Action Recognition with deep network features and dimension reduction

Lijun Li[1], Shuling Dai[1*]
[1] State Key Laboratory of Virtual Reality Technology and Systems, Beihang University
Beijing, China
[e-mail: lilijun1990@buaa.edu.cn, sldai@buaa.edu.cn]
*Corresponding author: Shuling Dai

Received March 4, 2017; revised May 15, 2018; revised June 25, 2018; accepted August 13, 2018; published February 28, 2019

Abstract

Action recognition has been studied in computer vision field for years. We present an effective approach to recognize actions using a dimension reduction method, which is applied as a crucial step to reduce the dimensionality of feature descriptors after extracting features. We propose to use sparse matrix and randomized kd-tree to modify it and then propose modified Local Fisher Discriminant Analysis (mLFDA) method which greatly reduces the required memory and accelerate the standard Local Fisher Discriminant Analysis. For feature encoding, we propose a useful encoding method called mix encoding which combines Fisher vector encoding and locality-constrained linear coding to get the final video representations. In order to add more meaningful features to the process of action recognition, the convolutional neural network is utilized and combined with mix encoding to produce the deep network feature. Experimental results show that our algorithm is a competitive method on KTH dataset, HMDB51 dataset and UCF101 dataset when combining all these methods.

Keywords: Action recognition, Convolutional neural network, Feature encoding, Mix encoding, Dimension reduction

A preliminary version of this paper appeared in IEEE ICVRV2016, September 23-25, Hangzhou, China. This version includes more novel ideas and supporting implement results on action recognition.

http://doi.org/10.3837/tiis.2019.02.019 **ISSN : 1976-7277**

1. Introduction

Action recognition is widely studied in computer vision research field and there have been lots of action recognition applications in our daily life in recent years. It is a challenging task and has attracted much attention in human-computer interaction (HCI), video searching and uncommon activity detection [1-9]. Due to the large variation in viewpoint, human appearance, background clutter, a huge number of action types and camera motion, but action recognition is still hard to implement in the above fields successfully.

Many researchers have contributed lots of efforts in recognizing complicated actions and improving the recognition accuracy. The dictionary of cuboid prototypes to recognize action robustly is proposed by Dollar et al. [10]. Laptev et al. employ Histogram of Gradient (HOG), Histogram of Flow (HOF) features and propose the spatio-temporal pyramid to replace the spatial pyramid method [11]. Wang et al. perform action recognition by using dense sampling which is borrowed from object recognition [12]. Zhang et al. propose spatial-temporal phrases to model spatial relationships among features [13].

Most studies in action recognition focus on feature extraction or feature encoding. There are only few works that study the effect of dimension reduction in action recognition. Murthy and Goecke propose to use Kernel PCA [14] as a dimension reduction method to recognize actions [15]. Xu et al. study the contribution of Linear discriminant analysis (LDA) and PCA in action recognition [16]. There are lots of feature encoding methods that have been studied to improve the performance of action recognition. These methods convert the local features to mid-level features. They actually compute the global features of video from a number of local features. Among the popular feature encoding methods in action recognition, Fisher vector (FV) has been widely used [17].

Although FV is widely used in the action recognition, it still has some drawbacks. It occupies more memory than other encoding methods because of the high dimensionality. So we bring up the dimension reduction method to preprocess the features which could reduce the dimensionality of FV. It is well known that some dimension reduction methods are efficient to reduce dimensionality and can preserve locality information. The goal of dimension reduction is obtaining lower dimensional feature when the original information is maintained as much as possible. In the paper, we explore Local Fisher Discriminant Analysis (LFDA) and use it to reduce the dimensionality of local features.

Convolutional neural network (CNN) has been widely used in the computer vision. It has made a huge success in ImageNet Large-Scale Visual Recognition Challenge. Various attempts to improve architecture have been made to improve performance in category classification, object detection, segmentation and it achieves significant progress in these research fields [18, 19]. Motivated by the great success of Inception model in image classification and object detection [19, 20], we attempt to use it as a feature extraction method to extract deep-learned feature to classify videos.

Our contributions are presented in the following. First, we propose modified Local Fisher Discriminant Analysis (mLFDA) as dimension reduction method to reduce the feature dimension. Second, we propose to concatenate FV and locality-constrained linear encoding (LLC) and then propose the mix encoding by learning from the previous encoding methods. Third, we utilize CNN to produce deep network feature from video frames and use the features to classify videos with the help of the encoding method. Fourth, we outperform most prevailing methods on popular action recognition datasets.

Our paper is organized as follows. Section 2 reviews the recent works of action recognition. Section 3 presents the mLFDA, FV, LLC, mix encoding and the CNN architecture to extract deep network feature. Experiment settings and our complete experiments are introduced in Section 4 on three popular datasets. In Section 5, we conclude our results.

2. Related Work

In recent years, research has progressed a lot in the field of action recognition. For the features extraction, features from image recognition have been imported to action recognition, such as STIP [21, 22], HOG3D [23], 3D SIFT [24], and spatio-temporal descriptor [25]. Trajectory-based descriptor [26], temporal pyramid [27] also perform well in the action recognition. Among all the features in action recognition, the features used in improved trajectories performs better than others [17]. When using improved trajectories, image features are extracted for each frame. HOG, HOF, trajectory feature and motion boundary histogram (MBH) are extracted. When preprocessing the extracted features, they use the PCA [28] to reduce the dimension. After dimension reduction, they use the reduced features to estimate the parameters of Gaussian Mixture Model (GMM) and then use the trained model to encode the features using FV. Afterward, they apply power normalization and L2 normalization to the encoded feature in order to get the final video representation. At last, they use the linear Support Vector Machine (SVM) to classify the actions.

As we know, Convolutional Neural Network (CNN) is a biologically inspired method and it has a long history in computer vision. Its first success can be traced back to digit recognition proposed by LeCun [29]. It was not until the AlexNet [30] came out and achieved great success in ImageNet that the CNN draws people's widely attention. Owing to the bigger datasets, deeper and wider architecture, algorithms, and powerful hardware, especially parallel computing GPUs, it has been proven to be useful in image recognition [30, 31], medical image segmentation [32], and object detection [33] in the recent years. As we know, the Long Short-Term Memory (LSTM) can model long-range temporal relationships. There are also some works combining CNN with LSTM. Donahue et al. [34] propose to use LSTM to encode CNN features and used the average pooling the LSTM features to get video classification scores. Wang et al. use both CNN and RNN in the framework for multi-label image recognition [35]. However, in order to recognize the actions well, CNNs require a large number of labeled data to train from scratch. Due to the limited number of videos in the public action recognition dataset, CNNs do not perform well [36, 37].

Among the remarkable CNN models, the Inception-v2 model performed excellently in the ILSVRC object detection task and classification task which proposed by Szegedy et al. [19, 20]. Their deep CNN is deeper and wider than the previous CNNs and they also propose the batch normalization which can improve the training speed and prevent overfitting. They proposed the inception layer which is a combination of different filter size convolution layers which can make the network more effective and outputs a single vector. We choose to use this model to extract deep network feature.

There are multiple of different encoding method to convert low-level features. When considering the characteristics of encoding methods, they could be classified to four groups mainly: histogram encoding, super vector encoding, reconstruction encoding, and Fisher vector encoding.

The bag of words method is a popular encoding method in natural language processing and it is a kind of histogram encoding. In the field of computer vision, Sivic et al. firstly

propose the bag of visual words to match the objects in videos which assigns features of the objects to single codeword in the codebook[38]. To overcome the disadvantages of hard assignment, kernel codebook encoding is proposed and it could be treated as the soft assignment of histogram encoding [39]. LLC is a kind of reconstruction encoding where features can be indicated as the integration of a few codewords in the codebook [40]. Super vector encoding is introduced to perform nonlinear feature transformation to the features in order to form sparse representation. FV is derived from the Fisher kernel [41]. It is a kind of soft assignment method which estimates the similarities between the feature and codewords. The computation of GMM is needed before FV in order to include the information on deviation and covariance.

