LLC (20 bases) [36]	89.7	SDL [25]	91.0
JDDRDL [50]	87.7	FDDL [45]	91.7
DSRC [49]	88.9	JNPDL	**92.1**

4) Action Recognition: The UCF50 dataset [31] has 50 classes and 6680 videos in total. Some action examples are given in Fig. 6. We directly use the action bank feature vector provided in [32] and run the JNPDL through 5-fold group-wise cross validation. We set the projection dimension as 800. The results in Table IX show that JNPDL achieves outperforms some competitive approaches and also validate the superiority of JNPDL in classifying actions.

Fig. 6. Samples from UCF 50 dataset.

TABLE IX. Recognition Accuracy (%) on 17 Oxford Flower Dataset.

Method	Accuracy	Method	Accuracy
SRC [40]	59.6	DLSI [30]	60.0
LC-KSVD [13]	53.6	COPAR [16]	52.5
Action Bank [32]	57.9	FDDL [45]	61.1
Wang [35]	47.9	JNPDL	**62.6**

C. Application to Image Set Classification

1) Classification strategy for image set classification: Applying the classification in Section 5 to each video frame altogether with a voting strategy, JNPDL can be easily extended to image set classification. Given a testing video $Y^{te} = \{\boldsymbol{y}_1^{te}, \boldsymbol{y}_2^{te}, \cdots, \boldsymbol{y}_{K_t}^{te}\}$ in which $\boldsymbol{y}_j^{te}$ is the jth frame and

K_t is the number of image frames in the video, we project each frame to a feature via the learned non-negative projection $\boldsymbol{P}$ and obtain its coding coefficients with Eq. (22). Thus the label of a video frame can be obtain by Eq. (23). After getting all the labels of frames, we perform a majority voting to decide the label of the given image set. For testing efficiency, we replace the l_1 norm $\|\boldsymbol{x}\|_1$ with a l_2 norm $\|\boldsymbol{x}\|_2^2$ and derive the decision:

$$label(\boldsymbol{y}_j^{te}) = \arg\min_i \{\|\boldsymbol{P}\boldsymbol{y}_j^{te} - \boldsymbol{D}_i\delta_i(\boldsymbol{D}^\dagger \boldsymbol{y}_j^{te})\|_2^2\}. \quad (24)$$

where $\boldsymbol{D}^\dagger = (\boldsymbol{D}^T\boldsymbol{D} + \lambda_2\boldsymbol{I})^{-1}\boldsymbol{D}^T$. Eventually we use the majority voting to decide the label of a video (image set).

2) Image set based face recognition: Three video face recognition benchmark dataset, including Honda/UCSD [20], CMU MoBo [7] and YouTube Celebrities (YTC) [14] are used to evaluate the proposed JNPDL. The Honda/UCSD dataset contains 59 face video of 20 individuals with large pose and expression variations. The average length of these videos are approximately 400 frames. The CMU MoBo dataset consists of 96 videos from 24 subjects. Each subject contains 5 videos corresponding to different walking pattern. For each video, there are around 300 frames. The YTC dataset collects 1910 video sequences of 47 celebrities from YouTube. Most videos contains noisy and low-resolution image frames. The number of frames in a video varies from 8 to 400. For fair comparison, we follows the experimental setup in [22]. We use Viola-Jones face detector to capture faces and then resize them to 30×30 intensity image. Each image frame is cropped into 30×30 according to the provided eye coordinates. Thus each video is represented as an image set. Following standard experiment protocol as in [3], [22], the detected face images are histogram equalized but no further preprocessing, and the image features are raw pixel values.

Comparison with state-of-the-art approaches. For both the Honda/UCSD and CMU MoBo datasets, we randomly select one face video per person as the training samples and the rest as testing samples. For YTC dataset, we equally divide the whole dataset into five folds, and each fold contains 9 videos per person. In each fold, we randomly select 3 face videos per person for training and use the rest for testing. We compare JNPDL with DCC [15], MMD [38], MDA [37], CHISD [3], SANP [9], LMKML [23] and SFDL [22]. The settings of these approaches are basically the same as [22]. We select the best accuracy that JNPDL achieves with projected dimension equal to 50, 100, 150, 200 and 300. Results in Table X show the superiority of the proposed method. JNPDL achieves best performance in Honda/UCSD, CMU MoBo datasets and is also comparable to the best performance in YTC dataset.

TABLE X. Recognition accuracy (%) on Honda, MoBo, YTC datasets.

Dataset	Honda/UCSD	CMU MoBo	YTC
DCC [15]	94.9	88.1	64.8
MMD [38]	94.9	91.7	66.7
MDA [37]	97.4	94.4	68.1
CHISD [3]	92.5	95.8	67.4
SANP [9]	93.6	96.1	68.3
LMKML [23]	98.5	96.3	**78.2**
SFDL [22]	100	96.7	76.7
JNPDL	**100**	**97.1**	**77.4**

Accuracy vs. Frame number. We evaluate the performance of JNPDL when videos contain different number of image frames on YTC dataset. We randomly select 50, 100 and 200 image frames from each face image set for training and select another 50, 100, 200 for testing. Note that if there is a image set that do not have enough image frames, we use all of the frames in the image set instead. We select the best accuracy that JNPDL achieves with projected dimension equal to 50, 100, 150, 200 and 300. Results are given in Fig. 7. One can see JNPDL achieves better accuracy than the other approaches. It can be learned that the discrimination power of JNPDL is strong even with small training set.

