Hindawi
Applied Computational Intelligence and Soft Computing
Volume 2017, Article ID 4019213, 7 pages
https://doi.org/10.1155/2017/4019213

*Research Article*

# Representation for Action Recognition Using Trajectory-Based Low-Level Local Feature and Mid-Level Motion Feature

**Xiaoqiang Li, Dan Wang, and Yin Zhang**

*School of Computer Engineering and Sciences, Shanghai University, Shanghai 200444, China*

Correspondence should be addressed to Xiaoqiang Li; xqli@i.shu.edu.cn

Received 4 August 2016; Revised 22 March 2017; Accepted 17 September 2017; Published 19 October 2017

Academic Editor: Francesco Carlo Morabito

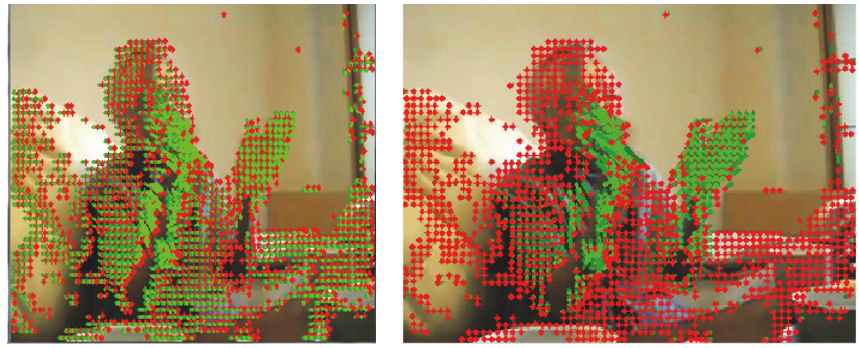
Copyright © 2017 Xiaoqiang Li et al. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

The dense trajectories and low-level local features are widely used in action recognition recently. However, most of these methods ignore the motion part of action which is the key factor to distinguish the different human action. This paper proposes a new two-layer model of representation for action recognition by describing the video with low-level features and mid-level motion part model. Firstly, we encode the compensated flow ($w$-flow) trajectory-based local features with Fisher Vector (FV) to retain the low-level characteristic of motion. Then, the motion parts are extracted by clustering the similar trajectories with spatiotemporal distance between trajectories. Finally the representation for action video is the concatenation of low-level descriptors encoding vector and motion part encoding vector. It is used as input to the LibSVM for action recognition. The experiment results demonstrate the improvements on J-HMDB and YouTube datasets, which obtain 67.4% and 87.6%, respectively.

## 1. Introduction

Human action recognition has become a hot topic in the field of computer vision. It has developed a practical system which will be applied to video surveillance, interactive gaming, and video annotation. Despite remarkable research efforts and many encouraging advances in recent years [1–3], action recognition is still far from being satisfactory and practical. There are large factors affecting accurate rate of the recognition such as cluttered background, illumination, and occlusion.

Most action recognition focuses on two important issues: extracting features within a spatiotemporal volume and modeling the action patterns. Many existing researches on human action recognition tend to extract features from whole 3D videos using spatiotemporal interest points (STIP) [4]. In recent years, optical flow is applied to extract the trajectory-based motion features, which have been widely used in local spatiotemporal features. Local trajectory-based features are pooled and normalized to a vector as the video global representation in action recognition. Meanwhile, a lot of work has focused on developing discriminative dictionary for image object recognition or video action recognition. The Bag of Feature (BOF) model generates simple video model by clustering spatiotemporal features of all the training samples and is trained using $\chi^2$-*kernel* Support Vector Machine (SVM). And the state of the art method is popular Fisher Vector (FV) [5] encoding model based on spatiotemporal local features. However, all these methods are not perfect, because they are only concerned about the low-level spatiotemporal features based on interest point and ignored the higher level features of motion part. For most actions, only a small subset of local motion features of the entire video is relevant to the action label. When a person is waving, only the movement around the arm or hand is responsible for the action clapping hand. Action Bank [6] and motionlets [7] adopt unsupervised learning to discover action parts. Many methods [8] cluster the trajectories and seek to understand spatiotemporal properties of movement to construct the mid-level action video representation. The Vector of Locally Aggregated Descriptors (VLAD) [9] is a descriptor encoding technique that aggregates the descriptors based on a locality criterion in the feature space. To keep more spatiotemporal characteristics of the processed motion part, VLAD encoding gets better results than BOF by [10]. Inspired by low-level local feature encoding and mid-level motion part model are key factors to distinguish the different human actions; we propose a new representation (depicted in Figure 2) for action



