Received: 17 June 2019 | Accepted: 31 July 2019
DOI: 10.1002/cpe.5507

WILEY

SPECIAL ISSUE PAPER

Real-time action feature extraction via fast PCA-Flow

Huafeng Chen[1] | Hongyang Li[2] | Zengmin Xu[3] | Yunhong Zhao[1] | Tigang He[1]

[1]School of Computer Engineering, Jingchu University of Technology, Jingmen, China
[2]College of Computer and Information Technology, China Three Gorges University, Yichang, China
[3]School of Mathematics and Computing Science, Guilin University of Electronic Technology, Guilin, China

Correspondence
Huafeng Chen, School of Computer Engineering, Jingchu University of Technology, Jingmen 448000, China.
Email: chenhuafeng@jcut.edu.cn

Funding information
National Natural Science Foundation of China, Grant/Award Number: 61773295 and 61862015

Summary

Action recognition is a research hotspot in the field of Internet of Things (IoT). Currently, local pixel-domain spatiotemporal feature extraction methods have reached the state-of-the-art action recognition performance on many challenging datasets. However, the poor computational complexity of these approaches prevents them from scaling up to real-time applications. For solving this problem, we present a novel real-time video feature extraction technique by exploiting the fast PCA-Flow algorithm. Firstly, we down-sample video images in form of grid. Based on the down-sampling images, PCA-Flow algorithm is used to calculate optical flow among adjacent images. The PCA-Flow matrices are then expanded to the original video image size by using efficient gCLSR super-resolution method to keep the inherent geometric structure of the optical flow. Finally, we compute action descriptors based on original pixel frames and the enlarged PCA-Flow images. The proposed approach is validated on three challenging datasets: UCF50, Hollywood2, and HMDB51. Experimental results indicate that the proposed method is more efficient in computation and can achieve competitive quality than the state-of-the-art methods.

KEYWORDS

action recognition, gCLSR, Internet of Things, PCA-Flow, real-time feature extraction

1 | INTRODUCTION

The Internet of Things (IoT) is becoming more deeply embedded in our lives at home and at work. It uses sensing technology and intelligent system to perceive physical world, realize interaction, and seamless link among people and objects. Driven by big data and artificial intelligence, IoT continues its good development momentum and provides new impetus for sustained and steady economic growth. Action recognition, as an important research area of IoT, has a wide application prospect in daily scenes such as automatic driving, video surveillance, etc. Much works have been recently devoted to action recognition, and among them, local spatiotemporal features are shown to be successful on a variety of challenging action recognition datasets.[1-3]

Two major groups of local spatiotemporal feature extraction methods, distinguished by the domain in which they operate, are pixel-domain[4-6] and compressed-domain approaches.[7-9] Pixel-domain algorithms generally offer good action recognition accuracy, but they also have the problem of high computational cost, which prevents their usage in real-time scenes. Contrastingly, the approaches of feature extraction in compressed-domain are able to significantly reduce computing time consumption by utilizing encoded motion vectors (MVs) information. Their downside is that they are greatly affected by the Bi-directional predictive frame (B-frame) because the backward predicted information in B-frame is incapable of reflecting the speed and direction of real object movement.

In this paper, we present a novel real-time pixel-based video feature extraction approach for real-time human action recognition. Its main motivation is to achieve the real-time purpose through acceleration of time-consuming optical flow calculation. In the first step, we down-sampling the video frames in the form of grid to reduce the computing time consumption of subsequent optical flow calculation. Based on the down-sampling images, we calculate the optical flow between adjacent frames through the PCA-Flow algorithm,[10] which is currently the fastest optical flow algorithm. The PCA-Flow matrices are expended to the original video image size by using efficient gCLSR[11] super-resolution approach because gCLSR can preserve the intrinsic geometric structure of the optical flow. At last, we calculate HOG, HOF, and MBH features based on original pixel frames and its enlarged PCA-Flow images. Experimental results on UCF50,[12] HMDB51,[13] and Hollywood2[14] benchmarks

show that our method can reach competitive action accuracy as state-of-the-art methods and our approach is more efficient in computation than some state-of-the-art methods.

The rest of the paper is organized as follows. The reviewing related work are detailed in Section 2, and the proposed method is presented in Section 3. We show the experimental results in Section 4 and presented our conclusions in Section 5.