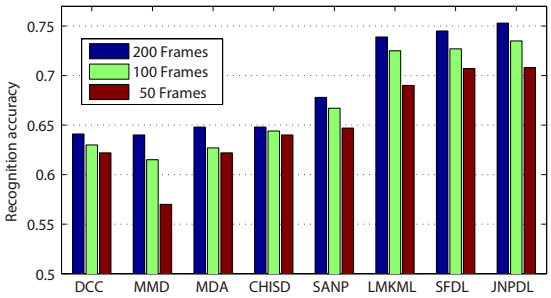

Fig. 7. Recognition accuracy with different number of image frames.

Efficiency. The computational time for JNPDL to recognize a query image set is approximately 5 seconds with a 3.4GHz Dual-core CPU and 16GB RAM, which is comparable to [22], [3] and slightly higher than [15], [38], [37].

VII. Concluding Remarks and Future Work

This paper proposes a novel joint non-negative projection and dictionary learning (JNPDL) where a non-negative projection and a dictionary are simultaneously learned with discriminative graph constraints. Discriminative graph constraints guarantee the discrimination of projected training samples and coding coefficients. The learned non-negative projection focuses on the parts-based representation of objects and can better work with the learned dictionary. JNPDL is optimized under the standard iterative framework, and we propose a multiplicative non-negative updating algorithm for the projection with theoretical convergence guarantee. Excellent experimental results show that JNPDL achieves promising performance on both image classification and image set classification.

JNPDL still has some weaknesses and limitations. First, if we input an extremely discriminative feature into the JNPDL, we may observe little improvement over the conventional DL, sometimes even a performance setback. It is because the projection fails to map the features to a more discriminative space. Second, because of the projections linear nature, it may suffer certain limitations as in linear dimensionality reduction methods. Possible future work includes handling nonlinear cases using methods like kernel trick or other non-linear mapping algorithms, adding more discriminative regularizations to learn the projection matrix and considering to learn a multiple-layered projection.

APPENDIX A
PROOF OF THE CONVERGENCE OF UPDATING RULES FOR $\boldsymbol{P}$ AND $\boldsymbol{M}$

Proof. Before proving the convergence of the updating rules, we first introduce some necessary preliminaries.

Definition 1. *Function $\mathcal{G}(\boldsymbol{A}, \boldsymbol{A}')$ is an auxiliary function for function $\mathcal{F}(\boldsymbol{A})$ if the conditions*

$$\mathcal{G}(\boldsymbol{A}, \boldsymbol{A}') \geq \mathcal{F}(\boldsymbol{A}), \quad \mathcal{G}(\boldsymbol{A}, \boldsymbol{A}) = \mathcal{F}(\boldsymbol{A}) \tag{25}$$

are satisfied.

Lemma 1. *If $\mathcal{G}(\boldsymbol{A}, \boldsymbol{A}')$ is an auxiliary function for $\mathcal{F}(\boldsymbol{A})$, then $\mathcal{F}(\boldsymbol{A})$ is non-increasing to $\boldsymbol{A}$ under the update*

$$\boldsymbol{A}^{t+1} = \arg\min_{\boldsymbol{A}} \mathcal{G}(\boldsymbol{A}, \boldsymbol{A}^t) \tag{26}$$

where t denotes the tth iteration.

Proof. From Eq.(26), we construct the following relation:

$$\mathcal{F}(\boldsymbol{A}^t) = \mathcal{G}(\boldsymbol{A}^t, \boldsymbol{A}^t) \geq \mathcal{G}(\boldsymbol{A}^{t+1}, \boldsymbol{A}^t). \tag{27}$$

Because $\mathcal{G}(\boldsymbol{A}, \boldsymbol{A}')$ is an auxiliary function for $\mathcal{F}(\boldsymbol{A})$, we can obtain the following inequality from Eq. (25):

$$\mathcal{G}(\boldsymbol{A}^{t+1}, \boldsymbol{A}^t) \geq \mathcal{F}(\boldsymbol{A}^{t+1}) \tag{28}$$

which leads to

$$\mathcal{F}(\boldsymbol{A}^{t+1}) \leq \mathcal{F}(\boldsymbol{A}^t). \tag{29}$$

Thus $\mathcal{F}(\boldsymbol{A})$ is non-increasing with respect to $\boldsymbol{A}$ under the updating rule in Eq. (26). The lemma is proved □

We first consider the scenario when $\boldsymbol{P}$ is fixed. With $\boldsymbol{P}$ fixed, we rewrite the optimization objective (Eq. (10) in the paper) as

$$\begin{aligned}\mathcal{F}(\boldsymbol{M}) =& \alpha_1 \|\boldsymbol{Y} - \boldsymbol{M}\boldsymbol{P}\boldsymbol{Y}\|_F^2 + Tr(\boldsymbol{M}\boldsymbol{G}_m\boldsymbol{M}^T) \\ &+ \alpha_1 \|\boldsymbol{M} - \boldsymbol{P}^T\|_F^2\end{aligned}. \tag{30}$$