(a) iDT vectors (b) $w$-flow DT vectors

Figure 1: A comparison of the iDT trajectories and $w$-flow dense trajectories. The red dot is the end point of the green optical flow vector in the current frame. (a) The optical flow vectors are tracked by the improve dense trajectories using SURF descriptors matching [12]. (b) Most of flow vectors due to camera motion are removed by $w$-flow method.

recognition based on local features and motion part in this paper. To reduce the background clutter noise, we extract the local trajectory-based features through a better compensated flow ($w$-flow) [11] dense trajectories method. Then we cluster the trajectories through the graph clustering algorithm and encode the group features to describe the different motion part. Finally, we represent the video through combining the low-level trajectory-based features encoding model with mid-level motion part model.

This paper is organized as follows. In Section 2, the local descriptors based on the $w$-flow dense trajectories and low-level video encoding with FV are introduced. Then we show clustering the motion part and introduce the representation for video in Section 3. We describe the evaluation of our approach and discuss the results in Section 4. Finally, the conclusion and future works are discussed in Section 5.

## 2. First Layer with FV

Trajectories are efficient in capturing object motions in videos. We extract spatiotemporal features along the $w$-flow dense trajectories to express low-level descriptors. In this section we introduce the $w$-flow dense trajectories and low-level descriptors with FV.

*2.1. w-Flow Dense Trajectory.* The idea of dense trajectory is based on tracking the interest points. The interest points are sampled on a grid spaced by $W = 5$ pixels and tracked in each frame. Points of subsequent frames are concatenated to form a trajectory: $(p_t, \ldots, p_{t+L})$. $p_t = (x_t, y_t)$ is the position of interest points at frame $t$. The length of a trajectory is $L = 15$ frames [1]. A recent work by Jain et al. [11] proposed the compensated flow ($w$-flow) dense trajectories which reduce the impact of the background trajectories. The $w$-flow dense trajectory is obtained by removing the affine flow vector from the original optical flow vector. The interest point of this method is tracked by $w$-flow [11] for compensating dominant motion (camera motion). It is beneficial for most of the existing descriptors used for action recognition. This method uses the 2D polynomial affine motion model for compensating camera motion. The affine flow $w_{\text{aff}}(p_t)$ is the main movement of the two consecutive images which is usually caused by the movement of the camera. We compute the affine flow with the publicly available Motion2D software (http://www.irisa.fr/vista/Motion2D/) which implements a real-time robust multiresolution incremental estimation framework. The final flow vector $w(p_t)$ at point $p_t = (x_t, y_t)$ is obtained by removing the affine flow vector $w_{\text{aff}}(p_t) = (u(p_t), v(p_t))$ from the original optical flow vector as follows.

$$w(p_t) = w(p_t) - w_{\text{aff}}(p_t). \tag{1}$$

Figure 1 shows the dense trajectories extracted by the iDT [12] method and the $w$-flow dense trajectories.

The shape of a trajectory encodes local motion patterns. The shape of a trajectory is described by concatenating a set of displacement vectors $\Delta P_t = (p_{t+1} - p_t) = (x_{t+1} - x_t, y_{t+1} - y_t)$. Meanwhile, to leverage the motion information in dense trajectories, we compute descriptors within a spatiotemporal volume around the trajectory. The size of a volume is $32 \times 32$. And the volume is divided into a $2 \times 2 \times 3$ spatiotemporal grid. The Histograms of Optical Flow (HOF and $w$-HOF) [1] descriptor captures the local motion information which is computed using the orientations and magnitudes of the flow field. Motion boundary histogram (MBH) [1] descriptor encodes the relative motion between pixels which is along both $x$ and $y$ image axis and describes the discriminatory features for the action recognition in background cluttering. The trajectory-based $w$-HOF features is computed on the compensated flow. For each trajectory, the descriptors combine motion information HOF, $w$-HOF, and MBH. The single trajectory feature is in the form of

$$F = \left(w\text{-HOF}, \text{HOF}, \text{MBH}_x, \text{MBH}_y, S\right). \tag{2}$$

The trajectory shape $S = (\Delta P_t, \ldots, \Delta P_{t+L-1}) / \sum_{j=t}^{t+L-1} \|\Delta P_j\|$ is normalized by the sum of the magnitudes of the displacement vectors and $L$ is the length of trajectories.