2 | RELATED WORK

In this section, we summarize the works on video local spatiotemporal action feature extraction. These methods can be classified into two categories: pixel-domain approaches and compressed-domain approaches.

2.1 | Pixel-domain feature extraction

The pixel-domain feature extraction method refers the video action features are extract from successive pixel images. Laptev[15] introduced the Harris detector into video area and proposed the spatiotemporal interest points (STIP) algorithm. More detailedly, the approach adds time constraints to Harris corners and detects the extreme points that change very strongly in time and space as local feature points. Scovanner et al[16] have proposed the 3-D version of the SIFT descriptor (3D SIFT). The algorithm maps the gradient and direction of each pixel in the neighborhood of the SIFT feature point to a M×M×M voxel unit with Gaussian weight and counts the spatiotemporal gradient values of each solid unit to form a one-dimensional vector. The corresponding vectors are normalized and stitched together to form a 3D SIFT feature descriptor. Willems et al[17] extend the image SURF descriptor to the video domain by computing weighted sums of uniformly sampled responses of spatiotemporal Haar wavelets. Klaser et al[18] noted that the 3D SIFT[16] raises the HOG3D feature descriptor because the singularity of the extreme points causes the histogram to become significantly smaller. HOG3D uses a 12-faceted representation of the directional angle feature calculation, and the direction within the neighborhood maps to the nearest side. Yeffet and Wolf[19] extended the LBP algorithm into the field of 3D action recognition through combining the appearance scale invariance and adaptability of image block matching algorithm and constructed the Local Trinary Patterns (LTP) action feature descriptor. The algorithm has good real-time performance and is suitable for online action recognition scenes. Wang et al[4] propose Dense Trajectories for action recognition. Among the local spatiotemporal features, DT[4] have been shown to perform the best on a variety of challenging datasets.

While their high action recognition accuracy, these pixel-domain methods are time-consuming algorithms. For example, DT[4] features run up to the highest action recognition accuracy at the speed of 1.1 fps on 640×480 pixels video images. The analysis in DT[4] shows that most of the time consumption (61%) is spent on the optical flow calculation. To counteract this problem, we alleviate the expense of optical flow calculation by exploiting the fastest PCA-Flow algorithm. Experimental outcomes indicate that our approach advances the speed of video feature extraction by around 40 times with a loss of around 4% in recognition accuracy compared with DT.

Yu et al[20] proposed a fast video (V-FAST) space-time salient point detection method, which uses the corner detection algorithm to perform fast corner detection[21] on the XY, XT, and YT planes, respectively, and obtains relatively dense spatiotemporal feature points. Shi et al[22] also increased the action feature extraction speed through random feature sampling and analyzed the effect of random sampling over dense grid for computational efficiency. We quantitatively compared our approach with the works of Yu et al[20] and Shi et al[22] and verified the improvements of the proposed method in both the speed and accuracy.

2.2 | Compressed-domain descriptor generation

To handle the high computing time consumption of local pixel-domain spatiotemporal descriptors, many researchers consider using compressed domain information of video in the action recognition process. Tom et al[23] has taken advantage of quantization parameters and motion vectors and extracted action feature from the compressed video sequence. They split action classes through Support Vector Machines (SVM). This approach is robust to illumination changes, scale, and appearance variations. Mu et al[8] proposed a fast suspicious action recognition method based on MV, which the characteristics of suspicious activities and the ways of obtaining motion vectors directly from video stream. Mu et al normalized the motion vectors by taking reference frames into account and extracted the action feature that display the interframe and intraframe information of the region of interest. Babu et al[9] proposed a H.264/AVC compressed domain action recognition system with projection-based metacognitive learning classifier (PBL-McRBFN). In the method of PBL-McRBFN, the features are extracted from the quantization parameters and the motion vectors of the compressed video stream for a time window.

Kantorov and Laptev[7] improved the DT by making the optical flow as substitution the motion vectors. MF[7] works well when there are only Intra coded frames (I-frames) and Predicted frames (P-frames) in the video compressed bitstream, as the motion vectors in P-frame are forward predicted information and roughly reflect the changes of object movement.[24] Once the video compressed bitstream abound with B-frames, the action recognition rate of MF decreases sharply (as shown in Section 4.3). The reason lies in the backward predicted information in B-frame, which cannot reflect the speed and direction of real object movement. The recognition accuracy rate of the our approach roughly equal MF (with no B-frame in video streams) and will not be affected by video encoding scheme.

