Action-Stage Emphasized Spatio-Temporal VLAD for Video Action Recognition

Zhigang Tu, *Member, IEEE,* Hongyan Li, Dejun Zhang, *Member, IEEE,* Justin Dauwels, *Senior Member, IEEE* Baoxin Li, *Senior Member, IEEE* and Junsong Yuan, *Senior Member, IEEE*

***Abstract*—Despite outstanding performance in image recognition, convolutional neural networks (CNNs) do not yet achieve the same impressive results on action recognition in videos. This is partially due to the inability of CNN for modeling long-range temporal structures especially those involving individual action stages that are critical to human action recognition. In this paper, we propose a novel action-stage (*Action*S) emphasized spatio-temporal Vector of Locally Aggregated Descriptors (*Action*S-ST-VLAD) method to aggregate informative deep features across the entire video according to adaptive video feature segmentation and adaptive segment feature sampling (AVFS-ASFS). In our *Action*S-ST-VLAD encoding approach, by using AVFS-ASFS, the key frame features are chosen and the corresponding deep features are automatically split into segments with the features in each segment belonging to a temporally coherent *Action*S. Then, based on the extracted key frame feature in each segment, a flow-guided warping technique is introduced to detect and discard redundant feature maps, while the informative ones are aggregated by using our exploited similarity weight. Furthermore, we exploit an RGBF modality to capture motion salient regions in the RGB images corresponding to action activity. Extensive experiments are conducted on four public benchmarks – HMDB51, UCF101, Kinetics and ActivityNet for evaluation. Results show that our method is able to effectively pool useful deep features spatio-temporally, leading to state-of-the-art performance for video-based action recognition.**

***Index Terms*—Action Recognition, Feature encoding, Adaptive video feature segmentation, Adaptive feature sampling, *Action*S-ST-VLAD.**

I. Introduction

Recognizing human actions in videos is one of the fundamental problems in computer vision [1], [2], [3], and has received much attention in both the research community and industry owing to its wide range of applications, e.g., video surveillance [10], robot navigation, and human behavior analysis [7], [9]. While a wide range of learning architectures are available, the paradigm which splits the action recognition problem into three main steps [11] – feature extraction, feature encoding, and classification – currently gives more favorable results than other designs. Each step has a significant influence on the accuracy, and thus many studies have been carried out to address these problems [2], [12], [22]. In this paper, we focus on deep feature encoding for videos to model long-range spatio-temporal structure for action prediction.

Traditionally, hand-crated features [23], [25], [26], [27] are exploited for action recognition. However, such features can be easily affected by illumination, weather condition, viewing angle, and other imaging conditions. Recent years, deep learning features obtained via convolutional neural networks (CNNs) demonstrated better robustness and discriminative power, and accordingly have become the new trend for video-based action recognition [2], [4], [5], [6], [18], [19].

Interestingly, although CNN approaches have achieved great success in recognizing objects in still images [31], [32], [33], [17], they have not yet gained significantly better performance over the hand-crated methods when dealing with video analytics tasks [7]. One major drawback is that CNNs are not directly suitable for videos since they lack the ability to capture long-range spatio-temporal information in the features [29].

Since videos include two types of complementary cues: appearance and motion, to extract spatio-temporal information for action recognition in videos, Simonyan *et al*. [2] presented a two-stream network (TS-Net). A TS-Net consists of two deep convolutional networks (ConvNets) to individually operate on RGB images to extract spatial features, and on optical flow images [35] to learn motion features. The TS-Nets are extensively studied and obtained competitive results compared to other deep architectures due to the following main characteristics [7], [12], [20]: 1) Employing new ultra-deep architectures, e.g., BN-Inception [36], Res-Nets [37], etc.; 2) Pre-training on large-scale image datasets; 3) Exploiting new complementary modalities to combine with the RGB appearance and optical flow [3], [21], [28]. Inspired by these effective good practices, we propose a three-stream network (3S-Net) that consists of three streams coming from three modalities (Figure 3), RGB appearance, optical flow and RGBF, where the third modality dubbed RGBF is formulated by fusing the RGB image and the optical flow image to form a third stream that attempts to capture motion salient regions [39].

More importantly, to address the video-level long-range feature encoding issue, we exploit an effective video representation by integrating the 3S-Net with trainable spatio-temporal

Zhigang Tu is with the State Key Laboratory of Information Engineering in Surveying, Mapping and Remote Sensing, Wuhan University, 430079 Wuhan, China, E-Mail: (tuzhigang@whu.edu.cn).

Hongyan Li is with School of Information Engineering, Hubei University of Economics, Wuhan, China, and State Key Laboratory of Information Engineering in Surveying, Mapping and Remote Sensing, Wuhan University, E-Mail: (hongyanli2000@126.com), (*Corresponding author*: Hongyan Li).

Dejun Zhang is with the School of Information Engineering, China University of Geosciences, 30074 Wuhan, China, E-Mail: (zhangdejun@cug.edu.cn).

Justin Dauwels are with the School of Electrical and Electronic Engineering, Nanyang Technological University, 637553 Singapore.

Baoxin Li is with the School of Computing, Informatics, Decision System Engineering, Arizona State University, AZ, 85287 USA, E-Mail: (baoxin.li@asu.edu).

Junsong Yuan is with the Computer Science and Engineering department, State University of New York at Buffalo, NY, 14260-2500, USA, E-Mail: (jsyuan@buffalo.edu).

Manuscript received Dec. 28, 2018.