We denote $\mathcal{F}_{ij}$ as the part of $\mathcal{F}(\boldsymbol{M})$ relevant to $\boldsymbol{M}_{ij}$, and then compute the first-order and the second-order derivative as follows:

$$\begin{aligned}\mathcal{F}'_{ij}(\boldsymbol{M}) =& \alpha_1(-2\boldsymbol{Y}\boldsymbol{Y}^T\boldsymbol{P}^T + 2\boldsymbol{M}\boldsymbol{P}\boldsymbol{Y}\boldsymbol{Y}^T\boldsymbol{P}^T)_{ij} \\ &+ (2\boldsymbol{M}\boldsymbol{G}_m)_{ij} + \alpha_1(2\boldsymbol{M} - 2\boldsymbol{P}^T)_{ij}\end{aligned}, \tag{31}$$

$$\mathcal{F}''_{ij}(\boldsymbol{M}) = 2\alpha_1(\boldsymbol{P}\boldsymbol{Y}\boldsymbol{Y}^T\boldsymbol{P}^T + \boldsymbol{G}_m + \boldsymbol{I})_{jj} \tag{32}$$

where $\boldsymbol{I}$ denotes an identity matrix with matched size. We construct the function $\mathcal{G}(\boldsymbol{M}_{ij}, \boldsymbol{M}_{ij}^t)$ as

$$\begin{aligned}&\mathcal{G}(\boldsymbol{M}_{ij}, \boldsymbol{M}_{ij}^t) = \mathcal{F}_{ij}(\boldsymbol{M}_{ij}^t) + \mathcal{F}'_{ij}(\boldsymbol{M}_{ij}^t)(\boldsymbol{M}_{ij} - \boldsymbol{M}_{ij}^t) \\ &+ \frac{\alpha_1(\boldsymbol{M}^t\boldsymbol{P}\boldsymbol{Y}\boldsymbol{Y}^T\boldsymbol{P}^T + \boldsymbol{M}^t\boldsymbol{G}_{m+} + \boldsymbol{M}^t)_{ij}}{\boldsymbol{M}_{ij}^t}(\boldsymbol{M}_{ij} - \boldsymbol{M}_{ij}^t)^2.\end{aligned} \tag{33}$$

Lemma 2. *$\mathcal{G}(\boldsymbol{M}_{ij}, \boldsymbol{M}_{ij}^t)$ in Eq.* (33) *is an auxiliary function for the function $\mathcal{F}_{ij}(\boldsymbol{M})$.*

Proof. Because it is easily obtained that $\mathcal{G}(\boldsymbol{M}_{ij}, \boldsymbol{M}_{ij}) = \mathcal{F}_{ij}(\boldsymbol{M}_{ij})$, we only need to prove that $\mathcal{G}(\boldsymbol{M}_{ij}, \boldsymbol{M}_{ij}^t) \geq \mathcal{F}_{ij}(\boldsymbol{M}_{ij})$. We first compute the Taylor series expansion of $\mathcal{F}_{ij}(\boldsymbol{M})$ as

$$\begin{aligned}\mathcal{F}_{ij}(\boldsymbol{M}_{ij}) =& \mathcal{F}_{ij}(\boldsymbol{M}_{ij}^t) + \mathcal{F}'_{ij}(\boldsymbol{M}_{ij}^t)(\boldsymbol{M}_{ij} - \boldsymbol{M}_{ij}^t) \\ &+ \frac{1}{2}\mathcal{F}''_{ij}(\boldsymbol{M}_{ij}^t)(\boldsymbol{M}_{ij} - \boldsymbol{M}_{ij}^t)^2\end{aligned}. \tag{34}$$

Because the following inequalities are satisfied:

$$\begin{aligned}(\boldsymbol{M}^t\boldsymbol{P}\boldsymbol{Y}\boldsymbol{Y}^T\boldsymbol{P}^T)_{ij} &= \sum_v \left(\boldsymbol{M}_{iv}^t(\boldsymbol{P}\boldsymbol{Y}\boldsymbol{Y}^T\boldsymbol{P}^T)_{vj}\right) \\ &\geq \boldsymbol{M}_{ij}^t(\boldsymbol{P}\boldsymbol{Y}\boldsymbol{Y}^T\boldsymbol{P}^T)_{jj},\end{aligned} \tag{35}$$

$$\begin{aligned}(\boldsymbol{M}^t\boldsymbol{G}_{m+})_{ij} &= \sum_v \left(\boldsymbol{M}_{iv}^t(\boldsymbol{G}_{m+})_{vj}\right) \\ &\geq \boldsymbol{M}_{ij}^t(\boldsymbol{G}_m)_{jj},\end{aligned} \tag{36}$$

$$\boldsymbol{M}_{ij}^t \geq \boldsymbol{M}_{ij}^t\boldsymbol{I}_{jj}, \tag{37}$$

we can let the following relation hold:

$$\begin{aligned}&\frac{\alpha_1(\boldsymbol{M}^t\boldsymbol{P}\boldsymbol{Y}\boldsymbol{Y}^T\boldsymbol{P}^T + \boldsymbol{M}^t\boldsymbol{G}_{m+} + \boldsymbol{M}^t)_{ij}}{\boldsymbol{M}_{ij}^t} \\ &\geq (\boldsymbol{P}\boldsymbol{Y}\boldsymbol{Y}^T\boldsymbol{P}^T + \boldsymbol{G}_m)_{jj}.\end{aligned} \tag{38}$$