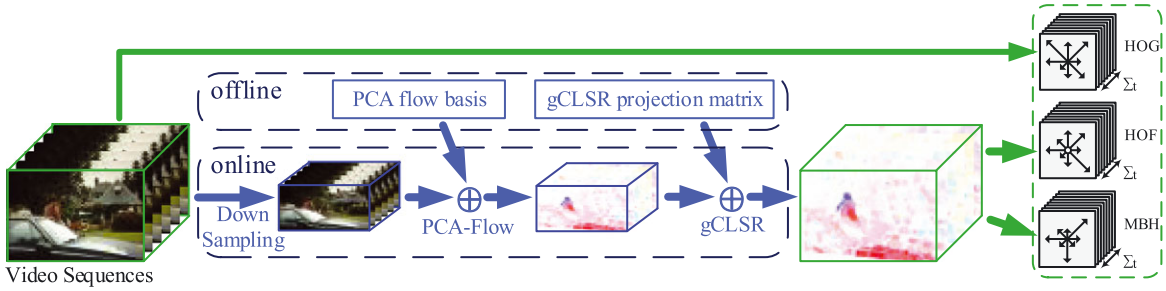

FIGURE 1 Illustrating the pipeline of proposed real-time video action feature extraction

3 | FEATURE EXTRACTION VIA PCA-FLOW

Figure 1 demonstrates the pipeline of our real-time feature extraction approach. After down-sampling the video images, we compute optical flow between adjacent down-sampling frames by using PCA-Flow.[10] Then, the PCA-Flow matrices are reconstructed to the original video image size by using gCLSR[11] super-resolution approach. We can save much online running time in the proposed framework because the PCA-Flow basis and the gCLSR projection matrix are offline computed upon massive video data. Based on the raw video images and enlarged PCA-Flow images, we compute HOG, HOF, and MBH descriptors by following the work of Kantorov and Laptev.[7]

3.1 | Down-sampling of images

Although the algorithm of PCA-Flow[10] is the fastest optical flow and itself has achieved the speed of 85 fps on 640×480 pixels video images, the whole feature extraction process (about 13 fps on 640×480 pixels video images) still does not reach the real-time requirement if we simply apply PCA-Flow to origin video images and extract video features by following the work of Kantorov and Laptev.[7]

We solve this problem and lower the original size of video images before PCA-Flow calculation to further reduce the computation complexity of optical flow. We adopt bilinear interpolation for the down-sampling process so that the smoothness of the images can be maintained.

3.2 | PCA-Flow of images

Based on sampling images, we compute PCA-Flow between adjacent frames by following the steps in the work of Wulff and Black.[10] The basic assumption of PCA-Flow is that optical flow territories can be fitted as a weighted sum over a small number of basic flow territories $\mathbf{b}_n, n = 1...N$, with homologous weights w_n

$$\mathbf{u} \approx \sum_{n=1}^{N} w_n \mathbf{b}_n, \tag{1}$$

where $\mathbf{u}$ and $\mathbf{b}_n$ are vectorized optical flow territories that contain horizontal and vertical motions and are stacked as column vectors $\mathbf{u} = (\mathbf{u}_x{}^T, \mathbf{u}_y{}^T)$.

We computed GPUFlow offline[25] on four Hollywood movies to learn the basis flow fields, and construct 8-hour flow data. Then, the basis that spans optical flow are computed through a robust PCA method,[26] and the first 250 eigenvectors are taken as the basic descriptors in both horizontal and vertical directions $\mathbf{b}_n$.

Given two adjacent sampling frames and the learned flow basic $\mathbf{b}_n$, we calculate the coefficients which define the optical flow.[10] Firstly, K sparse feature matches $(\{(p_k, q_k)\}, k = 1, 2, \ldots, K)$ across neighboring frames are computed by using the fast image matching method.[27] Each of these corresponding feature point induces a displacement vector $\mathbf{v}_k = q_k - p_k = (v_{k,x}, v_{k,y})^T$. Based on the matched points $(\{(p_k, q_k)\}, k = 1, 2, \ldots, K)$ and the basis vectors $\mathbf{b}_n$, the coefficients can be formulated as