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TIP.2018.2890749, IEEE Transactions on Image Processing

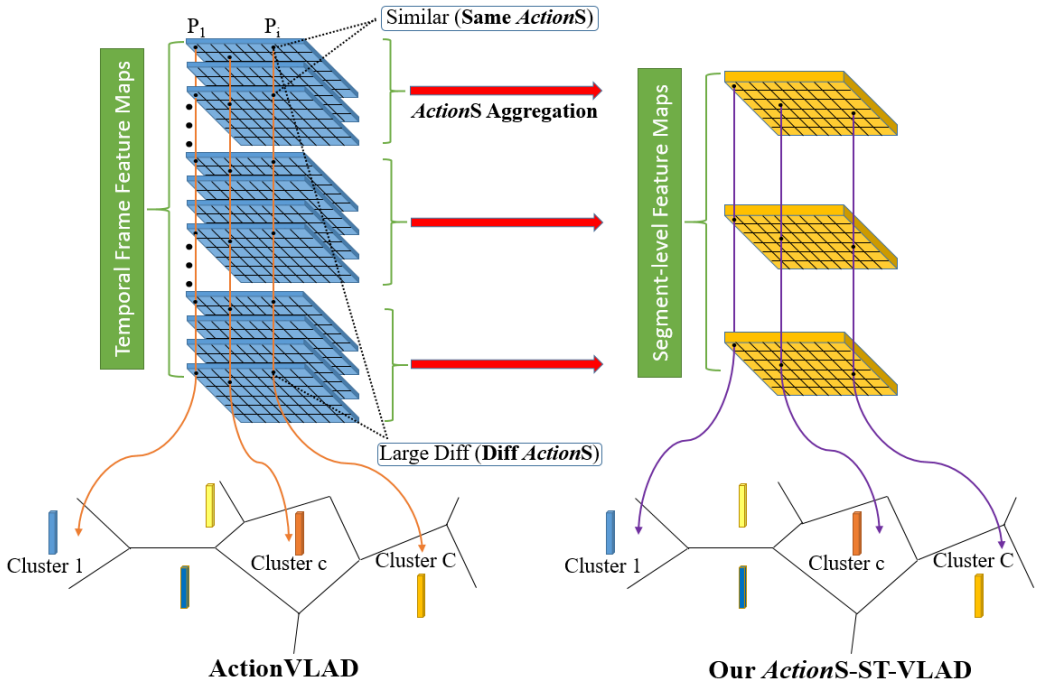

Fig. 1. The *Action*S-ST-VLAD strategy for deep features encoding. Compared to ActionVLAD [12] which concerns only the sub-actions, our *Action*S-ST-VLAD regards both different sub-actions spatially and different action-stages temporally. The local features are pooled in each *Action*S first, and then encoded across the whole video over all the *Action*S-based video-segments to form a video representation.

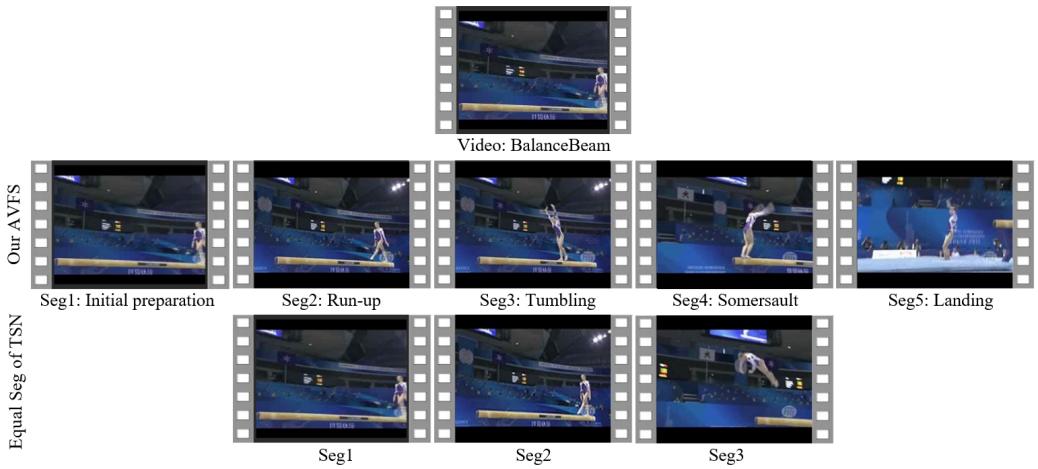

Fig. 2. The proposed AVFS method enables to temporally split the video deep features into *Action*S-based video-segments. For instance, a "BalanceBeam" action usually consists of multiple action-stages: (i) initial preparation, (ii) run-up, (iii) tumbling, (iv) somersault, and (v) landing.

deep feature aggregation through the following strategies:

1) **Building video-level representation via *Action*S-ST-VLAD**. Due to the limitations on GPU memory, for long videos, we are unable to load the entire video for training a CNN. Usually, one may process individual RGB frames and short snippet of motions (e.g., 10-staked optical flow [2], [4]). However, this method mostly ignores the long-term temporal structure of the video, which may be critical to recognizing realistic human actions. Our basic motivation, similar to [6], [12], [13], is to design a pooling strategy to learn a global video representation for modeling long-term video structure and for predicting complex human actions.

According to some recent research [11], [12], [13], [14], the super-vector-based encoding method – Vector of Locally Aggregated Descriptors (VLAD), which was initially presented in [7] – has demonstrated its superiority over other encoding approaches in many tasks involving aggregation of deep features. In this work, we present an action-stage (*Action*S) emphasized spatio-temporal VLAD (*Action*S-ST-VLAD) method, which is derived from the effective video encoding technique ActionVLAD [12], by incorporating adaptively captured action-stages (see Figure 1). One limitation of ActionVLAD is that it disregards the fact that each action is composed of multiple temporal coherent action-stages and every *Action*S may have its own effect on action recognition. To this end, we adaptively split the video into segments

based on *Action*S, dubbed adaptive video feature segmentation (AVFS; Figure 2). A spatial-temporal VLAD (ST-VLAD) is used to encode the segment-level deep feature maps $\{fV_{S_k}\}$ into a video representation. In this way, not only convolutional descriptors over the whole spatio-temporal extent of the video are aggregated, but also the specific cluster of features regarding their temporal location, i.e., *Action*S, is considered.