Therefore, we can prove that $\mathcal{G}(\boldsymbol{M}_{ij}, \boldsymbol{M}_{ij}^t) \geq \mathcal{F}_{ij}(\boldsymbol{M}_{ij})$ holds. The lemma is proved. □

Theorem 1. *The updating rule for $\boldsymbol{M}$ can be obtained by minimizing the auxiliary function $\mathcal{G}(\boldsymbol{M}_{ij}, \boldsymbol{M}_{ij}^t)$.*

Proof. We let the derivative of $\mathcal{G}(\boldsymbol{M}_{ij}, \boldsymbol{M}_{ij}^t)$ with respect to $\boldsymbol{M}_{ij}$ equal to zero, namely

$$\begin{aligned}&\frac{\partial \mathcal{G}(\boldsymbol{M}_{ij}, \boldsymbol{M}_{ij}^t)}{\partial \boldsymbol{M}_{ij}} \\ &= \frac{2\alpha_1(\boldsymbol{M}^t\boldsymbol{P}\boldsymbol{Y}\boldsymbol{Y}^T\boldsymbol{P}^T + \boldsymbol{M}^t\boldsymbol{G}_{m+} + \boldsymbol{M}^t)_{ij}}{\boldsymbol{M}_{ij}^t}(\boldsymbol{M}_{ij} - \boldsymbol{M}_{ij}^t) \\ &\quad + \mathcal{F}'_{ij}(\boldsymbol{M}_{ij}^t). \\ &= 0\end{aligned} \tag{39}$$

from which we can derive

$$\boldsymbol{M}_{ij}^{t+1} = \boldsymbol{M}_{ij}^t \frac{(\alpha_1\boldsymbol{Y}\boldsymbol{Y}^T\boldsymbol{P}^T + \boldsymbol{M}^t\boldsymbol{G}_{m-} + \alpha_1\boldsymbol{P}^T)_{ij}}{(\alpha_1\boldsymbol{M}^t\boldsymbol{P}\boldsymbol{Y}\boldsymbol{Y}^T\boldsymbol{P}^T + \boldsymbol{M}^t\boldsymbol{G}_{m+} + \alpha_1\boldsymbol{M}^t)_{ij}}. \tag{40}$$

which is identical to the updating rule that we use in the paper. Thus the lemma is proved □

Then we consider the other scenario when $\boldsymbol{M}$ is fixed. After updating the matrix $\boldsymbol{M}$ via Eq. (40), we normalize the column vectors $\boldsymbol{m}_i$ of $\boldsymbol{M}$ and consequently convey the norm to the projective matrix $\boldsymbol{P}$, namely

$$\begin{aligned}\boldsymbol{P}_i &\leftarrow \boldsymbol{P}_i \times \|\boldsymbol{m}_i\| \\ \boldsymbol{m}_i &\leftarrow \boldsymbol{m}_i / \|\boldsymbol{m}_i\|\end{aligned} \tag{41}$$

where $\boldsymbol{P}_i$ is the ith column vector of the projection matrix $\boldsymbol{P}$. Considering Eq. (41) and the fixed $\boldsymbol{M}$, we can rewrite the optimization objective (Eq. (10) in the paper) as

$$\begin{aligned}\mathcal{F}(\boldsymbol{P}) = &\|\boldsymbol{P}\boldsymbol{Y} - \boldsymbol{D}\boldsymbol{X}\|_F^2 + \sum \|\boldsymbol{P}\boldsymbol{Y}_i - \boldsymbol{D}_i\boldsymbol{X}_i^i\|_F^2 + \\ &\alpha_1\beta\mathrm{Tr}(\hat{\boldsymbol{P}}\boldsymbol{Y}\boldsymbol{L}_p\boldsymbol{Y}^T\hat{\boldsymbol{P}}^T) + \alpha_1\beta\mathrm{Tr}(\tilde{\boldsymbol{P}}\boldsymbol{Y}\boldsymbol{L}_p^p\boldsymbol{Y}^T\tilde{\boldsymbol{P}}^T). \\ &+ \alpha_1\|\boldsymbol{Y} - \boldsymbol{M}\boldsymbol{P}\boldsymbol{Y}\|_F^2 + \alpha_1\|\boldsymbol{M} - \boldsymbol{P}^T\|_F^2)\end{aligned} \tag{42}$$

By denoting $\mathcal{F}_{ij}$ as the part of $\mathcal{F}(\boldsymbol{P})$ relevant to $\boldsymbol{P}_{ij}$, we have the following derivatives:

$$\begin{aligned}&\mathcal{F}'_{ij}(\boldsymbol{P}) = 2(\boldsymbol{P}\boldsymbol{Y}\boldsymbol{Y}^T)_{ij} - 2(\boldsymbol{D}\boldsymbol{X}\boldsymbol{Y}^T))_{ij} + 2(\sum \boldsymbol{P}\boldsymbol{Y}_i\boldsymbol{Y}_i^T)_{ij} \\ &- 2\sum(\boldsymbol{D}_i\boldsymbol{X}_i^i\boldsymbol{Y}_i^T)_{ij} - 2\alpha_1(\boldsymbol{M}^T\boldsymbol{Y}\boldsymbol{Y}^T)_{ij} + 2\alpha_1(\boldsymbol{M}^T\boldsymbol{M}\boldsymbol{P}\boldsymbol{Y}\boldsymbol{Y}^T)_{ij}, \\ &+ 2\alpha_1\beta\begin{bmatrix}\hat{\boldsymbol{P}}\boldsymbol{Y}\boldsymbol{L}_p\boldsymbol{Y}^T \\ \tilde{\boldsymbol{P}}\boldsymbol{Y}\boldsymbol{L}_p^p\boldsymbol{Y}^T\end{bmatrix}_{ij} + \alpha_1(2\boldsymbol{P} - 2\boldsymbol{M}^T)_{ij}\end{aligned} \tag{43}$$