Dimension reduction is a traditional machine learning method which has been studied for a few decades. Normally, dimension reduction methods can be grouped into two groups. One is unsupervised dimension reduction, the other is unsupervised dimension reduction. PCA is the most widely used method which applies the orthogonal transformation to the feature to reduce dimensionality [28]. Kernel PCA is another unsupervised dimension reduction method which is the extension of PCA by using kernel methods. Isomap computes the low-dimension embedding of the features and is also an unsupervised method based on spectral theory [42]. Locality-preserving projection could be seen as an alternative dimension reduction method to PCA and can preserve the neighborhood data structure [43].

However, the videos in the datasets are labeled and supervised dimension reduction method could be more effective to reduce the feature dimensions. There are also various kinds of supervised dimension reduction method. For example, Fisher discriminant analysis is effective in reducing the labeled data and it preserves the class discriminant information as much as possible [44]. It is useful in classifying the data belonging to different class as a preprocessing step. LFDA is based on both Fisher discriminant analysis and locality-preserving projection [45]. Although LFDA performs better than Fisher discriminant analysis by introducing the affinity matrix, its computation requires not only lots of time but also lots of memory. In order to overcome the disadvantages of LFDA, we propose the mLFDA as the dimension reduction method to preprocess the feature.

This paper builds upon our previous work [46]. We test our results in two popular datasets.

3. Methodology

In the beginning, we present our action recognition framework. Second, the traditional LFDA is presented in details. Third, we present our mLFDA by the modification to the affinity matrix and making use of the randomized kd-tree. After that, we introduce our feature encoding method which is called mix encoding, based on the LLC and FV. At last, we describe the structure and effectiveness of Inception-v2 model and show how to extract deep network feature by using the deep network structure.

3.1 Framework

We treat the improved trajectories method as the baseline. The major steps are as follows: feature extraction, feature preprocessing, feature encoding, pooling and normalization, and recognition. The framework can be seen in **Fig. 1**.

For feature extraction, HOG [47], HOF [48], horizontal component of MBH (MBHx) [48], vertical component of MBH (MBHy) [48], and Trajectory feature [12] are extracted to describe detected regions in the video frames. We also combine MBHx and MBHy feature as

MBH feature. Moreover, we extract deep network feature from CNN in the meanwhile applying dimension reduction as the preprocessing method. mLFDA is used as the preprocessing step. It is helpful to improve performance and can save lots of memories. It is also much faster than the standard LFDA method. Then we propose the mix encoding to encode the features after dimension reduction. Moveover, we apply the pooling and normalization method to the features after feature encoding. At last, the final video representations are fed to SVM [49] classifier to train to recognize the actions of the video.

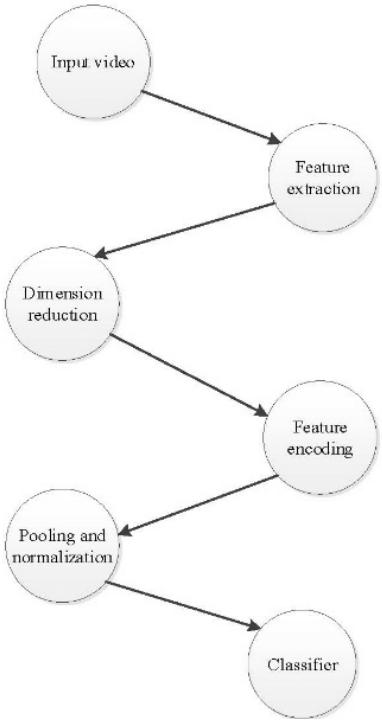


Fig. 1. Framework of our recognition process

3.2 LFDA

Since the low-level features usually have high dimensionality and are correlated, a dimension reduction method should be applied to the features to reduce the computation complexity. Since the data are all labeled, it is more superior to use supervised dimension reduction method to reduce the dimensionality. As LFDA combines the advantages of both LPP and FDA, we utilize LFDA to reduce the feature dimension. The standard LFDA method is presented as follows.

Assuming there is a group of HOG descriptors $x_i \in R^d (i = 1, 2, ...n)$ and the labels $y_i \in \{1, 2, ..., l\}$, where n denotes the total number of descriptors and l implies the number of categories. The matrix $X = (x_1 \mid x_2 \mid ... \mid x_n)$ consists of n original descriptors and the reduced descriptor $z_i \in R^r (1 \le r \le d)$ is applied dimension reduction on x_i, where r is the dimensionality of the reduced descriptor. We define the transformation matrix T and the reduced descriptor can be written as:

$$z_i = T^T x_i \tag{1}$$

The intra-class scatter matrix $S^{(w)}$ and inter-class scatter matrix $S^{(b)}$ are shown as follows.

$$S^{(w)} = \frac{1}{2}\sum_{i,j=1}^{n} W_{i,j}^{(w)}(x_i - x_j)(x_i - x_j)^T \tag{2}$$

$$S^{(b)} = \frac{1}{2}\sum_{i,j=1}^{n} W_{i,j}^{(b)}(x_i - x_j)(x_i - x_j)^T \tag{3}$$

Where

$$W_{i,j}^{(w)} = \begin{cases} A_{i,j}/n_c & if\ y_i = y_j = c \\ 0 & if\ y_i \neq y_j \end{cases} \tag{4}$$

$$W_{i,j}^{(b)} = \begin{cases} A_{i,j}(1/n - 1/n_c) & if\ y_i = y_j = c \\ 1/n & if\ y_i \neq y_j \end{cases} \tag{5}$$

n_c is the number of descriptors in class c. $A_{i,j}$ is the affinity matrix, and $A \in R^{n_c \times n_c}$. If x_i and x_j are close, $A_{i,j}$ is large, otherwise $A_{i,j}$ is small.

The dimension reduction transformation matrix could be defined as

$$T_{LFDA} = \arg\max_T \{tr((T^T S^{(w)} T)^{-1} T^T S^{(b)} T)\} \tag{6}$$

3.3 mLFDA

In the original paper [45], the affinity matrix is set as:

$$A_{i,j} = \exp(-\left\|x_i - x_j\right\|^2 / (\sigma_i \sigma_j)) \tag{7}$$

$$\sigma_i = \left\|x_i - x_i^{(7)}\right\| \tag{8}$$

$x_i^{(7)}$ is the seventh nearest neighbor of x_i in the same class. When extracting 480000 samples, 80000 per class, to perform dimension reduction, affinity matrix A would take about 50GB memory space. In our modication, the affinity matrix $A_{i,j}$ is set as follows.

$$A_{i,j} = \begin{cases} \exp(-\left\|x_i - x_j\right\|^2 / \sigma^2) & if\ x_j \in NN_i^{(7)} \\ 0 \end{cases} \tag{9}$$

$$\sigma = \left\|x_i - x_i^{(7)}\right\| \tag{10}$$

$NN_i^{(7)}$ is a set of the seventh nearest neighbor of x_i. Given our affinity matrix $A_{i,j}$, sparse matrix is used to save the matrix which would vastly reduce memory usage. It will take about more than 10 times lower memory space than normal LFDA.

There are various fast nearest neighbor search methods. Among them, hashing has proved to be an effective method by transforming features as binary codes to support retrieval and efficient storage [50-54]. While they are effective in image/video retrieval, they are rather complicated. We choose the randomized kd-tree which is both simple and effective. The FLANN library [55] is used to employ randomized kd-tree to search the nearest neighbors. By using the library, the time to search nearest neighbors can be largely reduced. Specifically, a group of randomized kd-tree is built and searched in parallel which can speed up the search process. There are two hyperparameters in this algorithm, kd-tree number and check number. Kd-tree number denotes the number of trees and the checknumber implies the maximum leaves to search. When we choose the proper hyperparameters, mLFDA can be used for dimension reduction. The mLFDA are used in our experiments as the dimension

reduction method.

