Semisupervised Discriminant Multimanifold Analysis for Action Recognition

Zengmin Xu®, Ruimin Hu®, Senior Member, IEEE, Jun Chen, Chen Chen®, Junjun Jiang®, Jiaofen Li, and Hongyang Li

Abstract—Although recent semisupervised approaches have proven their effectiveness when there are limited training data, they assume that the samples from different actions lie on a single data manifold in the feature space and try to uncover a common subspace for all samples. However, this assumption ignores the intraclass compactness and the interclass separability simultaneously. We believe that human actions should occupy multimanifold subspace and, therefore, model the samples of the same action as the same manifold and those of different actions as different manifolds. In order to obtain the optimum subspace projection matrix, the current approaches may be mathematically imprecise owing to the badly scaled matrix and improper convergence. To address these issues in unconstrained convex optimization, we introduce a nontrivial spectral projected gradient method and Karush–Kuhn–Tucker conditions without matrix inversion. Through maximizing the separability between different classes by using labeled data points and estimating the intrinsic geometric structure of the data distributions by exploring unlabeled data points, the proposed algorithm can learn global and local consistency and boost the recognition performance. Extensive experiments conducted on the realistic video data sets, including JHMDB, HMDB51, UCF50, and UCF101, have demonstrated that our algorithm outperforms the compared algorithms, including deep learning approach when there are only a few labeled samples.

Index Terms—Discriminant analysis, Karush–Kuhn–Tucker (KKT) conditions, manifold learning, semisupervised learning, spectral projected gradient (SPG).

I. INTRODUCTION

VISUAL recognition draws strong research interest in computer vision because of its promising applications for feature selection, image annotation, video concept detection, and so on [1]–[29]. With the developments in cloud storage technologies, the number of personal images/videos increases rapidly, and it becomes an important challenge to organize these resources effectively. Common approaches of visual recognition are to train supervised classifiers from large-scale labeled data. However, the amount of labeled data is extremely scarce compared with the unlabeled data in the real world. When confronted with huge amounts of unlabeled samples, manual annotation or labeling should be prohibitive. Consequently, semisupervised learning, which can make good use of both labeled and unlabeled data, is applied to explore feature correlation from the original feature space.

Motivated by the progress of semisupervised learning, a few research attention has been paid to semisupervised action recognition [30], [31]. A common limitation of the existing supervised and semisupervised action recognition algorithms is that they evaluate the importance of commonly shared structure between different actions, without considering intraclass compactness and interclass separability simultaneously [30], [31]. For example, even though legs motion appears in similar actions such as the SoccerJuggling and the SoccerPenalty, these within-class actions have much similar motion and dissimilar components simultaneously. Although the shared structural uncovering and label correlation mining have proven beneficial to action recognition in [30] and [31], the ways to learn discriminant features in a semisupervised framework for action recognition have not been largely addressed. To solve this problem, some state-of-the-art...
algorithms are proposed to take a discriminant analysis into consideration for visual recognition. For example, the works in [32]–[36] implement their methods in a supervised way. Another limitation of current semisupervised approaches is that they solve their nonconvex optimization by impressive derivation and alternating-least-squares-like iterative algorithm, which fails to discover the most valuable optimum in a mathematical way [3], [4], [30], [31], [37]. This is because the subproblem of objective function optimization is less rigorous, which have not discussed the singularity of the deduced matrix. In addition, the optimum is supposed to satisfy the Karush–Kuhn–Tucker (KKT) conditions, but they do not explain the KKT conditions of orthogonal constraint, and the accuracy of convergence optimum also lack of further analysis. Recent research studies have indicated that it is beneficial to obtain an optimal solution by projected gradient methods. Motivated by this fact, the projected gradient method has been introduced to the field of multimedia [38], [39]. Although spectral projected gradient (SPG) method [40] has been studied extensively in both theory and practice [38], [41], [42], so far no study has formally applied its techniques to action recognition in semisupervised way.

As mentioned above, it remains unclear how to manually define feature correlation in action recognition. Thus, we propose to model the intramanifold compactness and the intermanifold separability simultaneously and characterize high-level semantic pattern through the local action features by discriminant multimanifold analysis, as shown in Fig. 1. The proposed algorithm combines the strengths of semisupervised learning, discriminant analysis, multitask learning, and unconstrained optimization. Both labeled and unlabeled data are utilized for action recognition in classifiers’ training phase.

A. Motivation and Contributions

It is true that there is a trend to apply deep learning approaches to achieve good action recognition performance, by relying on large-scale labeled training data. Although there are a few large-scale data sets, e.g., Sports1M [43], YouTube8M [44], and ActivityNet [45], obtaining and annotating such data sets require a significant amount of time, resources, and effort. In contrast, collecting unlabeled videos is much easier. The semisupervised learning can effectively leverage the unlabeled data.

Moreover, videos in those data sets are limited to sports and/or daily activities. For real-world applications such as anomaly detection in surveillance and labeled data (videos contain rare anomalous events, e.g., crime related activities) are notoriously hard to obtain [46].

In addition, videos in surveillance applications are very different from other web-based multimedia videos, e.g., Sports1M, YouTube8M, and ActivityNet, due to content, background, device noise, action complexity, viewpoint, scale, and so on (see Fig. 2). The deep learning model on multimedia data set may not work well on surveillance data set, as the deep learning approach learned to exploit the specifics of a particular action from multimedia videos rather than the learning models of characters that are then used for parsing the action from other types of videos [47].

The deep learning approaches not only rely on large-scale labeled training data but also are restricted by the capacity of GPUs. Moreover, overfitting is still an unsolved problem, especially when there is limited training data. How to leverage the unlabeled data and how to pursue efficient learning methods trained on small data sets are worthwhile scientific questions of broad interest to the community [19], [48], [49].

The goal of this paper is to uncover the discriminative information by exploring action features and achieve the state-of-the-art action recognition performance based on the semisupervised setting, which uses only part of the labeled training data, as compared with other semisupervised approaches under the same setting. The distinction we want to make is that we do not aim to compete with fully supervised approaches,
which reported better performance than ours on the evaluated data sets. The contributions are summarized as follows:

1) Ours is the first work to consider both multimaniifold analysis and semisupervised learning in action recognition, given that samples (action videos) may lie in a multimaniifold subspace. By modeling a multimaniifold subspace, both intraclass compactness and interclass separability are taken into account.

