Deep Manifold Structure Transfer for Action Recognition

Ce Li, Baochang Zhang, Chen Chen, Qixiang Ye, Jungong Han, Guodong Guo, and Rongrong Ji

***Abstract*—While intrinsic data structure in subspace provides useful information for visual recognition, it has not yet been well studied in deep feature learning for action recognition. In this paper, we introduce a new spatio-temporal manifold network (STMN) that leverages data manifold structures to regularize deep action feature learning, aiming at simultaneously minimizing the intra-class variations of learned deep features and alleviating the over-fitting problem. To this end, the manifold prior is imposed from the top layer of a convolutional neural network (CNN), and is propagated across convolutional layers during forward-backward propagation. The observed correspondence of manifold structures in the data space and feature space validates that the manifold priori can be transferred across CNN layers. STMN theoretically recasts the problem of transferring the data structure prior into the deep learning architectures as a projection over the manifold via an embedding method, which can be easily solved by an Alternating Direction Method of Multipliers and Backward Propagation (ADMM-BP) algorithm. STMN is generic in the sense that it can be plugged into various backbone architectures to learn more discriminative representation for action recognition. Extensive experimental results show that our method achieves comparable or even better performance as compared with the state-of-the-art approaches on four benchmark datasets.**

***Index Terms*—Action Recognition, Manifold, Alternating Direction Method of Multipliers, Backward Propagation, ADMM-BP.**

I. Introduction

HUMAN action recognition has been extensively studied in the computer vision community [1]–[9], due to its broad range of applications in human computer interaction, video content analysis, and video surveillance. While many researchers view action recognition in constrained simple backgrounds as a well solved problem, action recognition in real-world complex scenes possess many hurdles driven by the change of human poses, viewpoints, and backgrounds.

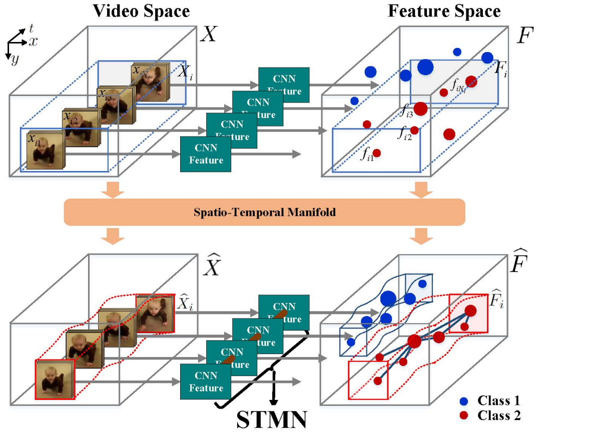


Fig. 1. Spatio-temporal manifold network (STMN) is a network designed by the intuition that action samples have intrinsic data structure, based on which an intra-class action space containing various samples is defined to be a spatio-temporal manifold. For action recognition, existing CNN features have been well studied to distinguish the inter-class variability, unfortunately ignoring the intrinsic data structure and intra-class variation. We model the intra-class action space as a spatio-temporal manifold, which is used as a regularization term in the loss function. Consequently, the manifold structure of intra-class actions remains in the resulting STMN approach. Two classes (blue/red) of samples in the CNN feature space are randomly distributed (upper). Differently, the manifold structure in STMN regularizes the samples in a compact space (bottom).

Action classification in video had been one of the most challenging problems next to the image classification [10]. Recent deep learning approaches including 3D CNN [11], two-stream CNNs [2], C3D [12], TDD [13], TSN [14], ST-ResNet+iDT [15], L^2STM [16], ST-VLMPF [17], P3D ResNet [18], I3D [19], 3D ResNeXt [20], R(2+1)D-TwoStream [7], CO2FI+ASYN [21], and DML [22] have shown state-of-the-art performances in action recognition. The recent development of CNNs with spatio-temporal 3D convolutional kernels (3D CNNs) rapidly grows and contributes to significant advances in video recognition [7], [18]–[20] because 3D CNNs can be used to directly extract spatio-temporal features from raw videos. However, most of these approaches aim at distinguishing the inter-class variability, but often ignore the intra-class distribution [23], and therefore could suffer from the over-fitting problem [22], particularly when the video datasets are relatively small compared to immense number of parameters in 3D CNNs [20] and intra-class variations in the training data are significant [22].

The work was supported in part by the Natural Science Foundation of China under Contract 61601466, 61672079 and 61473086. This work was supported by the National Basic Research Program of China (2015CB352501), Shenzhen Science and Technology Program, and Shenzhen Peacock Plan KQTD2016112515134654, and also National Key R&D Plan (2017YFC0821102). (Baochang Zhang and Jungong Han are the correspondences.)

C. Li is China University of Mining & Technology, Beijing, China. Email: celi@cumtb.edu.cn.

B. Zhang is with the School of Automation Science and Electrical Engineering, Beihang University, Beijing, China, and with the Shenzhen Academy of Aerospace Technology, Shenzhen, China. Email: bczhang@buaa.edu.cn.

C. Chen is with Department of Electrical and Computer Engineering, University of North Carolina at Charlotte, NC, USA. Email: chen.chen@uncc.edu.

Q. Ye is with School of Electronic, Electrical and Communication Engineering, University of Chinese Academy of Sciences, Beijing, China. Email: qxye@ucas.ac.cn.

J. Han is with the School of Computing and Communications, Lancaster University, Lancaster LA1 4YW, U.K. Email: jungonghan77@gmail.com.

G. Guo is with the Institute of Deep Learning, Baidu Research and National Engineering Laboratory for Deep Learning Technology and Application, Beijing, China.

R. Ji is with the School of Information Science and Engineering, Xiamen Univesity, Fujian, China. Email: rrji@xmu.edu.cn.

Manuscript received XX XX, 2018.

To alleviate the over-fitting problem, regularization techniques [8], [22]–[24] and prior knowledge, *e.g.*, 2D topological structure of input data [25], graph-based embedding [26], are explored in deep feature learning. Nonlinear structures, *e.g.*, Riemanian manifold [27], Grassmann manifold [28], [29] have been incorporated as constraints to balance the learned model [30], [31]. However, the problem about how to model the data structure priori into the deep learning architectures remains not being well solved.

In this paper, we propose a spatio-temporal manifold [1] network (STMN) approach for action recognition, to alleviate the above problems from the perspective of deep learning regularization. Fig. 1 depicts the basic idea, where the spatial manifold models the non-linearity of action samples while the temporal manifold considers the dependence structure of action video frames. With spatial and temporal manifolds defined, this paper aims to answer "how the spatio-temporal manifold can be embedded into CNN to preserve the data structure priori and thus improve the action recognition performance".

Specifically, our assumption is that the intrinsic data structure, *i.e.*, manifold structure, can be preserved in the deep learning pipeline, by being transferred from the input video sequences into the feature space. With this assumption, CNN is exploited to extract feature maps with respect to the overlapped clips of each video. Meanwhile, a new manifold constraint model is intuitively obtained and embedded into the loss function of CNN to reduce the structure variations in the high-dimensional data. The resulting constrained optimization problem is solved with an Alternating Direction Method of Multipliers and Backward Propagation (ADMM-BP) algorithm. This is based on the theoretical analysis that the manifold structure constraint can be seamlessly fused with the back propagation procedure through manifold embedding in the feature layer (the last layer of CNN). As a result, the optimization algorithm can be easily implemented by using a projection operation to introduce the manifold constraint. The main contributions of this paper include:

1. The deep manifold structure transfer is introduced into the loss function of a deep learning model as a regularization term for action recognition. The resulting STMN framework reduces the intra-class variations, improves the generalization capability, and alleviates the over-fitting problem of classifying action sequences.
2. An Alternating Direction Method of Multipliers and Backward Propagation (ADMM-BP) algorithm is developed to transfer the manifold structure between the input samples and deep features, which leads to a new framework to solve the theoretically reformulated optimization problem "how the structure of the *data* can be transferred to constrain the *variable* in learning 3D CNNs".
3. Extensive experimental results show that our method achieves comparable or better performance as compared with the state-of-the-art approaches on four benchmark datasets.

The rest of the paper is organized as follows. Section II introduces the related works, and Section III describes the details of the proposed method. Experiments and results are presented in Section IV, and Section V concludes the paper.

II. Related Work

Action recognition has attracted much attention in the past decade [22], [29]. The targets of action recognition evolute from scimple background to real-world video sequences, while the recognition methods shift from hand-crafted to learning based. Early methods represent human actions by hand-crafted features [32], [33], such as Harris-3D, SIFT-3D, HOG-3DHOF [34], ESURF [33] and MBH [35], and explicit motion modeling [36]. Recently, Wang *et al.* [37] proposed an improved dense trajectories (iDT) method, which is the state-of-the-art hand-crafted feature with densely sampled and tracked optical flow points along trajectories. However, it becomes intractable on large-scale dataset due to its expensive computation cost.

Recently, more effective action recognition approaches root in powerful learning methods, particulary the deep CNN approaches [25], [38]–[42]. The existing deep models to learn action representations from videos are categorized as four architectures, namely spatio-temporal networks, multiple stream networks, deep generative networks, and temporal coherency networks. The most dominant method is spatio-temporal networks including Stacked ISA [25], Gated Restricted Boltzmann [43], extended 3D CNN [11]. Despite of good performance achieved by deep CNN methods, they are usually adapted from image-based deep learning approaches, ignoring the temporal information of action video sequences. To arm the convolutional operation with temporal information, Ng *et al.* [44] explored temporal pooling and concluded that max pooling in the temporal domain is preferable, Karpathy *et al.* [41] proposed the concept of slow fusion to increase the temporal awareness of a CNN and trained deep structures on Sports-1M dataset. Tran *et al.* [12] performed 3D convolutions and 3D pooling to extract a generic video representation by exploiting the temporal information in the deep architecture named C3D, then Varol *et al.* [45] explored the effect of performing 3D convolutions over longer temporal durations as the input layer. Besides, some works tackle action recognition through a cascade of spatial CNN and a class of Long-Short Term Memory (LSTM) and Gated Recurrent Units (GRU) [46]–[48], such as ConvNet+LSTM (LRCN) [47], ConvNet+LSTM (unsup) [49], L^{2}STM [16], thereby selecting attention parts and incorporating temporal dependency between frames. However, the performance improvements over the spatial baselines have also been marginal in video classification [16], [44], [47], [49], [50].