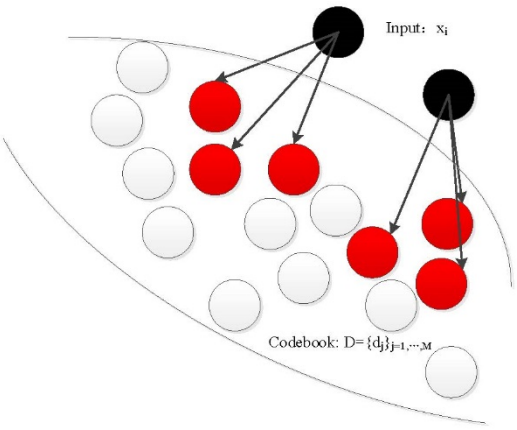


Fig. 2. Codeword generation of LLC

3.4 Encoding method

Two prevailing image encoding methods (LLC and FV) are introduced to lead to our proposed encoding method. They can also be used for the video encoding to get the representations of videos. In order to preserve the advantages of both methods, we propose to combine them as mix encoding.

LLC projects the feature vector to its local coordinate space, whose idea is based on the local coordinate coding [56]. We can see the codeword generation example in **Fig. 2**.

If there is a set of center vectors $d_1, d_2, ..., d_M$, it is computed by the k-means algorithm which makes up the codebook D. Among the center vectors, $d_{\sigma_1}, ..., d_{\sigma_N}$ are closer to x_i than the others. We set $B = [d_{\sigma_1}, ..., d_{\sigma_N}]$, then the feature encoding can be solved by the minimization problem.

$$\begin{aligned} &\min_C \| x_i - Bc_i \|^2 + \lambda \left\| e_i \circ c_i \right\|^2 \\ &st. 1^{\mathrm{T}} c_i = 1 \end{aligned} \tag{11}$$

In the above function, $\circ$ represents the Hadamard product, and $e_i \in R^M$ is proportional to the Euclidean distance between x_i and B.

$$e_i = \exp(\frac{dist(x_i, B)}{\theta}) \tag{12}$$

Where θ is an adjusting parameter, the distance function $dist(x_i, B) = [dist(x_i, b_1), ..., dist(x_i, b_N)]^{\mathrm{T}}$ is the concatenation of each element $dist(x_i, b_1)$ which denotes the Euclidean distance between x_i and b_1.

After feature encoding, feature x_i can be expressed as:

$$F = [c_1, c_2, ..., c_k] \tag{13}$$

Meanwhile, the FV method is presented in the following. It is based on the Fisher kernel [41]. With a probability density function p where θ denotes the parameters, the score function can be presented as:

$$G_\theta^x = \nabla_\theta \mathrm{In}\, p(x \mid \theta) \tag{14}$$

In FV, we choose the probability density function in GMM to represent p. GMM is based on the assumption that the data are sampled by a limited number of Gaussian distributions. It is commonly used as a parametric model of probability density distribution. The parameters of the K-component GMM are denoted by $\{\gamma_k, \mu_k, \sigma_k\}$, where μ_k denotes mean of the kth Gaussian distribution and σ_k is standard deviation of the kth component of GMM. γ_k is the probability of x_i belonging to kth Gaussian distribution and π_k is the prior probability.

$$\gamma_k = \frac{\pi_k p(x_i \mid \mu_k, \sigma_k)}{\sum_{k=1}^{K} \pi_k p(x_i \mid \mu_k, \sigma_k)} \tag{15}$$

Because the contribution of different feature x can be estimated by the gradient of Equation (14), it is needed to compute the derivatives of parameters. With a set of features $x_1, ..., x_n$, the derivatives can be computed as follows using Equation (14):

$$\mathcal{G}_{\mu,k}^{x_i} = \frac{1}{\sqrt{\pi_k}} \gamma_k \left(\frac{x_i - \mu_k}{\sigma_k}\right) \tag{16}$$

$$\mathcal{G}_{\sigma,k}^{x_i} = \frac{1}{\sqrt{2\pi_k}} \gamma_k \left[\frac{(x_i - \mu_k)^2}{\sigma_k^2} - 1\right] \tag{17}$$

FV concatenates the derivatives to perform the feature encoding. After that, x_i can be expressed as:

$$F = [\mathcal{G}_{\mu,1}^{x_i}, \mathcal{G}_{\sigma,1}^{x_i}, ..., \mathcal{G}_{\mu,K}^{x_i}, \mathcal{G}_{\sigma,K}^{x_i}] \tag{18}$$

Given the details of LLC and FV, it can be found that FV uses the first and second order difference in the meantime the LLC uses the soft assignment which denotes the probability to the codeword. Thus, we propose the mix encoding to take advantages of both LLC and FV. In fact, the mix encoding is the concatenation of both the LLC and FV which contains not only the zero order difference but also the first and second order difference. So mix encoding is shown as follows:

$$F = [c_1, c_2, ..., c_k, \mathcal{G}_{\mu,1}^{x_i}, \mathcal{G}_{\sigma,1}^{x_i}, ..., \mathcal{G}_{\mu,K}^{x_i}, \mathcal{G}_{\sigma,K}^{x_i}] \tag{19}$$

In Section 4, we will compare our mix encoding and the other feature encoding methods in action recognition.

3.5 CNN model

The Inception-v2 model performs excellently in ILSVRC 2014 in the field of classification and object detection. We suppose that its extracted features are discriminative enough as the feature descriptor. Thus, the Inception-v2 is used to extract deep network feature in the paper and we use the deep network features to describe the video. The structure is presented in **Fig. 3**.

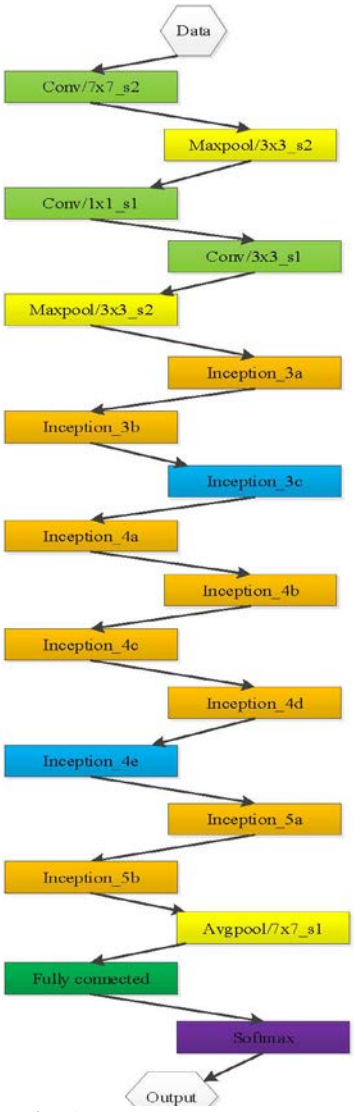


Fig. 3. Inception-v2 structure

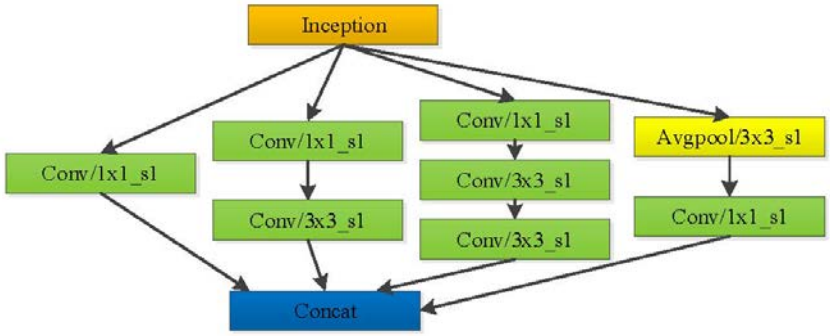


Fig. 4. Inception module in Inception-v2 model

The whole network is 18 layers deep and contains 10 Inception modules. The inception module is composed of convolutional layers with filter sizes of 1x1, 3x3 and pooling layers.

The inception module consists of these convolution types stacked together and two kinds of inception module in the model are presented in **Fig. 4** and **Fig. 5** respectively. In the two figures, s represents the stride in the convolutional and pooling layer in **Fig. 3**, **Fig. 4** and **Fig. 5**. The 1x1 convolutional layers are used to compute dimension reduction for the purpose of reducing the computational complexity. Each convolutional layers follows by a batch normalization layer and uses the RELU activation function [19].

We use the Inception-v2 model trained on ImageNet dataset [57, 58]. The model obtained top-5 accuracy 95.1% on ImageNet. The output of the average pooling layer is chosen as the deep network feature because it is a high-level feature which can be more discriminative than the low-level feature and the output size is suitable. The output feature vector dimension is 1024.

