Pattern Recognition Letters 148 (2021) 22–28

Contents lists available at ScienceDirect

Pattern Recognition Letters

journal homepage: www.elsevier.com/locate/patrec

T-VLAD: Temporal vector of locally aggregated descriptor for multiview human action recognition

Hajra Binte Naeem[a], Fiza Murtaza[b,c], Muhammad Haroon Yousaf[a,c,*], Sergio A. Velastin[d]

[a] Swarm Robotics Lab-NCRA, University of Engineering and Technology Taxila, Taxila 47050, Pakistan
[b] Sino-Pak Center for Artificial Intelligence (SPCAI), Pak-Austria Fachhochschule: Institute of Applied Sciences and Technology (PAF-IAST), Haripur, Pakistan
[c] Department of Computer Engineering, University of Engineering and Technology Taxila, Taxila 47050, Pakistan
[d] School of Electronic Engineering and Computer Science, Queen Mary University of London, London E1 4NS, UK

ARTICLE INFO

Article history:
Received 15 June 2020
Revised 20 April 2021
Accepted 23 April 2021
Available online 4 May 2021

Keywords:
Human action recognition
Multi-view
View-invariant
Temporal action sequence
VLAD
3D Convolutional neural network features
IXMAS
Muhavi
UCF101
Short segment features

ABSTRACT

Robust view-invariant human action recognition (HAR) requires effective representation of its temporal structure in multi-view videos. This study explores a view-invariant action representation based on convolutional features. Action representation over long video segments is computationally expensive, whereas features in short video segments limit the temporal coverage locally. Previous methods are based on complex multi-stream deep convolutional feature maps extracted over short segments. To cope with this issue, a novel framework is proposed based on a temporal vector of locally aggregated descriptors (T-VLAD). T-VLAD encodes long term temporal structure of the video employing single stream convolutional features over short segments. A standard VLAD vector size is a multiple of its feature codebook size (256 is normally recommended). VLAD is modified to incorporate time-order information of segments, where the T-VLAD vector size is a multiple of its smaller time-order codebook size. Previous methods have not been extensively validated for view-variation. Results are validated in a challenging setup, where one view is used for testing and the remaining views are used for training. State-of-the-art results have been obtained on three multi-view datasets with fixed cameras, IXMAS, MuHAVi and MCAD. Also, the proposed encoding approach T-VLAD works equally well on a dynamic background dataset, UCF101.

1. Introduction

A robust view-invariant HAR framework requires effective multi-view video representation. Influential multi-view methods [22,37] suggest transfer of view knowledge from one view to a high level virtual view. However, these methods require view-specific information. Recent CNN-based (Convolutional Neural Networks) HAR methods have shown significant progress in recognizing actions from single views. However, according to the review in [35], these methods are unable to overcome the challenge of viewpoint variation. So, there is a need to explore simpler CNN features methods that can handle view variation.

Different from hand-crafted methods for action representation [20,30], CNN-based methods learn trainable action representations directly from video. CNNs were originally intended to extract 2D or spatial features and not the 3D spatio-temporal characteristics of video. Hence, a major interest is on how to incorporate temporal information in CNNs. An important and popular method that encodes temporal information, takes stacked optical flow along with RGB frames as input in a two stream model [23]. In [8], the two stream model was extended by the addition of a novel spatial and temporal fusion layer using 3D Conv and 3D pooling. Tran et al. [26] factorized 3D Conv into two separate 2D spatial Conv and 1D temporal Conv which are easier to optimize. In [15] temporal excitation and aggregation (TEA) block introduced multiple temporal excitation module (MTA) that replaced 1D temporal Conv with a group of sub-convolutions. Gammulle et al. [9] exploited the function of memory cells in LSTM to learn the long-range temporal structure of video. However, TS-LSTM [17] suggests the use of pre-segmented data to fully exploit temporal information.

In [13,36], the authors investigated approaches to fuse and aggregate temporal frame-level information into video level information. Varol et al. [29] applies Long term Temporal Convolutions (LTC) on a large number of frames to capture the long range temporal structure of video, whereas [32] introduces a non-local neural network for this purpose. To capture visual content over the entire video, Temporal Relation Network (TRN) [38] exploits temporal relational reasoning and a Temporal Segment Network (TSN)

[*] Corresponding author at: Department of Computer Engineering, University of Engineering and Technology Taxila, Taxila 47050, Pakistan.
E-mail address: haroon.yousaf@uettaxila.edu.pk (M.H. Yousaf).

https://doi.org/10.1016/j.patrec.2021.04.023
0167-8655/© 2021 Elsevier B.V. All rights reserved.

[31] computes features from randomly selected snippets of video segments. ActionS-ST-VLAD [27] aggregates entire video spatio-temporal local CNN features, ignoring time order information. SeqVLAD [33], unlike NetVLAD [1] and ActionVLAD [10], encodes temporal relationships of the sequential frames, combining trainable VLAD and Recurrent Convolutional Network (RCN). [24] proposes a Temporal-Spatial Mapping (TSM) function that maps time order information and CNN features of densely sampled frames into a 2D feature map, referred as VideoMap. This dense sampling increases the computational cost of the CNN and fixed sampling rate limits temporal coverage locally.