$$\hat{\mathbf{w}} = \arg\min_{\mathbf{w}} \rho\left(\|\mathbf{A}\mathbf{w} - \mathbf{y}\|_2^2\right) + \lambda\|\mathbf{\Gamma}\mathbf{w}\|_2 \tag{2}$$

with

$$\mathbf{A} = \begin{pmatrix} b_{1,x(P_1)} & \cdots & b_{N,x(P_1)} \\ \vdots & \ddots & \vdots \\ b_{1,x(P_K)} & \ddots & b_{N,x(P_K)} \\ b_{1,y(P_1)} & \ddots & b_{N,y(P_1)} \\ \vdots & \ddots & \vdots \\ b_{1,y(P_K)} & \cdots & b_{N,y(P_K)} \end{pmatrix} \tag{3}$$

$$\rho(x^2) = \frac{\sigma^2}{2} \log\left[1 + \left(\frac{x}{\sigma}\right)^2\right], \tag{4}$$

where $\mathbf{y} = (v_{1,x}, v_{2,x}, \ldots, v_{K,x}, v_{1,y}, v_{2,y}, \ldots, v_{K,y})^T$ contains the motion of the matched points between two adjacent sampling frames. The function $\rho(\cdot)$ is the robust Cauchy function to be used for tolerating the outliers of matched points, and the parameter σ control the sensitivity to outliers. $\mathbf{\Gamma}$ is an inverse covariance matrix of the coefficients which computed by projecting the ground truth flow fields of KITTI[28] and MPI-Sintel[29] onto the flow basis $\mathbf{b}_n$.

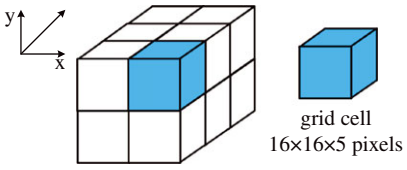

FIGURE 2 Video descriptor. Each $2 \times 2 \times 3$ descriptor grid is summed from quantized values[7]

We solve Equation (2) using Iterative Reweighted Least Squares and compute the PCA-Flow vectors **u** between adjacent sampling images.

3.3 | gCLSR super-resolution of PCA Flow images

In order to preserve the intrinsic geometric structure of PCA-Flow images, we adopt gCLSR super-resolution method to expand the PCA-Flow images to the original size of video images. Similar to the PCA-Flow algorithm, gCLSR[11] also obtains the fastest online running speed in super-resolution approaches by exploiting offline training of high resolution image patch manifold model.

In the first step, we adopt the offline training method in gCLSR[11] to calculate the gCLSR projection matrix $\mathbf{P}^*$ based on the learned flow basis $\mathbf{b}_n$ (and its bilinear down-sampling flow), which is offline trained in Section 3.2. The projection matrix $\mathbf{P}^*$ reflects the geometric structure relationship between the original optical flow image and its down-sampling image.

In the process of online feature extraction, on the strength of PCA-Flow vectors **u** and the gCLSR projection matrix $\mathbf{P}^*$, we use the online testing approach in gCLSR[11] to amplify the PCA-Flow to the original size of video.

3.4 | Video descriptor

By following the design scheme of local spatiotemporal descriptors,[7] we construct the action descriptor by histograms of the super-resolution PCA-Flow data **u** in the video block. Each block is split into grids as illustrated in Figure 2. We normalize the histograms of the block grids and concatenate them into an action feature vector.

The histograms of PCA-Flow are dispersed into eight orientation bins and a no-motion bin to construct HOF descriptors.[30] Then, we disperse the spatial gradients of the v_x and v_y vectors of the PCA-Flow into nine orientation bins and form the bins as the MBHx and MBHy descriptors. We concatenate the all bins of each grid of the $2 \times 2 \times 3$ descriptor grid, normalize the bins in every temporal slice with L_2-normalization, and take the normalized bins final action MBHx and MBHy feature. We compute the HOG descriptors the same sparse set of points. The final dimension of action descriptor vectors is 396.[30]

We compute a $32 \times 32 \times 15$ pixel action feature through the above scheme at each position of the video image by a spatial stride of 16 pixels and temporal step of 5 frames. Following by the method in the work of Kantorov and Laptev,[7] we also construct a $48 \times 48 \times 15$ picture element action feature extracted with the step of 24 pixels for multiple spatial scales sampling.