Different from the above-mentioned spatio-temporal networks using raw video inputs, the second main architecture, multiple stream networks, usually devise separate motion information (e.g., optical flow) from appearance input (i.e., video frames) by pre-processing. Simonyan *et al.* [2] introduced the first multiple-stream deep convolutional networks where two parallel CNNs are used for capturing motion information.

[1] The spatio-temporal structure is calculated based on sample sets from manifold.

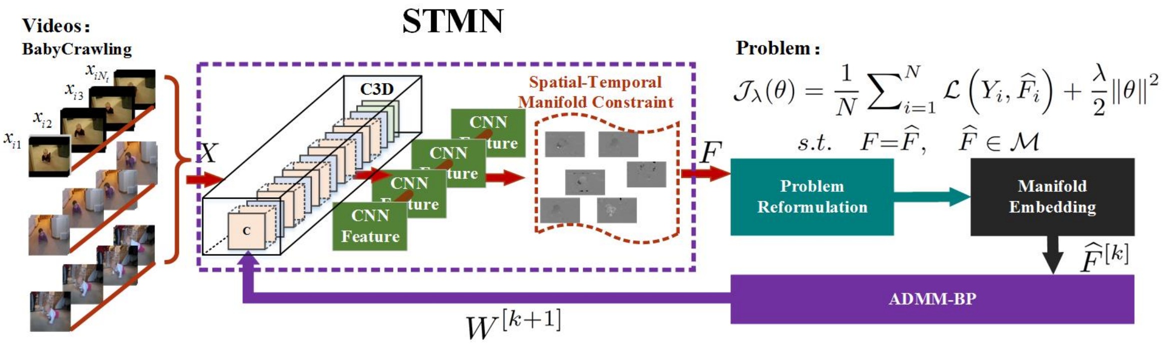

STMN
Videos:
BabyCrawling
x_{i1}
x_{i2}
x_{i3}
x_{iN_t}
X
C3D
c
CNN Feature
CNN Feature
CNN Feature
CNN Feature
Spatial-Temporal Manifold Constraint
F
Problem:
$\mathcal{J}_\lambda(\theta) = \frac{1}{N}\sum_{i=1}^{N} \mathcal{L}\left(Y_i, \widehat{F}_i\right) + \frac{\lambda}{2}\|\theta\|^2$
$s.t.\quad F=\widehat{F},\quad \widehat{F} \in \mathcal{M}$
Problem Reformulation
Manifold Embedding
$\widehat{F}^{[k]}$
ADMM-BP
$W^{[k+1]}$

Fig. 2. The STMN model is solved with an ADMM-BP algorithm, which leads to a chain of compact CNN features for action recognition. STMN is fine-tuned based the C3D model with manifold embedding in the back propagation procedure.

Similar to [2], Feichtenhofer *et al.* [15] showed that a two-stream fusion at an intermediate layer using RGB images and a stack of ten optical flow frames can improve the performance with less parameters. Extensions of two stream networks include Two-stream ConvNet(original) [2], Two-stream ConvPooling [44], TDD+FV [13], Two-stream Transformations [51], Two-stream ResNet [15], TSN (3 modalities) [14], KVMF [52], ST-ResNet [15], AdaScan [53], Three-stream sDTD [54], ST-VLMPF [17], SPN (BN-Inception) [55], and ActionVLAD [56]. Despite the good performance of multi-stream framework, it still remains unclear whether the deep learning based model can capture the subtle motion model and long-term motion dynamics for good performance without multi-stream fusion.

Unlike abovementioned approaches, we focus on 3D CNNs which have made significant advances in action recognition [20]. Based on the spatio-temporal 3D CNNs, some existing works are targeted at learning discriminative representations using only raw input videos, including with C3D [12], GRP [57], P3D ResNet [18], I3D [19], and 3D ResNeXt [20]. In this paper, a spatio-temporal manifold network which devises manifold structure in a deep neural network architecture for action recognition is proposed. The manifold structure is not explicitly exploited in existing deep learning approaches. In most deep learning approaches for action recognition, the learning objective involves only the inter-class discrimination, but ignores the intra-class variance and structure information. Our proposed approach inherits the advantages of the C3D method [12], while goes beyond it by introducing a new regularization term to exploit the manifold structure during the training process, in order to reduce intra-class variations and alleviate the over-fitting problem. Rather than simply combine the manifold and CNNs, we theoretically obtain the updating formula of our CNN model by preserving the structure of the data from the input space to the feature space.

Although using the manifold constraint among action videos, our work differs from the latest manifold work [22], [23], [26], [29] in the following aspects. First, our method is obtained from a theoretical investigation under the framework of ADMM, while [23] is empirical, [26] is based on semisupervised learning an optimal graph, [22] is based on incorporating the regularizer into RBM pretraining, and [29] is based on extrinsic least squares regression. Second, we regularize the spatio-temporal manifold embedding on a 3D CNN, while [22] applied the manifold on a restricted Boltzmann machine and a 2D CNN. Third, we are inspired from the fact that deep learning is so powerful that it can well discriminate the inter-class samples, and thus only intra-class manifold is considered to tackle the unstructured problem existed in the deep features (in Fig. 1). Differently, the method in [23] focused on learning a deep metric to 'pull' features to clusters considering intra-class and inter-class information based on the complicated manifold regularization terms. However, our study actually tend to 'pull' intra-class features and 'push' inter-class features away from each other by revealing the manifold structure information, which shows great advantages for action recognition in videos.

TABLE I
A BRIEF DESCRIPTION OF VARIABLES USED IN THE PAPER.

Variable	Description
X	video training set
$\mathcal{M}$	manifold constraint
F	C3D feature map
$\widehat{F}$	manifold embedding of F
$\widetilde{F}$	STMN feature map
W	convolution filters (exclude the last layer)
θ	weight for the last fully connected layer
Ω	the diagonal matrix of LLE weights

III. Spatio-Temporal Manifold Network

In this section, we present how the spatio-temporal manifold constraint can be introduced into CNN, i.e., C3D [12], for action recognition. Fig. 2 shows the framework of STMN, in which the intra-class manifold structure is embedded as a regularization term into the loss function. The manifold embedding leads to a new ADMM-BP learning algorithm to train the CNN model.

For simplicity of explanation, we briefly describe the variables in the problem definition in Tab. I. F is the CNN feature map, $\widehat{F}$ is the cloning of F, which formulates a manifold. θ is the weight for the last fully connected layer, while W represents convolution filters for other layers.

A. Model Formulation

Let $X = \{X_i\} \in \mathbb{R}, i \in [1, N]$ be a training set of videos, where N is the total number of videos and $X_i = \{x_{i1}, x_{i2}, ..., x_{iN_t}\}$ denotes the ith video with X_i divided into N_t clips (see Fig. 2). X is the input of C3D, and the output feature map is denoted as F. Given the convolution operator

$\odot$ and max pooling operator Ψ, the network performs convolutions in the spatio-temporal domain with a number of filter weights W and bias b. The function in the convolution layer is $f_W(X_i) = \Psi(W \odot X_i + b)$. In the last fully connected (FC) layer, the empirical loss function for the L-layers network is formulated as the average loss on the training set:

$$\mathcal{J}_\lambda(\theta) = \frac{1}{N}\sum\nolimits_{i=1}^{N} \mathcal{L}\left(Y_i, f_W(X_i)\right) + \frac{\lambda}{2}\|\theta\|^2, \tag{1}$$

where θ denotes the weight vector in the last FC layer, and all biases are omitted. In Eq. (1), the softmax loss term $\mathcal{L}\left(Y_i, f_W(X_i)\right)$ is

$$\mathcal{L}\left(Y_i, f_W(X_i)\right) = -\log \frac{e^{\theta_{Y_i}^T f_W(X_i)+b_{Y_i}}}{\sum_{j=1}^{m} e^{\theta_j^T f_W(X_i)+b_j}}, \tag{2}$$

where $f_W(X_i)$ denotes the deep feature for X_i, belonging to the Y_ith class. θ_j denotes the jth column of weights in the last FC layer, m is the number of classes. To simplify the notation, we denote the output feature map for video X_i as $F_i = \left\{F_{i1}^L, F_{i2}^L, ..., F_{iN_t}^L\right\}$, which is able to describe the nonlinear dependency of all features F_{ij}^L after L layers for video clips. As a result, the deep features are denoted as $F = \{F_i\}, i \in [1, N]$, and $F^{[k]}$ refers to the learned feature at the kth iteration (see Fig. 2).

The conventional objective function in Eq. (1) overlooks a property that the action video sequences usually formulate a specific manifold $\mathcal{M}$, which represents the nonlinear dependency of input videos. For example, in Fig. 1 the intra-class of video X with separated clips lies on a spatio-temporal manifold $\mathcal{M}$, which is supported by the evidence that video sequence with continuously moving and/or acting objects often lies on specific manifolds [30]. To take advantage of the property that the structure of the data can actually contribute to better solutions for lots of existing problems [23], we deploy a variable cloning technique $X = \widehat{X}$ with $\widehat{X} \in \mathcal{M}$ to explicitly add manifold constraint into the optimization objective Eq. (1). We then have a new problem:

$$\begin{aligned} \mathcal{J}_\lambda(\theta) &= \frac{1}{N}\sum\nolimits_{i=1}^{N} \mathcal{L}\left(Y_i, f_W(\widehat{X}_i)\right) + \frac{\lambda}{2}\|\theta\|^2 \\ s.t. \quad & X{=}\widehat{X}, \quad \widehat{X} \in \mathcal{M} \end{aligned}. \tag{P1}$$

The Problem (P1) is more reasonable since the intrinsic structure information is considered. However, it is unsolvable because θ is for the last FC layer of CNN and is not directly related to the input X.