$$\mathcal{F}''_{ij}(\boldsymbol{P}) = 2(\boldsymbol{Y}\boldsymbol{Y}^T)_{jj} + 2(\sum \boldsymbol{Y}_i\boldsymbol{Y}_i^T)_{jj} + 2\alpha_1(\boldsymbol{M}^T\boldsymbol{M})_{ii}(\boldsymbol{Y}\boldsymbol{Y}^T)_{jj} + 2\alpha_1\beta\begin{bmatrix}\boldsymbol{Y}\boldsymbol{L}_p\boldsymbol{Y}^T\\ \boldsymbol{Y}\boldsymbol{L}_p^p\boldsymbol{Y}^T\end{bmatrix}_{jj} + 2\alpha_1\boldsymbol{I}_{jj} \tag{44}$$

The auxiliary function of $\mathcal{F}_{ij}(\boldsymbol{P})$ is designed as

$$\mathcal{G}(\boldsymbol{P}_{ij}, \boldsymbol{P}_{ij}^t) = \mathcal{F}_{ij}(\boldsymbol{P}_{ij}^t) + \mathcal{F}'_{ij}(\boldsymbol{P}_{ij}^t)(\boldsymbol{P}_{ij} - \boldsymbol{P}_{ij}^t) + \frac{\begin{pmatrix}\boldsymbol{P}^t\boldsymbol{Y}\boldsymbol{Y}^T + \sum \boldsymbol{P}^t\boldsymbol{Y}_i\boldsymbol{Y}_i^T + \alpha_1(\boldsymbol{P}^t) + \\ \alpha_1\boldsymbol{M}^T\boldsymbol{M}\boldsymbol{P}\boldsymbol{Y}\boldsymbol{Y}^T + \alpha_1\beta\begin{bmatrix}\hat{\boldsymbol{P}}^t\boldsymbol{Y}\boldsymbol{B}_p\boldsymbol{Y}^T\\ \tilde{\boldsymbol{P}}^t\boldsymbol{Y}\boldsymbol{B}_p^p\boldsymbol{Y}^T\end{bmatrix}\end{pmatrix}_{ij}}{\boldsymbol{P}_{ij}^t}(\boldsymbol{P}_{ij} - \boldsymbol{P}_{ij}^t)^2 \tag{45}$$

Lemma 3. *$\mathcal{G}(\boldsymbol{P}_{ij}, \boldsymbol{P}_{ij}^t)$ in Eq. (45) is an auxiliary function for the function $\mathcal{F}_{ij}(\boldsymbol{P})$.*

Proof. Because obviously $\mathcal{G}(\boldsymbol{P}_{ij}, \boldsymbol{P}_{ij}) = \mathcal{F}_{ij}(\boldsymbol{P}_{ij})$, we only need to prove that $\mathcal{G}(\boldsymbol{P}_{ij}, \boldsymbol{P}_{ij}^t) = \mathcal{F}_{ij}(\boldsymbol{P}_{ij})$. We first obtain the Taylor series expansion of $\mathcal{F}_{ij}(\boldsymbol{P})$ as

$$\mathcal{F}_{ij}(\boldsymbol{P}_{ij}) = \mathcal{F}_{ij}(\boldsymbol{P}_{ij}^t) + \mathcal{F}'_{ij}(\boldsymbol{P}_{ij}^t)(\boldsymbol{P}_{ij} - \boldsymbol{P}_{ij}^t) + \frac{1}{2}\mathcal{F}''_{ij}(\boldsymbol{P}_{ij}^t)(\boldsymbol{P}_{ij} - \boldsymbol{P}_{ij}^t)^2 \tag{46}$$

Since the following relations hold:

$$(\boldsymbol{P}^t\boldsymbol{Y}\boldsymbol{Y}^T)_{ij} = \sum_v \left(\boldsymbol{P}_{iv}^t(\boldsymbol{Y}\boldsymbol{Y}^T)_{vj}\right) \geq \boldsymbol{P}_{ij}^t(\boldsymbol{Y}\boldsymbol{Y}^T)_{jj} \tag{47}$$

$$(\sum_i \boldsymbol{P}^t\boldsymbol{Y}_i\boldsymbol{Y}_i^T)_{ij} = \sum_v \left(\boldsymbol{P}_{iv}^t(\sum \boldsymbol{Y}_i\boldsymbol{Y}_i^T)_{vj}\right) \geq \boldsymbol{P}_{ij}^t(\sum \boldsymbol{Y}_i\boldsymbol{Y}_i^T)_{jj} \tag{48}$$