4 | EXPERIMENTS

The introduction of experimental datasets are enumerated in Section 4.1, and the experimental settings are presented in Section 4.2. Then, we evaluate the negative effect of B-frame upon MF in Section 4.3. In Section 4.4, we compare the proposed method to the traditional approaches and evaluate the impact of down-sampling interval upon proposed method.

4.1 | Experimental datasets

In this section, we briefly enumerate three datasets (Hollywood2,[14] HMDB51,[13] and UCF50[12]), which are the most challenging action datasets.

The **Hollywood2** dataset[14] (as shown in Figure 3A) has 12 action classes and is collected from different Hollywood movies. There are 1707 videos in Hollywood2 and are divided into a training set with 823 clips and a test set including 884 clips. Both the training clips and test clips in Hollywood2 are taken from different movies. The performance is measured by average precision (mAP) over all categories.[30]

The **HMDB51** dataset[13] (as shown in Figure 3B) has 51 action classes and 6766 video clips in total. The clips in HMDB51 are cut from a variety of YouTube videos. We use the original setup of three train-test splits for action recognition by following the work.[13] There are 70 videos for training and 30 videos for testing in every class and split. The average recognition performance is reported over all the three splits.[30]

The **UCF50** dataset[12] (as shown in Figure 3C) is collected from real-world videos of YouTube and has 50 action categories. The action clips range from common sports to daily living activities.[31] The action videos are divided into 25 groups in every class. There are not less than 4 action clips in every group. There are 6618 action video snippets in whole UCF50 dataset. Following by the recommendation in the work of Reddy and Shah,[12] we use the leave-one-group-out cross-validation and calculate the average accuracy over all categories.

4.2 | Experimental settings

We randomly extract a subset of 256 000 action feature descriptors from the training set and reduce feature dimensionality through the factor of two using Principal Component Analysis (PCA) as in the work of Perronnin et al.[32] We use whitening technique by following the process of

FIGURE 3 Datasets Examples. A, Hollywood2[14]; B, HMDB51[13]; C, UCF50[12]

	x264-B0	x264-B1	x264-B2	x264-B3
Hollywood2	**56.2%**	52.9%	47.9%	41.3%
HMDB51	**46.7%**	41.5%	36.8%	30.7%
UCF50	**82.6%**	77.3%	71.7%	65.8%

TABLE 1 Illustration of negative impact of B-frame on MF. The x264-B0 indicates none B-frame in x264 stream; x264-B1(2,3) represents that there are not more than 1 (or 2,3) successive B-frame(s) in x264 stream

PCA to ensure the descriptors have equal variance among different dimensions, and L2 normalization is applied to reduce the correlations among vector elements. The GMM model with a 256 dimensional Gaussians is trained. We then calculate the action descriptor of each video through a 2*DK* dimensional Fisher vector, where *D* is the descriptor dimension after performing PCA. Finally, the L2 normalization is taken on the Fisher vector once again. We montage the normalized Fisher vectors to construct different feature types and use a linear SVM ($C = 100$) for the action recognition training and testing.

4.3 | Impact of B-frame on MF

The fastest video feature extraction method MF[7] is easily affected by B-frame in video code stream. To investigate this aspect, we transcode the original videos to videos with specified coding scheme by using common video codec x264. We set the interval of I-frame at 15 in video stream and fix bit-rate at 500. The parameter of adaptive B-frame placement decision is forbidden. We then set the maximum number of concurrent B-frames at 0, 1, 2, and 3 separately and get video code streams with different number of B-frame. MF descriptors are extracted from the specified format videos and action recognition performs based on the settings in Section 4.2.

Table 1 indicates the performance of action recognition under different number of B-frame in video streams. The action recognition accuracies of MF achieve maximum when there is no B-frame in video streams on three datasets. Moreover, with the increase of the amount of B-frames (the number of P-frame reduced correspondingly) in video streams, the recognition accuracies of MF are declined by 3% to 7%.

4.4 | Evaluation of proposed method

We first evaluate our approach comparing with recent methods.[4,7,20,22] Then, we evaluate the impact of down-sampling interval on the proposed method.