In the deep learning approach with error propagation from the top layer, it is more favorable to impose the manifold constraint on the deep layer features. This is also inspired from the idea of manifold on the structure for preserving in different spaces, i.e. the high-dimensional and the low-dimensional spaces. Similarly, the manifold structure of X in the input space is assumed to be preserved in the feature F of CNN in order to reduce variation in the higher-dimensional feature space (Fig. 1). That is to say, an alternative manifold constraint is obtained as $F \in \mathcal{M}$, and evidently F is more related to CNN training. In the problem of action recognition, the intra-class variation at data level is interpreted as the data shift in the same category of videos that have temporal evolution and appearance variation, and the intra-class manifold at feature level can be recognized as the similar structural information in the same category of representation features. Ultimately, the action variation in the same category may come from the illumination, resolution, view point, action speed, and styles of people performing the same action. To use $F \in \mathcal{M}$ to solve the problem (P1), we perform variable replacement, i.e. $F = \widehat{F}$, alternatively formulate $\widehat{F}$ as a manifold, and achieve a new problem (P2),

$$\begin{aligned} \mathcal{J}_\lambda(\theta) &= \frac{1}{N}\sum\nolimits_{i=1}^{N} \mathcal{L}\left(Y_i, \widehat{F}_i\right) + \frac{\lambda}{2}\|\theta\|^2 \\ s.t. \quad & F{=}\widehat{F}, \quad \widehat{F} \in \mathcal{M} \end{aligned}, \tag{P2}$$

It is obvious that the objective in problem (P2) is learnable, because F is the convolution result based on the learned filter (W and θ) and θ is directly related to F. Indeed, the manifold constraint adds weighted feature maps of the same class to the original output, this regularization will make the data points of the same class are further cluttered and become more discriminative. The solution to the constraint $\widehat{F} \in \mathcal{M}$ is elaborated in the next section.

B. ADMM-BP Solution (P2)

Based on the augmented lagrangian multiplier (ALM) method, we have a new objective for the problem (P2) as

$$\mathcal{J}_{\lambda,\sigma}(\widehat{F}, F; \theta, R) = \mathcal{J}_\lambda(\theta) + R^T(\widehat{F} - F) + \frac{\sigma}{2}\|\widehat{F} - F\|^2, \tag{3}$$

where R^T denotes the Lagrange multiplier vector, σ is the corresponding regularization factor. Optimizing the above objective involves complex neural network training problem. Eq. (3) is solved based on ADMM and backward propagation algorithm, named ADMM-BP, which integrates CNN training with manifold embedding in an unified framework.

Specifically, we solve each variable in each sub-problem. ADMM-BP is described from the kth iteration, and $\widehat{F}^{[k]}$ is first solved based on $F^{[k]}$. Next $F^{[k]}$, $R^{[k+1]}$, $\theta^{[k+1]}$ and $W^{[k+1]}$ are solved step by step. Finally $F^{[k+1]}$ is obtained, which is then used to calculate $\widehat{F}^{[k+1]}$ similar to that in the kth iteration. We have

$$\begin{aligned} \widehat{F}^{[k]} &= \arg\min \mathcal{J}_{\lambda,\sigma}(\widehat{F}|F^{[k]}) \\ s.t. \quad & \widehat{F} \in \mathcal{M} \end{aligned}, \tag{4}$$

which is described in the next section. And then

$$R^{[k+1]} = R^{[k]} + \sigma^{[k]}(\widehat{F}^{[k]} - F^{[k]}). \tag{5}$$

For the FC layer, we use the gradient descend method,

$$\theta^{[k+1]} = \theta^{[k]} - \alpha\frac{\partial \mathcal{J}_{\lambda,\sigma}^{[k]}}{\partial \theta^{[k]}} = \theta^{[k]} - \alpha\frac{\partial \mathcal{J}_{\lambda}^{[k]}}{\partial \theta^{[k]}}, \tag{6}$$

and we update the parameters for convolutional layers W by stochastic gradient descent in the backward propagation as

$$W^{[k+1]} = W^{[k]} - \alpha\frac{\partial \mathcal{J}_{\lambda,\sigma}^{[k]}}{\partial \widehat{F}^{[k]}} \cdot \frac{\partial \widehat{F}^{[k]}}{\partial W^{[k]}}, \tag{7}$$

where α is the learning rate, k is the iterative number, and

$$\frac{\partial \mathcal{J}^{[k]}_{\lambda,\sigma}}{\partial \widehat{F}^{[k]}} = \frac{\partial \mathcal{J}^{[k]}_{\lambda}}{\partial \widehat{F}^{[k]}} + \sigma^{[k]}\left(\widehat{F}^{[k]} - F^{[k]}\right) + R^{[k]T}. \tag{8}$$

Now we have an updated CNN model to calculate the feature map $F^{[k+1]}$, which is then deployed to calculate $\widehat{F}^{[k+1]}$ via Eq. (4) (replacing k by $k+1$).

C. Manifold Embedding

In the ADMM-BP algorithm, only Eq. (4) is unsolved because of an unknown manifold constraint $\mathcal{M}$ isometrically embedded in $\mathbb{R}$. Based on Eq. (3), we can rewrite Eq. (4) by dropping the constant terms and the index of variables,

$$\begin{aligned} \widehat{F} &= \arg\min[R^T(\widehat{F} - F) + \frac{\sigma}{2}\|\widehat{F} - F\|^2] \\ &= \arg\min \|\widehat{F} - (F - \frac{R}{\sigma})\|^2 \qquad . \\ s.t. &\quad \widehat{F} \in \mathcal{M} \end{aligned} \tag{9}$$

In the kth iteration, we have $\widehat{F}^{[k]} = A_{\mathcal{M}}(F^{[k]} - \frac{R^{[k]}}{\sigma^{[k]}})$[2], where $A_{\mathcal{M}}$ is the projection matrix related to the manifold $\mathcal{M}$. This is the key part of the proposed algorithm where the constraint manifold $\mathcal{M}$ arises. Replacing $\mathcal{M}$ equals replacing the projection $A_{\mathcal{M}}$. This is the modularity which we alluded previously. To calculate $A_{\mathcal{M}}$, we exploit the locally linear embedding (LLE) method [58] in order to find a structure-preserving solution for our problem based on the embedding technique. By considering intrinsic manifold structure of the input data, the algorithm can stop on a manifold, $A_{\mathcal{M}}$, in the kth iteration as

$$A_{\mathcal{M}} = F_{[1:H]}\Omega^{[k]}, \tag{10}$$

where $\Omega^{[k]}$ is a diagonal matrix defined as $\Omega^{[k]} = \text{diag}(\omega_1^{[k]}, ..., \omega_N^{[k]})$. $F_{[i1:iH]}$ are the H neighborhoods of the sample and $\omega_N^{[k]}$ are the corresponding weight vector calculated in LLE.

D. Algorithmic Implementation

Based on the analysis above, we present a summary of the ADMM-BP algorithm for STMN in Alg. 1, where the key step defined by Eq. (11) is respectively solved in Sec. III-B and Sec. III-C. Note that Eq. (10) is straightforward to implement, and Eq. (11) is an optimization problem solved via ADMM and then in the similar procedure to back-propagate the gradient. Although the convergence of the ADMM optimization problem with multiple variables remains an open problem, the learning procedure experimentally never diverge. The reason is that the new variables related to the manifold constraint are solved following the similar pipeline of back propagation, which essentially leads to no extra computational cost in the stochastic gradient descent update.

We also give a brief analysis on the complexity of manifold embedding in training STMN. To identify the H nearest neighbors, each data point and its neighbors are assumed to lie on a locally linear patch of the manifold, and the $\Omega^{[k]}$ in

Algorithm 1 ADMM-BP for the problem (P2)

```
Set $t = 0$ and $\epsilon_{\text{best}} = +\infty$
Initialize $\alpha$, $\lambda$, $\sigma^{[0]}$, $R^{[0]}$, and $0 < \eta <= 1$
Initialize $\theta^{[0]}$, $W^{[0]}$, $\widehat{F}^{[0]}$, and $\Omega^{[0]}$
repeat
        $(\widehat{F}^{[k+1]}, R^{[k+1]}, \theta^{[k+1]}, W^{[k+1]}) =$
            $\arg\min \; \mathcal{J}_{\lambda,\sigma^{[k]}}(\widehat{F}, F; \theta^{[k]}, R^{[k]} | \Omega^{[k]})$
            $s.t. \; \widehat{F} \in \mathcal{M},$   (11)
    Update $\Omega^{[k]}$ and $\widehat{F}^{[k]}$ by LLE
    $\epsilon = \|\widehat{F}^{[k+1]} - \widehat{F}^{[k]}\|^2$
    if $\epsilon < \eta\, \epsilon_{\text{best}}$
        $R^{[k+1]} = R^{[k]} + \sigma^{[k]}(\widehat{F}^{[k]} - F^{[k]})$
        $\sigma^{[k+1]} = \sigma^{[k]}$
        $\epsilon_{\text{best}} = \epsilon$
    else
        $R^{[k+1]} = R^{[k]}$
        $\sigma^{[k+1]} = 2 \cdot \sigma^{[k]}$
    endif
    $k \leftarrow k + 1$
until maximum iteration step or $\epsilon \leq 0.001$
```

Eq. (10) is calculated by LLE. Note that LLE has the fast optimization taking advantage of eigenvalue decomposition by sparse matrix algorithms [58]. Next, the parameters θ and W are updated by Eq. (6) and Eq. (7) in back propagation training, which changes relatively in each iteration as the loss $\mathcal{J}_{\lambda,\sigma}$ with the regularization of manifold embedding. That is to say, it is not any deep learning network exclusive, but can be widely applied to many fancy CNN models. In this paper, we select the C3D model as the baseline due to its applicability for the task of action recognition.