*2.2. Low-Level Video Encoding.* The representation of video is a vital problem in action recognition. We first encode the low-level $w$-flow trajectory-based descriptors using the Fisher Vector (FV) encoding method which was proposed for image

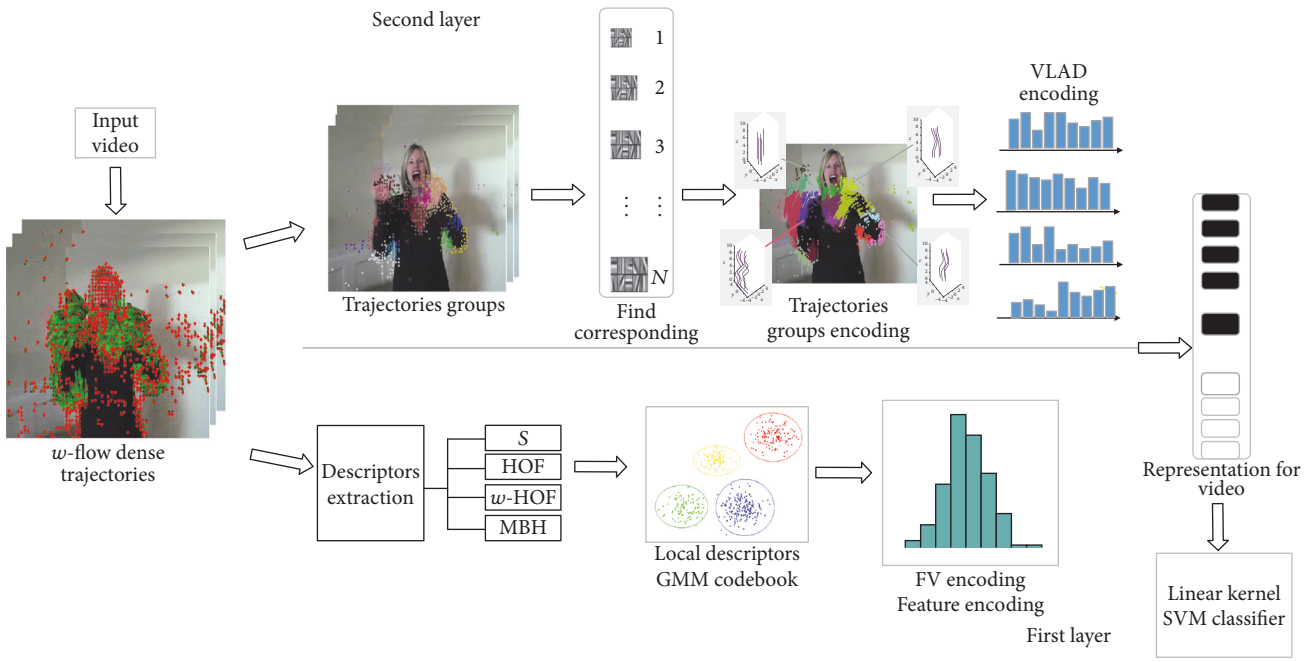



FIGURE 2: The recognition framework of the two-layer model. The first layer encodes the low-level $w$-flow trajectory-based descriptors using the Fisher Vector (FV). The second layer describes motion part with trajectories groups.

categorization [13]. FV is derived from Fisher kernel which encodes the statistics between video descriptors and Gaussian Mixture Model (GMM). We reduce the low-level features ($w$-HOF, HOF, and MBH) dimensionality by PCA keeping the 90% energy. The local descriptors $X$ can be modeled by a probability density function $p(X, \theta)$ with parameters $\theta$, which is usually modeled by GMM.

$$G_\theta^X = \frac{1}{N}\nabla_\theta \log p\,(X;\theta)\,, \qquad \theta = w_1, \mu_1, \delta_1, \ldots, w_k, \mu_k, \delta_k, \tag{3}$$

where $w, \mu, \delta$ are the model parameters denoting the weights, means, and diagonal covariances of GMM. $N$ is the number of local descriptors. $k$ is the number of mixture components and we set $k$ to 256 [5]. We can compute the gradient of the log likelihood with respect to the parameters of the model to represent a video. FV requires a GMM of the encoded feature distribution. The Fisher Vector is the concatenation of these partial derivatives and describes in which direction the parameters of the model should be modified to best fit the data [14]. To keep the low-level feature, we encode each video with the FV encoding feature.