Most of the approaches [27,29,31,33] proposed to model action long-range temporal structure are limited to two stream CNN features. These two stream models require computationally expensive optical flow (or other motion vectors) as additional input, and training of two deep networks. A single stream 3D-Convolutional network (C3D) [25] replaces 2D Conv kernel with 3D Conv kernel representing actions at short-segment-level as generic, efficient and compact spatio-temporal features. These features are simply aggregated, thus ignoring temporal relationship of sequential short segments. Bidirectional encoder representations from transformers (BERT) [12] added a layer at the end of 3DCNN that incorporates temporal information with attention mechanism. Temporal pyramid network (TPN) [34] forms a feature hierarchy that capture visual characteristics at various temporal scale of an action. Temporal shift module (TSM) [16] inserted in 2DCNN shifts part of the channels along the temporal dimension to facilitate information exchange among neighbouring frames. X3D [7] extended tiny spatial network along temporal duration, frame rate, spatial resolution, width, and depth to achieve efficient video network with target complexity. Temporal 3D Convolutional network (T3D) [5] extends the 2D DenseNet architecture, adding a 3D temporal transition layer (TTL) which functions on different temporal depths, capturing short, mid, and long-range temporal information. Although the computation of 3D Conv burdens GPU memory, they do not involve the computation of the co-relation between appearance and motion as in two stream models.

This work proposes a novel multi-view framework to capture temporal sequences of complete videos, exploiting the time order of non-overlapping video segments. Video segments spatio-temporal features are pooled employing second order statistics of VLAD based on action class membership and segment time order membership. Formation of sequential VLAD from a codebook is a simpler approach, as compared to end-to-end trainable networks SeqVLAD [33] and T3D [5] that capture temporal information through additional top layers SGRU-RCN and TTL, respectively. These layers increase network training time due to backpropagation in RCN, and more parameters for convolution at multiple temporal depths in TTL.

Motivation: The proposed method is inspired by Spatio Temporal VLAD (ST-VLAD) [6] that incorporates spatial location of each frame local features, T-VLAD includes temporal location of each video segment global features, hence extending frame-level information into video-level information. Moreover, human actions are combination of several common primitive motions and a discriminating motion. VLAD recognizes action based on discriminating motion. Common primitive motion occurs in a specific time order and discriminating motion may become occluded due to view variations. T-VLAD timestamps these primitive motions facilitating view invariant action recognition. This is illustrated in Fig. 5 and explained in detail in Section 3.6 with an example.

It is worth highlighting the following contributions: (1) We propose a long-term temporal structure of actions using single stream C3D features from sparsely sampled short segments of video. The computation of features from short video segments is computationally inexpensive [17]. (2) We include time-order information for encoding temporal sequence of complete video in a single feature vector. (3) The proposed action representation from short segment features is robust to view changes and works well with variable background sequences.

2. Proposed T-VLAD for multiview action recognition

This section describes the details of the proposed T-VLAD encoding approach for view-invariant action representation, as illustrated in Fig. 1. C3D features are computed over fixed sampling intervals of video to form a feature descriptor X_f. An additional time-order descriptor X_t is formed to show the order of segments. X_f and X_t are then used to learn codebooks C_f and C_t, respectively. Finally, assignments from C_t are used to pool residuals R_f and the membership information S_f is evaluated from C_f, resulting in encoded T-VLAD features.

2.1. Feature extraction and codebooks generation

The video is sampled at a rate δ before feature extraction. For a video V with N number of frames, the number of non-overlapping segments is $\frac{N}{\delta}$. Two kinds of information are collected from these segments. First, per segment features are stacked to form a feature descriptor $X_f \in \mathbb{R}^{\frac{N}{\delta} \times d}$, where d is the dimensionality of the descriptor. Second, the time order of segments is expressed by a single vector as $X_t = [1, 2, 3 \cdots N/\delta] \in \mathbb{R}^{\frac{N}{\delta} \times 1}$. This vector is termed as 'time-order descriptor'.

For a fixed value of δ, the number of segments per video varies and so does the size of descriptor. Two codebooks are constructed, $C_f \in \mathbb{R}^{k_f \times d}$ from X_f and $C_t \in \mathbb{R}^{k_t \times 1}$ from X_t, where, k_f and k_t are the number of codewords in each codebook respectively. K-means clustering with Euclidean distance is used to learn the two codebooks. Codewords of C_f are related to the visual details of video segments whereas codewords of C_t indicate their temporal order.

2.2. T-VLAD encoding

After computing codebooks C_f and C_t, video features and time-order descriptors i.e. X_f and X_t are assigned to their nearest codewords from the respective codebooks. Similar to VLAD, a first order distance between feature descriptor X_f and assigned codeword is computed, known as residuals, as given in Eq. (1).

$$R_f = [r_1 | r_2 | r_3 | \cdots | r_{\frac{N}{\delta}}] \tag{1}$$

where, r_j is the residual of the jth feature x_j from the assigned codeword c of C_f, given as $r_j = x_j - c$. Along with residuals, the binary matrix $S_f \in \mathbb{R}^{\frac{N}{\delta} \times k_f}$ encrypts segment feature membership. Membership of a segment feature with an assigned codeword is denoted by '1' at the corresponding position in codebook. For instance, Eq. (2) shows that first, second and last segment features are assigned to first, third and second codewords respectively i.e. the rows of S_f represent the feature segments numbers and the columns represent codewords numbers.

$$S_f = \begin{bmatrix} 1 & 0 & 0 & 0 \cdots & 0 \\ 0 & 0 & 1 & 0 \cdots & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 0 & 1 & 0 & 0 \cdots & 0 \end{bmatrix} \tag{2}$$

Therefore, residuals encode the similarity of features and membership represents the association with a codebook. R_f and S_f are aggregated to get a final encoded feature vector.