Based on the learned STMN model, we obtain a chain of geometrically meaningful CNN features denoted as

$$\begin{aligned} \widetilde{F} &= \left\{\widetilde{F}_i\right\} \in \mathbb{R}, i \in [1, N] \\ \widetilde{F}_i &= \left\{\widetilde{F}_{i1}, \widetilde{F}_{i2}, ..., \widetilde{F}_{iN_t}\right\} \end{aligned}, \tag{12}$$

where N is the number of videos, and $\widetilde{F}_i$ is the STMN feature for the video X_i with N_t clips for action classification.

IV. Experimental Results

In this section ,we first present popular benchmark datasets, then describe the details of our method, and finally present and experimental results and analyze them compared with the state-of-the-art results.

A. Datasets and Implementation

Three popular small-scale benchmark datasets UCF101 [59], HMDB51 [60], and Hollywood2 [61] are used to validate our approach for action recognition. Example frames from the datasets are shown in Fig. 3.

UCF101 dataset contains 13320 videos from 101 action classes with each class having at least 100 videos, which are divided into 25 groups for each class. We follow the evaluation scheme of the THUMOS13 Challenge [62] to use the three training/testing splits. **HMDB51 dataset** consists of $6,766$

[2] We have $\widehat{F} = (F - \frac{R}{\sigma})$ without manifold constraint.

Fig. 3. Example frames from Hollywood2, HMDB51 and UCF101 datasets. (Numbers indicate the improvements of STMN over C3D for corresponding actions.)

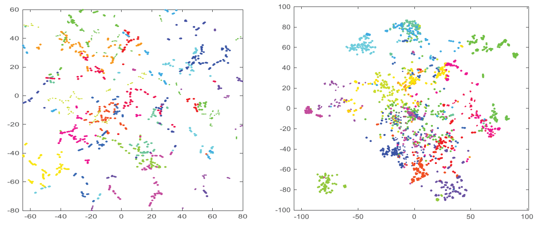
realistic videos from 51 action categories with each category containing at least 100 videos. We follow the evaluation scheme in [60] to report the average accuracy over three different training/testing splits. **Hollywood2 dataset** provides 12 classes of human actions over 2,517 videos from 69 movies with each category containing at least 210 videos [61]. The clean training subset and testing subset are with action labels manually verified to be correct, respectively. In our experiments, we use 823 training sequences and 884 testing sequences from different movies.

TABLE II
EXPLORATION OF DIFFERENT NEIGHBORHOODS AS MANIFOLD CONSTRAINTS FOR STMN+SVM ON DIFFERENT DATASETS (ACCURACY %).

Neighborhoods #	UCF101	HMDB51
$H = 5$	75.0	68.2
$H = 10$	79.8	68.6
$H = 15$	86.1	68.7
$H = 20$	92.5	69.7

Experimental settings. We employed the parallel computing strategy to implement our approach on Caffe [67] with four Titan X Pascal GPUs and Xeon(R) E5-2620 V2 CPU. We use C3D [12] as the baseline, which is the 3D version of CNN designed to extract spatial and temporal features, to obtain a chain of CNN features for video recognition. We initially use the same pretrained model as C3D from sport1M [41] to train the STMN model on UCF101 dataset, then further deploy and finetune it on the other two datasets. Similar to C3D, each video is divided into 16-frame clips with 8-frame overlapped between two consecutive clips as the input of the STMN. The frame resolution is set to 128×171, and input sizes are $3 \times 16 \times 128 \times 171$ (channels×frames×height×width). The network uses 5 convolution layers, 5 pooling layer, 2 FC layers and a softmax loss layer to predict action labels. The filter numbers from the first to fifth convolutional layer respectively are 64, 128, 256, 256 and 256. The sizes of convolution filter kernels and the pooling layers respectively are $3 \times 3 \times 3$ and $2 \times 2 \times 2$. The output feature size of each FC layer is 4096. During the STMN training, the batch size, initial learning rate, weight decay parameters, and maximum iteration are set to be 24, 0.001, 0.9999, 60000, respectively. Following to the three-net deployment for higher performance on C3D [12], we also extract STMN features from all clips and finally concatenate them for video classification.

Model settings and baseline. To evaluate the effectiveness of our STMN ($\widetilde{F}$), we follow the same protocols of training/testing splits as used in TDD [13], TSN [14] and C3D [12] on UCF101, HMDB51, and Hollywood2, in order to have a fair comparison with other methods. Since many works reported results by performing the fusion with hand-crafted iDT features [37], we also conduct three different settings for classification models for a fair comparison: (A) STMN – it uses STMN features directly with a softmax layer for classification, (B) STMN+SVM – it combines the STMN features with a multi-class linear SVM for classification, (C) STMN+iDT+SVM – it concatenates the STMN features and iDT features with multi-class linear SVM for final classification, which follows the same protocol as used in C3D [12]. Table III compares the mean average accuracy over splits of our approach with the state-of-the-art methods. In Table III, the methods in the first block are based on hand-crafted features, the second block are based on recent deep learning features, and the third block are based on the combination of various deep learning features and the hand-crafted iDT features. We also recognize each method whether or not using multi-stream architectures in the fifth column, and also list each method whether or not using deep learning in the sixth column. Noted that the methods without multi-stream architectures only using raw video input but others using video and motion input. As a confirmation in the table, the proposed STMN model can learn compact feature representation, and the performance on datasets will be compared in Section IV-C.



(a) (b)

Fig. 4. Feature visualization of twenty difficult classes on the UCF101 dataset. (a) are the C3D features, and (b) are the STMN features. The STMN feature is more discriminative than the C3D feature.

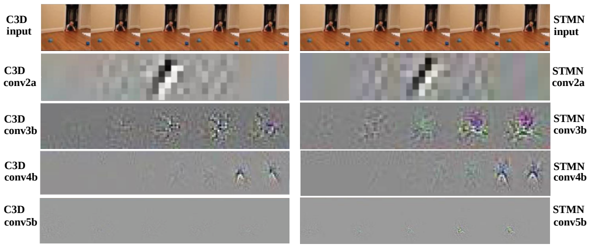


Fig. 5. Visualization of feature maps in conv2a, conv3b, conv4b and conv5b layer. Second row: the learned filters detect moving edges, the third row: the learned feature maps detect moving textures, the fourth row: the learned feature maps detect moving body parts, the last row: the learned feature maps detect moving motions of crawling. Left group is C3D and right is STMN. Best viewed in color.

B. Experimental Analysis

Parameter analysis. To investigate the performance of SMTN using different numbers of neighborhoods in LLE, we first study the average recognition accuracies of STMN on the same trail of UCF101 and HMDB51 datasets in Tab. II. Due to limited number of training videos in each class, we

TABLE III
COMPARISON TO THE STATE-OF-THE-ART RESULTS ON UCF101 DATASET (3 SPLITS), HMDB51 (3 SPLITS), AND HOLLYWOOD2 DATASET.

Method	UCF101 (%)	HMDB51 (%)	Hollywood2 (%)	Multi-stream	Deep Learning	Year
STIP+BoVW [60]	43.9	23.0	32.6	N	N	2011
DT+BoVW [63]	79.9	46.6	63.0	N	N	2013
DT+MVSV [64]	83.5	55.9	–	N	N	2014
iDT+FV [37]	84.7	57.2	64.3	N	N	2013
DT+BOW [65]	–	60.9	63.0	N	N	2016
3D ConvNet [41]	65.4	–	–	N	Y	2014
ConvNet+LSTM (LRCN) [47]	82.9	–	–	N	Y	2015
ConvNet+LSTM (unsup) [49]	84.3	–	–	N	Y	2015
Two-stream ConvNet(original) [2]	88.0	59.4	–	Y	Y	2014
Two-stream ConvPooling [44]	88.2	–	–	Y	Y	2015
TDD+FV [13]	90.3	63.2	–	Y	Y	2015
Two-stream Transformations [51]	92.4	63.4	–	Y	Y	2016
Two-stream ResNet [15]	93.4	–	–	Y	Y	2016
TSN (3 modalities) [14]	94.2	69.4	66.8	Y	Y	2016
KVMF [52]	93.1	63.3	–	Y	Y	2016
ST-ResNet [15]	93.4	66.4	–	Y	Y	2016
AdaScan [53]	89.4	54.9	–	Y	Y	2017
GRP [57]	91.9	65.4	–	N	Y	2017
Three-stream sDTD [54]	92.2	65.2	–	Y	Y	2017
L^{2}STM [16]	93.6	66.2	–	Y	Y	2017
ST-VLMPF [17]	93.6	69.5	–	Y	Y	2017
ActionVLAD [56]	92.7	66.9	–	Y	Y	2017
P3D ResNet [18]	88.6	–	–	N	Y	2017
3D ResNeXt [20]	90.7	63.8	–	N	Y	2017
SPN (BN-Inception) [55]	94.6	68.9	–	Y	Y	2017
MiCT [6]	89.1	–	–	N	Y	2018
CO2FI+ASYN [21]	94.3	69.0	–	Y	Y	2018
R(2+1)D-TwoStream [7]	95.0	**72.7**	–	Y	Y	2018
DML without additional training data [22]	94.7	65.2	–	Y	Y	2018
DML [22]	**96.7**	72.5	–	Y	Y	2018
C3D [12] (baseline)	79.4	49.3	55.7	N	Y	2015
STMN (ours)	83.7	56.2	58.5	N	Y	–
C3D+SVM [12] (baseline)	85.2	50.3	60.6	N	Y	2015
STMN+SVM (ours)	**92.5**	**69.7**	63.2	N	Y	–
TDD+iDT [13]	91.5	65.9	–	Y	Y	2015
Dynamic Image Networks+iDT [66]	89.1	65.2	–	Y	Y	2016
AdaScan+iDT [53]	91.3	61.0	–	Y	Y	2017
GRP+iDT [57]	92.3	67.0	–	N	Y	2017
ActionVLAD+iDT [56]	93.6	69.8	–	Y	Y	2017
P3D ResNet+iDT [18]	93.7	–	–	N	Y	2017
ST-ResNet+iDT [15]	94.6	70.3	–	Y	Y	2016
CO2FI+ASYN+iDT [21]	95.2	72.6	–	Y	Y	2018
C3D+iDT [12] (baseline)	90.4	63.5	67.7	N	Y	2015
STMN+iDT (ours)	**92.8**	**70.2**	**70.1**	N	Y	–