2) **Selecting informative deep features**. Since the image content varies slowly across video frames, successive frames are highly redundant. To reduce the redundancy, current CNNs usually utilize sampled video frames, by uniform or random sampling [6], [7], [12]. For example, ActionVLAD used just 25 sparse sampled frames per video for processing. However, human action recognition is inherently different from a problem like object verification in video [24], in that different frames may correspond to different states of an action. Therefore, uniform or random sampling may miss some important frames while keeping some frames that are not informative (redundant/insignificant) or even harmful (e.g., noisy frames) for recognition. To handle this problem, in each *Action*S derived video-segment, we propose a novel sampling method, called adaptive segment feature sampling (ASFS), which uses the deep feature flow warping method [38] to select valid feature maps. The preserved feature maps after ASFS in each segment are aggregated to form a video-segment feature map fV_{S_k}.

By integrating the AVFS-ASFS with ST-VLAD, the overall performance of our *Action*S-ST-VLAD on recognizing human actions is significantly enhanced compared to ActionVLAD.

In summary, our approach has the following contributions:

- A novel feature encoding technique, *Action*S-ST-VLAD, which considers different action-stages, is designed to capture the long-term temporal structure by pooling deep features over the entire video to obtain a video representation for action classification.
- Taking into account action-stages, an AVFS approach is proposed to group deep features in temporal coherent clusters adaptively based on the special action stages of individual videos.
- An ASFS method is presented to sample every *Action*S based video-segment to select discriminative feature maps and discard redundant/insignificant/noisy ones. The AVFS-ASFS method can be extended to other video tasks which require segmentation and sampling.
- A new RGBF modality, which takes into account motion distinctive regions associated with human actions, is designed to formulate a third stream to learn complementary features for other two streams to boost the performance of action recognition.

II. Related Work

Recent years have seen significant efforts devoted to action recognition in videos, especially those employing CNNs. It is beyond the scope of this work to discuss all the related works. Instead, we focus on deep CNN methods in three aspects: 1) flow-guided feature warping; 2) exploiting effective stream networks; 3) learning a video-level representation to model long-term temporal structure for action prediction.

Flow-guided feature warping: Zhu *et al*. [38] proposed a deep feature flow method for video recognition. By selecting sparse key frames, the deep features of other frames can be obtained by propagating the features from them via flow-guided warping. This achieves better computational tractability as the flow estimation and feature propagation are much faster than the computation of deep features. To solve the difficulties of object detection in videos due to motion blur, video defocus, etc., Zhu *et al*. [45] modified the per-frame features by aggregating nearby features along the motion paths through flow-guided feature warping.

Stream Networks for action recognition: To model spatial and temporal information jointly, Simonyan *et al*. [2] introduced the TS-Net to process an optical flow stream and an appearance stream. To improve the performance of [2], some semantic cues have been exploited. For example, Cheron *et al*. [40] utilized human pose positions to detect informative regions and extract deep features from these body joints. Wang *et al*. [7] exploited two visual cues – RGB difference and warped optical flow fields, to produce two additional modalities for the TS-Net. In some other work, the semantic cues were used to construct new streams of features. Singh *et al*. [42] employed a state-based tracker to detect the bounding box of the person, and two person-centric appearance and motion streams are constructed. Tu *et al*. [3] applied the IB-RPCA [3] to obtain two human-related regions. Based on the detected regions, two other TS-Nets are formed. We propose an RGBF modality to construct a third stream to pay attention to the motion salient part of the RGB image.

Different ways were studied in [4] to combine the appearance and motion features to take advantage of the spatio-temporal information in a best way. Wang *et al*. [5] exploited a multi-layer pyramid fusion strategy to replace the fusion method of [4], which is able to integrate the spatial and temporal features at multiple levels. Following [12], we fused the video representations of the three streams at the last convolutional layer through weighted sum.

Learning a video-level deep representation: To capture the long-range temporal structure for video-based action recognition, many schemes have been proposed [5], [6], [7], [11], [12]. Duta *et al*. [11] exploited Spatio-Temporal Vector of Locally Max Pooled Features (ST-VLMPF) to learn a general video representation that combines the deep features over the whole video with two different assignments, and they performed two max-poolings and one sum-pooling for each assignment. Kar *et al*. [6] presented an AdaScan approach, which pools the sampled video frames across the video based on the predicted discriminative importance of each frame. To learn a more global video representation, Wang *et al*. [5] sampled optical flow frames by multi-path temporal subnetworks that shared network parameters in a longer sequence. An STCB operator is designed to integrate the spatial and temporal features effectively.

The works most related to ours are [7], [12]. Wang *et al*. [7] designed a temporal segment network (TSN), which uses a sparse sampling scheme to extract short snippets in manually obtained video segments over the video. A consensus function is utilized to aggregate the preliminary prediction of each

snippet. However, the artificial equal segmentation method is unable to capture the video segments with the concern of *Action*S flexibly. In addition, the sparse sampling strategy is unable to remove insignificant frames while preserving useful ones automatically. Girdhar *et al*. [12] modified the NetVLAD [43] with a trainable spatio-temporal extension to aggregate the deep features both spatially and temporally. A vocabulary of "action words" is used to aggregate descriptors into a video-level fixed-length vector. However, since they group the features into clusters only based on the position of the anchor points spatially without regard for the temporal coherence, one cluster descriptors usually spans several action-stages temporally. We address this issue by dividing the video into action-stages and encoding the frame features in each *Action*S separately (see Figure 2).