2) To solve the unconstrained convex optimization in our problem, we propose to incorporate SPG and KKT conditions to avoid matrix inversion, as done in [26] and [27], which may suffer from the singularity, thereby leading to better convergence and a more accurate solution. In addition, we provide experimental justification on the convergence.

3) We not only introduce a new idea (i.e., multimaniifold analysis) in the problem formulation of semisupervised action recognition but also develop an effective and efficient algorithm to solve the optimization of the objective function.

4) Extensive experiments have validated that our method achieves the best recognition performances on four benchmarks in the semisupervised setting, while has the fastest training speed as compared with the state of the art [e.g., subfeature uncovering with sparsity (SFUS), semisupervised feature correlation mining (SFCM), and multiple feature correlation uncovering (MFCU)]. We believe our work provides valuable insights into video action analysis in a semisupervised manner.

II. RELATED WORK

In this section, we review the related research on manifold learning, semisupervised learning, and multitask learning.

A. Discriminant Analysis

Previous works have stated that manifold learning is capable of mining geometry structures information by regarding a space of probabilities as a manifold [33], [34], [36], [50]–[55].

Cai and He [33] perform an active learning algorithm which lies on the data manifold adaptive kernel space by using graph Laplacian, which can reflect the underlying geometry of the data. Harandi et al. [51] develop a discriminant analysis approach on Grassmannian manifolds by characterizing intraclass compactness and interclass separability. Li et al. [52] contribute a novel coclustering algorithm based on symmetric nonnegative matrix trifactorization by manifold ensemble learning. Yan et al. [53] propose a novel multitask learning framework for multiview action recognition by multitask linear discriminant analysis. Jiang et al. [34]–[36] try to match a low resolution or poor quality face image to a gallery of high-resolution face images by discriminant analysis on multimaniifold. Yu and Zhao [50], [54] introduce the penalty of a lasso or elastic net into the exponential discriminant analysis so that the key variables responsible for fault diagnosis can be automatically selected. In [56]–[58], they exploit the local manifold structure to capture the discrimination features when reconstructing the face images. Ma et al. [55] exploit the intrinsic geometrical structures among the feature points for shape registration based on manifold regularization. Inspired by these research studies, we try to join the idea of discriminant analysis into a semisupervised framework and use the labeled data points to maximize the separability between different classes.

B. Semisupervised Learning

Semisupervised learning has been widely used for its promising performance in different applications [1], [3], [4], [30], [31], [37], [59]–[62]. Given labeling a large amount of training data is time-consuming and expensive, unlabeled samples can be exploited to learn data correlation by semisupervised learning. Thus, semisupervised learning is beneficial in terms of both the data analysis performance and human laboring cost.

Graph Laplacian-based semisupervised learning has shown its simplicity and efficiency in visual concept recognition [63]. Nie et al. [59] propose a manifold learning framework based on graph Laplacian and compared its performance with other algorithms. Ma et al. [4] develop a novel feature selection method and apply it to automatic image annotation. Yang et al. [1] present a framework for multimedia content analysis and retrieval which consists of two independent algorithms. Chang and Yang [3] build a semisupervised feature selection framework by mining correlations among multiple tasks and apply it to different multimedia applications. Wang et al. [30], [31] point out that action recognition can be improved by a complicated formulation and iterative algorithm. In addition, semisupervised learning has also been applied to solving the problems of face recognition [60], image matching [61], image fusion [62], and so on. Motivated by these papers, we design a semisupervised learning algorithm with graph embedding discriminant analysis, then the intrinsic geometric structure of the data distributions can be estimated by exploring the unlabeled data points.

C. Multitask Learning

Multitask learning has gained increasing interest in many applications for its advantage, which can learn multiple related tasks with a shared representation [2], [64], [65]. Recent research studies have indicated that learning multiple related tasks jointly always outperforms learning them independently. Inspired by the progress of multitask learning, researchers have introduced it to the field of multimedia and demonstrated its promising performance on multimedia analysis. For example, Yang et al. [2] study a novel multitask feature selection algorithm in a batch mode by leveraging shared information among multiple related tasks. Ma et al. [64] design a multitask learning framework to jointly optimize the classifiers for both laboratory and real-world data sets. Yang et al. [65] learn a novel clustering model to capture correlations among the related clustering tasks and/or within an individual task. Despite their good performances, these classical algorithms are all implemented only with labeled training data. Following the related works, the proposed framework can learn the...
global consistency and the local geometric, and hence, the performance can be improved by mining correlations between multiple related tasks.

There are two aforementioned research studies close to our work, which are both proposed by Wang et al. [30], [31]. They assume that the samples from different actions define a single data manifold in the feature space, visual words of different action videos may share a common structure in a low-dimensional space. They introduce a transformation matrix to characterize the shared information and employ a regularization term of the shared information among different features. They solve their constrained nonconvex optimization problem by comprehensive derivation and alternating-least-squares-like iterative algorithm. However, the deduced inverse matrix is close to singular or badly scaled during the optimization process, which may make the results inaccurate.

To solve the above-mentioned issues, we model the samples of the same action as the same manifold and those of different actions as different manifolds. As described before, we claim that multimanifold mapping can maximize the discriminatory power while preserving local geometry, mining shared structure is not our purpose, so we discard the shared structure regularization term, and model the local geometrical structure of manifolds by building a within-class similarity graph $A_w$ and a between-class similarity graph $A_b$. We also remove the selection matrix $U$ in our function. Since the proposed optimization solution in [30] and [31] may be mathematically imprecise, we introduce the SPG method and the KKT conditions to avoid matrix inversion and improper convergence.

III. PROPOSED APPROACH

This section begins with an elaboration of the formulation of the proposed approach. Our method incorporates several techniques including the least-square loss function, graph-based semisupervised learning, feature correlation mining, SPG, KKT, and discriminant multimaniifold analysis. It is named semisupervised discriminant multimaniifold analysis (SDMM). Following this, we describe how to obtain the classifiers in detail.