Clustering divides the feature descriptor into cells. Standard VLAD [37] aggregates feature residuals within these cells and then

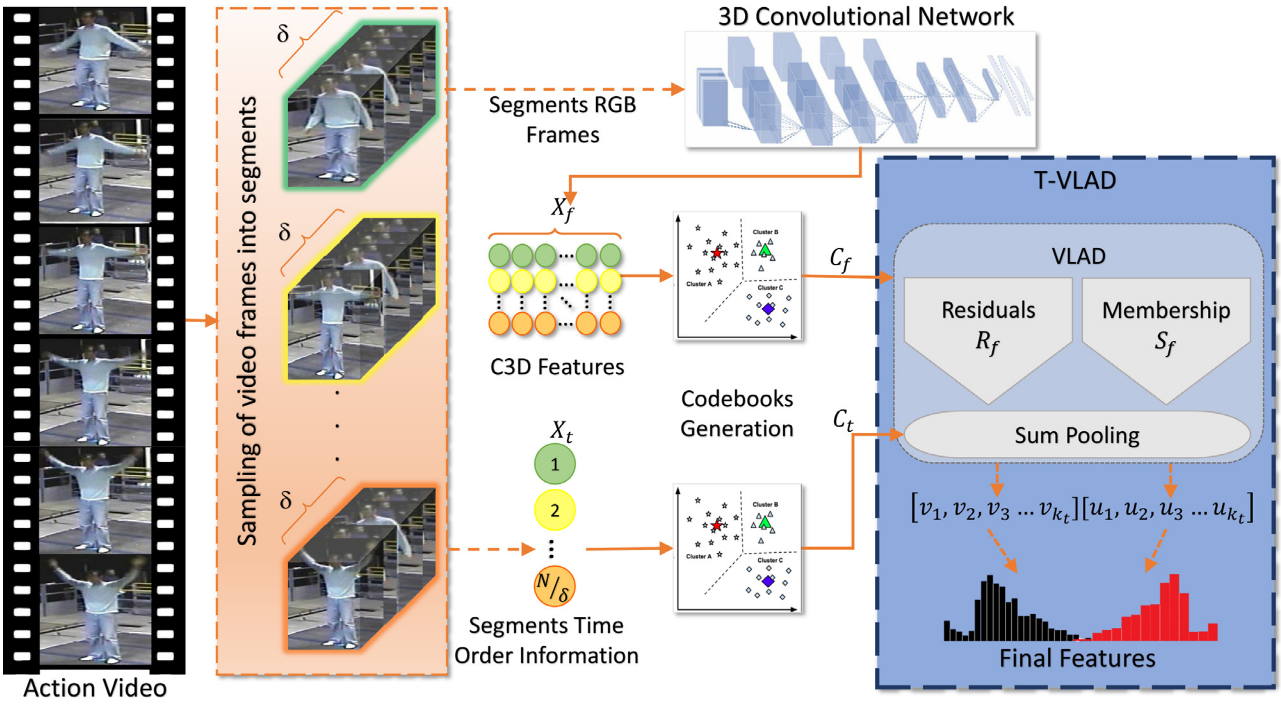

Fig. 1. Proposed T-VLAD approach to encode features of full video.

concatenates them to form an encoded vector. Additional clustering of the time-order descriptor introduces a new form of aggregation. Instead of aggregating residuals within cells of C_f, they are aggregated within cells of C_t. Hence, they are named 'temporal VLAD' (T-VLAD). Each cell of C_t is time-order of feature occurrence and residuals R_f record features similarity with respect to cells of C_f. R_f differentiates primitive and discriminating motion in segments. As discussed in motivation, common primitive motion occurs in specific time order and discriminating motion may become occluded due to view variation. Aggregation within cells of C_t records this time-order, which assists action recognition when discriminating motion is occluded. Let M_i and N_i be residuals and membership of set of features assigned to the ith codeword of C_t, respectively. Then, sum pooling of residuals forms a new code v_i given by:

$$v_i = \sum M_i \tag{3}$$

where, each residual vector is of size d. Concatenation of all v_i results in an encoded vector of size $d \times k_t$. Therefore, pooling is enforced for features existing in similar time order. Similarly, sum pooling of N_i records feature assignment with C_f as u_i given by:

$$u_i = \sum N_i \tag{4}$$

where each membership vector is of size k_f. Concatenation of all u_i results in a vector of size $k_f \times k_t$. The first part of pooling arranges features according to time order and the latter part assists in distinguishing their action class. Both vectors, one from v_i's and another from u_i's are concatenated to form the final T-VLAD vector of size $(d \times k_t) + (k_f \times k_t) = (d + k_f) \times k_t$. Addition of time-order information does not increase the overall vector size, as shown in 3.3.