## 3. Representation for Video

Motion part encoding has already been identified as a successful method to represent the video for action recognition. In this section, we use a graph clustering method to cluster the similarity trajectories into groups. Then representation for action video is concatenation of low-level local descriptors encoding and high-level motion part encoding.

*3.1. Trajectories Group.* To better describe the motion, we cluster the similarity trajectories into groups, because critical regions of the video are relevant to a specific action. In the method of [22], they compute a hierarchical clustering on trajectories to yield trajectories group of action parts. Then we apply that efficient greedy agglomerative hierarchical clustering procedure to group the trajectories. There are a large number of trajectories in a video; that is, there are a large number of nodes in graph. By removing trajectories distance which is not spatially close will get a sparse trajectories graph. Greedy agglomerative hierarchical clustering is a fast, scalable algorithm, with almost linear complexity in the number of nodes for relatively sparse trajectories' graph. To group the trajectories we set $N \times N$ trajectories distance matrix for a video containing $N$ trajectories. We use a distance metric between trajectories taking into consideration their spatial and temporal relations to cluster. Given two trajectories $P_a(t)_{t=t_a}^{T_a}$ and $P_b(t)_{t=t_b}^{T_b}$,

$$d\,(a,b) = \max d_s\,(t) \cdot \frac{1}{T_2 - T_1}\sum_{t=T_1}^{T_2} d_{\text{vel}}\,(t)\,, \qquad t \in [T_1, T_2]\,, \tag{4}$$

$$d_s\,(t) = \left|P_a\,(t) - P_b\,(t)\right|_2\,,$$
$$d_{\text{vel}} = \left|\Delta P_a\,(t) - \Delta P_b\,(t)\right|_2\,,$$

where $d_s$ and $d_{\text{vel}}$ are the $L2$ distances of the trajectory points at corresponding time instances. We just calculate the distance between the trajectories $P_a$ and $P_b$ simultaneously existing in $[T_1, T_2]$. To ensure the spatial compactness of the estimated groups, we enforce the above affinity to be zero for trajectory pairs that are not spatially close $d_s \geq 30$. The number of clusters in a video is set as the number used in [22] and the number of trajectories in a cluster is below the 100 based on empirical value.

*3.2. Second Layer with VLAD.* The trajectory group describing the motion part in the same action categories will have similarities. To capture the coarser spatiotemporal characteristics of the descriptors in the group $k$, we compute the mean of group descriptors ($w$-HOF, HOF, and MBH) and trajectory shapes. Then, we concatenate all the group descriptors ($w$-HOF, HOF, and MBH) as $G_r$ and group shape as the group descriptors $G_g$. So the group is described as $G = \{G_r, G_g\}$; VLAD [9] is a descriptor encoding technique that aggregates the descriptors based on a locality criterion in the feature space. As we know, the classic BOF uses the clustering centers statistics to represent the sample which will result in the loss of the lots of information. In group encoding, we denote the code words in the group codebook as $c_1, c_2, \ldots, c_k$. The group descriptors $G_i^k = \{G_r, G_g\}$ are all the group descriptors that belong to the $k$th word. The video will be encoded as a vector:

$$v = \left( \sum_{i=1}^{n_1} \left( G_i^1 - c_1 \right), \ldots, \sum_{i=1}^{n_k} \left( G_i^k - c_k \right) \right), \tag{5}$$

where $k$ is the size of codebook learned by the $k$-means clustering. So the VLAD keeps more information than the BOF.

*3.3. Video Encoding.* We encode each video from the group descriptors of motion part using VLAD model. The codebook for each kind of group descriptors ($w$-HOF, HOF, MBH, and $S$) was separately constructed by using $K$-means cluster. According to the average number of groups in every video, we set the number of visual words to 50. In order to find the nearest center quickly we construct a KD-tree when each group descriptors are mapped to the codebook. We describe video encoding vector with the group model for different descriptors. Then, motion part model is encoded by the concatenation of different descriptors of the group VLAD model. Finally, the representation for action recognition is encoded by the concatenation of low-level local descriptors encoding and mid-level motion part encoding. Figure 2 shows an overview of our pipeline for action recognition.

## 4. Experiments

In this section, we implement some experiments to evaluate the performance of representation for action. We validate our model on several action recognition benchmarks and compare our results with different methods.

*4.1. Datasets.* We validate our model on three standard datasets for human action: KTH, J-HMDB, and YouTube dataset. The KTH dataset views actions in front of a uniform background, whereas the J-HMDB dataset [10] and YouTube dataset [16] are collected from a variety of sources ranging from digitized movies to YouTube. They cover different scales and difficulty levels for action recognition. We summarize them and the experimental protocols as follows.