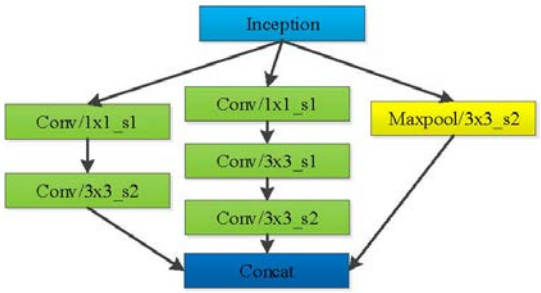


Fig. 5. Inception module in Inception-v2 model

4. Experimental Evaluation

This section describes the action recognition datasets and the experiment results on the datasets.

4.1 Datasets

In this section, we introduce the KTH dataset, HMDB51 dataset and UCF101 dataset. They are widely used in the evaluation of action recognition algorithms.

Fig. 6. Examples on KTH dataset

We conduct major experiments on KTH dataset. It is made up of six hundred videos and there are 6 kinds of actions, such as walking, running in 4 kinds of scenarios [59]. Each video is divided into several sequences. In total, there are 2391 sequences in the dataset. We follow the original setup of the authors to assign sequences to the training set and testing set. In line with the original paper, we report the average accuracy on all the categories. Sample frames from the dataset are shown in **Fig. 6**.

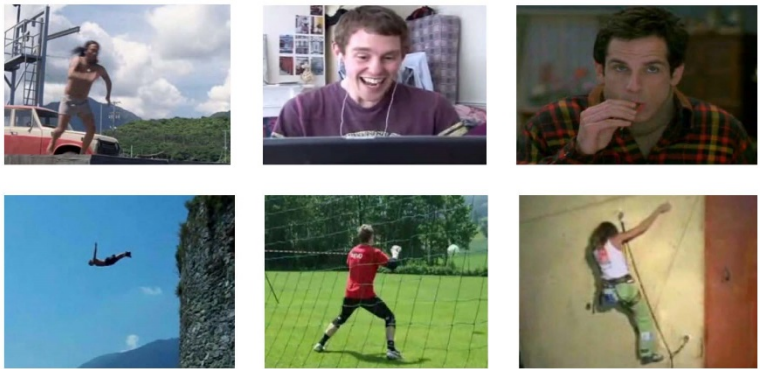


Fig. 7. Examples on HMDB51 dataset

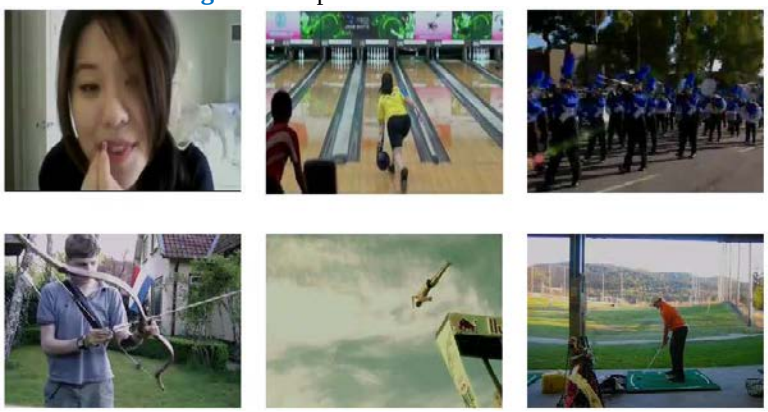


Fig. 8. Examples on UCF101 dataset

We also compare our method with other works on HMDB51 dataset. It is made up of 6766 videos and fifty-one classes. Examples of HMDB51 dataset are shown in **Fig. 7**. It contains not only body movement but also human-object interaction. We follow the original training/testing splits [60] and present the average accuracy of the splits to measure recognition performance. Moreover, UCF101 dataset is used in our experiments [61]. It has 101 action categories and 13320 videos in total. We also report the average accuracy on all the classes. Example videos are shown in **Fig. 8**.

4.2 Experimental details

In the experiment, we follow major steps introduced in Section 3.1. To extract the deep network feature, we firstly resize the video frames to 224x224 to fit the input of the Inception-v2 model. As to reduce the computational complexity, we sample the video frames every 3 frames. Then we feed the frames into the Inception and extract feature from the last pooling layer as feature vector whose feature dimension is 1024. After that, we collect the feature vectors of the entire video. When training the LLC codebook, we use 256000

randomly sampled features. The size of LLC codebook is 2048. When training the GMM, we also randomly sample 256000 features and the size of Gaussian distributions is 256.

While for deep network feature, we train GMM with 72000 randomly sampled features. For the pooling method, we use sum pooling to sum up the features in one video. While for the normalization method, we use power and L2 normalization to the features. Then we can obtain the video representations of different features. After that, the video representations are concatenated to get the final video representation. In the end, linear SVM is used for action recognition.

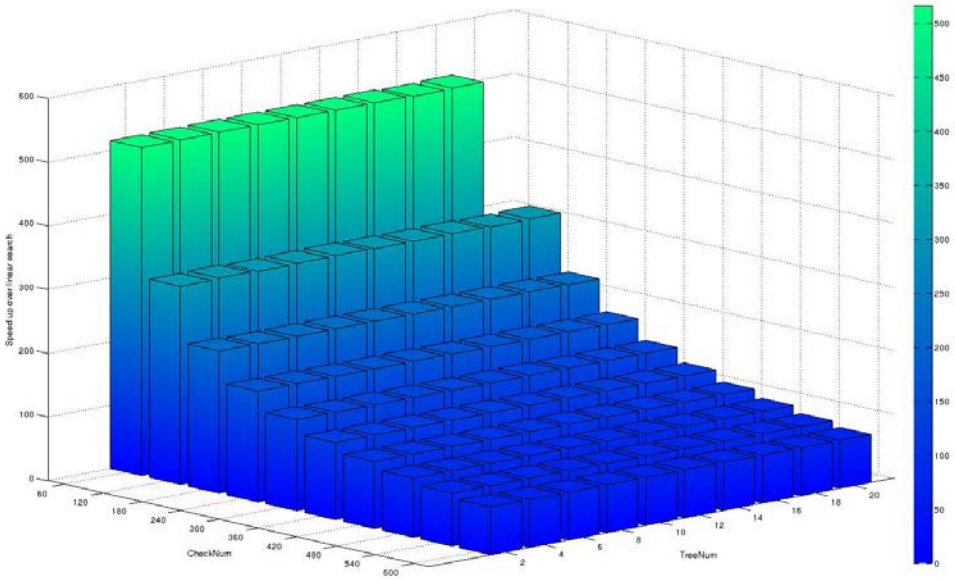


Fig. 9. Performance of randomized kd-tree nearest neighbor search on KTH dataset

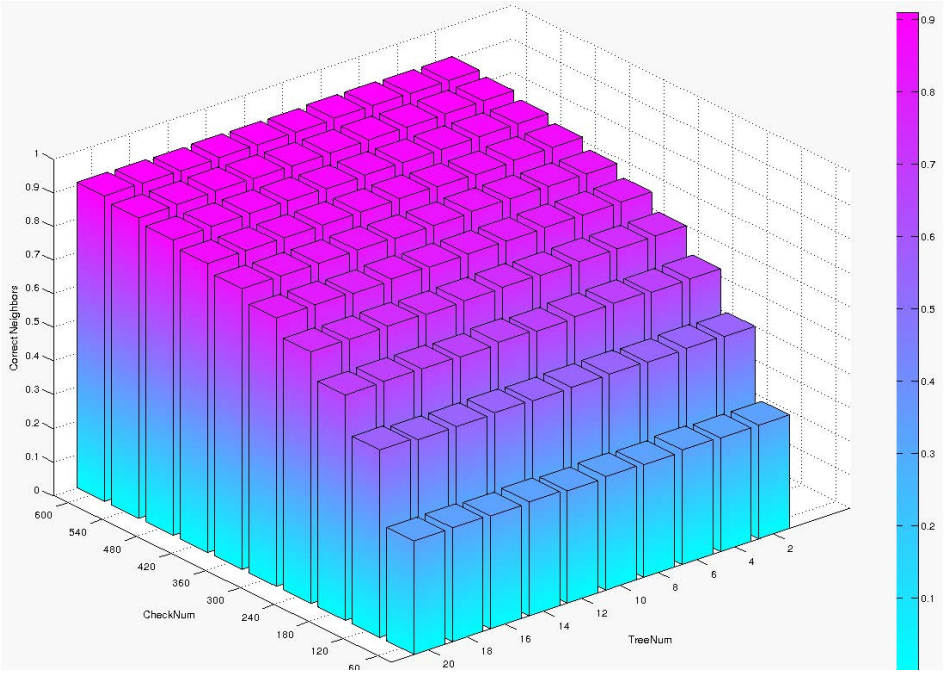


Fig. 10. Correct neightbors by searching in randomized kd-tree on KTH dataset

4.3 Evaluation of dimension reduction method

The significance of dimension reduction is evaluated in the section. We compare the performance of no dimension reduction method, mLFDA and PCA on KTH dataset and HMDB51 dataset.