3. Experimental evaluation

3.1. Datasets

The proposed approach has been evaluated on three fixed camera setup multi-view human actions datasets: IXMAS, MuHAVi and MCAD. IXMAS includes 1,650 videos from 5 camera views of 10 actors performing 11 actions. MuHAVi contains 3,443 videos from 8 different views of 7 actors performing 17 actions. MCAD is more complex as it contains 14,298 videos of 20 actors performing 18 actions recorded with 3 static cameras and 2 PTZ cameras. In this study only static cameras videos are used for evaluation. IXMAS and MuHAVi results are computed using one view for testing with the remaining views for training. For comparison of MCAD results, one view is used for training with the remaining for testing. In all cases the testing view samples are never used in the training phase. This experimental setup tests the view-invariant nature of the method. In this paper average accuracy on all camera views refers to average of all testing view accuracy. Further experimentation is performed on an actions dataset with varying backgrounds, UCF101, to asses the suitability of the approach for such cases. It includes 13,320 trimmed YouTube video clips from 101 action classes.

Table 1
Parameters setting for finetuning C3D.

Parameter	Value
Pre-trained model	Sport1m
Input dimensions	3 × 16 × 128 × 171
Optimizer	Gradient descent
Learning rate	0.001
Epochs	50

3.2. Implementation details

C3D [25] was finetuned for feature extraction. Parameters selected for finetuning are shown in Table 1. Only training videos are used to finetune the network. After finetuning, per segment features are extracted from the fully connected layer fc6 i.e. $d = 4096$. As a standard practice [25], the value of δ is set to 16 and frames are resized to 128 × 171. The proposed method can, in principle, be used with any global features. Due to lack of space, results are only presented here with C3D, considered a state-of-the-art approach.

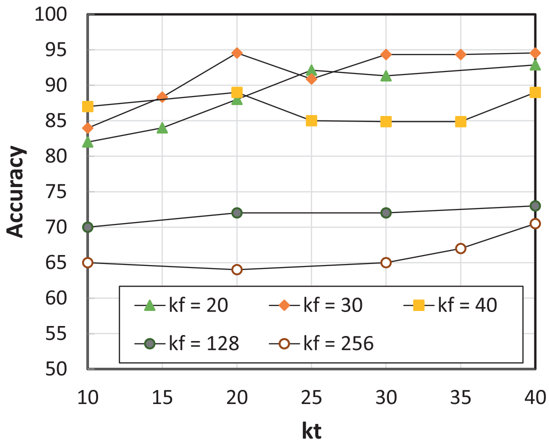

Fig. 2. Evaluation of feature and time order codebook size for MuHAVi.

To avoid a negative influence on the classification step of T-VLAD vectors, power normalization is applied followed by L2 normalization. Power normalization is given by Eq. (5).

$$\|sign(x)|x|^{\alpha}\|, 0 \le \alpha \le 1 \tag{5}$$

As a standard practice [6], the value of α is set to 0.1 for CNN based features. Later, a linear Support Vector Machine (SVM) is trained for action recognition because it is suitable for L2 normalized features [20]. To facilitate multiclass action recognition, we use one-vs-all SVM.

3.3. Evaluation of codebook size

Experimentation has been carried out to tune values of k_f and k_t. The value of k_t is tuned to half of the maximum number of segments present in the videos in the dataset. For instance, in MuHAVi the maximum number of non-overlapping video segments are 40 and the value of k_t and k_f is 20 and 30 respectively, as shown in Fig. 2. However, vector size increases with the number of video segments. Unlike standard VLAD with a standard feature codebook size (k_f) 256, small codebook shows remarkable performance in this modified version of VLAD. Technically, VLAD encodes local features, whereas the modified version encodes temporal sequence of actions from global or high-level features. Fully connected layers are rich in non-spatial or high-level features. Therefore, a small codebook is appropriate for global features. With $k_f = 256$ and $d = 4096$, VLAD is of size 1,048,576, whereas T-VLAD vector size with $k_f = 30, k_t = 20$ and $d = 4096$ is 82,520. Hence, VLAD vector size has been reduced by a factor of 12 approximately, even with addition of the extra time order information.

3.4. Performance on IXMAS and MuHAVi

For IXMAS, the T-VLAD approach is compared with multi-view methods in Table 2, outperforming the state-of-the-art methods including RNKTM [22], TDL [37], STF [28] and 4Stream 3DCNN [3]. It is worth mentioning that robust non-linear knowledge transfer model (RNKTM) [22] and transferable dictionary learning (TDL) [37] exploit neural network and dictionary learning that transfer videos from different views to a high level virtual view. On the other hand, the proposed method is best in withstanding view-variation without having to transfer any view information and by simple time order-based encoding of C3D features. Space time filtering (STF) [28] is based on 3D discrete tensor Fourier transform (3D-DTFT) that trains one view-invariant Action-DCCF (3D distance classifier correlation filter) for all views and action classes. Unlike the C3D model, Action-DCCF is trained from all views performing better for variable viewpoints, particularly for a top view (i.e C4) with distinct action appearance (as shown in Table 2). However, considering the computational cost constraint, real time applications prefer not to work in the frequency domain [2]. Chenarlogh et al. [3] proposed a multi-view method for action recognition from four stream 3DCNN using raw frame, optical flow and gradient as input. Lower recognition accuracy shows that, unlike T-VLAD 4Stream 3DCNN method lacks compact view-invariant action representation.