A. Formulation

To exploit the feature correlation for action recognition, we define the training set as $X = \{x_1, \ldots, x_n\} \in \mathbb{R}^{d \times n}$ and then associate it with its ground truth labels matrix $Y = [y_1, \ldots, y_n]^T \in \{0, 1\}^{n \times c}$. Note that $x_i \in \mathbb{R}^{d \times 1}$ is the $i$-th datum, and $n$ is the size of $X$. We aim to learn $c$ prediction functions (classifiers) $\{f_c\}_{c=1}^c$, with one for each class. $c$ stands for the class number. Usually, the prediction function $f$ is defined as

$$f(x) = w^T x \quad (1)$$

where $x$ is a datum and $w \in \mathbb{R}^{d \times 1}$ is weight vectors. By denoting $W = [w_1, \ldots, w_c] \in \mathbb{R}^{d \times c}$, the above-mentioned function becomes

$$f(X) = X^T W. \quad (2)$$

As indicated in [66], the least-square loss function achieves comparable performance to other loss functions, e.g., hinge loss or logistic loss. To obtain the projection matrix $W$, we employ least square regression to solve the following optimization problem:

$$\min_w \|X^T W - Y\|_F^2 + \alpha \|W\|_F^2 \quad (3)$$

where $\alpha$ is the regularization parameters. $\| \cdot \|_F$ denotes Frobenius norm, $\|W\|_F^2$ controls the complexity of the model to avoid overfitting.

Following the assumption of [31], the nearby data points are likely to have the same label, and the edges of graph $A$ refers to connect pairs of data points. $A$ denotes the symmetric matrix with elements describing the similarity between the pairs of data points. However, unlike [31], utilizing one graph model to approximate the density and manifold information,
in this paper, we model the local geometrical structure of manifolds by building a within-class similarity graph $A_w$ and a between-class similarity graph $A_b$. For simplicity, $A_w$ and $A_b$ are defined based on the nearest neighbor graphs as follows:

$$A_w(i, j) = \begin{cases} 1, & x_i \in N_w(x_j) \text{ or } x_j \in N_w(x_i) \\ 0, & \text{otherwise} \end{cases}$$ (4)

$$A_b(i, j) = \begin{cases} 1, & x_i \in N_b(x_j) \text{ or } x_j \in N_b(x_i) \\ 0, & \text{otherwise}. \end{cases}$$ (5)

In (4), $N_w(x_j)$ is the set of neighbors $x_j$, sharing the same label with $x_j$. $N_b(x_j)$ contains some neighbors having different labels in (5). We note that intraclass and interclass distances between points can be encoded on manifold by using similarity graphs [36].

### B. Discriminant Analysis

Our goal is to maximize discriminatory power while preserving local geometry, by mapping the points to new manifold, i.e., $w : X \rightarrow F$. To better demonstrate the relationship between the data distribution on manifold and feature correlation mining, we define a predicted label matrix $F = [F_1, \ldots, F_n]^T \in \mathbb{R}^{n \times c}$ for all the training videos in $X$, where $F_i \in \mathbb{R}^{c \times 1}$ is the predicted label vector of the $i$-th datum $x_i \in X$.

Inspired by the manifold discriminant analysis [34], [51], [67], we aim to minimize the intramanifold compactness and maximize the intermanifold separability simultaneously. A suitable transform would place the connected points of $A_w$ as close as possible, while moving the connected points of $A_b$ as far as possible. This goal can be achieved by optimizing the following two objective functions:

$$f_1 = \min \frac{1}{2} \sum_{\ell=1}^{c} \sum_{i,j=1}^{n} (F_{i\ell} - F_{j\ell})^2 A_w(i, j)$$ (6)

$$f_2 = \max \frac{1}{2} \sum_{\ell=1}^{c} \sum_{i,j=1}^{n} (F_{i\ell} - F_{j\ell})^2 A_b(i, j)$$ (7)

where $F_{i\ell}$ is the $\ell$-th element of $F_i$. $f_1$ punishes neighbors in the same class if they are mapped far away, while $f_2$ punishes samples of different classes if they are mapped close together. Hence, the overall discriminative information can be represented as

$$f = \frac{1}{2} \sum_{\ell=1}^{c} \sum_{i,j=1}^{n} (F_{i\ell} - F_{j\ell})^2 A_w(i, j) - \frac{1}{2}\beta \sum_{\ell=1}^{c} \sum_{i,j=1}^{n} (F_{i\ell} - F_{j\ell})^2 A_b(i, j)$$ (8)

where $\beta$ is a regularization parameter which controls the tradeoff between the intramanifold compactness term and the intermanifold separability term. Note that

$$\frac{1}{2} \sum_{\ell=1}^{c} \sum_{i,j=1}^{n} (F_{i\ell} - F_{j\ell})^2 A_w(i, j) = \frac{1}{2} \sum_{i,j=1}^{n} A_w(i, j)(F_{i}^T F_i + F_{j}^T F_j - 2F_{i}^T F_j)$$

$$= \text{tr}(F^T (D_w - A_w) F) = \text{tr}(F^T L_w F)$$ (9)

where tr$(\cdot)$ denotes trace operator, $D_w$ is a diagonal matrix with $D_w(i,i) = \sum_{j=1}^{n} A_w(i,j)$, and $L_w = D_w - A_w$ is the Laplacian matrix [68]. Similarly, (7) can be simplified to

$$\frac{1}{2} \sum_{\ell=1}^{c} \sum_{i,j=1}^{n} (F_{i\ell} - F_{j\ell})^2 A_b(i, j) = \text{tr}(F^T (D_b - A_b) F) = \text{tr}(F^T L_b F)$$ (10)

where $D_b$ is a diagonal matrix with $D_b(i,i) = \sum_{j=1}^{n} A_b(i,j)$. Therefore, equation (8) can be rewritten as

$$f = \frac{1}{2} \sum_{\ell=1}^{c} \sum_{i,j=1}^{n} (F_{i\ell} - F_{j\ell})^2 A_w(i, j)$$

$$- \frac{1}{2}\beta \sum_{\ell=1}^{c} \sum_{i,j=1}^{n} (F_{i\ell} - F_{j\ell})^2 A_b(i, j)$$

$$= \text{tr}(F^T (L_w - \beta L_b) F).$$ (11)

### C. Multitask Learning

To alleviate the tedious work in supervised learning, we extend the above-mentioned function to a graph-based semisupervised method for leveraging both labeled and unlabeled data as shown in [4] and [37]. Most existing semisupervised learning methods assume that the nearby data points are likely to have the same label. Specifically, the data points which can be connected via a path through high-density regions on the data manifold are likely to have the same label [4], [30], [37]. Nevertheless, the density and manifold information are inadequate due to limited labeled data. To relieve this problem, we utilize the graph model mentioned in Section III-B to approximate the density and manifold information.