$$\begin{aligned}(\boldsymbol{M}^T\boldsymbol{M}\boldsymbol{P}^t\boldsymbol{Y}\boldsymbol{Y}^T)_{ij} &= \sum_v \left((\boldsymbol{M}^T\boldsymbol{M}\boldsymbol{P}^t)_{iv}(\boldsymbol{Y}\boldsymbol{Y}^T)_{vj}\right)\\ &\geq (\boldsymbol{M}^T\boldsymbol{M}\boldsymbol{P}^t)_{ij}(\boldsymbol{Y}\boldsymbol{Y}^T)_{jj}\\ &= \sum_v \left((\boldsymbol{M}^T\boldsymbol{M})_{iv}\boldsymbol{P}_{vj}^t\right)(\boldsymbol{Y}\boldsymbol{Y}^T)_{jj}\\ &\geq \boldsymbol{P}_{ij}^t(\boldsymbol{M}^T\boldsymbol{M})_{ii}(\boldsymbol{Y}\boldsymbol{Y}^T)_{jj}\end{aligned} \tag{49}$$

$$\begin{aligned}\begin{bmatrix}\hat{\boldsymbol{P}}^t\boldsymbol{Y}\boldsymbol{B}_p\boldsymbol{Y}^T\\ \tilde{\boldsymbol{P}}^t\boldsymbol{Y}\boldsymbol{B}_p^p\boldsymbol{Y}^T\end{bmatrix}_{ij} &= \begin{cases}\sum_v \left(\hat{\boldsymbol{P}}_{iv}^t(\boldsymbol{Y}\boldsymbol{B}_p\boldsymbol{Y}^T)_{vj}\right), & \text{if} j \leq q\\ \sum_v \left(\tilde{\boldsymbol{P}}_{iv}^t(\boldsymbol{Y}\boldsymbol{B}_p^p\boldsymbol{Y}^T)_{vj}\right), & \text{otherwise}\end{cases}\\ &\geq \begin{cases}\hat{\boldsymbol{P}}_{ij}^t(\boldsymbol{Y}\boldsymbol{B}_p\boldsymbol{Y}^T)_{jj}, & \text{if} j \leq q\\ \tilde{\boldsymbol{P}}_{ij}^t(\boldsymbol{Y}\boldsymbol{B}_p^p\boldsymbol{Y}^T)_{jj}, & \text{otherwise}\end{cases}\\ &\geq \begin{cases}\hat{\boldsymbol{P}}_{ij}^t(\boldsymbol{Y}\boldsymbol{L}_p\boldsymbol{Y}^T)_{jj}, & \text{if} j \leq q\\ \tilde{\boldsymbol{P}}_{ij}^t(\boldsymbol{Y}\boldsymbol{L}_p^p\boldsymbol{Y}^T)_{jj}, & \text{otherwise}\end{cases}\\ &= \boldsymbol{P}_{ij}^t\begin{bmatrix}\boldsymbol{Y}\boldsymbol{L}_p\boldsymbol{Y}^T\\ \boldsymbol{Y}\boldsymbol{L}_p^p\boldsymbol{Y}^T\end{bmatrix}_{jj}\end{aligned} \tag{50}$$

$$\boldsymbol{P}_{ij}^t \geq \boldsymbol{P}_{ij}^t\boldsymbol{I}_{jj}, \tag{51}$$

we can have $\mathcal{G}(\boldsymbol{P}_{ij}, \boldsymbol{P}_{ij}^t) \geq \mathcal{F}(\boldsymbol{P}_{ij})$. Therefore the lemma is proved. □

Theorem 2. *The updating rule for $\boldsymbol{P}$ can be obtained by minimizing the auxiliary function $\mathcal{G}(\boldsymbol{P}_{ij}, \boldsymbol{P}_{ij}^t)$.*

Proof. Let $(\partial\mathcal{G}(\boldsymbol{P}_{ij}, \boldsymbol{P}_{ij}^t))/(\partial\boldsymbol{P}_{ij}) = 0$, and we have

$$2\frac{\begin{pmatrix}\boldsymbol{P}^t\boldsymbol{Y}\boldsymbol{Y}^T + \sum \boldsymbol{P}^t\boldsymbol{Y}_i\boldsymbol{Y}_i^T + \alpha_1(\boldsymbol{P}^t) + \\ \alpha_1\boldsymbol{M}^T\boldsymbol{M}\boldsymbol{P}\boldsymbol{Y}\boldsymbol{Y}^T + \alpha_1\beta\begin{bmatrix}\hat{\boldsymbol{P}}^t\boldsymbol{Y}\boldsymbol{B}_p\boldsymbol{Y}^T\\ \tilde{\boldsymbol{P}}^t\boldsymbol{Y}\boldsymbol{B}_p^p\boldsymbol{Y}^T\end{bmatrix}\end{pmatrix}_{ij}}{\boldsymbol{P}_{ij}^t}(\boldsymbol{P}_{ij} - \boldsymbol{P}_{ij}^t) + \mathcal{F}'_{ij}(\boldsymbol{P}_{ij}^t) = 0 \tag{52}$$