4.4.1 | Comparison of proposed method with recent methods

We evaluate the proposed method and compare the speed and accuracy of action recognition with the state-of-the-art methods.[4,7,20,22] The down-sampling interval is reported at 4 pixels. The speed of feature extraction with different methods is reported in fps and run at a single module

TABLE 2 Comparison of proposed method with recent methods in action recognition accuracy and speed. The speed of descriptor calculation is presented with spatial resolution 640 × 480 (Hollywood2), 360 × 240 (HMDB51), and 320 × 240 (UCF50) pixels

	Ours		DT[4]		MF[7]		V-FAST[20]		MBH[22]	
	mAP	fps	mAP	fps	mAP	fps	mAP	fps	mAP	fps
Hollywood2	55.6%	40.9	59.4%	1.1	56.2%	158.2	52.3%	32.5	51.3%	9.4
HMDB51	45.0%	87.8	49.3%	2.9	46.7%	419.4	42.9%	81.3	40.8%	33.5
UCF50	81.1%	104.1	85.9%	3.9	82.6%	593.7	78.5%	97.6	77.6%	37.7

of AMD Opteron(tm) Processor (2.2 GHz). Table 2 gives the results of recognition performance and feature extraction speed on Hollywood2, HMDB51, and UCF50 datasets.

Proposed method vs DT[4]. For the performance, DT[4] (59.4%, 49.3%, 85.9%) is about four percent higher compared to our descriptors (55.6%, 45.0%, 81.1%). However, if the speed of action descriptor extraction is acted on both approaches measured in Hollywood2 videos with the picture element of 640 × 480, the proposed method gets 40.9 fps, which is around 40 times faster compared to DT[4] on Hollywood2 dataset. We have got more faster speed on the datasets of HMDB51 and UCF50, as the videos in these two datasets have smaller spatial resolution.

Proposed method vs MF[7]. The running speed of proposed method (40.9, 87.8, 104.1 fps) is obviously slower than MF[7] (158.2, 419.4, 593.7 fps), and the recognition accuracy of our method (55.6%, 45.0%, 81.1%) is also slightly less than MF[7] (56.2%, 46.7%, 82.6%). However, our method meets the real-time requirements as well and will not be affected by video encoding scheme.

Proposed method vs V-FAST[20]. The performance of our method (55.6%, 45.0%, 81.1%) is approximately three percent higher than V-FAST[20] (52.3%, 42.9%, 78.5%), and the running speed of ours (40.9, 87.8, 104.1 fps) is slightly faster than V-FAST[20] (32.5, 81.3, 97.6 fps).

Proposed method vs MBH[22]. As MBH[22] only achieves the real-time speed on small size videos such as HMDB51 and UCF50, the running speed of our method (40.9, 87.8, 104.1 fps) obviously faster than MBH[22] (9.4, 33.5, 37.7 fps). Moreover, our method (55.6%, 45.0%, 81.1%) is approximately four percent higher than MBH[22] (51.3%, 40.8%, 77.6%) in recognition accuracy.

4.4.2 | Impact of down-sampling interval on proposed method

For estimating the effect of down-sampling interval size on proposed method, we test the recognition accuracy and the speed of proposed method under different down-sampling grid sizes of 2, 4, 8, and 16 pixels. Figure 4 illustrates the PCA-Flow images of consecutive frames under different sampling interval, and Table 3 presents the speed and accuracy of recognition of proposed method. From Figure 4, we can see that the PCA-Flow motion images severely distort when down-sampling interval exceed 8 pixels and the accuracy of recognition also decreased at a great lick. Thus, the appropriate down-sampling interval size of the proposed method is 2 ∼ 4 pixels.