III. The Proposed Method

In this section, we give a detailed description of our encoding method – *Action*S-ST-VLAD, which can be decomposed into two main steps: 1) Adaptive video-level *Action*S emphasized deep features segmentation and segment-level features sampling, and 2) Learning an entire video representation via ST-VLAD in each stream, and fusing the representations of all steams for final prediction. To the best of our knowledge, selecting informative deep features has not be done before in video-based action recognition. Additionally, our approach employs a third stream RGBF (Figure 5), which attempts to capture motion-salient regions.

A. Deep Features Segmentation and Sampling

Since most of the ConvNet frameworks [2], [3], [4] are operated on single RGB frame or stacked flow frames in a short snippet, they are unable to model the long-range temporal structure of the video actions. However, realistic human actions usually span a long time. It is difficult to distinguish two actions that have similar appearance over short horizons [5]. Moreover, complex actions usually consist of multiple action-stages where each *Action*S has coherent motion pattern with a specific intention [30]. As shown in Figure 2, "BalanceBeam" normally includes the action-stages of (1) initial preparation, (2) run-up, (3) tumbling, (4) somersault, and (5) landing. For action prediction, the features should be pooled in each *Action*S and then further aggregated across all the action-stages temporally. Consequently, we need to improve the ConvNet to enable it not only to learn the visual representation over the whole video, but also to analyze the spatio-temporal characteristic of the video with regard to (1) action-stages and (2) the discriminative importance of each video frame.

Drawbacks of general action recognition methods: 1) Almost all the previous action recognition methods ignore the important segmental coherent characteristic. Recently, to handle this problem, the recent TSN [7] divides a video into 3 segments with equal duration $V = \{S_1, S_2, S_3\}$ to address the temporal segmentation issue. This scheme is useful to improve the recognition performance. However, it disregards the facts that different *Action*S lasts different time duration, and different actions contain different numbers of action-stages. Thus equal segmentation (ES) is not an optimal choice.

2) Consecutive video frames are highly redundant and the redundancy is more severe for deep features [38], and thus most of them do not contribute much to video-based action recognition. An even greater concern is that some noisy frames would degrade the accuracy of action recognition.

To reduce the redundancy, the popular way is to use a uniform or random sampling strategy [2], [4], [6], [12] to down-sample the videos. In ActionVLAD [12], they select $T = 25$ frames per video randomly to learn and evaluate the video representation. In TSN [7], they utilize a sparse temporal sampling method to down-sample each video segment S_k into a short snippet T_k where the samples are uniformly distributed along the timeline. However, such a uniform or random down-sampling strategy does not consider the discriminative importance of each frame, e.g., some informative frames would be removed while some redundant/insignificant/noisy frames would still be preserved. To handle the noisy frames, AdaScan [6] proposed an 'Adaptive Scan Pooling (AdaScan)' method to weighted mean pooling feature maps with the weights represented by the predicted discriminative importance scores. Since the 'Adaptive Pooling' is conducted by recursively predicting a score that measures the distinctiveness of the current frame across the video, it is computational expensive. To decrease the cost time, [6] samples 25 frames from each video uniformly and conduct the 'Adaptive Pooling' on these sampled 25 frames. In this case, compared to [7], [12], the redundant/insignificant/noisy frames can all be tackled. However, some informative frames may be discarded. Consequently, exploiting a method that (1) *is able to quantify the importance of each frame* and (2) *has a proper running speed* is a desirable task.

We propose a novel AVFS-ASFS method to address these two issues as follows:

1) **Adaptive Video Feature Segmentation (AVFS)**. For a input video *V*, we first extract the deep feature of each frame f_n:

$$F = \{f_1, ..., f_n, ..., f_N\} \tag{1}$$

where *N* is the frame number of the video *V*.

According to the extracted deep features, we apply *k-means* to learn a Codebook *C* [44] with a vocabulary of K' "visual words" $C = \{c_1, c_2, ..., c_{K'}\}$ based on the feature similarity, and each frame feature will be assigned a label without the temporal consistency constraint (see Figure 4). Generally, for the first *Action*S segment S_1, the first frame f_1 will be selected as the key frame f_{SK_1}, and so on for other *Action*S segment S_k. There is an elemental principle for two local features f_i and f_j: if they are located in two different action-stages, their difference is large; conversely, if they are distributed in a same *Action*S, their difference is small. To determine whether the feature maps from f_2 to f_i belong to a same *Action*S as f_{SK_1}, we utilize the cosine similarity (CS) for evaluation (taking the first *Action*S segment S_1 as the example):