Table 3 presents a comparison of MuHAVi average recognition accuracy from all views. Where one view out of eight is used for testing and remaining views are used for training. This provides per view testing accuracy and average of these signify overall view-invariant nature of the method. As far we can tell, we are first to present results on the RGB version of this dataset. Working on RGB data doesnot require segmentation to extract human silhouttes. T-VLAD encoded features outperforms state-of-the-art human silhouttes based handcrafted [4] and deep learning [19] approaches. Moreover, the results show the effectiveness of T-VLAD for multi-view action recognition.

3.5. Performance on UCF101

The previous section shows that the proposed method is robust to view-point variation. However, multi-view datasets tend to be recorded in fixed camera setups. UCF101 is typical of single camera action datasets with dynamic arbitrary viewpoints. Table 4 shows average accuracy of various methods over all of UCF101's three standard splits. The proposed encoding approach works equally well in this context, as T-VLAD outperforms C3D (discussed in 3.6). T-VLAD shows 11% better recognition accuracy than ST-VLAD for RGB-only input. End-to-end trainable networks ActionVLAD [10], SeqVLAD [33] and T3D [5] perform better than forming sequential VLAD from a codebook. Overall, UCF-101 contain diverse action classes and end-to-end training instead of formation of codebook is better. Girdhar et al. [10] and Xu et al. [33] are two stream networks that extract local temporal information from additional CNN input i.e. stacked optical flow. Single stream T3D [5] aggregates local Conv feature maps to extract temporal information. However, T-VLAD aggregates global temporal information performing better than C3D [25] and ST-VLAD (2St) [6] that does not aggregate temporal information specifically. Proposed encoding approach lacks local temporal information due to which it did not perform better than ActionVLAD, SeqVLAD and T3D. In future, T3D single stream network can be fused with T-VLAD global feature encoding approach, aggregating temporal information both locally and globally.

3.6. Comparison with average pooled features

In Tables 2,3 and 4, results are also compared with simple average pooling of C3D features as used in [25]. T-VLAD recognition accuracy is approximately 5% higher than simple averaging. Figs. 3 and 4 show comparisons between C3D and T-VLAD for per action accuracy. One can observe a significant improvement in recognition accuracy for 'Walk' in IXMAS and 'WalkTurnBack' in MuHAVi, which are primitive actions that can be easily confused with any action having a walking sequence of motion. Consecutive segment features emphasize human walking motion and direction. When these features are simply averaged, information about direction of motion in a complete action diminishes. In T-VLAD however, segment features are added in their respective time order, highlighting motion direction. Hence, other actions e.g. 'WalkFall' that are performed in combination with the primitive action 'Walk' are not confused.

'SmashObject', 'PullHeavyObject' and 'PickUpThrowObject' in MuHAVi involve similar primitive 'Bending' motion, along with dis-

Table 2
Comparison of IXMAS recognition accuracy with other multi-view methods for each camera (Cx is accuracy when camera x is used for testing and all other cameras are used for training). The last column is average accuracy of all cameras.

Method	C0	C1	C2	C3	C4	Average accuracy (%)
NKTM [21]	77.8	75.2	80.3	74.7	54.6	72.5
RNKTM [22]	78.4	78.0	80.7	75.8	57.8	74.1
TDL [37]	73.2	78.5	74.9	76.1	52.9	71.1
STF [28]	**98.6**	90.5	89.8	88.0	**82.7**	**88.1**
4Stream 3DCNN [3]	68.6	68	63.3	67.8	37.7	61
Standard VLAD	82.6	92.1	86.9	74.1	32.9	73.72
C3D average	93.3	93.9	93	89.6	23.3	78.9
T-VLAD	95.4	**96.0**	**95.4**	**90.9**	47.6	84.8

Table 3
Comparison of MuHAVi average accuracy from all camera views.

Method	Average accuracy (%)
ELM [11]	74.75
MHI/HOG [18]	54.40
Box and cloud features [4]	81.43[a]
Deep features/ELM [19]	82.04
Standard VLAD	76.67
C3D average	82.27
T-VLAD	**87.65**

[a] for only five actions, CrawlOnKnees, Punch, RunStop, WalkTurnBack and WaveArms

Table 4
Average accuracy on UCF101 over all three splits.

Method	Input of the CNN	Average accuracy (%)
ST-VLAD (2St) [6]	RGB + Flow	90.2
ActionVLAD [10]	RGB + Flow	92.7
SeqVLAD [33]	RGB + Flow	94.5
ST-VLAD (SCN) [6]	RGB	78.0
T3D [5]	RGB	91.7
C3D average [25]	RGB	85.2
T-VLAD	RGB	89.0

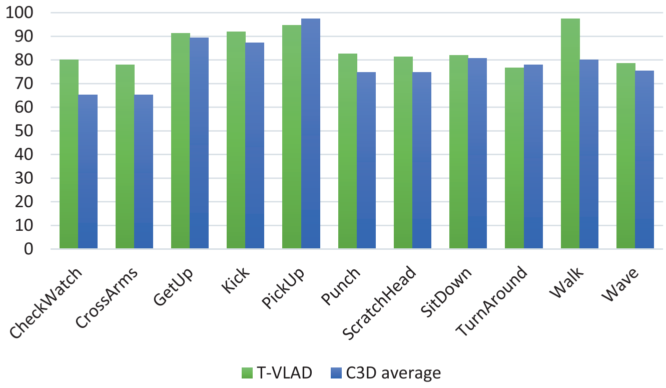

Fig. 3. Comparison of per action accuracy of C3D averaged features and T-VLAD encoded features for IXMAS.