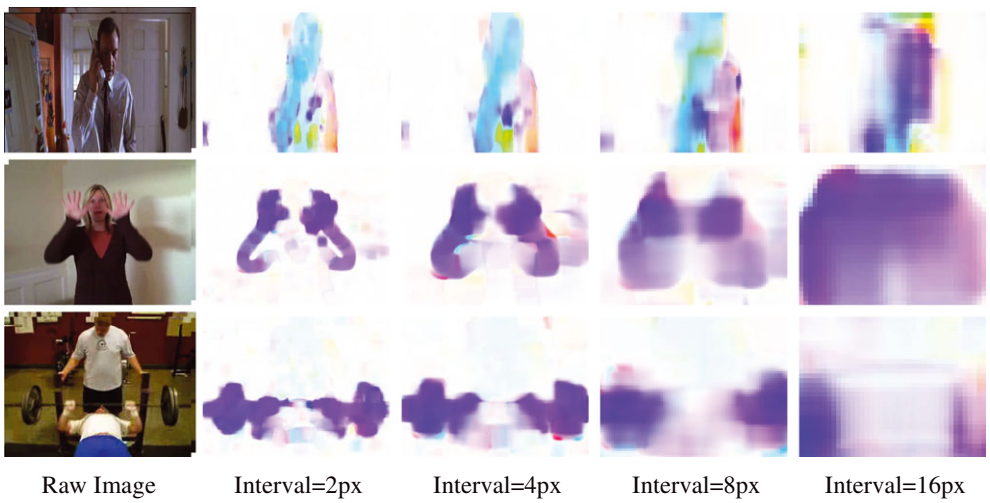

FIGURE 4 Consecutive frames in action recognition datasets (left) and their PCA-Flow images under different down-sampling interval size

TABLE 3 Impact of down-sampling interval on proposed method. The speed of descriptor calculation is computed with spatial resolution 640 × 480 (Hollywood2), 360 × 240 (HMDB51), and 320 × 240 (UCF50) pixels

	Hollywood2		HMDB51		UCF50	
Step	mAP	fps	mAP	fps	mAP	fps
2px	56.4%	24.3	**45.9%**	**64.1**	**82.3%**	**86.2**
4px	**55.6%**	**40.9**	45.0%	87.8	81.1%	104.1
8px	52.1%	56.1	41.2%	102.9	77.9%	122.9
16px	47.7%	70.5	36.8%	121.4	71.5%	141.0

5 | CONCLUSIONS

In this paper, we proposed a real-time action video descriptor extraction approach for real-time action recognition which is totally different from traditional approaches. Its main idea is that we take full advantage of the current fastest PCA-Flow algorithm and down-sampling the origin video frames to accelerate the process of optical flow calculation. We down-sampling video frames in form of grid firstly, and based on the sampling images, we calculate PCA-Flow between adjacent frames. Then, the PCA-Flow images are expanded to the original video image size by using efficient gCLSR super-resolution approach. Finally, we compute video descriptors on account of the original video images and enlarged PCA-Flow images. Experimental results show that our approach achieves competitive quality as traditional methods and completely meets the real-time applications.

ACKNOWLEDGMENTS

This work was partially supported by the National Natural Science Foundation of China under grants 61773295 and 61862015. The numerical calculations in this paper have been partially done on the supercomputing system in the Supercomputing Center of Wuhan University.