We firstly evaluate the performance of randomized kd-tree used in mLFDA and we can see that the nearest neighbor searching speed is greatly improved comparing with linear search. It can also greatly reduce the memory consumption which is explained in Section 3.3. In **Fig. 9**, when tuning the kd-tree number and check number, the searching speed can boost to 500 times faster than linear search. While in **Fig. 10**, when tuning the check number and kd-tree number, the correctness of searching nearest neighbor can be more than 90%. Given the above figures, we can see that larger values of check number can show better search accuracy. On the contrary, larger values of check number can lead to more computation time. However, the number of kd-tree seems less important for the searching correctness and speed. By summarizing from the figures, we choose to assign the kd-tree number to eight and check number to six hundred as the hyperparameters in the following experiment. While our mLFDA can not only reduce the computational time but also greatly reduce memory consumption as described in Section 3.3. Comparing with the normal LFDA, our mLFDA use 10 times less memory in computation.

Table 1. Comparison of different dimension reduction method on KTH dataset

Feature Type	No Dimension Reduction	mLFDA	PCA
HOG	87.03%	89.81%	88.89%
HOF	93.05%	93.52%	93.05%
MBH	94.50%	95.60%	94.91%
Trajectory	88.89%	89.81%	89.81%
Combined	94.79%	95.72%	95.21%

Table 2. Comparison of different dimension reduction method on HMDB51 dataset

Feature Type	No Dimension Reduction	mLFDA	PCA
HOG	30.58%	41.61%	41.11%
HOF	42.81%	47.11%	46.76%
MBH	47.90%	51.58%	50.89%
Trajectory	25.01%	26.64%	27.25%
Combined	56.70%	58.40%	57.60%

Then we compare the different dimension reduction methods and no dimension reduction method on both KTH dataset and HMDB51 dataset. The descriptor dimension is reduced by a factor of two, as in the paper [62] for both dimension reduction methods. From the results in **Table 1**, it can be found that the recognition accuracy can be improved greatly with the help of dimension reduction method, as a preprocessing step to the features. When comes to

HOG and MBH feature, mLFDA improves the accuracy much more than the other features about 2.8% and 1.5%. For Trajectory feature, mLFDA slightly improves the recognition performance. As to HOF features, mLFDA can also slightly improve the performance. When the features are combined, the recognition accuracy is much higher than every single feature for all dimension reduction methods. The recognition performance of PCA and mLFDA are also compared in **Table 1**. The result suggests that it is useful to use dimension reduction as the preprocessing step. Our mLFDA performs better than PCA in almost every feature. Generally, the mLFDA is effective to preprocess the features in order to recognize the actions in the video. When the output feature dimensionality varies, the recognition performance does not change greatly. When the feature vector is mapping to low dimensional space, the recognition accuracy is also relative a slightly lower. When it is mapping to higher dimensional space, the recognition accuracy is improving a little bit and reach a limit. The same trend can be seen in **Table 2**. Action recognition with dimension reduction method outperforms recognition without dimension reduction. Although mLFDA gets slightly lower accuracy in Trajectory feature, it can get higher accuracy in other features on HMDB51 dataset than PCA.

4.4 Comparison of feature encoding method

In this section, the original FV method and our mix encoding are compared on the KTH dataset. From the results in **Table 3**, the mix encoding not only performs better than FV about 0.25% but also outperform LLC about 1.16%.

Table 3. Comparison of different feature encoding techniques on KTH dataset

Feature Type\Encoding Method	Fisher vector	LLC	Mix encoding
HOG	89.81%	84.25%	89.81%
HOF	93.52%	92.93%	93.73%
MBH	95.60%	95.02%	95.80%
Trajectory	89.81%	88.20%	89.58%
Combined	95.72%	94.79%	95.95%

For HOF, and MBH features, our encoding method performs better than other methods about 0.21% and 0.2% respectively. While for HOG and Trajectory feature, mix encoding performs almost the same as FV. FV aggregates information with first and second order statistics, while LLC models zero order statistics. The improvement in HOF and MBH features may be because the flow motion information can benefit from the zero order statistics than HOG and Trajectory features. When comes to the accuracy of the combined feature, our method performs 0.23% better than FV. From the results above, it can be concluded that the proposed mix encoding method could actually improve the recognition performance more than FV in a way.

Table 4. Comparison of different encoding methods in CNN feature on KTH dataset

Method	Accuracy
Mean pooling CNN	84.70%
LCC encoding CNN	80.93%
FV encoding CNN	88.00%
Mix encoding CNN	89.04%

Table 5. Comparison of HOG feature and deep feature on KTH dataset

Feature Type	Accuracy(%)
HOG+HOF	94.55%
HOG+MBH	95.83%
HOG+Trajectory	93.63%
Deep feature+HOF	95.02%
Deep feature+MBH	96.18%
Deep feature+Trajectory	93.05%

4.5 Deep network feature

As described in the previous section, we utilize Inception-v2 model pretrained from ImageNet to extract CNN feature using the output of the last pooling layer. In order to reduce the computational complexity, we sample the entire video every 3 frames to feed into the network. After applying mLFDA to reduce dimensionality, the Fisher vector encoded CNN feature's recognition accuracy can reach 70.83%. If we perform mean pooling on the CNN features instead of using FV, the recognition accuracy is 67.12%. It can be seen that our CNN fisher vector performs better than mean pooling CNN features.

On the other hand, after pretraining on KTH dataset, the network can produce more specific features that suit KTH dataset. After pretraining, the mean pooling of CNN features can reach 84.70% in recognition accuracy. When the CNN features are encoded by mix encoding, the recognition accuracy can get 89.04% which still performs better than Fisher vector encoding. The comparison of the different encoding method of CNN is presented in **Table 4**. Meanwhile, the performance of deep network feature is almost the same as HOG feature which is about 89.80%.

Since HOG feature and deep network feature both are features that describe the appearance of image frames, we compare the HOG feature and deep network feature with their combination with HOF feature, MBH feature, and Trajectory feature.

We can clearly see from **Table 5** that combination of the deep network feature with motion description feature can be slightly more discriminative than the combination of HOG feature with the motion description feature. The combination of deep network feature with MBH feature can outperform the combination of HOG feature with MBH feature about 0.4% improvements in recognition accuracy. While the results in **Table 6** clearly show that the combination of deep network feature with the hand-crafted features can improve the recognition accuracy of hand-crafted features. The combination of deep network feature with all the other features can reach the highest recognition accuracy of 96.53%. We can conclude from the experiment that the deep-learned feature can represent some latent features that the hand-crafted feature cannot represent.

Table 6. Combination of deep network feature with the other features on KTH dataset

Feature Type	Accuracy(%)
Deep feature+HOG	90.16%
Deep feature+HOF	95.02%
Deep feature+MBH	96.18%
Deep feature+Trajectory	93.05%
Deep feature+All	96.53%

Table 7. Comparison with the state of the art methods on KTH dataset

KTH	Accuracy
Laptev et al. (2008) [11]	91.80%
Liu et al. (2009) [63]	93.80%
Kovashka et al. (2010) [64]	94.50%
Zhang et al. (2012) [13]	94.00%
Baumann et al. (2013) [65]	94.31%
Wang et al. (2013) [12]	94.40%
Veriahh et al. (2015) [66]	93.28%
Ours	96.53%

4.6 Comparison with the state-of-the-art

In this section, our method combines the mLFDA, mix encoding, and the deep network feature. Our final method is compared with the state-of-the-art methods on KTH dataset, HMDB51 dataset, and UCF101 dataset.