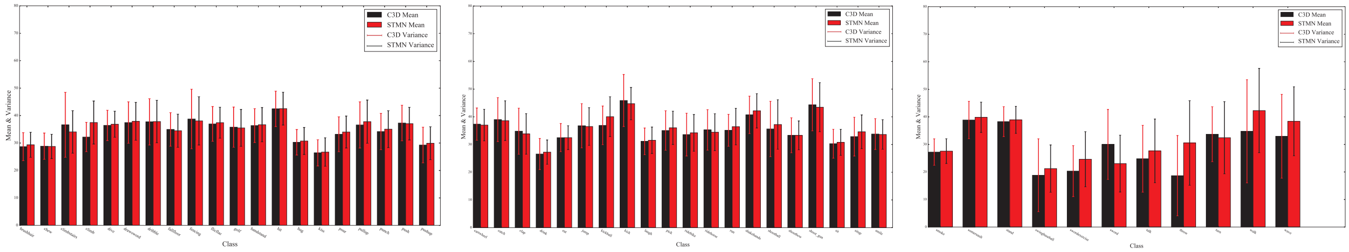


Fig. 6. Comparison of intra-class means and variances of C3D (black) and STMN (red) features. 51 classes are from HMDB51 dataset.

learned the STMN on the UCF101 using $H = 5, 10, 15, 20$ neighbor samples and extracted features for the SVM classifier. By the comparison of the second column and the third column, STMN+SVM achieves the best accuracies of 92.5% and 69.7% on UCF101 dataset and HMDB51 dataset, respectively, when $H = 20$. Note that the value of H has to be smaller than the batch size. In our experiment, we can only evaluate the performance of STMN with three different classification models by setting H up to 20 due to memory limitation of GPUs.

Feature visualization. To better understand our network, we analyze and visualize the learned features based on C3D and STMN on the UCF101 dataset. Fig. 4 shows the embedding feature visualizations on UCF101 dataset by t-SNE [68]. The C3D features of twenty difficult classes on UCF101 are visualized in Fig. 4(a), while the STMN features are illustrated in Fig. 4(b), respectively. It is shown that the STMN features in Fig. 4(b) can be better discriminated for action classification than the C3D features in Fig. 4(a). As another verification, the quantitative evaluation is performed based on intra-class mean and variance in the next section.

Model effect. To compare the baseline and STMN learned

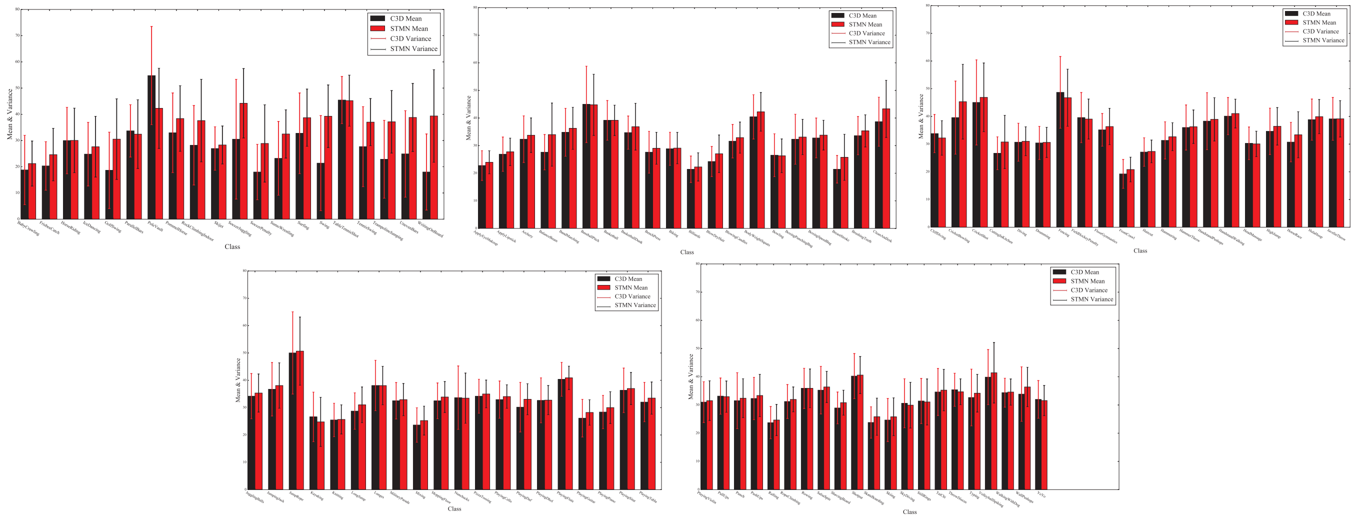


Fig. 7. Comparison of intra-class means and variances of C3D (black) and STMN (red) features. 101 classes are from UCF101 dataset.

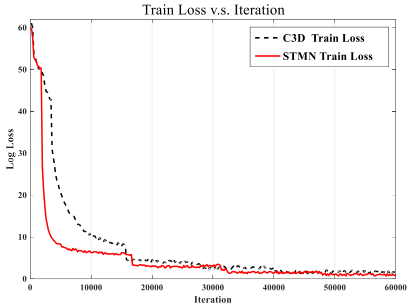
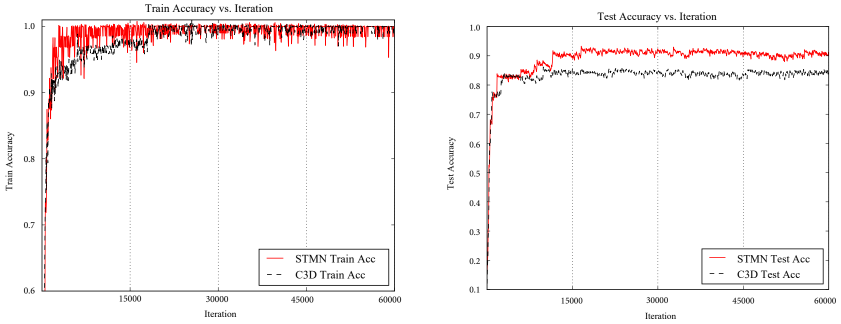


Fig. 8. The analysis of training loss curves of STMN and C3D on the UCF101 dataset.



(a) (b)

Fig. 9. The training and testing accuracy curves of STMN and C3D on the UCF101 dataset.

feature maps in convolution layer of deep networks, we select a clip of frames in *BabyCrawling* example from UCF101 and visualize the deconvolutions of feature maps from the layers of conv2a, conv3b, conv4b and conv5b in Fig. 5, where the top activations are projected into image space [42]. As shown in the visualizations, STMN learns low level motion patterns such as moving edges, edge orientation changes and color changes at early layer conv2a, higher level motion patterns such as textures, body parts and trajectories at convolution layer conv3b and conv4b, and complicated motion patterns of moving objects at the conv5b layer. It also shows that the feature maps at deeper convolutional layer are stronger to capture the motions.

Intra-class variation. Considering that the manifold regularization aims to make the data point of the same class are further cluttered in the feature space, we evaluate the intra-class variation at the feature level in terms of the mean and variance values of features' similarity. The variance values are smaller, and the learnt features are more cluttered. We calculate the pairwise Euclidean distances of STMN features and C3D features, and then compute the intra-class mean and variance values. The quantitative intra-class means and variances of STMN features and C3D features on HMDB51 and UCF101 are shown in Fig. 6 and Fig. 7, respectively. We observe that the variance value of STMN features for most of the classes are less than those of C3D. Fig. 4, Fig. 6 and Fig. 7 demonstrate that STMN can exploit the manifold structure to better eliminate the randomness of samples in the feature space. Especially as shown in Fig. 6 and Fig. 7, the quantitative intra-class means and variances [3] of STMN features are much smaller than those of C3D, e.g. the total mean on the UCF101 dataset has decreased from 14.02 to 11.15. We can also observe that 21.22 (STMN mean) versus 18.78 (C3D mean) and 8.58 (STMN variance) versus 13.23 (C3D variance) for the specific action *BabyCrawling*.

Learning convergence. We plot the training loss curves in Fig. 8, the training accuracy curves in Fig. 9(a), and testing accuracy curves in Fig. 9(b) of STMN and C3D networks on a same trail of UCF101. It is clear that the training loss of STMN (red line) converges much faster than C3D (black line) from the 1000th iteration to the 15000th iteration. We also can observe that STMN curve stops increasing much early than C3D curve for the training accuracy, and it obtains higher testing accuracy than C3D. These curves show that STMN obviously converge faster and gain better performance than C3D, which proves that the proposed manifold regularization can alleviate the over-fitting problem by taking the instrinsic input data structure information into account in the same framework.

C. Results and Comparisons

To provide comprehensive evaluation, we follow the same evaluation scheme to compare our STMN with several rep-

[3] The statistics are computed using pairwise Euclidean distance.

resentative human action recognition methods. Tab. III shows the performance comparison of three settings of classification models for the STMN architecture, including with (A) STMN, (B) STMN+SVM, and (C) STMN+iDT, with other state-of-the-art approaches. Note that all our STMN-based approaches under different classification models are able to achieve better performance than the baseline C3D-based approaches (C3D, C3D+SVM, C3D+iDT), demonstrating the superiority of our spatio-temporal manifold regularization.