To begin with, we redefine the training data set as $X = [X_1^T, X_2^T]^T$, where $X_1 = [x_1, \ldots, x_m]^T$ and $X_2 = [x_{m+1}, \ldots, x_n]^T$ are the two subsets of the data with labels and without labels, respectively. The label matrix of $X$ is $Y = [Y_1^T, Y_2^T]^T$, where $Y_1 = [y_1, \ldots, y_m]^T \in 0, 1^{m \times c}$ and $Y_2 = [y_{m+1}, \ldots, y_n]^T \in \mathbb{R}^{(n-m) \times c}$ is a matrix with all zeros. According to [30], [37], and [69], the graph embedded label prediction matrix $F$ should be consistent with similarity graphs $A_w$ and $A_b$, and the ground-truth labels $Y$. The idea of multimanihood and label consistency can be generalized as

$$\min_{F} \sum_{\ell=1}^{c} \left[ \frac{1}{2} \sum_{i,j=1}^{n} (F_{i\ell} - F_{j\ell})^2 A_w(i, j) - \frac{1}{2}\beta \sum_{i,j=1}^{n} (F_{i\ell} - F_{j\ell})^2 A_b(i, j) + \sum_{i=1}^{n} (F_{i\ell} - y_{i\ell})^2 \right]$$

$$\Rightarrow \min_{F} \text{tr}(F^T (L_w - \beta L_b) F) + \text{tr}(F - Y)^T (F - Y).$$ (12)
Different from previous shared structure learning algorithms [4], [30], [31], [37], we do not take shared structure learning into account in semisupervised learning framework. Instead, we propose a novel joint framework by incorporating graph embedding method on multitask manifold and the classifiers, which can be formulated as

$$\min_{F, W} \text{tr}(F^T (L_w - \beta L_b) F) + \text{tr}(F - Y)^T (F - Y)$$

$$+ \mu \sum_{t=1}^c \left( \sum_{i=1}^n \text{loss}(f_I(x_i), F_{it}) + \alpha \| w_t \|^2 \right)$$

(13)

where $\mu > 0$, $\alpha > 0$, and $\beta > 0$ are the regularization parameters. As discussed in Section III-A, we employ the Frobenius norm regularized loss function and then rewrite our objective as

$$\min_{F, W} \text{tr}(F^T (L_w - \beta L_b) F) + \text{tr}(F - Y)^T (F - Y)$$

$$+ \mu \| X^T W - F \|^2 + \alpha \| W \|^2).$$

(14)

There are three issues worthy of consideration. First, our objective function (14) is an unconstrained convex optimization problem. It does not contain the shared subspace information regularization term, hence the global optimum can be obtained by performing alternating least squares or SPG method [38], [42]. Second, the solution of objective functions shown in [1], [4], [30], and [31] have not discussed the singularity of the matrix. In [38] and [42], the SPG method has been proved that it can handle the aforementioned issues without matrix inversion. Third, the convergence conditions in [4], [30], and [31] merely depend on monotone decreasing. Since the objective function value becoming stable may be mathematically improper convergence, we utilize KKT conditions to deal with this matter.

D. Optimization

For reducing the dimension of $X$, we follow [37] to perform singular value decomposition (SVD), in which all the eigenvectors corresponding to the nonzero eigenvalues of the covariance matrix is preserved.

After that, according to [38] and [42], a general unconstrained minimization problem can be solved iteratively by introducing the SPG method and the trace operator. Therefore, we define a function $g(F, W)$ as a new objective problem instead of (14)

$$g(F, W) = \min_{F, W} \text{tr}(F^T (L_w - \beta L_b) F) + \text{tr}(F - Y)^T (F - Y)$$

$$+ \mu \text{tr}(X^T W - F)^T (X^T W - F) + \mu \alpha \text{tr}(W^T W).$$

(15)

By setting the derivative of (15) with respect to $F$ and $W$, respectively, we have

$$\nabla_F g = \frac{\partial g(F, W)}{\partial F}$$

$$= 2(L_w - \beta L_b) F + 2(F - Y) - 2 \mu (X^T W - F)$$

$$\nabla_W g = \frac{\partial g(F, W)}{\partial W}$$

$$= 2 \mu X (X^T W - F) + 2 \mu \alpha W.$$  

(16)

(17)

If $(F^*, W^*)$ is an approximate stationary point of (15), it is supposed to meet the KKT condition of (15) like this

$$\nabla g_F (F^*, W^*) = 0, \quad \nabla g_W (F^*, W^*) = 0.$$  

(18)

Then, the iteration stopping criterion becomes

$$\| \nabla g_F (F^*, W^*) \|^2 + \| \nabla g_W (F^*, W^*) \|^2 \leq \varepsilon$$  

(19)

where $\varepsilon$ is a nonnegative small constant. In summary, the classifiers training process of the proposed method is detailed in Algorithm 1.

**Algorithm 1 SDMM Algorithm**

**Input:**
- The training data $X \in \mathbb{R}^{d \times n}$
- The training data labels $Y \in \mathbb{R}^{n \times c}$
- Semi-supervised Parameters $\alpha, \beta$ and $\mu$.
- SPG Parameters $M, \alpha^+_{min}, \alpha^+_{max}, \gamma, \delta_1$ and $\delta_2$.

**Output:**
- Optimized $W^* \in \mathbb{R}^{d \times c}$

1. Perform SVD to reduce $X$’s dimension according to [37] to compute the within-class similarity graph $L_w \in \mathbb{R}^{n \times n}$.
2. Compute the between-class similarity graph $L_b \in \mathbb{R}^{n \times n}$.
3. Initialize $t = 0$, $a_0^+ \in (\alpha^+_{min}, \alpha^+_{max})$, $\lambda = 1$.
4. Initialize $F^{(0)} \in \mathbb{R}^{n \times c}$ randomly.
5. Initialize $W^{(0)} \in \mathbb{R}^{d \times c}$ randomly.