Comparing with other state-of-the-art methods in **Table 7**, the evaluation can be concluded in the following. Laptev et al. get 91.80% accuracy by extracting HOG feature and HOF feature and using the pyramid method [11]. By mining the significant features and ranking via PageRank, Liu et al. obtain 93.80% accuracy [63]. Given training videos, Kovashka et al. obtain the accuracy of 94.50% by extracting motion and appearance feature and building a hierarchy of composite vocabularies[64]. By proposing the bag of spatio-temporal phrase and 3D correspondence transformation, Zhang et al. reported 94.00% accuracy [13]. By learning separate Random Forest classifier for each feature and then combining the classifiers, Baumann et al. can get the 94.31% accuracy [65]. Wang et al. obtain 94.40% accuracy with their dense trajectory method and different kinds of features [12]. Veeriah et al. suggested to use the differential recurrent neural network and reached 93.28% on the KTH dataset [66]. Among all the methods, our method performs the best with the accuracy of 96.53%.

Table 8. Comparison with the state of the art methods on HMDB51 dataset

HMDB51	Accuracy
Kuehne et al. (2011) [60]	23.18%
Sadanand et al. (2012) [67]	26.90%
Wang et al. (2013) [17]	57.20%
Cai et al. (2014) [68]	55.90%
Park et al. (2016) [69]	56.20%
Xu et al. (2016) [16]	56.47%
Murthy and Roland (2016) [15]	57.80%
Fernando et al. (2017) [70]	61.80%
Our method	59.40%

Table 9. Comparison with the state of the art methods on UCF101 dataset

UCF101	Accuracy
Khurram et al. (2011) [61]	43.90%
Karpthy et al. (2014) [73]	63.90%
Wu et al. (2014) [71]	81.23%
Donahue et al. (2015) [34]	82.66%
Tran et al. (2015) [74]	76.40%
Xu et al. (2016) [16]	84.77%
Wang et al. (2016) [75]	86.00%
Murthy and Roland (2016) [15]	86.30%
Our method	87.60%

Our method is compared with competitive methods on the HMDB51 dataset in **Table 8**. Kuehne et al. introduced the HMDB51 dataset and reported 23.18% accuracy utilized HOG and HOF for action classification [60]. Sadanand et al. proposed action bank method and got 26.9% accuracy [67]. Wang et al. reported 57.2% by proposing improved trajectories [17]. Cai et al. got 55.9% accuracy by using multi-view super vector [68]. Park et al. proposed feature amplification and different methods to fuse CNN features with the recognition accuracy of 56.2% [69]. Murthy and Goecke applied KPCA as the dimension reduction method in action recognition to improve the recognition accuracy and reported 57.8% accuracy on HMDB51 dataset and 86.3% on UCF101 dataset [15]. Xu et al. used LDA and PCA as the dimension reduction method and classified the feature using extreme learning machine [16]. They got 56.47% on HMDB51 dataset and 84.77% on UCF101 dataset. Fernando et al. [70] proposed rank pooling to capture the temporal dynamics of video and reached the accuracy of 61.8%. It is higher than ours in the HMDB51 dataset because they

capture the dynamics of the CNN feature. Our method is better than the others on the HMDB51 dataset. We compare ours with others on UCF101 dataset in **Table 9**. Khurram et al. proposed UCF101 dataset and report 43.9% recognition accuracy [61]. Donahue et al. combined LSTM with CNN and the classification accuracy is 82.66% [34]. Wu et al. propose bimodal encoding method built upon VLAD [71] to boost the accuracy to 81.23% [72]. Karphthy et al. [73] investigate several temporal fusion in CNN and reported 63.9% by averaging them. Tran et al. propose a 3D convolution network for action recognition and get 76.4% [74]. Wang et al. reported 86% on UCF101 dataset [75]. While our method can get 87.6% which is the best in the table. In brief, through the experiments on these datasets, our method shows competitive performance.

5. Conclusion

In this paper, we propose mLFDA method and use it as our dimension reduction method with success. With the help of randomized kd-tree, the searching speed of our mLFDA method could be more than one hundred times faster than linear search. In the experiments, it is shown that dimension reduction method is indispensable in action recognition.

Our dimension reduction method could be used in other applications with labeled data. Moreover, a new feature encoding method, mix encoding, is proposed in the paper. The mix encoding takes advantage of both FV and LLC. From the experiments, it can be seen our encoding method shows high recognition accuracy on the public datasets and our method outperform both FV and LLC. The combination of our mLFDA and mix encoding shows better performance than the baseline method.

The prominent Inception-v2 model is used to extract deep network feature through video frames. It is also a discriminative appearance feature and can improve recognition accuracy by combining itself with improved trajectories feature. When combining the deep network feature with improved trajectories feature, the recognition accuracy is higher than the prevailing methods on the KTH dataset. While on the HMDB51 and UCF101 dataset, our method also performs better than most methods.

In the meantime, the accuracy of action recognition can be improved with other kinds of methods. It is noticed that the binary quantization has shown encouraging performance in large-scale image retrieval [76]. In the future, we will do some experiments to check whether the binary quantization methods can be integrated for better performance. On the other hand, we can make use of the 3D CNN to extract the features with temporal information to further increase the recognition accuracy.

References

[1] X. Wang, L. Gao, J. Song, and H. Shen, "Beyond frame-level cnn: Saliency-aware 3-d cnn with lstm for video action recognition," *IEEE Signal Processing Letters*, vol. 24, pp. 510-514, 2017. Article (CrossRef Link)

[2] X. Wang, L. Gao, J. Song, X. Zhen, N. Sebe, and H. T. Shen, "Deep appearance and motion learning for egocentric activity recognition," *Neurocomputing*, vol. 275, pp. 438-447, 2018. Article (CrossRef Link)

[3] S. Herath, M. Harandi, and F. Porikli, "Going deeper into action recognition: A survey," *Image and vision computing*, vol. 60, pp. 4-21, 2017. Article (CrossRef Link)

[4] J. Song, L. Gao, F. Nie, H. T. Shen, Y. Yan, and N. Sebe, "Optimized graph learning using partial tags and multiple features for image and video annotation," *IEEE Transactions on Image*