Results of three small-scale datasets. We analyze the results of STMN-based approaches from the second column to the fourth column in Tab. III. For the UCF101 dataset, STMN performs better than the hand-crafted STIP+BoVW [60], DT+BoVW [63], DT+MVSV [64] methods in the first block of table and the deep learning based 3D ConvNet [41] in the second block. STMN+SVM achieves better performance than all hand-crafted methods and most of deep learning based methods, such as 3D ConvNet [41], ConvNet+LSTM (LRCN) [47], ConvNet+LSTM (unsup) [49], Two-stream ConvNet(original) [2], Two-stream ConvPooling [44], TDD+FV [13], Two-stream Transformations [51], AdaScan [53], AdaScan [53], Three-stream sDTD [54], and C3D+SVM [12]. STMN+iDT also gains good result comparing to other methods based on the combination of various deep learning features and the hand-crafted iDT features in the third block. The improved results of STMN+SVM and STMN+iDT suggest that our STMN can exhibit better feature modeling ability than other models by the manifold regularization. For the HMDB51 dataset and Hollywood2, STMN, STMN+SVM, and STMN+iDT also lead to comparable results than the state-of-the-art methods. Limited by the problem of hollywood movie data, such as multi-objects, object parts and various background *etc.*, the results of existing methods have not been as good at Hollywood2 as UCF101 and HMDB51. Noted that STMN+iDT performs better than MiCT [6] which mixes the 3D CNNs and 2D CNNs in the convolutional tube for intergrating feature representation.

To demonstrate the enhancement of STMN, We plot the recognition confusion matrices for the best 30 classified classes on HMDB51 and UCF101 dataset in Fig. 10, which show most of the classes achieve high recognition accuracies. We take five classes including *Run, drawsword, pullup, BabyCrawling, and GolfSwing* as examples for more detailed analysis as shown in Fig. 3. We can see that the recognition accuracies of STMN+SVM for these five actions are 88.4%, 100%, 98%, 99% and 100%, and the improvements over baseline are 9.6%, 17.2%, 10.0%, 10.4% and 15.3%, respectively. It would be interesting to look into the comparison on the manifold strctures of the input data, C3D features, and STMN features. As illustrated in Fig. 11, STMN has a similar structure as that of the original input data, whilst the manifold structure of C3D is obviously different. Together with quantitative evaluation in Fig. 6 and Fig. 7, we believe that manifold constraint can decrease the intra-class variance in these classes, which is important for action recognition. As a further confirmation in Tab. III, the proposed STMN model can learn compact feature representation, and therefore it achieves comparable results with the spatio-temporal architectures and multi-stream networks.

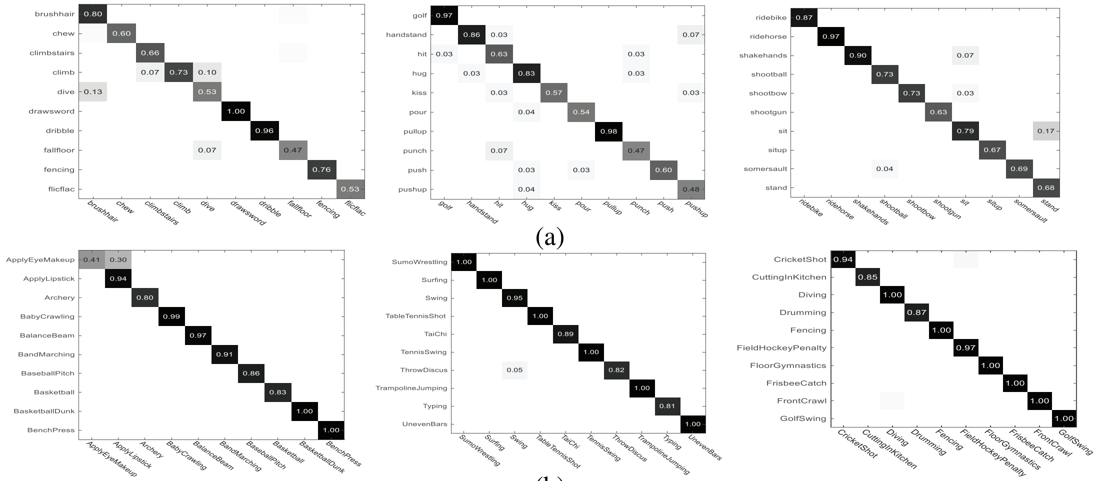
TABLE IV
COMPARISON OF DIFFERENT BACKBONE ARCHITECTURES ON UCF101 DATA SET.

Architecture	Layer	Dim	UCF101
STMN+SVM (ours)	10	3D	92.5
STMN ResNet+SVM (ours)	152	3D	93.6
STMN+iDT (ours)	10	3D	**92.8**
STMN ResNet+iDT (ours)	152	3D	**94.5**

Analysis of backbone architectures. It is worth mentioning that our results of STMN+SVM and STMN+iDT achieve the comparable performance on HMDB51 and they are lower on UCF101 than some recent methods, such as ST-ResNet+iDT [15], L^{2}STM [16], ST-VLMPF [17], R(2+1)D-TwoStream [7], DML [22], and CO2FI+ASYN+iDT [21], which use sophisticated deep models (BN-Inception+RBM, ResNet), and multi-stream modalities of fusion (RGB, Optical Flow, Warped Flow and audio), while we only use the RGB and flow features (*i.e.*, original video frames, iDT) in our STMN with the simple 10-layer C3D backbone architecture. Note that our manifold regularization scheme is very general and could be used in any architecture where input data constraints are natural, and the similar idea of multi-stream can be used to improve STMN in our future work.

Inspired by the recent successful application of ResNet backbone for action recognition, e.g. ST-ResNet+iDT [15], R(2+1)D-TwoStream [7], P3D ResNet [18], and 3D ResNeXt [20], we also use the ResNet backbone in 3D CNN to learn the spatio-temporal representation. Following the mixing residual blocks of P3D ResNet [18] and name this special STMN as STMN ResNet, we implement it in Tensorflow and initially use the same pretrained model from Sport-1M as P3D ResNet, embed the manifold regularization in the output feature space of convolutional architecture, and extract a generic 2048 dimensional feature for final classification. In Tab. IV, we compare the original 10-layer C3D backbone and the ResNet 3D backbone on UCF101 dataset. Benefied from the spatio and temporal residual connections throughout the architecture, STMN ResNet+SVM outperforms C3D+SVM when using only RGB features, and STMN ResNet+iDT achieves better recognition accuracy than C3D+iDT when fusing RGB and flow features. The results indicate that the deeper backbone architecture makes our method boost the performance and it leads to consistent improvement combing with flow features. In addition, by performing the ResNet backbone on the 3D spatio-temporal convolutional network, the accuracy of our STMN ResNet+iDT gets close to the state-of-the-art ST-ResNet+iDT [15], SPN (BN-Inception) [55], DML [22], and CO2FI+ASYN+iDT [21]. This makes us to believe that using 3D CNNs together with complex deeper architectures, more huge pretraining dataset, and multiple stream inputs will further develop the spatio-temporal video representation in the future work.

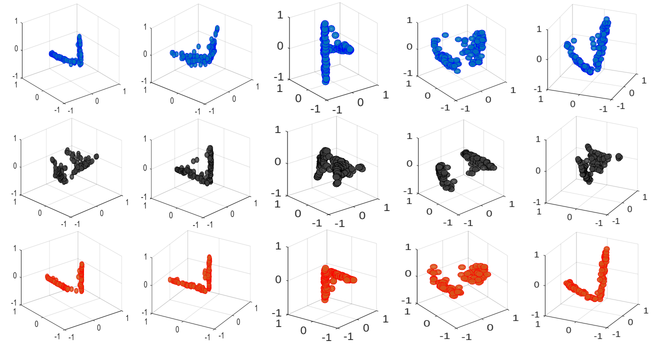
Results of large-scale ActivityNet-200 dataset. To further evaluate our method, we also conduct the experiment on ActivityNet-200 dataset. The ActivityNet-200 dataset (v1.3) [70] is one of the most popular large-scale video benchmarks for human activity classification. It contains 19,994 videos from 200 activity categories, which are divided into 10,024,



(a)

(b)

Fig. 10. Recognition confusion matrices of top 30 classes with most improvements than baseline on (a) HMDB51 and (b) UCF101 datasets.



(a) (b) (c) (d) (e)

Fig. 11. Manifold structure visualization of input data (blue), C3D features (black) and STMN (red) for the examples in Fig. 3, including (a) *Run*, (b) *drawsword*, (c) *pullup*, (d)*BabyCrawling*, and (e) *GolfSwing*.

TABLE V
PERFORMANCE COMPARISONS IN TERMS OF TOP-1 CLASSIFICATION ACCURACY ON ACTIVITYNET-200 DATASET.

Method	top-1
C3D+SVM [12]	65.8
ResNet+SVM [69]	71.4
P3D ResNet+SVM [18]	75.1
STMN ResNet+SVM (ours)	**76.3**

4,926 and 5,044 videos for training, validation and test set, respectively. This dataset is very challenging since it suffers from complex temporal evolution, untrimmed non-action frames, appearance variation and background clutters. Following ResNet+SVM and P3D ResNet+SVM [18], we implement the STMN ResNet+SVM using ResNet-152 as the backbone architecture and report the performance in terms of top-1 classification accuracy on the validation set. As shown in Tab. V, STMN ResNet+SVM outperforms C3D+SVM, ResNet+SVM and P3D ResNet+SVM by 10.5%, 4.9%, and 1.2% in terms of top-1 accuracy, respectively. The performance improvements validate that our method is indeed effective and gets better spatio-temporal representation than others on the large-scale dataset due to the use of the manifold regularization on 3D CNN.