1. **repeat**
   2. Compute $d F^{(t)} = -\alpha^+_t \nabla g_F (F^{(t)}, W^{(t)})$
   3. Compute $d W^{(t)} = -\alpha^+_t \nabla g_W (F^{(t)}, W^{(t)})$
   4. Compute $\tilde{F} = F^{(t)} + \lambda d F^{(t)}$
   5. Compute $\tilde{W} = W^{(t)} + \lambda d W^{(t)}$
   6. if $g(\tilde{F}, \tilde{W}) < \gamma \{ (d F^{(t)}, \nabla g_F (F^{(t)}, W^{(t)})) + (d W^{(t)}, \nabla g_W (F^{(t)}, W^{(t)})) \}$
      max$
      0 \leq j \leq \text{min}(M - 1)$
      $g(F^{(t-j)}, W^{(t-j)})$
      then
    7. $F^{(t+1)} = \tilde{F}$, $W^{(t+1)} = \tilde{W}$
    8. $s_1^{(t)} = F^{(t+1)} - F^{(t)}, s_2^{(t)} = W^{(t+1)} - W^{(t)}$
    9. $\gamma_1^{(t)} = \nabla g_F (F^{(t+1)}, W^{(t+1)}) - \nabla g_F (F^{(t)}, W^{(t)})$
    10. $\gamma_2^{(t)} = \nabla g_W (F^{(t+1)}, W^{(t+1)}) - \nabla g_W (F^{(t)}, W^{(t)})$
    11. Compute $b_t = (\gamma_1^{(t)}, \gamma_1^{(t)}) + (s_2^{(t)}, s_2^{(t)})$
    12. if $b_t \leq 0$ then $a_{t+1}^+ = \alpha^+_{max}$
    13. else
    14. Compute $a_t = (s_1^{(t)}, s_2^{(t)}) + (s_2^{(t)}, s_2^{(t)})$
    15. Compute $a^+_{t+1} = \min(a^+_{max}, \max(a^+_{min}, a^+_t))$
    16. **end if**
    17. $t = t + 1$
    18. **else**
    19. $\lambda_{new} \in [\delta_1 \lambda, \delta_2 \lambda]$
    20. $\lambda = \lambda_{new}$
    21. **end if**
22. **until Convergence according to (19)**

Return $W^*$

**IV. EXPERIMENTS**

To validate our method for action recognition in videos, we first demonstrate Fisher vector (FV) used for data representation. Then, we conduct extensive experiments on challenging data sets to test our framework’s performance.
TABLE I

<table>
<thead>
<tr>
<th>Data Set</th>
<th>Ours</th>
<th>IDT-Based</th>
<th>UC50</th>
<th>UCF101</th>
</tr>
</thead>
<tbody>
<tr>
<td>JHMDB</td>
<td>0.4234</td>
<td>0.1628</td>
<td>0.6814</td>
<td>0.5975</td>
</tr>
<tr>
<td>HMDB51</td>
<td>0.3258</td>
<td>0.1973</td>
<td>0.5465</td>
<td>0.4715</td>
</tr>
<tr>
<td>UCF50</td>
<td>0.3440</td>
<td>0.2246</td>
<td>0.5051</td>
<td>0.4386</td>
</tr>
<tr>
<td>UCF101</td>
<td>0.3552</td>
<td>0.2405</td>
<td>0.5803</td>
<td>0.4858</td>
</tr>
<tr>
<td>SVM-linear</td>
<td>0.3324</td>
<td>0.2019</td>
<td>0.5420</td>
<td>0.4209</td>
</tr>
<tr>
<td>SVM-classifier</td>
<td>0.3737</td>
<td>0.2551</td>
<td>0.5831</td>
<td>0.4681</td>
</tr>
</tbody>
</table>

TABLE II

<table>
<thead>
<tr>
<th>Data Set</th>
<th>Ours</th>
<th>IDT-Based</th>
<th>UC50</th>
<th>UCF101</th>
</tr>
</thead>
<tbody>
<tr>
<td>JHMDB</td>
<td>0.4665</td>
<td>0.1787</td>
<td>0.7443</td>
<td>0.5963</td>
</tr>
<tr>
<td>HMDB51</td>
<td>0.3775</td>
<td>0.2045</td>
<td>0.5905</td>
<td>0.5265</td>
</tr>
<tr>
<td>UCF50</td>
<td>0.4135</td>
<td>0.2639</td>
<td>0.6375</td>
<td>0.5446</td>
</tr>
<tr>
<td>UCF101</td>
<td>0.3748</td>
<td>0.2616</td>
<td>0.6067</td>
<td>0.5127</td>
</tr>
<tr>
<td>SVM-linear</td>
<td>0.4120</td>
<td>0.2902</td>
<td>0.6620</td>
<td>0.5551</td>
</tr>
<tr>
<td>SVM-classifier</td>
<td>0.5497</td>
<td>0.3177</td>
<td>0.7135</td>
<td>0.6280</td>
</tr>
</tbody>
</table>

TABLE III

<table>
<thead>
<tr>
<th>Data Set</th>
<th>Ours</th>
<th>IDT-Based</th>
<th>UC50</th>
<th>UCF101</th>
</tr>
</thead>
<tbody>
<tr>
<td>JHMDB</td>
<td>0.5479</td>
<td>0.1819</td>
<td>0.8007</td>
<td>0.6643</td>
</tr>
<tr>
<td>HMDB51</td>
<td>0.4836</td>
<td>0.2015</td>
<td>0.7139</td>
<td>0.6280</td>
</tr>
<tr>
<td>UCF50</td>
<td>0.5143</td>
<td>0.2071</td>
<td>0.7558</td>
<td>0.6696</td>
</tr>
<tr>
<td>UCF101</td>
<td>0.4618</td>
<td>0.2325</td>
<td>0.7384</td>
<td>0.6057</td>
</tr>
<tr>
<td>SVM-linear</td>
<td>0.4957</td>
<td>0.2218</td>
<td>0.7570</td>
<td>0.6226</td>
</tr>
<tr>
<td>SVM-classifier</td>
<td>0.7284</td>
<td>0.2015</td>
<td>0.8427</td>
<td>0.8477</td>
</tr>
</tbody>
</table>

TABLE IV

<table>
<thead>
<tr>
<th>Data Set</th>
<th>Ours</th>
<th>IDT-Based</th>
<th>UC50</th>
<th>UCF101</th>
</tr>
</thead>
<tbody>
<tr>
<td>JHMDB</td>
<td>0.5822</td>
<td>0.2093</td>
<td>0.8576</td>
<td>0.7218</td>
</tr>
<tr>
<td>HMDB51</td>
<td>0.5341</td>
<td>0.2012</td>
<td>0.7811</td>
<td>0.6437</td>
</tr>
<tr>
<td>UCF50</td>
<td>0.5278</td>
<td>0.1918</td>
<td>0.7935</td>
<td>0.6329</td>
</tr>
<tr>
<td>UCF101</td>
<td>0.5334</td>
<td>0.2001</td>
<td>0.8014</td>
<td>0.6701</td>
</tr>
<tr>
<td>SVM-linear</td>
<td>0.5909</td>
<td>0.2016</td>
<td>0.7782</td>
<td>0.6324</td>
</tr>
<tr>
<td>SVM-classifier</td>
<td>0.5341</td>
<td>0.2085</td>
<td>0.8104</td>
<td>0.6796</td>
</tr>
</tbody>
</table>

A. Data Sets

In the experiments, three data sets are used, including the JHMDB data set [70], the HMDB51 data set [71], the UCF50 data set [72], and the UCF101 data set [73]. The JHMDB data set is a subset of HMDB51 with 928 clips comprising 21 action categories. The HMDB51 data set contains 6766 video sequences recording 51 action categories. The UCF50 data set has 50 action categories, consisting of real-world videos taken from YouTube. There are 6618 video clips in UCF50. The UCF101 data set collects 13,320 video clips including 101 action categories. As far as the testing set, we use the standard testing set provided by the authors on JHMDB and HMDB51 data sets, and the testing set of the first split on UCF50 and UCF101 data sets. Due to the random training samples selection, we repeat the experiment for 10 trials to avoid any bias. The average accuracy and standard deviation are reported.