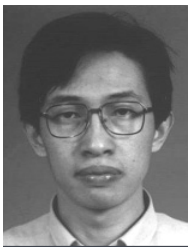
Processing, vol. 25, pp. 4999-5011, 2016. Article (CrossRef Link)
[5] X. Wang, L. Gao, P. Wang, X. Sun, and X. Liu, "Two-stream 3-D convNet fusion for action recognition in videos with arbitrary size and length," *IEEE Transactions on Multimedia*, vol. 20, pp. 634-644, 2018. Article (CrossRef Link)
[6] L. Gao, Z. Guo, H. Zhang, X. Xu, and H. T. Shen, "Video captioning with attention-based lstm and semantic consistency," *IEEE Transactions on Multimedia*, vol. 19, pp. 2045-2055, 2017. Article (CrossRef Link)
[7] L. Li and B. Gong, "End-to-end video captioning with multitask reinforcement learning," *arXiv preprint arXiv:1803.07950*, 2018.
[8] Y. Li, T. Yang, and B. Gong, "How Local is the Local Diversity? Reinforcing Sequential Determinantal Point Processes with Dynamic Ground Sets for Supervised Video Summarization," *arXiv preprint arXiv:1807.04219*, 2018.
[9] C. Gan, Y. Li, H. Li, C. Sun, and B. Gong, "Vqs: Linking segmentations to questions and answers for supervised attention in vqa and question-focused semantic segmentation," in *Proc. IEEE Int. Conf. Comp. Vis*, 2017. Article (CrossRef Link)
[10] P. Dollár, V. Rabaud, G. Cottrell, and S. Belongie, "Behavior recognition via sparse spatio-temporal features," in *Proc. of IEEE International Workshop on Visual Surveillance and Performance Evaluation of Tracking and Surveillance*, pp. 65-72, 2005. Article (CrossRef Link)
[11] I. Laptev, M. Marszałek, C. Schmid, and B. Rozenfeld, "Learning realistic human actions from movies," in *IEEE Conference on Computer Vision and Pattern Recognition*, pp. 1-8, 2008. Article (CrossRef Link)
[12] H. Wang, A. Kläser, C. Schmid, and C.-L. Liu, "Dense trajectories and motion boundary descriptors for action recognition," *International journal of computer vision*, vol. 103, pp. 60-79, 2013. Article (CrossRef Link)
[13] Y. Zhang, X. Liu, M.-C. Chang, W. Ge, and T. Chen, "Spatio-temporal phrases for activity recognition," in *Proc. of European Conference on Computer Vision*, ed: Springer, pp. 707-721, 2012. Article (CrossRef Link)
[14] B. Schölkopf, A. Smola, and K.-R. Müller, "Nonlinear component analysis as a kernel eigenvalue problem," *Neural computation*, vol. 10, pp. 1299-1319, 1998. Article (CrossRef Link)
[15] V. R. M. Oruganti and R. Goecke, "Dimensionality reduction of Fisher vectors for human action recognition," *IET Computer Vision*, vol. 10, pp. 392-397, 2016. Article (CrossRef Link)
[16] H. Xu, Q. Tian, Z. Wang, and J. Wu, "A joint evaluation of different dimensionality reduction techniques, fusion and learning methods for action recognition," *Neurocomputing*, vol. 214, pp. 329-339, 2016. Article (CrossRef Link)
[17] H. Wang and C. Schmid, "Action recognition with improved trajectories," in *Proc. of Computer Vision (ICCV), 2013 IEEE International Conference on*, pp. 3551-3558, 2013. Article (CrossRef Link)
[18] J. Long, E. Shelhamer, and T. Darrell, "Fully convolutional networks for semantic segmentation," in *Proc. of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 3431-3440, 2015. Article (CrossRef Link)
[19] C. Szegedy, W. Liu, Y. Jia, P. Sermanet, S. Reed, D. Anguelov, *et al.*, "Going deeper with convolutions," in *Proc. of the IEEE conference on computer vision and pattern recognition*, pp. 1-9, 2015. Article (CrossRef Link)
[20] S. Ioffe and C. Szegedy, "Batch normalization: Accelerating deep network training by reducing internal covariate shift," in *Proc. of International Conference on Machine Learning*, pp. 448-456, 2015.
[21] D. D. Dawn and S. H. Shaikh, "A comprehensive survey of human action recognition with spatio-temporal interest point (STIP) detector," *The Visual Computer*, vol. 32, pp. 289-306, 2016. Article (CrossRef Link)
[22] Y. Li, J. Ye, T. Wang, and S. Huang, "Augmenting bag-of-words: a robust contextual representation of spatiotemporal interest points for action recognition," *The Visual Computer*, vol. 31, pp. 1383-1394, 2015. Article (CrossRef Link)

[23] A. Klaser, M. Marszałek, and C. Schmid, "A spatio-temporal descriptor based on 3d-gradients," in *British Machine Vision Conference*, pp. 1-10, 2008. Article (CrossRef Link)
[24] P. Scovanner, S. Ali, and M. Shah, "A 3-dimensional sift descriptor and its application to action recognition," in *Proc. of the 15th international conference on Multimedia*, pp. 357-360, 2007. Article (CrossRef Link)
[25] L. Li and S. Dai, "Action recognition with spatio-temporal augmented descriptor and fusion method," *Multimedia Tools and Applications*, pp. 1-17, 2016.
[26] Y. Yi and H. Wang, "Motion keypoint trajectory and covariance descriptor for human action recognition," *The Visual Computer*, pp. 1-13, 2017.
[27] J. Wu, D. Hu, and F. Chen, "Action recognition by hidden temporal models," *The Visual Computer*, vol. 30, pp. 1395-1404, 2014. Article (CrossRef Link)
[28] I. Jolliffe, *Principal component analysis*: Wiley Online Library, 2002.
[29] Y. LeCun, B. Boser, J. S. Denker, D. Henderson, R. E. Howard, W. Hubbard, *et al.*, "Backpropagation applied to handwritten zip code recognition," *Neural computation*, vol. 1, pp. 541-551, 1989. Article (CrossRef Link)
[30] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "Imagenet classification with deep convolutional neural networks," in *Advances in neural information processing systems*, pp. 1097-1105, 2012.
[31] S. Xie, T. Yang, X. Wang, and Y. Lin, "Hyper-class augmented and regularized deep learning for fine-grained image classification," in *Proc. of IEEE Conference on Computer Vision and Pattern Recognition*, 2015. Article (CrossRef Link)
[32] L. Bi, J. Kim, A. Kumar, M. Fulham, and D. Feng, "Stacked fully convolutional networks with multi-channel learning: application to medical image segmentation," *The Visual Computer*, pp. 1-11, 2017. Article (CrossRef Link)
[33] R. Girshick, F. Iandola, T. Darrell, and J. Malik, "Deformable part models are convolutional neural networks," in *IEEE Conference on Computer Vision and Pattern Recognition*, pp. 437-446, 2015. Article (CrossRef Link)
[34] J. Donahue, L. Anne Hendricks, S. Guadarrama, M. Rohrbach, S. Venugopalan, K. Saenko, *et al.*, "Long-term recurrent convolutional networks for visual recognition and description," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 2625-2634, 2015. Article (CrossRef Link)
[35] J. Wang, Y. Yang, J. Mao, Z. Huang, C. Huang, and W. Xu, "Cnn-rnn: A unified framework for multi-label image classification," in *Proc. of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 2285-2294, 2016. Article (CrossRef Link)
[36] A. Karpathy, G. Toderici, S. Shetty, T. Leung, R. Sukthankar, and L. Fei-Fei, "Large-scale video classification with convolutional neural networks," in *IEEE Conference on Computer Vision and Pattern Recognition*, pp. 1725-1732, 2014. Article (CrossRef Link)
[37] L. Sun, K. Jia, D.-Y. Yeung, and B. E. Shi, "Human Action Recognition using Factorized Spatio-Temporal Convolutional Networks," in *Proc. of the IEEE International Conference on Computer Vision*, pp. 4597-4605, 2015. Article (CrossRef Link)
[38] J. Sivic and A. Zisserman, "Video Google: A text retrieval approach to object matching in videos," in *Proc. of IEEE International Conference on Computer Vision*, pp. 1470-1477, 2003. Article (CrossRef Link)
[39] J. Philbin, O. Chum, M. Isard, J. Sivic, and A. Zisserman, "Lost in quantization: Improving particular object retrieval in large scale image databases," in *Proc. of IEEE Conference on Computer Vision and Pattern Recognition*, pp. 1-8, 2008. Article (CrossRef Link)
[40] J. Wang, J. Yang, K. Yu, F. Lv, T. Huang, and Y. Gong, "Locality-constrained linear coding for image classification," in *Proc. of IEEE Conference on Computer Vision and Pattern Recognition*, pp. 3360-3367, 2010. Article (CrossRef Link)
[41] F. Perronnin, J. Sánchez, and T. Mensink, "Improving the fisher kernel for large-scale image classification," in *Proc. of European Conference on Computer Vision*, ed: Springer, pp. 143-156, 2010. Article (CrossRef Link)
[42] J. B. Tenenbaum, V. De Silva, and J. C. Langford, "A global geometric framework for nonlinear dimensionality reduction," *Science*, vol. 290, pp. 2319-2323, 2000. Article (CrossRef Link)