$$SF(SK_1, i) = \frac{|f_{SK_1} \cdot f_i|}{\|f_{SK_1}\| \|f_i\|} \tag{2}$$

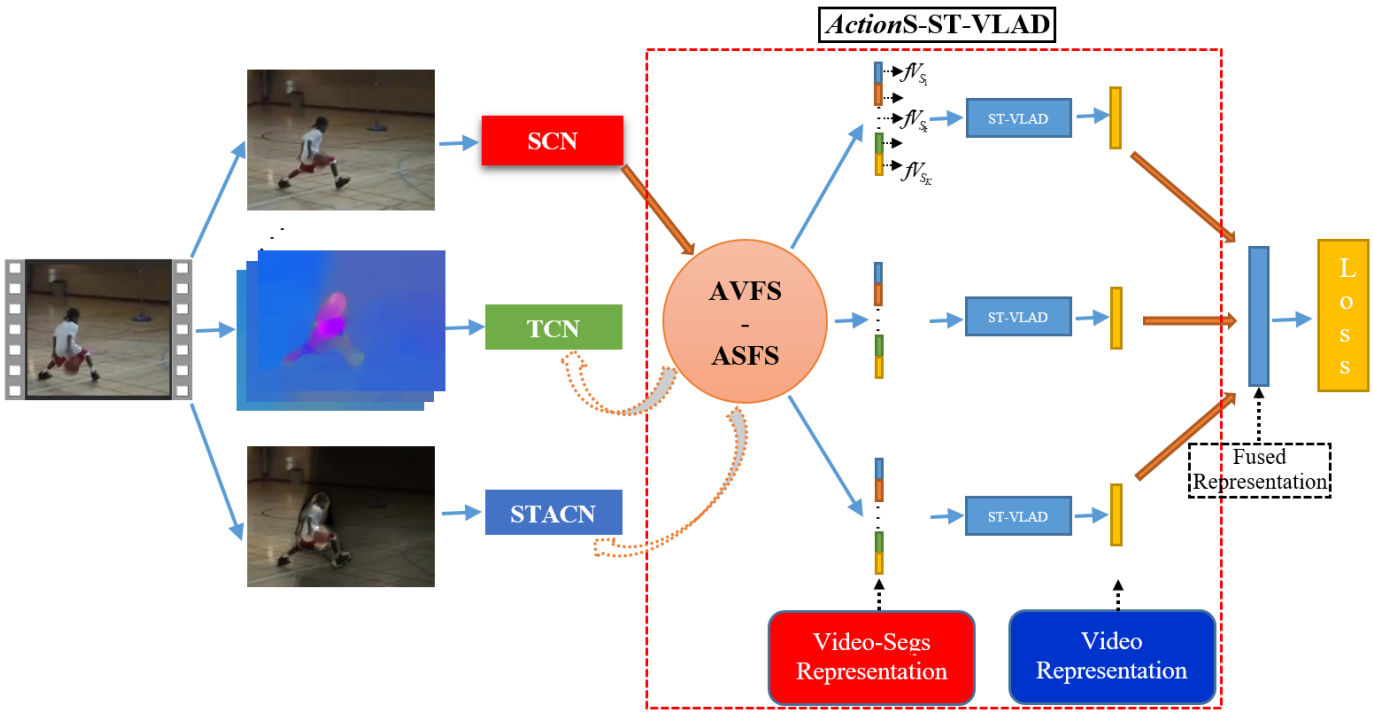

Fig. 3. The network architecture of our method, which consists of four main components: 1) Deep feature extraction. Three stream ConvNets, i.e., SCN, TCN, and STACN, are constructed to extract three types of deep features. 2) Deep feature clustering and sampling. An AVFS-ASFS is proposed to adaptively divide the extracted deep features into *Action*S-based video-segments and sample informative feature maps in each segment. 3) Deep feature encoding. A ST-VLAD method is used to learn a whole video-representation to model long-range spatio-temporal structure of video actions in each stream. 4) Video representations are fused for final action prediction.

If the deep feature similarity $SF(SK_1, i)$ is smaller than τ_1, the frames from f_1 to f_{i-1} will be grouped as the first segment $S_1 = \{f_1, ..., f_{i-1}\}$, and f_i is classified to the next *Action*S segment S_2 and considered as the second key frame f_{SK_2}.

Segmentation updating: As shown in Figure 4, for *k-means* operator, if f_{i-1} and f_i have the same label, we will update f_i and re-classify it to S_1. Besides, the neighbors of f_i, like f_{i+1}, will also be grouped into S_1 if they have the same labels as f_i. By following this pattern, all the key frame features, their corresponding *Action*S segments, and the selective feature maps in each segment will be automatically determined:

$$\begin{aligned} f_{SK} &= \{f_{SK_1}, ..., f_{SK_k}, ..., f_{SK_K}\} \\ V &= \{S_1, ..., S_k..., S_K\} \\ S_k &= \{f_{S_1}, ..., f_{S_m}, ..., f_{S_M}\} \end{aligned} \tag{3}$$

2) **Adaptive Segment Feature Sampling (ASFS)**. It is necessary to sample the video to compress redundant frames which have no contribution to improved action recognition. In each *Action*S segment, after locating the key frame feature map f_{SK_k}, we apply the flow-guided warping strategy of [45] to select the discriminative local features by comparing the similarity between f_{SK_k} and the warped feature maps of other frames.

For two consecutive video frames I_i and I_{i+1}, we compute their optical flow $\mathbf{F}(i, i+1)$, which describes the pixel correspondence between I_i and I_{i+1} [46], according to the deep CNN-based method FlowNet2 [47]. For the non-consecutive frames I_i and I_j, their optical flow $\mathbf{F}(i, j)$ is calculated by compositing the intermediate flow fields $\mathbf{F}(i, i+1)$.

To execute propagation, we first bilinearly resize the flow field $\mathbf{F}(i, j)$ to the same resolution of the feature maps. Then, in one segment S_k, each feature map f_{S_i} is warped to the key frame f_{SK_j} with the resized $\mathbf{F}(i, j)$:

$$f_{S_{i\to j}} = W(f_{S_i}, \mathbf{F}(i, j)) \tag{4}$$

where $W(\cdot)$ is the spatial warping function that works on all the positions for every channel in the feature maps by bilinear interpolation [38]. j and i respectively denote the indices of the key frame and the other frames in S_k.

To sample S_k, a similarity measure (SM), which evaluates the importance of the non-key frames to the key frame, is proposed by computing the comparability between $f_{S_{i\to j}}$ and f_{SK_j}:

$$SM(i, j) = \frac{exp(-\|f_{S_{i\to j}} - f_{SK_j}\|^2)}{\sum_{i'} exp(-\|f_{S_{i'\to j}} - f_{SK_j}\|^2)} \tag{5}$$

where i' denotes the index of any frame in S_k. If $i' = j$, $f_{S_{i'\to j}} = f_{SK_j}$, and $SM(j, j)$ is also computed according to Eq. (5). $SM(i', j) \in (0, 1)$.