For the JHMDB and HMDB51 data sets, we follow [30] and [31] that randomly split each data set into training and testing sets, we only use the first split provided by author due to computation complexity and limited memory resource. In addition, we randomly select 30 videos per category as the training data including the labeled and unlabeled samples and apply the original testing sets for comparison in a more fair way.

B. Features

For hand-crafted features, we extract improved dense trajectories (IDTs)-based features with HOG + HOF + MBH descriptors [74]. The dimension D is reduced to 198 by performing PCA and L2-normalization. After training a GMM codebook with K Gaussians based on 256 000 randomly sampled features, each action video is represented by a 2DK = 6336 dimensional FV with Power L2-normalization, if K = 16 as Tables–IV.

For deep-learned features, the convolutional neural networks (CNN)-based features are selected, e.g., the trajectory-pooled
deep-convolutional descriptors (TDDs) [75] and temporal segment networks (TSNs) [76]. We follow [75] to concatenate eight normalized deep-learned features from spatial conv4 + conv5 and temporal conv3 + conv4 layers, let the dimension of combined TDDs becomes \( D = 64 \times 8 = 512 \), since each TDDs’ dimension of a video is decorrelated to 64 by PCA. Then, we encode the combined TDDs into FV representation, and the final dimension of each video can be changed to \( 2DK = 16384 \) when \( K = 16 \), as shown in Fig. 3. Meanwhile, the TSN models of \( 3 \times c, 5 \times c, 10 \times c, \) and \( 15 \times c \) are retrained according to [76], then we extract the global pool features of \( 3 \times c, 5 \times c, 10 \times c, \) and \( 15 \times c \), respectively, by corresponding trained TSN model, concatenate rgb + flow into 2048 dimension with Power L2-normalization, as shown in Tables V–VIII.

C. Experimental Setup

To evaluate the performance of our approach, the proposed algorithm is compared to the five state-of-the-art methods which include SVM with \( \chi^2 \) kernel, SVM with linear kernel, SFU, SFCM, SFUS, MFCU, and SDMM [30] and [31]. Note that SFCM, SFUS, and MFCU are semisupervised learning approaches. SFCM and MFCU also exploit the data manifold and are designed for action recognition. To demonstrate the superiority of our method, we employed these related state-of-the-art methods for comparison. Also, with the available source codes, we can run experiments on different data sets and settings to facilitate fair comparisons.

For training phase, we denote \( c \) as the class number for each data set (\( c = 21, 51, 50, \) and 101 for JHMDB, HMDB51, UCF50, and UCF101, respectively). As semisupervised training set contains both labeled and unlabeled data, we randomly select 30 videos per category in the training set, where \( m \) labeled videos (\( m = 3, 5, 10, \) and 15) per category are sampled, thus resulting in \( 3 \times c, 5 \times c, 10 \times c, \) and \( 15 \times c \) randomly labeled videos, while the remaining training videos are unlabeled.

For testing phase, we use the standard testing set provided by the author on JHMDB and HMDB51 data sets, and the first split of the testing set on UCF50 and UCF101 data sets due to the limited memory resource.

For semisupervised parameters, including SFUS, SFCM, MFCU, and our SDMM’s \( \alpha, \beta, \mu \), we follow the same setting utilized in [30] and [31] and range from \( \{10^{-4}, 10^{-3}, 10^{-2}, 10^{-1}, 1, 10^1, 10^2, 10^3, 10^4\} \).

For SPG parameters, since they are not sensitive to our algorithm, we follow [38] and set \( M = 10, \varepsilon_{\text{min}} = 10^{-15}, \) and \( a_{\text{max}} = 10^{15} \), sufficient decrease parameter \( \gamma = 10^{-4} \), safeguarding parameters \( \delta_1 = 0.1, \delta_2 = 0.9 \), and \( \lambda_{\text{new}} = (1/2)(\delta_1 \lambda + \delta_2 \lambda). \) Initially, \( a_0 \in [a_{\text{min}}, a_{\text{max}}] \) is arbitrary, we set \( a_0 = 1 \) in our experiments. Technically, since the dimension of FV is relatively high, it is hard to stop iteration for merely subtracting the last two objective function values, we regard the relative error of the objective function values as iteration stopping criterion in Algorithm 1. The nonnegative small constant \( \varepsilon \) of (19) is suggested to be set \( 10^{-6} \).

D. Comparison Results

Tables I–VIII show the action recognition results on four challenging data sets with respect to different number of labeled training data. Specifically, we compare the proposed method to those other approaches that only apply a single type of feature, i.e., FV representation.

1) Performance on Action Recognition: We observe the following.

1) Our method consistently obtains the best recognition performance, the recognition of our semisupervised classifiers even better than the popular supervised classifiers such as linear SVM.

2) We verify the effectiveness of the proposed method with IDT-and CNN-based representations beyond Bag-of-Words.

3) All methods achieve worse results on HMDB51 compared with those on another three data sets. This is probably owed to the complexity of HMDB51.

4) The recognition accuracy of all methods is improved with the increase of the number of labeled training videos.

5) Our method gains better performance when the amount of labeled data is small. For example, when only \( 3 \times c \) (63 out of 660 training data for JHMDB) training data are labeled, our method achieves the recognition accuracy of 42.38%, which is better than others.

These results indicate that our algorithm benefits from the multimaniifold analysis of feature correlations.

The fully supervised linear SVM is taken as baseline, we average the accuracy of \( 3 \times c, 5 \times c, 10 \times c, \) and \( 15 \times c \) cases totally. Using the IDT features, the average accuracy of our SDMM on JHMDB, HMDB51, UCF50, and
labeled video data from daily life, cannot adapt to large-scale samples. However, small-scale data sets such as deep layers, the spatial net, and the temporal net rely well in the small labeled data case. Finally, the deep learning KKT conditions mathematically, thus our multimanifold works.