The KTH dataset [23] contains 6 action categories: walking, handclapping, hand waving, jogging, running, and walking. The background is homogeneous and static in most sequences. We follow the experimental setting [23] dividing the dataset into the train set and test set. We train a multiclass classifier and report average accuracy over all classes as performance measure.

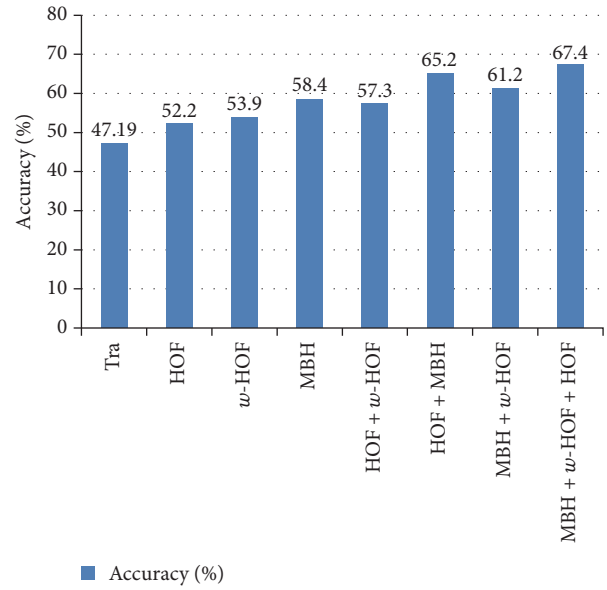


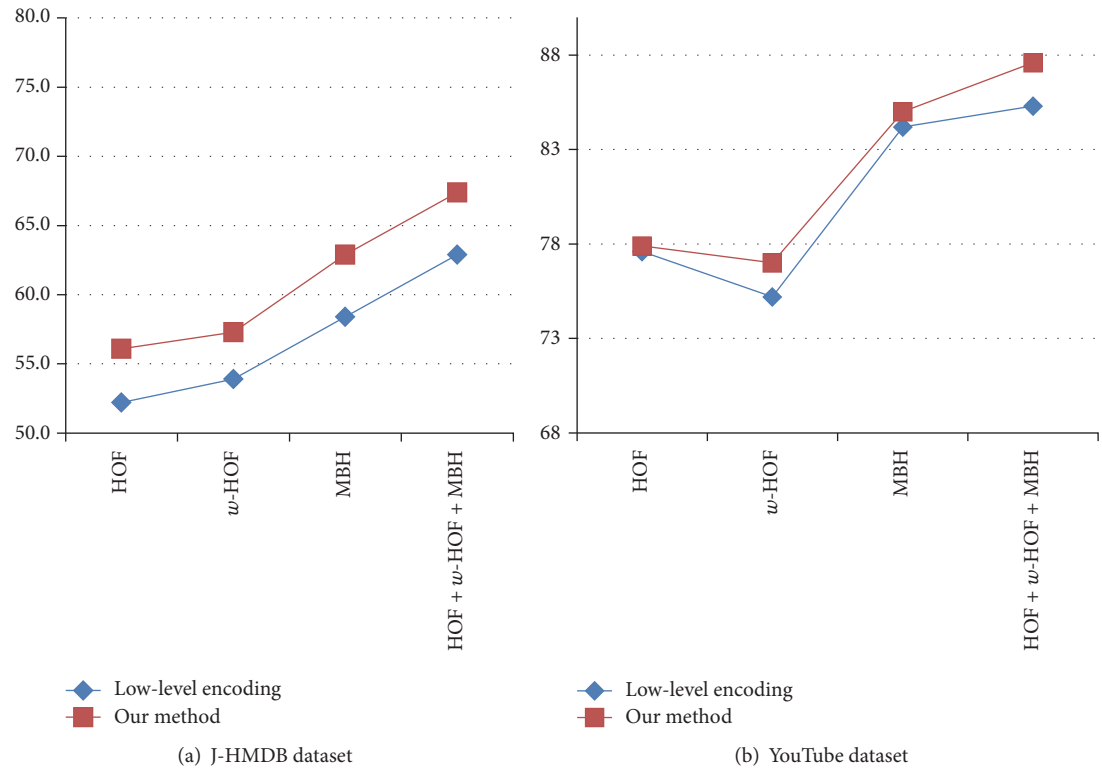

Figure 3: Illustration on the effect of our descriptors with FV encoding. Each bar corresponds to one of the feature descriptors or feature combinations.