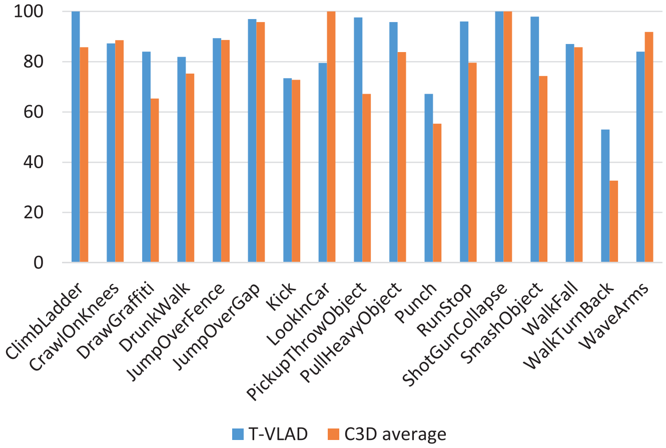

Fig. 4. Comparison of per action accuracy of C3D averaged features and T-VLAD encoded features for Muhavi.

Table 5
Comparison of MCAD average accuracy from all camera views.

Method	Average accuracy (%)
IDT_FV [14]	75.9
Standard VLAD	69.9
C3D average	75.6
T-VLAD	75.3
C3D average + T-VLAD	**78.6**

criminating motion like ‘smashing’ ‘pulling’ and ‘throwing’. They are confused when their discriminating motion is occluded due to view variation. T-VLAD indicates in what time sequence primitive ‘Bending’ motion occurs, distinguishing actions effectively. Fig. 5 shows the time order of the ‘Bending’ motion in three actions. Including time order information of video segments along with segment features benefits the encoding of long-term temporal structure of actions across variable viewpoints. However, actions like ‘LookInCar’and ‘WaveArms’that have similar primitive motion across all segments. T-VLAD marks time-order of these segments, but in this case it does not include extra information in action recognition. Therefore, simple averaging of these actions C3D segment features shows better recognition accuracy.

3.7. Performance on MCAD

In Table 5 results are compared with [14] where first Gaussian Mixture Model(GMM) is used to learn codebook. Then Fisher Vector (FV) encoding is applied on IDT features to generate the descriptor. This descriptor is high dimensional twice the dimension of VLAD. Fusion of average pooled features and T-VLAD results in higher accuracy of 78.6%. As T-VLAD recognition accuracy is approximately equal to average pooled C3D features. Figure 6 shows that T-VLAD performs better for ‘Walk’which is primitive motion common among other actions. Also, actions with little discriminative motion like ‘ObjectPut’and ‘ObjectLeft’are confused when features are average pooled. Whereas timestamping results in better recognition accuracy for these actions. On the other hand, average pooling shows remarkable performance on other actions like ‘PersonRun’and ‘ObjectThrow’as compared to T-VLAD. Therefore, overall fusion of both is better view-invariant representation for all actions.

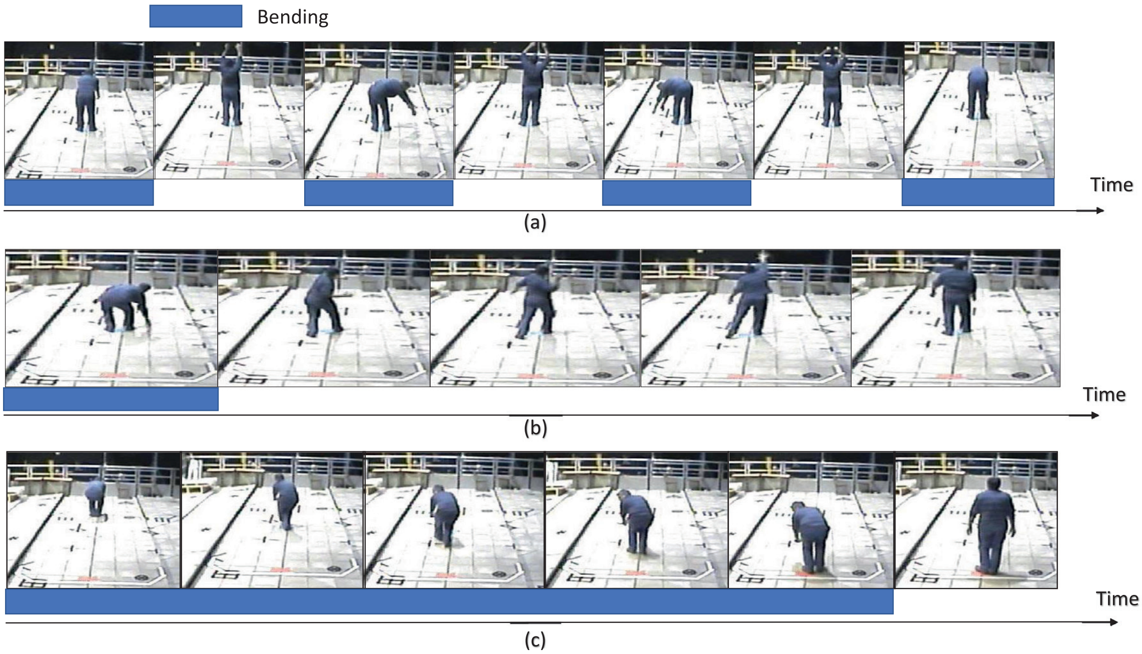

Fig. 5. Time order of bending motion for actions, (a) SmashObject (b) PickupThrowObject (c) PullHeavyObject.