ORCID

Huafeng Chen https://orcid.org/0000-0002-2816-1020

REFERENCES

1. Peng X, Wang L, Wang X, Qiao Y. Bag of visual words and fusion methods for action recognition: comprehensive study and good practice. *Comput Vis Image Und*. 2016;150:109-125.
2. Wang Q, Gong D, Qi M, Shen Y, Lei Y. Temporal sparse feature auto-combination deep network for video action recognition. *Concurrency Computat Pract Exper*. 2018;30(23):e4487.
3. Akila K, Chitrakala S. An efficient method to resolve intraclass variability using highly refined HOG description model for human action recognition. *Concurrency Computat Pract Exper*. 2019;31:e4856.
4. Wang H, Klaser A, Schmid C, Liu CL. Dense trajectories and motion boundary descriptors for action recognition. *Int J Comput Vis*. 2013;103(1):60-79.
5. Xu Z, Hu R, Chen J, et al. Action recognition by saliency-based dense sampling. *Neurocomputing*. 2017;236:82-92.
6. Xu Z, Hu R, Chen J, et al. Semisupervised discriminant multimanifold analysis for action recognition. *IEEE Trans Neural Netw Learn Syst*. 2019:1-12. In press.
7. Kantorov V, Laptev I. Efficient feature extraction, encoding, and classification for action recognition. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR); 2014; Columbus, OH.
8. Mu C, Xie J, Yan W, Liu T, Li P. A fast recognition algorithm for suspicious behavior in high definition videos. *Multimedia Systems*. 2016;22(3):275-285.
9. Babu RV, Rangarajan B, Sundaram S, Tom M. Human action recognition in H. 264/AVC compressed domain using meta-cognitive radial basis function network. *Applied Soft Computing*. 2015;36:218-227.
10. Wulff J, Black MJ. Efficient sparse-to-dense optical flow estimation using a learned basis and layers. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR); 2015; Boston, MA.
11. Jiang J, Hu R, Han Z, Lu T. Efficient single image super-resolution via graph-constrained least squares regression. *Multimed Tools Appl*. 2014;72(3):2573-2596.
12. Reddy KK, Shah M. Recognizing 50 human action categories of web videos. *Mach Vis Appl*. 2013;24(5):971-981.
13. Kuehne H, Jhuang H, Garrote E, Poggio T, Serre T. HMDB: a large video database for human motion recognition. In: Proceedings of the IEEE International Conference on Computer Vision (ICCV); 2011; Barcelona, Spain.
14. Marszałek M, Laptev I, Schmid C. Actions in context. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR); 2009; Miami, FL.
15. Laptev I, Marszałek M, Schmid C, Rozenfeld B. Learning realistic human actions from movies. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR); 2008; Anchorage, AK.
16. Scovanner P, Ali S, Shah M. A 3-dimensional sift descriptor and its application to action recognition. In: Proceedings of the 15th ACM International Conference on Multimedia; 2007; Augsburg, Germany.
17. Willems G, Tuytelaars T, Van Gool L. An efficient dense and scale-invariant spatio-temporal interest point detector. In: Proceedings of the European Conference on Computer Vision (ECCV); 2008; Marseille, France.
18. Klaser A, Marszaek M, Schmid C. A spatio-temporal descriptor based on 3D-gradients. In: Proceedings of the British Machine Vision Conference (BMVC); 2008; Leeds, UK.
19. Yeffet L, Wolf L. Local trinary patterns for human action recognition. In: Proceedings of the IEEE International Conference on Computer Vision (ICCV); 2009; Kyoto, Japan.
20. Yu T-H, Kim T-K, Cipolla R. Real-time action recognition by spatiotemporal semantic and structural forests. In: Proceedings of the British Machine Vision Conference (BMVC); 2010; Aberystwyth, UK.
21. Rosten E, Drummond T. Machine learning for high-speed corner detection. In: Proceedings of the European Conference on Computer Vision (ECCV); 2006; Graz, Austria.
22. Shi F, Petriu E, Laganiere R. Sampling strategies for real-time action recognition. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR); 2013; Portland, OR.

23. Tom M, Babu RV, Praveen RG. Compressed domain human action recognition in H. 264/AVC video streams. *Multimed Tools Appl*. 2015;74(21):9323-9338.
24. Chen H, Chen J, Li H, Xu Z, Hu R. Compressed-domain based camera motion estimation for realtime action recognition. In: Proceedings of the Pacific Rim Conference on Multimedia (PCM); 2015; Gwangju, South Korea.
25. Werlberger M, Trobin W, Pock T, Wedel A, Cremers D, Bischof H. Anisotropic Huber-L1 optical flow. In: Proceedings of the British Machine Vision Conference (BMVC); 2009; London, UK.
26. Hauberg S, Feragen A, Black MJ. Grassmann averages for scalable robust PCA. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR); 2014; Columbus, OH.
27. Geiger A, Ziegler J, Stiller C. StereoScan: dense 3D reconstruction in real-time. Paper presented at: Intelligent Vehicles Symposium (IV); 2011; Baden-Baden, Germany.
28. Geiger A, Lenz P, Stiller C, Urtasun R. Vision meets robotics: the KITTI dataset. *Int J Robot Res*. 2013;32(11):1231-1237.
29. Butler DJ, Wulff J, Stanley GB, Black MJ. A naturalistic open source movie for optical flow evaluation. In: Proceedings of the European Conference on Computer Vision (ECCV); 2012; Florence, Italy.
30. Wang H, Schmid C. Action recognition with improved trajectories. In: Proceedings of the IEEE International Conference on Computer Vision (ICCV); 2013; Sydney, Australia.
31. Xu Z, Hu R, Chen J, Chen H, Li H. Global contrast based salient region boundary sampling for action recognition. In: Proceedings of the International Conference On MultiMedia Modeling (MMM); 2016; Miami, FL.
32. Perronnin F, Sánchez J, Mensink T. Improving the Fisher kernel for large-scale image classification. In: Proceedings of the European Conference on Computer Vision (ECCV); 2010; Crete, Greece.

How to cite this article: Chen H, Li H, Xu Z, Zhao Y, He T. Real-time action feature extraction via fast PCA-Flow. *Concurrency Computat Pract Exper*. 2019;e5507. https://doi.org/10.1002/cpe.5507