The J-HMDB [10] contains 21 action categories: brush hair, catch, clap, climb stairs, golf, jump, kick ball, pick, pour, pull-up, push, run, shoot ball, shoot bow, shoot gun, sit, stand, swing baseball, throw, walk, and wave. J-HMDB is a subset of the HMDB51 which is collected from the movies or Internet. This dataset excludes categories from HMDB51 that contain facial expressions like smiling and interactions with others such as shaking hands and focuses on single body action. We evaluate the J-HMDB which contains 11 categories involving one single body action. For multiclass classification, we use the one-vs-rest approach.

The YouTube Action dataset [16] contains 11 action categories: basketball, biking, diving, golf swinging, horse riding, soccer juggling, swinging, tennis swinging, trampoline jumping, volleyball spiking, and walking with a dog. Because of the large variations in camera motion, appearance, and pose, it is a challenging dataset. Following [16], we use leave-one-group-out cross-validation and report the average accuracy over all classes.

*4.2. Experiment Result.* The proposed method extract one-scale $w$-flow trajectory-based local features through tracking dense sampling interest points and, then, cluster the trajectories into groups to encode motion part.

In order to choose a discriminative combination of features to represent the low-level local descriptors, we evaluate the low-level local descriptors based on $w$-flow dense trajectories with Fisher Vector encoding in the first baseline experiment. GMM with 256 components is learned from a subset of 256,000 randomly selected trajectory-based local descriptors. Linear SVM with $c = 100$ is used as classifier. We compare different feature descriptors in Figure 3 where




(a) J-HMDB dataset (b) YouTube dataset

Figure 4: The accuracy of different features comparisons between low-level and two-layer model. (a) The comparison on J-HMDB dataset. (b) The comparison on YouTube dataset.

Table 1: The accuracy comparisons of representation for action recognition on J-HMDB dataset and YouTube dataset.

| Datasets | Features | Low-level encoding | Two-layer model |
|---|---|---|---|
| JHMDB | HOF | 52.2% | 56.1% |
| | $w$-HOF | 53.9% | 57.3% |
| | MBH | 58.4% | 62.9% |
| | HOF + $w$-HOF + MBH | 62.9% | 67.4% |
| YouTube | HOF | 77.6% | 77.9% |
| | $w$-HOF | 75.2% | 77.0% |
| | MBH | 84.2% | 85.0% |
| | HOF + $w$-HOF + MBH | 85.3% | 87.6% |

the average accuracy on J-HMDB dataset is reported. It can be seen that MBH descriptors, encoding the relative motion between pixels, work better than other descriptors. Figure 3 also shows that the combination of HOF, $w$-HOF, and MBH descriptors achieves 67.4%, which is the highest precision among all kinds of the low-level local descriptors. So, we use this combination in the second experiment.

In the second baseline experiment, the proposed two-layer model of the representation for action is the concatenation of low-level local descriptors and motion part descriptors encoding. Table 1 and Figure 4 compare the two-layer method with the low-level method for J-HMDB and YouTube datasets. It can be seen that the two-layer model had better performance than the low-level encoding using different descriptors. In addition, we compare the proposed method with a few classic methods on KTH, J-HMDB, and YouTube datasets, such as DT + BoVW [1], mid-level parts [21], traditional FV [17], stacked FV [17], DT + BOW [10], and IDT + FV [17]. As shown in Table 2, the two-layer model obtains 67.4% and 87.6% accuracy on J-HMDB and YouTube datasets, respectively. And the recognition accuracy is improved by 4.6% on J-HMDB dataset and 2.2% on YouTube dataset compared with other state of the art methods. However, the performance on KTH dataset of the proposed method is not the same better as on the J-HMDB and YouTube datasets, because the KTH dataset is collected by the fixed camera with homogeneous background and the advantage of the $w$-flow trajectories is not shown in this case.