V. CONCLUSIONS

This paper focuses on a new insight into deep feature extraction for action recognition using raw videos from the perspective of spatio-temporal manifold constraint. By transferring the structure of the data to a new constraint of the variable, we have proposed a new spatio-temporal convolutional manifold network (STMN) to solve the action classification problem. STMN is solved based on the proposed ADMM-BP algorithm. Experimental results on four benchmark datasets have demonstrated that STMN provides an effective way to reduce intra-class variation while preserve the inter-class discrimination of high-dimensional spatio-temporal deep features, which effectively alleviates the over-fitting problem of classifying action sequences. Based on the experimental evaluation, STMN achieves comparable or better performance as compared with the state-of-the-art approaches on four benchmark datasets. Our manifold regularization scheme is general and could be adapted to most existing architectures where input data constraints are natural. The idea of multi-stream will be also applied to improve the effectiveness of STMN in future work.

REFERENCES

[1] L. Wang, Y. Qiao, and X. Tang, "Motionlets: Mid-level 3D parts for human motion recognition," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, vol. 9, no. 4, 2013, pp. 2674–2681. 1

[2] K. Simonyan and A. Zisserman, "Two-stream convolutional networks for action recognition in videos," *Advances in Neural Information Processing Systems*, vol. 1, no. 4, pp. 568–576, 2014. 1, 2, 3, 7, 9

[3] L. Wang, Y. Qiao, and X. Tang, "Latent hierarchical model of temporal structure for complex activity classification," *IEEE Transactions on Image Processing*, vol. 23, no. 2, pp. 810–822, 2014. 1

[4] B. Zhang, Y. Yang, C. Chen, L. Yang, J. Han, and L. Shao, "Action recognition using 3d histograms of texture and a multi-class boosting classifier," *IEEE Transactions on Image Processing*, vol. 26, no. 10, pp. 4648–4660, 2017. 1

[5] C. Feichtenhofer, A. Pinz, R. P. Wildes, and A. Zisserman, "What have we learned from deep representations for action recognition," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2018. 1

[6] Y. Zhou, X. Sun, Z.-J. Zha, and W. Zeng, "Mict: Mixed 3d/2d convolutional tube for human action recognition," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2018, pp. 449–458. 1, 7, 9

[7] D. Tran, H. Wang, L. Torresani, J. Ray, Y. LeCun, and M. Paluri, "A closer look at spatiotemporal convolutions for action recognition," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2018. 1, 7, 9

[8] B. Mahasseni and S. Todorovic, “Regularizing long short term memory with 3d human-skeleton sequences for action recognition,” in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2016, pp. 3054–3062. 1, 2

[9] F. Wan, P. Wei, Z. Han, J. Jiao, and Q. Ye, “Min-entropy latent model for weakly supervised object detection,” *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2019. 1

[10] B. Zhang, Z. Li, A. Perina, A. D. Bue, and V. Murino, “Adaptive local movement modeling for robust object tracking,” *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 27, no. 7, pp. 1515–1526, 2017. 1

[11] S. Ji, W. Xu, M. Yang, and K. Yu, “3D convolutional neural networks for human action recognition,” *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 35, no. 1, pp. 221–231, 2013. 1, 2

[12] D. Tran, L. Bourdev, R. Fergus, L. Torresani, and M. Paluri, “Learning spatiotemporal features with 3d convolutional networks,” in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2015, pp. 4489–4497. 1, 2, 3, 6, 7, 9, 10

[13] L. Wang, Y. Qiao, and X. Tang, “Action recognition with trajectory-pooled deep-convolutional descriptors,” in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2015, pp. 4305–4314. 1, 3, 6, 7, 9

[14] L. Wang, Y. Xiong, Z. Wang, Y. Qiao, D. Lin, X. Tang, and L. V. Gool, “Temporal segment networks: towards good practices for deep action recognition,” in *Proceedings of European Conference on Computer Vision*, 2016, pp. 20–26. 1, 3, 6, 7

[15] C. Feichtenhofer, A. Pinz, and R. P. Wildes, “Spatiotemporal residual networks for video action recognition,” in *Neural Information Processing Systems*, 2016, pp. 3468–3476. 1, 3, 7, 9

[16] L. Sun, K. Jia, K. Chen, D. Y. Yeung, B. E. Shi, and S. Savarese, “Lattice long short-term memory for human action recognition,” in *Proceedings of the IEEE International Conference on Computer Vision*, 2017, pp. 2166–2175. 1, 2, 7, 9

[17] I. C. Duta, B. Ionescu, K. Aizawa, and N. Sebe, “Spatio-temporal vector of locally max pooled features for action recognition in videos,” in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2017, pp. 3205–3214. 1, 3, 7, 9

[18] Z. Qiu, T. Yao, and T. Mei, “Learning spatio-temporal representation with pseudo-3d residual networks,” in *Proceedings of the IEEE International Conference on Computer Vision*. IEEE, 2017, pp. 5534–5542. 1, 3, 7, 9, 10

[19] J. Carreira and A. Zisserman, “Quo vadis, action recognition? A new model and the kinetics dataset,” in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. IEEE, 2017, pp. 4724–4733. 1, 3

[20] K. Hara, H. Kataoka, and Y. Satoh, “Can spatiotemporal 3D CNNs retrace the history of 2D cnns and imagenet,” in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2018, pp. 18–22. 1, 3, 7, 9

[21] W. Lin, C. Zhang, K. Lu, B. Sheng, J. Wu, B. Ni, X. Liu, and H. Xiong, “Action recognition with coarse-to-fine deep feature integration and asynchronous fusion,” in *Thirty-Second AAAI Conference on Artificial Intelligence*, 2018. 1, 7, 9

[22] X. Chen, J. Weng, W. Lu, J. Xu, and J. Weng, “Deep manifold learning combined with convolutional neural networks for action recognition,” *IEEE Transactions on Neural Networks and Learning Systems*, vol. 29, no. 9, pp. 3938–3952, 2018. 1, 2, 3, 7, 9

[23] J. Lu, G. Wang, W. Deng, P. Moulin, and J. Zhou, “Multi-manifold deep metric learning for image set classification,” in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2015, pp. 1137–1145. 1, 2, 3, 4

[24] X. Ma, D. Tao, and W. Liu, “Effective human action recognition by combining manifold regularization and pairwise constraints,” *Multimedia Tools and Applications*, pp. 1–17, 2017. 2

[25] Q. V. Le, W. Y. Zou, S. Yeung, and A. Y. Ng, “Learning hierarchical invariant spatio-temporal features for action recognition with independent subspace analysis,” in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2011, pp. 3361–3368. 2

[26] W. Wang, Y. Yan, F. Nie, S. Yan, and N. Sebe, “Flexible manifold learning with optimal graph for image and video representation,” *IEEE Transactions on Image Processing*, vol. 27, no. 6, pp. 2664–2675, 2018. 2, 3

[27] T. Zhang, W. Zheng, Z. Cui, and C. Li, “Deep manifold-to-manifold transforming network,” in *2018 25th IEEE International Conference on Image Processing*. IEEE, 2018, pp. 4098–4102. 2

[28] Z. Huang, J. Wu, and L. Van Gool, “Building deep networks on Grassmann manifolds,” in *Thirty-Second AAAI Conference on Artificial Intelligence*, 2018. 2

[29] Y. Wang, L. Wang, D. Kong, and B. Yin, “Extrinsic least squares regression with closed-form solution on product Grassmann manifold for video-based recognition,” *Mathematical Problems in Engineering*, vol. 2018, 2018. 2, 3

[30] B. Zhang, A. Perina, V. Murino, and A. D. Bue, “Sparse representation classification with manifold constraints transfer,” in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2015, pp. 4557–4565. 2, 4

[31] Y. Wen, K. Zhang, Z. Li, and Y. Qiao, “A discriminative feature learning approach for deep face recognition,” in *Proceedings of European Conference on Computer Vision*, 2016, pp. 11–26. 2

[32] P. Dollár, V. Rabaud, G. W. Cottrell, and S. Belongie, “Behavior recognition via sparse spatio-temporal features,” *VS-PETS*, pp. 65–72, 2005. 2

[33] G. Willems, T. Tuytelaars, and L. Van Gool, “An efficient dense and scale-invariant spatio-temporal interest point detector,” in *Proceedings of European Conference on Computer Vision*, 2008, pp. 650–663. 2

[34] I. Laptev, M. Marszalek, C. Schmid, and B. Rozenfeld, “Learning realistic human actions from movies,” in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2008, pp. 1–8. 2

[35] N. Dalal, B. Triggs, and C. Schmid, “Human detection using oriented histograms of flow and appearance,” in *Proceedings of European Conference on Computer Vision*, vol. 3952, 2006, pp. 428–441. 2

[36] Y.-G. Jiang, Q. Dai, W. Liu, X. Xue, and C.-W. Ngo, “Human action recognition in unconstrained videos by explicit motion modeling,” *IEEE Transactions on Image Processing*, vol. 24, no. 11, pp. 3781–3795, 2015. 2

[37] H. Wang and C. Schmid, “Action recognition with improved trajectories,” in *Proceedings of IEEE International Conference on Computer Vision*, 2013, pp. 3551–3558. 2, 6, 7

[38] G. Huang, H. Lee, and E. Learnedmiller, “Learning hierarchical representations for face verification with convolutional deep belief networks,” in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2012, pp. 2518–2525. 2

[39] A. Krizhevsky, I. Sutskever, and G. E. Hinton, “Imagenet classification with deep convolutional neural networks,” in *Neural Information Processing Systems*, 2012, pp. 1097–1105. 2

[40] A. Jain, J. Tompson, M. Andriluka, G. W. Taylor, and C. Bregler, “Learning human pose estimation features with convolutional networks,” in *International Conference on Learning Representations*, 2014. 2