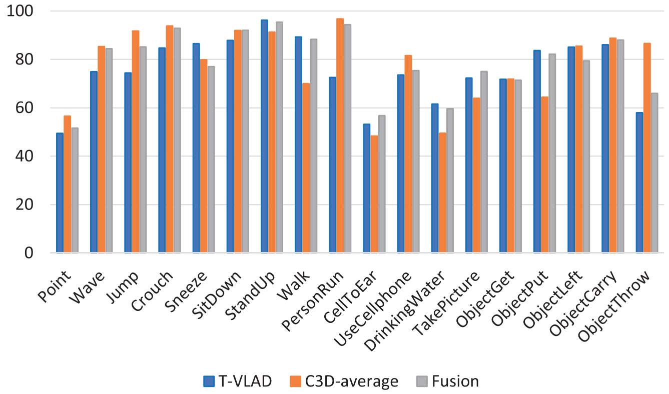

Fig. 6. Comparison of per action accuracy of C3D averaged features and T-VLAD encoded features for MCAD.

4. Conclusion

This study has proposed a novel modification of VLAD to capture temporal sequence of complete action. T-VLAD encodes video non-overlapping segment global features using their time order information. Experiments on the MuHAVi, IXMAS,MCAD and UCF101 demonstrate that T-VLAD is an efficient view-invariant representation of complete action long-term temporal sequence that also performs equally well on a dynamic background. For densely sampled video, high sampling rate increases the number of segments and T-VLAD vector size. In the future, this encoding scheme can be made independent of sampling rate without significant influence on recognition accuracy. Additionally, it is planned to modify T-VLAD to incorporate feature spatial information. That way, it can be used with local features like improved dense trajectories.

Declaration of Competing Interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Acknowledgement

Muhammad Haroon Yousaf has received funding from Higher Education Commission, Pakistan, for Swarm Robotics Lab under National Centre for Robotics and Automation (NCRA).

References

[1] R. Arandjelovic, P. Gronat, A. Torii, T. Pajdla, J. Sivic, NetVLAD: CNN architecture for weakly supervised place recognition, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2016, pp. 5297–5307.

[2] D.R. Beddiar, B. Nini, M. Sabokrou, A. Hadid, Vision-based human activity recognition: a survey, Multimed. Tools Appl. 79 (41) (2020) 30509–30555.

[3] V.A. Chenarlogh, F. Razzazi, N. Mohammadyahya, A multi-view human action recognition system in limited data case using multi-stream CNN, in: 2019 5th Iranian Conference on Signal Processing and Intelligent Systems (ICSPIS), IEEE, 2019, pp. 1–11.

[4] K.-P. Chou, M. Prasad, D. Wu, N. Sharma, D.-L. Li, Y.-F. Lin, M. Blumenstein, W.-C. Lin, C.-T. Lin, Robust feature-based automated multi-view human action recognition system, IEEE Access 6 (2018) 15283–15296.

[5] A. Diba, M. Fayyaz, V. Sharma, A. Hossein Karami, M. Mahdi Arzani, R. Yousefzadeh, L. Van Gool, Temporal 3D convnets using temporal transition layer, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops, 2018, pp. 1117–1121.

[6] I.C. Duta, B. Ionescu, K. Aizawa, N. Sebe, Spatio-temporal VLAD encoding for human action recognition in videos, in: International Conference on Multimedia Modeling, Springer, 2017, pp. 365–378.

[7] C. Feichtenhofer, X3d: expanding architectures for efficient video recognition, in: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2020, pp. 203–213.

[8] C. Feichtenhofer, A. Pinz, A. Zisserman, Convolutional two-stream network fusion for video action recognition, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2016, pp. 1933–1941.

[9] H. Gammulle, S. Denman, S. Sridharan, C. Fookes, Two stream LSTM: a deep fusion framework for human action recognition, in: 2017 IEEE Winter Conference on Applications of Computer Vision (WACV), IEEE, 2017, pp. 177–186.

[10] R. Girdhar, D. Ramanan, A. Gupta, J. Sivic, B. Russell, ActionVLAD: learning spatio-temporal aggregation for action classification, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2017, pp. 971–980.

[11] A. Iosifidis, A. Tefas, I. Pitas, Minimum variance extreme learning machine for human action recognition, in: 2014 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), IEEE, 2014, pp. 5427–5431.

[12] M.E. Kalfaoglu, S. Kalkan, A.A. Alatan, Late temporal modeling in 3D CNN architectures with bert for action recognition, in: A. Bartoli, A. Fusiello (Eds.), Computer Vision – ECCV 2020 Workshops, Springer International Publishing, Cham, 2020, pp. 731–747.

[13] A. Karpathy, G. Toderici, S. Shetty, T. Leung, R. Sukthankar, L. Fei-Fei, Large-scale video classification with convolutional neural networks, in: Proceedings of the IEEE conference on Computer Vision and Pattern Recognition, 2014, pp. 1725–1732.