1) Unlabeled Samples: We leverage the unlabeled data, the recognition performance can be improved.

The deep learning approaches, which trained on large-scale labeled data, have shown promising performance on image classification and action recognition. To validate the performance of the deep learning approaches on small-scale data set, we follow the experimental setup of [75] to extract TDDs, then encode the TDDs of every video into FV representation and recognize actions by linear SVM on JHMDB. In order to train GMMs with $K (K = 256)$, we decorrelate TDDs with PCA and reduce its dimension to $D = 64$. Note that we utilize the combined TDDs from spatial conv4 + conv5 and temporal conv3 + conv4 nets.

We compare our SDMM algorithm with linear SVM and TDD in case of $15 \times c$, the comparison results on JHMDB data set with respect to different gmmSize is shown in Fig. 3. Note that this figure contains both semisupervised learning and fully supervised learning which use 15\ resides. The number of labeled training samples is set to $15 \times c$. Note that we utilize the combined TDDs from spatial conv4 + conv5 and temporal conv3 + conv4 nets.

We compare our SDMM algorithm with linear SVM and TDD in case of $15 \times c$, the comparison results on JHMDB data set with respect to different gmmSize is shown in Fig. 3. Note that this figure contains both semisupervised learning and fully supervised learning which use 15\ resides. The number of labeled training samples is set to $15 \times c$. Note that we utilize the combined TDDs from spatial conv4 + conv5 and temporal conv3 + conv4 nets.

We compare our SDMM algorithm with linear SVM and TDD in case of $15 \times c$, the comparison results on JHMDB data set with respect to different gmmSize is shown in Fig. 3. Note that this figure contains both semisupervised learning and fully supervised learning which use 15\ resides. The number of labeled training samples is set to $15 \times c$. Note that we utilize the combined TDDs from spatial conv4 + conv5 and temporal conv3 + conv4 nets.

These results may account for many reasons. First, our method not only takes the advantage of compared semisupervised approaches in [4], [30], [31], and [37] but also leverages the intraclass compactness and interclass separability simultaneously, hence our performance gain over other methods is more significant when the labeled data are small. Second, we enlarge the geometric structure information of feature subspace by increasing training samples with many unlabeled samples for discriminant learning, and the objective function optimization is solved by the SPG method and the KKT conditions mathematically, thus our multimanifold works well in the small labeled data case. At last, the deep learning approaches that are trained such as TDD built on CNN with deep layers, the spatial net, and the temporal net rely on large-scale samples. However, small-scale data set such as real-world surveillance applications, which are hard to collect labeled video data from daily life, cannot adapt to deep learning approaches, because the scale of network weights, which are learned by using fine-tuned network structure based on large-scale data sets, may be larger than the scale of action features.

2) Convergence Study: To validate the proposed algorithm that can derive optimum solution by the SPG method and the KKT conditions, we conduct experiments on all four data sets by applying convergence curves of the objective function values. The number of labeled training samples is set to $15 \times c$ for each data set, and the parameters are set to the median value of the tuned range. The results in Fig. 4 demonstrate that the objective function values converge after only a few iterations. Note that there are oscillations caused by the SPG method in our convergence curves, the objective values are not monotonically decreasing before iterations stop.

3) Computation Speed: We also set a practical example for comparing the computation speed of the aforementioned semisupervised algorithms. We consider the case of $15 \times c$ labeled samples for JHMDB, and use the training–testing set given in Section IV-C, train GMMs with $K = 16$ and then compute the average run time of algorithms over the standard splits. Given the high dimension of raw features are utilized in SFUS, SFCM, and MFCU, we first perform SVD to reduce raw features’ dimension according to [37]. Nevertheless, our SDMM still obtains the fastest speed due to the trait of the SPG method. Compared with the SFUS, SFCM, and MFCU, the run time of SDMM gains 1.06\, 4.15\, and 2.50\ faster, respectively, as shown in Table IX.

4) Parameter Sensitivity Study: Our algorithm involves two types of parameters, i.e., semisupervised parameters and SPG parameters. To learn how they affect the analysis performance and iteration process on action recognition, we conduct extensive experiments on the parameter sensitivity.

For semisupervised parameters, we first verify that SDMM benefits from intramanifold and intermanifold by multimanifold discriminant analysis in Fig. 5(a) and (b). The JHMDB and HMDB51 data sets are taken to study the impact of multimanifold learning. We fix $\alpha$ and $\mu$ at their optimal values over the second split, i.e., $10^{-3}$ and $10^3$, respectively, for $15 \times c$ labeled training data. It can be seen that as $\beta$ varies from $10^{-4}$ to $10^{-2}$, the accuracy increases accordingly and reaches the peak value when $\beta = 10^{-2}$. Note that Fig. 5(a) and (b) can be regarded as the influence of both intramanifold and intermanifold structure proportion on accuracy.

UCF101 is improved by 5.02\%, 3.82\%, 4.16\%, and 4.24\%, respectively. While using the TSN features, the average accuracy of our SDMM on JHMDB, HMDB51, UCF50, and UCF101 is improved by 4.06\%, 3.92\%, 5.06\%, and 3.39\%, respectively, as compared with linear SVM. It is evident that the performance of the deep learning approaches on small-scale data can be improved.

These results may account for many reasons. First, our method not only takes the advantage of compared semisupervised approaches in [4], [30], [31], and [37] but also leverages the intraclass compactness and interclass separability simultaneously, hence our performance gain over other methods is more significant when the labeled data are small. Second, we enlarge the geometric structure information of feature subspace by increasing training samples with many unlabeled samples for discriminant learning, and the objective function optimization is solved by the SPG method and the KKT conditions mathematically, thus our multimanifold works well in the small labeled data case. At last, the deep learning approaches that are trained such as TDD built on CNN with deep layers, the spatial net, and the temporal net rely on large-scale samples. However, small-scale data set such as real-world surveillance applications, which are hard to collect labeled video data from daily life, cannot adapt to deep learning approaches, because the scale of network weights, which are learned by using fine-tuned network structure based on large-scale data sets, may be larger than the scale of action features.