[43] He, Xiaofei, and Partha Niyogi. "Locality preserving projections." In *Advances in neural information processing systems*, pp. 153-160. 2004.
[44] B. Scholkopft and K.-R. Mullert, "Fisher discriminant analysis with kernels," *Neural networks for signal processing IX*, vol. 1, p. 1, 1999.
[45] M. Sugiyama, "Dimensionality reduction of multimodal labeled data by local fisher discriminant analysis," *The Journal of Machine Learning Research*, vol. 8, pp. 1027-1061, 2007.
[46] L. Li and S. Dai, "Action recognition based on local fisher discriminant analysis and mix encoding," in *Proc. of Virtual Reality and Visualization (ICVRV), 2016 International Conference on*, pp. 16-23, 2016. Article (CrossRef Link)
[47] N. Dalal and B. Triggs, "Histograms of oriented gradients for human detection," in *Proc. of IEEE Conference on Computer Vision and Pattern Recognition*, pp. 886-893, 2005. Article (CrossRef Link)
[48] N. Dalal, B. Triggs, and C. Schmid, "Human detection using oriented histograms of flow and appearance," in *Proc. of European Conference on Computer Vision*, ed: Springer, pp. 428-441, 2006. Article (CrossRef Link)
[49] C. Cortes and V. Vapnik, "Support-vector networks," *Machine learning*, vol. 20, pp. 273-297, 1995. Article (CrossRef Link)
[50] C. Deng, H. Deng, X. Liu, and Y. Yuan, "Adaptive multi-bit quantization for hashing," *Neurocomputing*, vol. 151, pp. 319-326, 2015. Article (CrossRef Link)
[51] Z. Li, X. Liu, J. Wu, and H. Su, "Adaptive Binary Quantization for Fast Nearest Neighbor Search," in *ECAI*, pp. 64-72, 2016.
[52] X. Liu, B. Du, C. Deng, M. Liu, and B. Lang, "Structure sensitive hashing with adaptive product quantization," *IEEE transactions on cybernetics*, vol. 46, pp. 2252-2264, 2016. Article (CrossRef Link)
[53] J. Song, H. Zhang, X. Li, L. Gao, M. Wang, and R. Hong, "Self-supervised video hashing with hierarchical binary auto-encoder," *IEEE Transactions on Image Processing*, vol. 27, pp. 3210-3221, 2018. Article (CrossRef Link)
[54] J. Song, L. Gao, L. Liu, X. Zhu, and N. Sebe, "Quantization-based hashing: a general framework for scalable image and video retrieval," *Pattern Recognition*, vol. 75, pp. 175-187, 2018. Article (CrossRef Link)
[55] M. Muja and D. G. Lowe, "Scalable nearest neighbor algorithms for high dimensional data," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 36, pp. 2227-2240, 2014. Article (CrossRef Link)
[56] K. Yu, T. Zhang, and Y. Gong, "Nonlinear learning using local coordinate coding," in *Advances in Neural Information Processing Systems*, pp. 2223-2231, 2009.
[57] O. Russakovsky, J. Deng, H. Su, J. Krause, S. Satheesh, S. Ma, *et al.*, "Imagenet large scale visual recognition challenge," *International Journal of Computer Vision*, vol. 115, pp. 211-252, 2015. Article (CrossRef Link)
[58] Y. Jia, E. Shelhamer, J. Donahue, S. Karayev, J. Long, R. Girshick, *et al.*, "Caffe: Convolutional architecture for fast feature embedding," in *Proc. of the 22nd ACM international conference on Multimedia*, pp. 675-678, 2014. Article (CrossRef Link)
[59] C. Schüldt, I. Laptev, and B. Caputo, "Recognizing human actions: a local SVM approach," in *Proc. of the 17th International Conference on Pattern Recognition*, pp. 32-36, 2004. Article (CrossRef Link)
[60] H. Kuehne, H. Jhuang, E. Garrote, T. Poggio, and T. Serre, "HMDB: a large video database for human motion recognition," in *Proc. of Computer Vision (ICCV), 2011 IEEE International Conference on*, pp. 2556-2563, 2011. Article (CrossRef Link)
[61] K. Soomro, A. R. Zamir, and M. Shah, "Ucf101: A dataset of 101 human actions classes from videos in the wild," *arXiv preprint arXiv:1212.0402*, 2012.
[62] F. Perronnin, J. Sánchez, and T. Mensink, "Improving the fisher kernel for large-scale image classification," *Computer Vision–ECCV 2010*, ed: Springer, pp. 143-156, 2010. Article (CrossRef Link)

[63] J. Liu, J. Luo, and M. Shah, "Recognizing realistic actions from videos “in the wild”," in *Proc. of IEEE Conference on Computer Vision and Pattern Recognition*, pp. 1996-2003, 2009. Article (CrossRef Link)
[64] A. Kovashka and K. Grauman, "Learning a hierarchy of discriminative space-time neighborhood features for human action recognition," in *Proc. of IEEE Conference on Computer Vision and Pattern Recognition*, pp. 2046-2053, 2010. Article (CrossRef Link)
[65] F. Baumann, "Action recognition with HOG-OF features," *Pattern Recognition*, ed: Springer, pp. 243-248, 2013. Article (CrossRef Link)
[66] V. Veeriah, N. Zhuang, and G.-J. Qi, "Differential recurrent neural networks for action recognition," in *Proc. of the IEEE International Conference on Computer Vision*, pp. 4041-4049, 2015. Article (CrossRef Link)
[67] S. Sadanand and J. J. Corso, "Action bank: A high-level representation of activity in video," in *Proc. of Computer Vision and Pattern Recognition (CVPR), 2012 IEEE Conference on*, pp. 1234-1241, 2012. Article (CrossRef Link)
[68] Z. Cai, L. Wang, X. Peng, and Y. Qiao, "Multi-view super vector for action recognition," in *Proc. of the IEEE conference on Computer Vision and Pattern Recognition*, pp. 596-603, 2014. Article (CrossRef Link)
[69] E. Park, X. Han, T. L. Berg, and A. C. Berg, "Combining multiple sources of knowledge in deep cnns for action recognition," in *Proc. of Applications of Computer Vision (WACV), 2016 IEEE Winter Conference on*, pp. 1-8, 2016. Article (CrossRef Link)
[70] B. Fernando, E. Gavves, J. Oramas, A. Ghodrati, and T. Tuytelaars, "Rank pooling for action recognition," *IEEE transactions on pattern analysis and machine intelligence*, vol. 39, pp. 773-787, 2017. Article (CrossRef Link)
[71] D. Oneata, J. Verbeek, and C. Schmid, "Action and event recognition with Fisher vectors on a compact feature set," in *Proc. of Computer Vision (ICCV), 2013 IEEE International Conference on*, pp. 1817-1824, 2013. Article (CrossRef Link)
[72] J. Wu, Y. Zhang, and W. Lin, "Towards good practices for action video encoding," in *Proc. of the IEEE conference on Computer Vision and Pattern Recognition*, pp. 2577-2584, 2014. Article (CrossRef Link)
[73] A. Karpathy, G. Toderici, S. Shetty, T. Leung, R. Sukthankar, and L. Fei-Fei, "Large-scale video classification with convolutional neural networks," in *Proc. of Computer Vision and Pattern Recognition (CVPR), 2014 IEEE Conference on*, pp. 1725-1732, 2014. Article (CrossRef Link)
[74] D. Tran, L. Bourdev, R. Fergus, L. Torresani, and M. Paluri, "Learning spatiotemporal features with 3d convolutional networks," in *Proc. of Computer Vision (ICCV), 2015 IEEE International Conference on*, pp. 4489-4497, 2015. Article (CrossRef Link)
[75] H. Wang, D. Oneata, J. Verbeek, and C. Schmid, "A robust and efficient video representation for action recognition," *International Journal of Computer Vision*, vol. 119, pp. 219-238, 2016. Article (CrossRef Link)
[76] Y. Gong, S. Lazebnik, A. Gordo, and F. Perronnin, "Iterative quantization: A procrustean approach to learning binary codes for large-scale image retrieval," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 35, pp. 2916-2929, 2013. Article (CrossRef Link)

Lijun Li is now a Ph.D candidate of State Key Laboratory of Virtual Reality Technology and Systems in Beihang University. He received his B.Sc. from China University of Geosciences in 2012. His research interests include computer vision and computer graphics.



Shuling Dai is a Professor in the State Key Laboratory of Virtual Reality Technology and Systems at Beihang University. He received his Ph.D from Beihang University in 1997. His research interests include virtual reality, control, computer graphics, etc.