The sampling is operated in terms of two conditions based on $SM(i, j)$. **Cond 1**: If the warped feature $f_{S_{i\to j}}$ is closely similar to the key frame feature f_{SK_j}, i.e., $SM(i, j) > \tau_2$, we should remove its corresponding local feature f_{S_i} in the subsequent procedure. **Cond 2**: If the warped feature $f_{S_{i\to j}}$ is largely different to f_{SK_j}, i.e., $SM(i, j) < \tau_3$, we treat its

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TIP.2018.2890749, IEEE Transactions on Image Processing

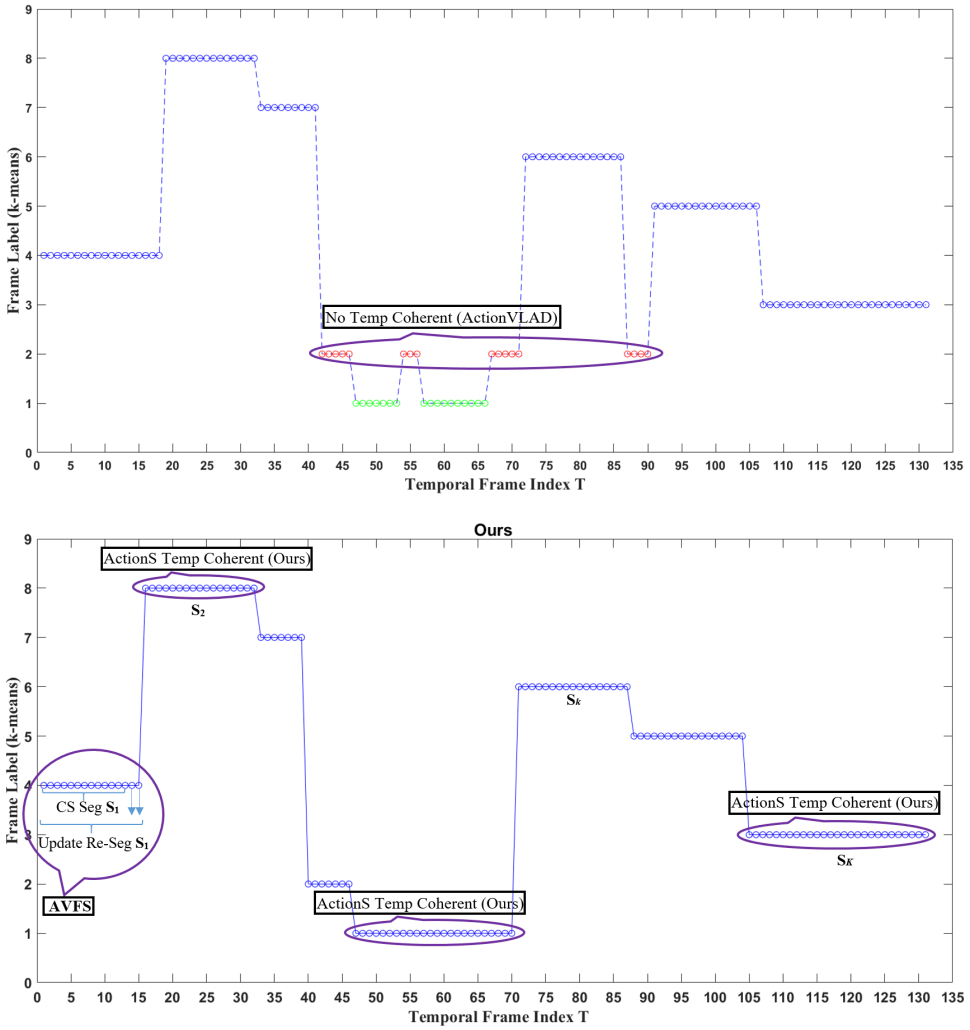

Fig. 4. The working scheme of our deep feature segmentation – AVFS (Using the action video "Lunges" in UCF101 [48] for explain). At the bottom, a segmentation updating strategy is designed by us: f_i will be updated and re-classified to the segment of f_{i-1} if f_{i-1} and f_i have the same label; the neighbors of f_i will also be classified to the segment of f_i if they have the same labels.

corresponding frame as noise and discard its deep features. (We set $\tau_2 >= 0.5$ and $\tau_3 \in (0, 0.05]$ experimentally)

The obtained informative local deep features in each video *Action*S segment S_k are pooled via weighted average, using the similarity weight to form a segment-level deep feature map fV_{S_k}:

$$fV_{S_k} = \frac{1}{\sum_{i'} SM(i', j)} (\sum_{i'} SM(i', j) f_{S_{i'}}) \tag{6}$$

The fV_{S_k} is then L2-normalized [6] to obtain the final segment-level feature map. The deep features for the entire video can be expressed as:

$$fV = \{fV_{S_1}, ..., fV_{S_k} ..., fV_{S_K}\} \tag{7}$$

Since we select the deep features of the last convolutional layer for processing, the proposed method can be applied to nearly all CNN architectures, such as VGG-Net [32], BN-Inception [36], and Res-Nets [37].