2) Convergence Study: To validate the proposed algorithm that can derive optimum solution by the SPG method and the KKT conditions, we conduct experiments on all four data sets by applying convergence curves of the objective function values. The number of labeled training samples is set to $15 \times c$ for each data set, and the parameters are set to the median value of the tuned range. The results in Fig. 4 demonstrate that the objective function values converge after only a few iterations. Note that there are oscillations caused by the SPG method in our convergence curves, the objective values are not monotonically decreasing before iterations stop.

3) Computation Speed: We also set a practical example for comparing the computation speed of the aforementioned semisupervised algorithms. We consider the case of $15 \times c$ labeled samples for JHMDB, and use the training–testing set given in Section IV-C, train GMMs with $K = 16$ and then compute the average run time of algorithms over the standard splits. Given the high dimension of raw features are utilized in SFUS, SFCM, and MFCU, we first perform SVD to reduce raw features’ dimension according to [37]. Nevertheless, our SDMM still obtains the fastest speed due to the trait of the SPG method. Compared with the SFUS, SFCM, and MFCU, the run time of SDMM gains 1.06\, 4.15\, and 2.50\ faster, respectively, as shown in Table IX.

4) Parameter Sensitivity Study: Our algorithm involves two types of parameters, i.e., semisupervised parameters and SPG parameters. To learn how they affect the analysis performance and iteration process on action recognition, we conduct extensive experiments on the parameter sensitivity.

For semisupervised parameters, we first verify that SDMM benefits from intramanifold and intermanifold by multimanifold discriminant analysis in Fig. 5(a) and (b). The JHMDB and HMDB51 data sets are taken to study the impact of multimanifold learning. We fix $\alpha$ and $\mu$ at their optimal values over the second split, i.e., $10^{-3}$ and $10^3$, respectively, for $15 \times c$ labeled training data. It can be seen that as $\beta$ varies from $10^{-4}$ to $10^{-2}$, the accuracy increases accordingly and reaches the peak value when $\beta = 10^{-2}$. Note that Fig. 5(a) and (b) can be regarded as the influence of both intramanifold and intermanifold structure proportion on accuracy.
Since we perceive the proportion of intramanifold structure as constant $\mu$, hence a larger $\frac{\mu}{\mu}$ means a larger proportion of intermanifold structural consideration, and vice versa. When $\beta = 0$, no intramanifold structure is utilized, thus, if $\beta \rightarrow +\infty$, no intramanifold structure is contained. The results illustrate that appropriately exploiting intraclass compactness and interclass separability simultaneously in multimanifold subspace can further improve the performance. Then, we keep $\beta = 10^{-2}$, and show the parameter sensitivity results in Fig. 5(c) and (d). From these figures, we can see that mining correlations between multiple related tasks are beneficial to improve the performance. More specifically, we conduct extensive experiments on HMDB51 using TSN features, as shown in Fig. 5(b) and (d). Fig. 5 shows that the recognition can achieve stable high accuracy when all the hyperparameters are selected in certain range, e.g., $\alpha$ ranges in $\{10^{-2}, 10^{-1}, 1\}$, $\beta$ ranges in $\{10^{-3}, 10^{-2}, 10^{-1}\}$, and $\mu$ ranges in $\{10^{-1}, 1, 10^1\}$. In other words, there is flexibility in choosing the parameters in order to achieve optimal performance.

For SPG parameters, accuracy and $M$ are used to reflect the performance and iteration variation, respectively, where $M$ denotes the number of former iteration which is designed for inequality calculation. In algorithm 1, the step 6 of SPG method, new objective function value $g(\hat{F}, \hat{W})$ is supposed to compare with the former $M$th objective function values. Fig. 4 illustrates the iteration variation with respect to $M$ on four databases. In Fig. 4, the iteration process changes slightly corresponding to different values of $M$. The impact of different values of these parameters is supposed to be related to the trait of the feature representation. Generally speaking, $M$ is not sensitive to the iteration of SDMM.

V. Conclusion

In this paper, a novel algorithm is proposed to categorize human actions in videos by exploring data distribution and feature correlation. Using a multimanifold-based joint framework, our method discovers the intrinsic relationship of midlevel features to improve recognition performance. Second, the SPG method and the KKT conditions are applied to optimize the objective function for training robust classifiers. Finally, we extend the classifier into the semisupervised scenario to exploit both labeled and unlabeled videos. We evaluate our framework for action recognition on four challenging data sets. The experimental results show that our approach outperforms all compared algorithms, especially when the amount of labeled data is relatively small. Since semisupervised learning methods based on generative adversarial networks (GANs) have obtained strong empirical results, we prepare to discover discriminative information via GANs with shallow layers in the future.

REFERENCES


Jun Chen received the B.S. degree in automation from Beijing Forestry University, Beijing, China, in 2009, the M.S. degree of electrical engineering from Mississippi State University, Starkville, MS, USA, in 2012, and the Ph.D. degree from the Department of Electrical Engineering, The University of Texas at Dallas, Richardson, TX, USA, in 2016. He was a Post-Doctoral Research Associate with the Center for Research in Computer Vision, University of Central Florida, Orlando, FL, USA, from 2016 to 2018. He is currently an Assistant Professor of Electrical and Computer Engineering, University of North Carolina at Charlotte, Charlotte, NC, USA. His current research interests include signal and image processing, computer vision, and deep learning. He published over 50 papers in refereed journals and conferences in these areas.

Junjun Jiang received the B.S. degree from the Department of Mathematics, Huaqiao University, Quanzhou, China, in 2009, and the Ph.D. degree from the School of Computer Science, Wuhan University, Wuhan, China, in 2014. From 2015 to 2018, he was an Associate Professor with the China University of Geosciences, Wuhan. Since 2016, he has been a Project Researcher with the National Institute of Informatics, Tokyo, Japan. He is currently a Professor with the School of Computer Science and Technology, Harbin Institute of Technology, Harbin, China. His current research interests include image processing and computer vision.

Jiaofen Li received the B.S. degree from the Changsha University of Science and Technology, Changsha, China, in 2005, and the M.S. and Ph.D. degrees from the College of Mathematics and Econometrics, Hunan University, Changsha, in 2007 and 2010, respectively. He is currently a Professor with the School of Mathematics and Computational Science, Guilin University of Electronic Technology, Guilin, China. His current research interests include inverse eigenvalue problem, linear matrix equation, and matrix algebra in control theory.

Hongyang Li received the M.S. degree from the Huazhong University of Science and Technology, Wuhan, China, in 2005. He is currently pursuing the Ph.D. degree with the National Engineering Research Center for Multimedia Software, School of Computer Science, Wuhan University. His current research interests include computer vision and pattern recognition.