## 5. Conclusions

This paper proposed a two-layer model of representation for action recognition based on local descriptors and motion part descriptors, which achieved an improvement compared to the low-level local descriptors. Not only did it consider making use of low-level local information to encoding the video, but also it combined the motion part to represent the video. It also presented a discriminative and compact representation for action recognition. However, there is still room for improvement. First, the proposed method cannot determine the number of groups in different datasets while the number of groups affects the performance of mid-level encoding a

Table 2: Accuracy comparisons of different methods on KTH, YouTube and J-HMDB datasets.

| KTH | | YouTube | | J-HMDB | |
|---|---|---|---|---|---|
| ISA [15] | 86.5% | Liu et al. [16] | 71.2% | Traditional FV [17] | 62.83% |
| Yeffet and Wolf [18] | 90.1% | Ikizler-Cinbis and Sclaroff [19] | 75.21% | Stacked FV [17] | 59.27% |
| Cheng et al. [20] | 89.7% | DT + BoVW [1] | 85.4% | DT + BOW [10] | 56.6% |
| Le et al. [15] | 93.9% | Mid-level parts [21] | 84.5% | IDT + FV [17] | 62.8% |
| Two-layer model | 92.6% | Two-layer model | 87.6% | Two-layer model | 67.4% |

lot. Second, many groups in video do not represent the action part; it is needed to develop a method to learn the discriminately groups for better representation of the video. In the future, we will do research on new group clustering method which can find the more discriminative groups of action part.

## Conflicts of Interest

The authors declare that there are no conflicts of interest regarding the publication of this paper.

## References


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

[2] Y. Wang, B. Wang, Y. Yu, Q. Dai, and Z. Tu, "Action-Gons: Action recognition with a discriminative dictionary of structured elements with varying granularity," *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics): Preface*, vol. 9007, pp. 259–274, 2015.

[3] I. Laptev, M. Marszałek, C. Schmid, and B. Rozenfeld, "Learning realistic human actions from movies," in *Proceedings of the 26th IEEE Conference on Computer Vision and Pattern Recognition (CVPR '08)*, June 2008.

[4] I. Laptev, "On space-time interest points," *International Journal of Computer Vision*, vol. 64, no. 2-3, pp. 107–123, 2005.

[5] J. Wu, Y. Zhang, and W. Lin, "Towards good practices for action video encoding," in *Proceedings of the 27th IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2014*, pp. 2577–2584, Columbus, OH, USA, June 2014.

[6] S. Sadanand and J. J. Corso, "Action bank: a high-level representation of activity in video," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR '12)*, pp. 1234–1241, June 2012.

[7] L. M. Wang, Y. Qiao, and X. Tang, "Motionlets: Mid-level 3D parts for human motion recognition," in *Proceedings of the 26th IEEE Conference on Computer Vision and Pattern Recognition (CVPR '13)*, pp. 2674–2681, June 2013.

[8] W. Chen and J. J. Corso, "Action detection by implicit intentional motion clustering," in *Proceedings of the 15th IEEE International Conference on Computer Vision, ICCV 2015*, pp. 3298–3306, chl, December 2015.

[9] H. Jégou, F. Perronnin, M. Douze, J. Sánchez, P. Pérez, and C. Schmid, "Aggregating local image descriptors into compact codes," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 34, no. 9, pp. 1704–1716, 2012.

[10] H. Jhuang, J. Gall, S. Zuffi, C. Schmid, and M. J. Black, "Towards understanding action recognition," in *Proceedings of the 2013 14th IEEE International Conference on Computer Vision, ICCV 2013*, pp. 3192–3199, aus, December 2013.

[11] M. Jain, H. Jegou, and P. Bouthemy, "Better exploiting motion for better action recognition," in *Proceedings of the 26th IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2013*, pp. 2555–2562, Portland, OR, USA, June 2013.

[12] H. Wang and C. Schmid, "Action recognition with improved trajectories," in *Proceedings of the 14th IEEE International Conference on Computer Vision (ICCV '13)*, pp. 3551–3558, Sydney, Australia, December 2013.

[13] F. Perronnin, J. Sánchez, and T. Mensink, "Improving the fisher kernel for large-scale image classification," in *Proceedings of the 11th European Conference on Computer Vision (ECCV '10)*, vol. 6314 of *Lecture Notes in Computer Science*, pp. 143–156, Crete, Greece, 2010.

[14] G. Csurka and F. Perronnin, "Fisher vectors: Beyond bag-of-visual-words image representations," *Communications in Computer and Information Science*, vol. 229, pp. 28–42, 2011.

[15] Q. V. Le, W. Y. Zou, S. Y. 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 (CVPR '11)*, pp. 3361–3368, June 2011.

[16] J. Liu, J. Luo, and M. Shah, "Recognizing realistic actions from videos in the wild," in *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR '09)*, pp. 1996–2003, IEEE, Miami, Fla, USA, June 2009.