[41] A. Karpathy, G. Toderici, S. Shetty, T. Leung, R. Sukthankar, and F. F. Li, “Large-scale video classification with convolutional neural networks,” in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2014, pp. 1725–1732. 2, 6, 7, 9

[42] M. D. Zeiler and R. Fergus, “Visualizing and understanding convolutional networks,” in *Proceedings of the European Conference on Computer Vision*, 2013, pp. 818–833. 2, 8

[43] G. W. Taylor, R. Fergus, Y. Lecun, and C. Bregler, “Convolutional learning of spatio-temporal features,” in *Proceedings of the European Conference on Computer Vision*, 2010, pp. 140–153. 2

[44] Y. H. Ng, M. Hausknecht, S. Vijayanarasimhan, O. Vinyals, R. Monga, and G. Toderici, “Beyond short snippets: Deep networks for video classification,” in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2015, pp. 4694–4702. 2, 3, 7, 9

[45] G. Varol, I. Laptev, and C. Schmid, “Long-term temporal convolutions for action recognition,” *IEEE Transactions on Pattern Analysis and Machine Intelligence*, pp. 1–1, 2017. 2

[46] K. Cho, B. Van Merrienboer, C. Gulcehre, D. Bahdanau, F. Bougares, H. Schwenk, and Y. Bengio, “Learning phrase representations using rnn encoder-decoder for statistical machine translation,” *Empirical Methods in Natural Language Processing*, pp. 1724–1734, 2014. 2

[47] J. Donahue, L. A. Hendricks, S. Guadarrama, M. Rohrbach, S. Venugopalan, T. Darrell, and K. Saenko, “Long-term recurrent convolutional networks for visual recognition and description,” in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2015, pp. 2625–2634. 2, 7, 9

[48] W. Huang, F. Sun, L. Cao, D. Zhao, H. Liu, and M. T. Harandi, “Sparse coding and dictionary learning with linear dynamical systems,” in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2016, pp. 3938–3947. 2

[49] N. Srivastava, E. Mansimov, and R. Salakhutdinov, “Unsupervised learning of video representations using lstms,” in *Proceedings of the*

International Conference on Machine Learning, 2015, pp. 843–852. 2, 7, 9

[50] N. Ballas, L. Yao, C. Pal, and A. Courville, "Delving deeper into convolutional networks for learning video representations," in *Proceedings of the International Conference on Learning Representations*, 2016. 2

[51] X. Wang, A. Farhadi, and A. Gupta, "Actions transformations," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2016, pp. 2658–2667. 3, 7, 9

[52] W. Zhu, J. Hu, G. Sun, X. Cao, and Y. Qiao, "A key volume mining deep framework for action recognition," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2016, pp. 1991–1999. 3, 7

[53] A. Kar, N. Rai, K. Sikka, and G. Sharma, "Adascan: Adaptive scan pooling in deep convolutional neural networks for human action recognition in videos," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2017, pp. 5699–5708. 3, 7, 9

[54] Y. Shi, Y. Tian, Y. Wang, and T. Huang, "Sequential deep trajectory descriptor for action recognition with three-stream CNN," *IEEE Transactions on Multimedia*, 2017. 3, 7, 9

[55] Y. Wang, M. Long, J. Wang, and P. S. Yu, "Spatiotemporal pyramid network for video action recognition," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2017, pp. 2097–2106. 3, 7, 9

[56] R. Girdhar, D. Ramanan, A. Gupta, J. Sivic, and B. C. Russell, "Actionvlad: Learning spatio-temporal aggregation for action classification," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2017, pp. 3165–3174. 3, 7

[57] A. Cherian, B. Fernando, M. Harandi, and S. Gould, "Generalized rank pooling for activity recognition," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2017, pp. 1581–1590. 3, 7

[58] S. Roweis and L. K. Saul, "Nonlinear dimensionality reduction by locally linear embedding," *Science*, vol. 290, no. 5500, pp. 2323–2326, 2000. 5

[59] K. Soomro, A. R. Zamir, and M. Shah, "UCF101: A dataset of 101 human actions classes from videos in the wild," *CRCV-TR-12-01*, 2012. 5

[60] H. Kuehne, H. Jhuang, E. Garrote, T. Poggio, and T. Serre, "HMDB: A large video database for human motion recognition," in *Proceedings of the IEEE International Conference on Computer Vision*, 2011, pp. 2556–2563. 5, 6, 7, 9

[61] M. Marszalek, I. Laptev, and C. Schmid, "Actions in context," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2009, pp. 2929–2936. 5, 6

[62] Y. G. Jiang, J. Liu, A. Roshan Zamir, I. Laptev, M. Piccardi, M. Shah, and R. Sukthankar, "THUMOS challenge: Action recognition with a large number of classes," 2013. 5

[63] H. Wang, A. Klaser, C. Schmid, and C. Liu, "Dense trajectories and motion boundary descriptors for action recognition," *International Journal of Computer Vision*, vol. 103, no. 1, pp. 60–79, 2013. 7, 9

[64] Z. Cai, L. Wang, X. Peng, and Y. Qiao, "Multi-view super vector for action recognition," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2014, pp. 596–603. 7, 9

[65] D. Tran and L. Torresani, "Exmoves: Mid-level features for efficient action recognition and video analysis," *International Journal of Computer Vision*, vol. 119, no. 3, pp. 239–253, 2016. 7

[66] H. Bilen, B. Fernando, E. Gavves, A. Vedaldi, and S. Gould, "Dynamic image networks for action recognition," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2017. 7

[67] Y. Jia, E. Shelhamer, J. Donahue, S. Karayev, J. Long, R. Girshick, S. Guadarrama, and T. Darrell, "Caffe: Convolutional architecture for fast feature embedding," *ACM Multimedia*, pp. 675–678, 2014. 6

[68] L. V. D. Maaten, E. o. Postma, and H. J. V. D. Herik, "Dimensionality reduction: A comparative review," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 10, 2009. 7

[69] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2016, pp. 770–778. 10

[70] F. Caba Heilbron, V. Escorcia, B. Ghanem, and J. Carlos Niebles, "Activitynet: A large-scale video benchmark for human activity understanding," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2015, pp. 961–970. 9

Ce Li received the M.S.and Ph.D. degrees in Computer Science from the University of Chinese Academy of Sciences, Beijing, China, respectively. She is a lecturer with China University of Mining & Technology, Beijing, and is currently a visiting scholar with Beihang University, Beijing, China. Her current interests include computer vision, video analysis and machine learning.

Baochang Zhang received the B.S., M.S. and Ph.D. degrees in Computer Science from Harbin Institue of the Technology, Harbin, China, in 1999, 2001, and 2006, respectively. From 2006 to 2008, he was a research fellow with the Chinese University of Hong Kong, Hong Kong, and with Griffith University, Brisban, Australia. Currently, he is an associate professor with the Science and Technology on Aircraft Control Laboratory, School of Automation Science and Electrical Engineering, Beihang University, Beijing, China. He was supported by the Program for New Century Excellent Talents in University of Ministry of Education of China. His current research interests include pattern recognition, machine learning, face recognition, and wavelets.

Chen Chen received the B.E. degree in automation from the Beijing Forestry University, Beijing, China, in 2009, and the M.S. degree in electrical engineering from the Mississippi State University, Starkville, MS, USA, in 2012, and the Ph.D. degree in the Department of Electrical Engineering, University of Texas at Dallas, Richardson, TX, USA, in 2016. He is currently an assistant professor in the Department of Electrical and Computer Engineering, University of North Carolina at Charlotte, NC, USA. He has published more than 50 papers in refereed journals and conferences in these areas. His research interests include compressed sensing, signal and image processing, pattern recognition, and computer vision.

Qixiang Ye received the B.S. and M.S. degrees in mechanical and electrical engineering from the Harbin Institute of Technology, Harbin, China, in 1999 and 2001, respectively, and the Ph.D. degree from the Institute of Computing Technology, Chinese Academy of Sciences, Beijing, China, in 2006. Since 2015, he has been a Full Professor with the University of Chinese Academy of Sciences, and was a Visiting Assistant Professor with the Institute of Advanced Computer Studies, University of Maryland at College Park, College Park, MD, USA, until 2013. He has published over 50 papers in refereed conferences and journals. His current research interests include image processing, visual object detection, and machine learning. Dr. Ye was a recipient of the Sony Outstanding Paper Award. He is a member of the IEEE.

Jungong Han is currently a Senior Lecturer with the Department of Computer Science and Digital Technologies at Northumbria University, Newcastle, UK. Previously, he was a Senior Scientist (2012-2015) with Civolution Technology (a combining synergy of Philips Content Identification and Thomson STS), a Research Staff (2010-2012) with the Centre for Mathematics and Computer Science (CWI), and a Senior Researcher (2005-2010) with the Technical University of Eindhoven (TU/e) in Netherlands. Dr. Han's research interests include Multimedia Content Identification, Multi-Sensor Data Fusion, Computer Vision and Multimedia Security. He is an Associate Editor of Elsevier Neurocomputing (IF 2.4) and an Editorial Board Member of Springer Multimedia Tools and Applications (IF 1.4). He has been (lead) Guest Editor for five international journals, such as IEEE-T-SMCB, IEEE-T-NNLS. Dr. Han is the recipient of the UK Mobility Award Grant from the UK Royal Society in 2016.

Guodong Guo is currently the deputy head of Institue of Deep Learning, Baidu Reserach and National Engineering Laboratory for Deep Learning Technology and Application, Beijing, China. His research interests include computer vision, biometrics, machine learning, and multimedia.

Rongrong Ji received the Ph.D. degree in computer science from the Harbin Institute of Technology, Harbin, China. He has been a Post-Doctoral Research Fellow with the Department of Electrical Engineering, Columbia University, New York, NY, USA, since 2011, with Prof. S.-F. Chang. He is currently a Professor with Xiamen University, Xiamen, China.