B. ActionS-ST-VLAD for Video Representation Aggregation

In each stream, how to combine multiple *Action*S-based segment-level deep features $fV_{S_k} (fV_{S_k} \in fV)$ to construct a single video-level representation for the entire video is crucial for final action recognition. ActionVLAD [12], which is end-to-end trainable, is able to aggregate deep features frame by frame over space and time to form a video representation. However, ActionVLAD only concerns sub-actions: dividing the descriptor space into cells according to a vocabulary of "action words" spatially, and then applying the *Sum* to aggregate the residual vectors inside each of the clusters over the whole video. It ignores that one cluster usually crosses multiple action-stages temporally. As shown in Figure 1, at a position P_i in cluster C_k, if the feature of frame f_{ni} and frame f_{mi} are located in different action-stages, their difference is large and their action intentions are different. Therefore, directly aggregating them only based on position over time is not good for formulating a feature representation to model the long-term temporal structure of a video action. With the consideration of *Action*S, the encoded segment-level

feature map fV_{S_k} avoids this drawback of ActionVLAD, as the pooled features in fV_{S_k} follow one action intention. Significantly, we exploit an *Action*S-ST-VLAD which applies the spatio-temporal VLAD (ST-VLAD) of [12] to aggregate the calculated fV_{S_k} to obtain a final video representation FV:

$$FV(j,l) = \sum_{S_1}^{S_K} \sum_{i=1}^{N} \frac{e^{(-\alpha \|x_{iS_k} - c_l\|^2)}}{\sum_{l'} e^{(-\alpha \|x_{iS_k} - c'_l\|^2)}} (x_{iS_k}(j) - c_l(j)) \tag{8}$$

where $x_{iS_k}(j)$ and $c_l(j)$ are respectively the j-th components of the local deep feature x_{iS_k} and the anchor point c_l. $x_{iS_k} \in \Re^D$, is a D-dimensional local descriptor extracted from spatial location $i \in \{1, ..., N\}$ and temporal S_k-th segment-level deep feature map of a video ($S_k \in \{S_1, ..., S_K\}$, see Eq. (7)). The descriptor space $\Re^D$ is divided into L clusters using a vocabulary of L "action words" represented by anchor points $\{c_l\}$. α is a tunable hyper-parameter. The second term $x_{iS_k}(j) - c_l(j)$ is the residual between the descriptor and the anchor point of cell l in the video segment S_k. The aggregated descriptor are L2-normalized as [12] to form a single descriptor with the vector size of $L \times D$. In our framework, the size of the final video representation is $FV = 64 \times 1024$, where we set $L = 64$ following [12], and 1024 is the dimension of the learned local deep feature. The parameters, e.g., the deep feature extractor, anchor points $\{c_l\}$, and classifier, can be learnt end-to-end jointly.

In general, our aggregation strategy is an extension of ST-VLAD [12] with the improvements of *Action*S-based segmentation, sampling, and pooling. Since our *Action*S-ST-VLAD considers not only the spatial coherence of sub-actions but also their temporal consistency, it is able to select the distinctive deep features, leading to a performance much better than ActionVLAD.

C. Spatio-temporal Attention Modality – RGBF

To model the dynamic motion of human actions effectively, we design an RGBF modality (see Figure 5), which can localize the motion salient regions that correspond to human activities, by combing the RGB image and a movement confidence map (MCM) derived from the optical flow:

$$RGBF(x,y) = Im(x,y) \times \frac{MF(x,y) - MF_{min}}{MF_{max} - MF_{min}} \tag{9}$$

where (x, y) represents a pixel location. Im denotes one RGB frame of a video, and MF is the magnitude of its corresponding flow field $\mathbf{F}$. MF_{max} and MF_{min} represent the maximum and minimum flow magnitude of $\mathbf{F}$. Before combination, we linearly transfer MF to a MCM (the second term of Eq. (9)), which indicates the probability of where and how strong the motion occurs, by re-scaling each of the flow magnitude value to a range of $[0, 1]$. Figure 5 shows that RGBF highlights the motion discriminative part of the human body related to action, and hence potentially supplies complementary information to the appearance and motion modalities. For example, in contrast to RGB image, insignificant features, e.g., the static background information that is irrelevant the action, is greatly compressed. In contrast to optical flow, valid appearance information is added.

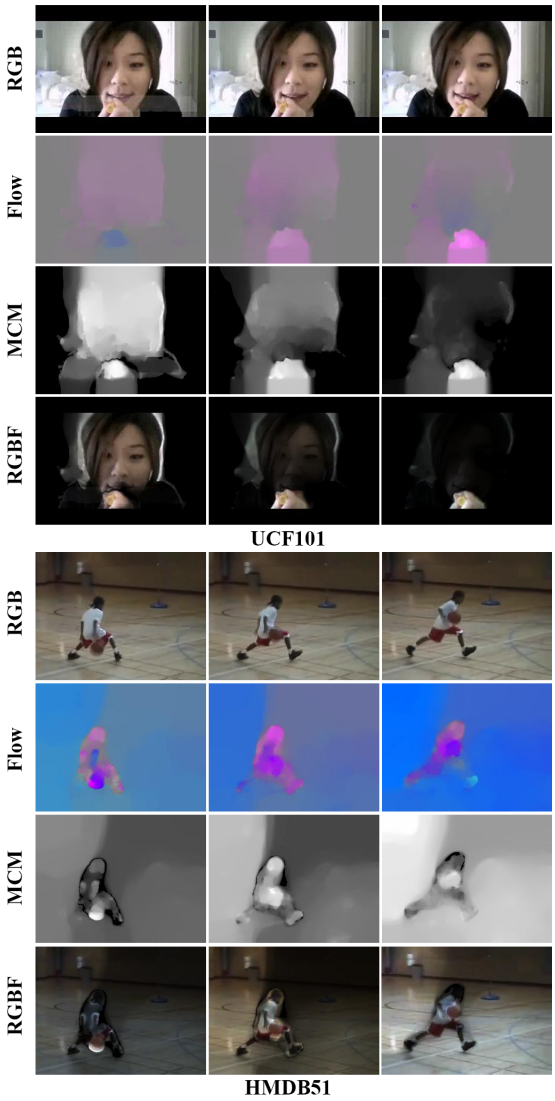

Fig. 5. The comparison of the RGB, Flow, MCM, and RGBF on the UCF101 and HMDB51 datasets.