from which we can derive the updating rule for $\boldsymbol{P}$

$$\boldsymbol{P}_{ij}^{(t+1)} = \boldsymbol{P}_{ij}^{(t)}\frac{\begin{pmatrix}\boldsymbol{D}\boldsymbol{X}\boldsymbol{Y}^T + \sum \boldsymbol{D}_i\boldsymbol{X}_i^i\boldsymbol{Y}_i^T + \alpha_1\boldsymbol{M}^T\boldsymbol{Y}\boldsymbol{Y}^T \\ + \alpha_1\boldsymbol{M}^T + \alpha_1\beta\begin{bmatrix}\hat{\boldsymbol{P}}^t\boldsymbol{Y}\boldsymbol{W}_p\boldsymbol{Y}^T\\ \tilde{\boldsymbol{P}}^t\boldsymbol{Y}\boldsymbol{W}_p^p\boldsymbol{Y}^T\end{bmatrix}\end{pmatrix}_{ij}}{\begin{pmatrix}\boldsymbol{P}^t\boldsymbol{Y}\boldsymbol{Y}^T + \sum \boldsymbol{P}^t\boldsymbol{Y}_i\boldsymbol{Y}_i^T + \alpha_1\boldsymbol{P}^t + \\ \alpha_1\boldsymbol{M}^T\boldsymbol{M}\boldsymbol{P}^t\boldsymbol{Y}\boldsymbol{Y}^T + \alpha_1\beta\begin{bmatrix}\hat{\boldsymbol{P}}^t\boldsymbol{Y}\boldsymbol{B}_p\boldsymbol{Y}^T\\ \tilde{\boldsymbol{P}}^t\boldsymbol{Y}\boldsymbol{B}_p^p\boldsymbol{Y}^T\end{bmatrix}\end{pmatrix}_{ij}}. \tag{53}$$

Thus the theorem is proved. □

According to **Lemma 1**, **Theorem 1** and **Theorem 2**, we have proved that the convergence of the updating rules for $\boldsymbol{P}$ and $\boldsymbol{M}$ can be theoretically guaranteed. □

REFERENCES

[1] M. Aharon, M. Elad, and A. Bruckstein. K-svd: An algorithm for designing overcomplete dictionaries for sparse representation. *IEEE T. SP*, 54(11):4311–4322, 2006.

[2] A. Castrodad and G. Sapiro. Sparse modeling of human actions from motion imagery. *IJCV*, 100(1):1–15, 2012.

[3] H. Cevikalp and B. Triggs. Face recognition based on image sets. In *CVPR*, 2010.

[4] M. Elad and M. Aharon. Image denoising via learned dictionaries and sparse representation. In *CVPR*, 2006.

[5] Z. Feng, M. Yang, L. Zhang, Y. Liu, and D. Zhang. Joint discriminative dimensionality reduction and dictionary learning for face recognition. *Pattern Recognition*, 46(8):2134–2143, 2013.

[6] B. Fernando, E. Fromont, and T. Tuytelaars. Effective use of frequent itemset mining for image classification. In *ECCV*. 2012.

[7] R. Gross and J. Shi. The cmu motion of body (mobo) database. 2001.

[8] T. Guha and R. K. Ward. Learning sparse representations for human action recognition. *IEEE T. PAMI*, 34(8):1576–1588, 2012.

[9] Y. Hu, A. S. Mian, and R. Owens. Sparse approximated nearest points for image set classification. In *CVPR*, 2011.

[10] G. B. Huang, M. Ramesh, T. Berg, and E. Learned-Miller. Labeled faces in the wild: A database for studying face recognition in unconstrained environments. Technical report, 2007.

[11] K. Huang and S. Aviyente. Sparse representation for signal classification. In *NIPS*, 2006.

[12] J. J. Hull. A database for handwritten text recognition research. *IEEE T. PAMI*, 16(5):550–554, 1994.

[13] Z. Jiang, Z. Lin, and L. S. Davis. Label consistent k-svd: learning a discriminative dictionary for recognition. *IEEE T. PAMI*, 35(11):2651–2664, 2013.

[14] M. Kim, S. Kumar, V. Pavlovic, and H. Rowley. Face tracking and recognition with visual constraints in real-world videos. In *CVPR*, 2008.

[15] T.-K. Kim, J. Kittler, and R. Cipolla. Discriminative learning and recognition of image set classes using canonical correlations. *IEEE T. PAMI*, 29(6):1005–1018, 2007.

[16] S. Kong and D. Wang. A dictionary learning approach for classification: Separating the particularity and the commonality. In *ECCV*. 2012.

[17] D. D. Lee and H. S. Seung. Learning the parts of objects by non-negative matrix factorization. *Nature*, 401(6755):788–791, 1999.

[18] H. Lee, A. Battle, R. Raina, and A. Y. Ng. Efficient sparse coding algorithms. In *NIPS*, 2006.

[19] K.-C. Lee, J. Ho, and D. Kriegman. Acquiring linear subspaces for face recognition under variable lighting. *IEEE T. PAMI*, 27(5):684–698, 2005.

[20] K.-C. Lee, J. Ho, M.-H. Yang, and D. Kriegman. Video-based face recognition using probabilistic appearance manifolds. In *CVPR*, 2003.

[21] X. Liu, S. Yan, and H. Jin. Projective nonnegative graph embedding. *IEEE T. IP*, 19(5):1126–1137, 2010.

[22] J. Lu, G. Wang, W. Deng, and P. Moulin. Simultaneous feature and dictionary learning for image set based face recognition. In *ECCV*. 2014.

[23] J. Lu, G. Wang, and P. Moulin. Image set classification using holistic multiple order statistics features and localized multi-kernel metric learning. In *ICCV*, 2013.