[14] W. Li, Y. Wong, A.-A. Liu, Y. Li, Y.-T. Su, M. Kankanhalli, Multi-camera action dataset for cross-camera action recognition benchmarking, in: 2017 IEEE Winter Conference on Applications of Computer Vision (WACV), IEEE, 2017, pp. 187–196.

[15] Y. Li, B. Ji, X. Shi, J. Zhang, B. Kang, L. Wang, TEA: temporal excitation and aggregation for action recognition, in: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2020, pp. 909–918.

[16] J. Lin, C. Gan, S. Han, TSM: temporal shift module for efficient video understanding, in: Proceedings of the IEEE/CVF International Conference on Computer Vision, 2019, pp. 7083–7093.

[17] C.-Y. Ma, M.-H. Chen, Z. Kira, G. AlRegib, TS-LSTM and temporal-inception: exploiting spatiotemporal dynamics for activity recognition, Signal Process. Image Commun. 71 (2019) 76–87.

[18] F. Murtaza, M.H. Yousaf, S.A. Velastin, Multi-view human action recognition using 2D motion templates based on MHIs and their hog description, IET Comput. Vis. 10 (7) (2016) 758–767.

[19] N. Nida, M.H. Yousaf, A. Irtaza, S.A. Velastin, Instructor activity recognition through deep spatiotemporal features and feedforward extreme learning machines, Math. Probl. Eng. 2019 (2019).

[20] X. Peng, L. Wang, X. Wang, Y. Qiao, Bag of visual words and fusion methods for action recognition: comprehensive study and good practice, Comput. Vis. Image Underst. 150 (2016) 109–125, doi:10.1016/j.cviu.2016.03.013.

[21] H. Rahmani, A. Mian, Learning a non-linear knowledge transfer model for cross-view action recognition, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2015, pp. 2458–2466.

[22] H. Rahmani, A. Mian, M. Shah, Learning a deep model for human action recognition from novel viewpoints, IEEE Trans. Pattern Anal. Mach.Intell. 40 (3) (2018) 667–681, doi:10.1109/TPAMI.2017.2691768.

[23] K. Simonyan, A. Zisserman, Two-stream convolutional networks for action recognition in videos, in: Advances in Neural Information Processing Systems, 2014, pp. 568–576.

[24] X. Song, C. Lan, W. Zeng, J. Xing, X. Sun, J. Yang, Temporal-spatial mapping for action recognition, IEEE Trans. Circuits Syste. Video Technol. (2019).

[25] D. Tran, L. Bourdev, R. Fergus, L. Torresani, M. Paluri, Learning spatiotemporal features with 3D convolutional networks, in: The IEEE International Conference on Computer Vision (ICCV), 2015.

[26] D. Tran, H. Wang, L. Torresani, J. Ray, Y. LeCun, M. Paluri, A closer look at spatiotemporal convolutions for action recognition, in: Proceedings of the IEEE conference on Computer Vision and Pattern Recognition, 2018, pp. 6450–6459.

[27] Z. Tu, H. Li, D. Zhang, J. Dauwels, B. Li, J. Yuan, Action-stage emphasized spatiotemporal VLAD for video action recognition, IEEE Trans. Image Process. 28 (6) (2019) 2799–2812.

[28] A. Ul Haq, X. Yin, J. He, Y. Zhang, On space-time filtering framework for matching human actions across different viewpoints, IEEE Trans. Image Process. 27 (3) (2018) 1230–1242, doi:10.1109/TIP.2017.2765821.

[29] G. Varol, I. Laptev, C. Schmid, Long-term temporal convolutions for action recognition, IEEE Trans. Pattern Anal. Mach.Intell. 40 (6) (2018) 1510–1517, doi:10.1109/TPAMI.2017.2712608.

[30] H. Wang, C. Schmid, Action recognition with improved trajectories, in: Proceedings of the IEEE International Conference on Computer Vision, 2013, pp. 3551–3558.

[31] L. Wang, Y. Xiong, Z. Wang, Y. Qiao, D. Lin, X. Tang, L. Van Gool, Temporal segment networks for action recognition in videos, IEEE Trans. Pattern Anal. Mach.Intell. (2018).

[32] X. Wang, R. Girshick, A. Gupta, K. He, Non-local neural networks, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2018, pp. 7794–7803.

[33] Y. Xu, Y. Han, R. Hong, Q. Tian, Sequential video VLAD: training the aggregation locally and temporally, IEEE Trans. Image Process. 27 (10) (2018) 4933–4944.

[34] C. Yang, Y. Xu, J. Shi, B. Dai, B. Zhou, Temporal pyramid network for action recognition, in: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2020, pp. 591–600.

[35] G. Yao, T. Lei, J. Zhong, A review of convolutional-neural-network-based action recognition, Pattern Recognit. Lett. 118 (2019) 14–22.

[36] S. Yu, Y. Cheng, S. Su, G. Cai, S. Li, Stratified pooling based deep convolutional neural networks for human action recognition, Multimed. Tools Appl. 76 (11) (2017) 13367–13382.

[37] J. Zhang, H.P. Shum, J. Han, L. Shao, Action recognition from arbitrary views using transferable dictionary learning, IEEE Trans. Image Process. 27 (10) (2018) 4709–4723.

[38] B. Zhou, A. Andonian, A. Oliva, A. Torralba, Temporal relational reasoning in videos, in: Proceedings of the European Conference on Computer Vision (ECCV), 2018, pp. 803–818.