[17] X. Peng, C. Zou, Y. Qiao, and Q. Peng, "Action recognition with stacked fisher vectors," in *Computer Vision—ECCV 2014: 13th European Conference, Zurich, Switzerland, September 6–12, 2014, Proceedings, Part V*, vol. 8693 of *Lecture Notes in Computer Science*, pp. 581–595, Springer, Berlin, Germany, 2014.

[18] L. Yeffet and L. Wolf, "Local trinary patterns for human action recognition," in *Proceedings of the 12th International Conference on Computer Vision (ICCV '09)*, pp. 492–497, Kyoto, Japan, October 2009.

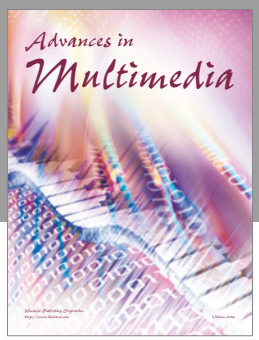
[19] N. Ikizler-Cinbis and S. Sclaroff, "Object, scene and actions: Combining multiple features for human action recognition," *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics): Preface*, vol. 6311, no. 1, pp. 494–507, 2010.

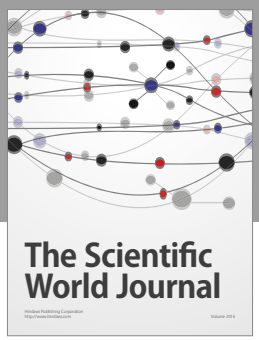
[20] G. Cheng, Y. Wan, W. Santiteerakul, S. Tang, and B. P. Buckles, "Action recognition with temporal relationships," in *Proceedings of the 2013 IEEE Conference on Computer Vision and Pattern Recognition Workshops, CVPRW 2013*, pp. 671–675, Portland, OR, USA, June 2013.

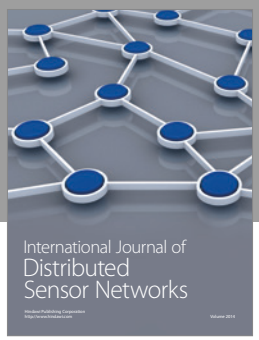
[21] M. Sapienza, F. Cuzzolin, and P. H. S. Torr, "Learning discriminative space-time action parts from weakly labelled videos," *International Journal of Computer Vision*, vol. 110, no. 1, pp. 30–47, 2014.

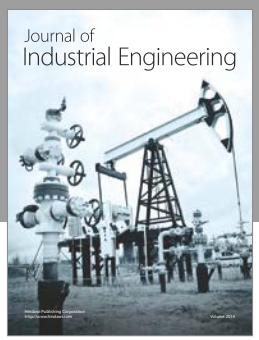
[22] M. Raptis, I. Kokkinos, and S. Soatto, "Discovering discriminative action parts from mid-level video representations," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR '12)*, June 2012.

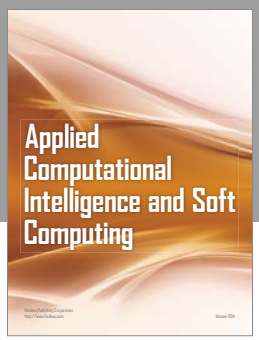
[23] C. Schüldt, I. Laptev, and B. Caputo, "Recognizing human actions: a local SVM approach," in *Proceedings of the 17th International Conference on Pattern Recognition (ICPR '04)*, pp. 32–36, August 2004.

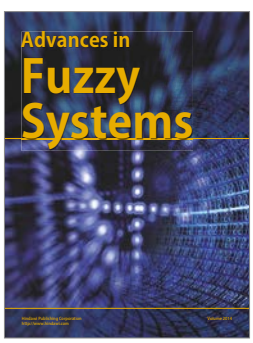


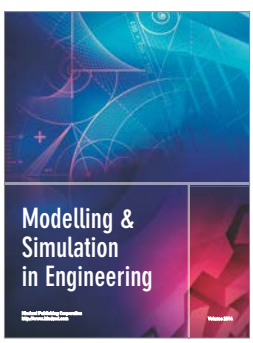


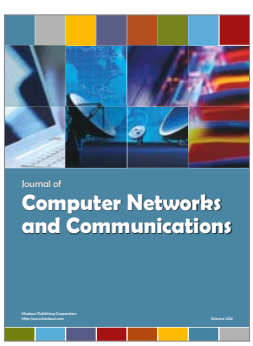












Hindawi
Submit your manuscripts at
https://www.hindawi.com