[24] J. Mairal, F. Bach, and J. Ponce. Task-driven dictionary learning. *IEEE T. PAMI*, 34(4):791–804, 2012.

[25] J. Mairal, J. Ponce, G. Sapiro, A. Zisserman, and F. R. Bach. Supervised dictionary learning. In *NIPS*, 2009.

[26] A. M. Martinez. The ar face database. *Technical Report*, 1998.

[27] M.-E. Nilsback and A. Zisserman. A visual vocabulary for flower classification. In *CVPR*, 2006.

[28] Q. Qiu, Z. Jiang, and R. Chellappa. Sparse dictionary-based representation and recognition of action attributes. In *ICCV*, 2011.

[29] Q. Qiu, V. Patel, and R. Chellappa. Information-theoretic dictionary learning for image classification. *IEEE T. PAMI*, 36(11):2173–2184, 2014.

[30] I. Ramirez, P. Sprechmann, and G. Sapiro. Classification and clustering via dictionary learning with structured incoherence and shared features. In *CVPR*, 2010.

[31] K. K. Reddy and M. Shah. Recognizing 50 human action categories of web videos. *Machine Vision and Applications*, 24(5):971–981, 2013.

[32] S. Sadanand and J. J. Corso. Action bank: A high-level representation of activity in video. In *Computer Vision and Pattern Recognition (CVPR), 2012 IEEE Conference on*, pages 1234–1241. IEEE, 2012.

[33] A. Wang, J. Lu, G. Wang, J. Cai, and T.-J. Cham. Multi-modal unsupervised feature learning for rgb-d scene labeling. In *ECCV*. 2014.

[34] C. Wang, Z. Song, S. Yan, L. Zhang, and H.-J. Zhang. Multiplicative nonnegative greph embedding. In *CVPR*, 2009.

[35] H. Wang, C. Yuan, W. Hu, and C. Sun. Supervised class-specific dictionary learning for sparse modeling in action recognition. *Pattern Recognition*, 45(11):3902–3911, 2012.

[36] J. Wang, J. Yang, K. Yu, F. Lv, T. Huang, and Y. Gong. Locality-constrained linear coding for image classification. In *CVPR*, 2010.

[37] R. Wang and X. Chen. Manifold discriminant analysis. In *CVPR*, 2009.

[38] R. Wang, S. Shan, X. Chen, and W. Gao. Manifold-manifold distance with application to face recognition based on image set. In *CVPR*, 2008.

[39] L. Wolf, T. Hassner, and Y. Taigman. Effective unconstrained face recognition by combining multiple descriptors and learned background statistics. *IEEE T. PAMI*, 33(10):1978–1990, 2011.

[40] J. Wright, A. Y. Yang, A. Ganesh, S. S. Sastry, and Y. Ma. Robust face recognition via sparse representation. *IEEE T. PAMI*, 31(2):210–227, 2009.

[41] S. Yan, D. Xu, B. Zhang, H.-J. Zhang, Q. Yang, and S. Lin. Graph embedding and extensions: a general framework for dimensionality reduction. *IEEE T. PAMI*, 29(1):40–51, 2007.

[42] J. Yang, K. Yu, Y. Gong, and T. Huang. Linear spatial pyramid matching using sparse coding for image classification. In *CVPR*, 2009.

[43] M. Yang, D. Dai, L. Shen, and L. V. Gool. Latent dictionary learning for sparse representation based classification. In *CVPR*, 2014.

[44] M. Yang, D. Zhang, and X. Feng. Fisher discrimination dictionary learning for sparse representation. In *ICCV*, 2011.

[45] M. Yang, L. Zhang, X. Feng, and D. Zhang. Sparse representation based fisher discrimination dictionary learning for image classification. *IJCV*, 109(3):209–232, 2014.

[46] M. Yang, L. Zhang, J. Yang, and D. Zhang. Metaface learning for sparse representation based face recognition. In *ICIP*, 2010.

[47] Z. Yang and E. Oja. Linear and nonlinear projective nonnegative matrix factorization. *IEEE T. NN*, 21(5):734–749, 2010.

[48] X.-T. Yuan, X. Liu, and S. Yan. Visual classification with multitask joint sparse representation. *IEEE T. IP*, 21(10):4349–4360, 2012.

[49] H. Zhang, Y. Zhang, and T. S. Huang. Simultaneous discriminative projection and dictionary learning for sparse representation based classification. *Pattern Recognition*, 46(1):346–354, 2013.

[50] Q. Zhang and B. Li. Discriminative k-svd for dictionary learning in face recognition. In *CVPR*, 2010.

[51] M. Zheng, J. Bu, C. Chen, C. Wang, L. Zhang, G. Qiu, and D. Cai. Graph regularized sparse coding for image representation. *IEEE T. IP*, 20(5):1327–1336, 2011.

[52] N. Zhou, Y. Shen, J. Peng, and J. Fan. Learning inter-related visual dictionary for object recognition. In *CVPR*, 2012.

[53] W. Zou, S. Zhu, K. Yu, and A. Y. Ng. Deep learning of invariant features via simulated fixations in video. In *NIPS*, 2012.

[54] Z. Zuo, G. Wang, B. Shuai, L. Zhao, Q. Yang, and X. Jiang. Learning discriminative and shareable features for scene classification. In *ECCV*. 2014.