IV. Experiments

In this section, to evaluate the proposed AVFS-ASFS based *Action*S-ST-VLAD method, four challenging publicly available human action recognition benchmarks are selected for experimenting: UCF101 [48], HMDB51 [49], Kinetics [50] and ActivityNet [51]. Extensive experiments are conducted to test the effect of various parts of our approach, e.g., (1) the respectively influence of AVFS, ASFS and AVFS-ASFS; (2) the performance of the ActionS-ST-VLAD pooling; (3) the effect of the RGBF modality. Finally, we compare our method with the state-of-the-arts on both trimmed and untrimmed videos to verify whether our proposed method is effective to recognize human actions.

A. Datasets and Implementations

UCF101 contains 13320 videos belong to 101 action categories. For one class, it includes more than 100 video clips, and for one video clip, it contains an average length of

180 frames. They span a large range of activities such as sports and human-object interaction. It is a complex dataset as the captured videos varies significantly in scale, viewpoint, illumination, and camera motion.

HMDB51 consists of 6766 action videos which have been grouped to 51 action classes. The videos are collected from a large number of sources, including movies and online videos. It is challenging for action recognition due to the issues of videos: 1) the quality is poor; 2) a wide range of variations in which actions occur [56]; 3) contain strong camera motions.

Kinetics is a large-scale well-labeled video dataset used for action recognition. It includes more than 300000 trimmed videos covering 400 human action classes, and for each action, there are more than 400 video clips. The videos are collected from realistic, challenging YouTube, and each video is temporally trimmed last around 10 seconds.

ActivityNet (V1.3) is an untrimmed video dataset, which contains 849 hours of video and 28108 action instances. 200 human action categories with an average of 137 videos per category. ActivityNet is split into three different subsets randomly, i.e., training, validation, and testing. In particular, 50%, 25% and 25% videos are respectively utilized for training, validation and testing.

On HMDB51 and UCF101, we follow the original evaluation measure using three training/testing splits, and the final results are obtained by averaging the accuracy over the three splits. On ActivityNet, the mean average precision (mAP) is applied to evaluate the performance.

Deep Feature Extraction. We design an architecture consists of three streams to extract three types of deep features: a spatial convolutional network (SCN) is used to extract the appearance information on RGB images, a temporal convolutional network (TCN) is employed to capture the motion information on flow fields, and a spatio-temporal attention convolutional network (STACN) is utilized to learn the appearance information with attention to motion salient regions. For each input video, we downsample it equally with a ratio of 0.5. Specifically, 1) the spatial-stream model of [7] with fine-tuning is introduced for our SCN to extract features of every frame. The features at the last convolutional layer (Inception5b) are selected for processing. For each frame, the output of Inception5b is a descriptor with a spatial size of 7×7 with 1024-dimensional descriptors. For a video, before our AVFS-ASFS, we get in total $\#\lfloor 0.5 \times frames \rfloor \times 49$ deep features. 2) The temporal-stream model of [7] with fine-tuning is applied to our TCN. FlowNet2 [47] is used to compute the optical flow. For each input flow image to our TCN, it contains 20 channels which is formulated by stacking 10 consecutive *x* and *y* direction flow components. We can obtain $\#(\lfloor 0.5 \times frames \rfloor - 9) \times 49$ feature maps with each has 1024 dimensions. 3) We use the cross modality pre-training technique to train a STACN model with the BN-Inception architecture as [7]. In which we use the appearance-stream model of [7] as the initialization and then finetuned with our RGBF images. At Inception5b, we can get $\#(\lfloor 0.5 \times frames \rfloor - 1) \times 49$ features in total.

AVFS-ASFS for Flow and RGBF. For the other two modalities – optical flow and RGBF, their temporal *Action*S segments $\{S_k\}$, the key frame feature maps $\{f_{SK_k}\}$, and the selected informative feature maps after sampling in each segment, are directly obtained according to their corresponding information in the RGB modality. To aggregate their feature maps in a segment, the weight between the key frame and one of the other frames is computed via the CS measure without the flow guided warping (refer to Eq. (2)).

Fusing video representations of the three streams. The video representations of the three streams are fused at the last convolutional layer via weighted average as [12] followed by L2-normalization, which enables our architecture to be optimized by a unified spatio-temporal loss function ene-to-end learnable.

For fair comparison and analysis, from sub-section *B* to *C*, we select the two-stream architecture with the pre-trained SCN and TCN models of [7] as the baseline.

B. Effect of AVFS-ASFS

1) **Evaluation of AVFS**. For this testing, in each segment, the deep features are encoded by mean pooling. Figure 6 shows the performance of AVFS with the setting of different τ_1 on the appearance-stream. If τ_1 is small, the number of feature segments is small. For most of the videos in UCF101, the number of segments reduced to 1 if $\tau_1 < 0.5$. The best result is obtained when $\tau_1 = 0.85$, and we set $\tau_1 = 0.85$ for all the other experiments. Figure 7 shows the obtained segments due to our AVFS. Each segment corresponds to a different *Action*S, which is different from other segments visually if an action contains multiple action-stages. For example, with our AVFS, "LongJump" is divided to 6 action-stages, e.g., Run-up, Long-jump, Landing, Clawing on the ground, Standing, and Leaving. In contrast, with the normally used ES method of TSN, segments S_1 and S_2 are confused, some frames in S_2 share the same action phase (Run-up) as S_1, and some other frames in S_2 spans several other action-stages. For "Smile", as all frames in the video have one only action intention, our AVFS treats the entire video as one *Action*S, while the ES separates it into three repeated segments.

Table I compares the performance of two segmentation strategies on the spatial-stream (*Spa-Stream*): our AVFS and the ES approach. Our AVFS outperforms the ES at least by 0.5% (86.3% vs 85.8% (S=7)). If the number of segments is small, e.g., S=1, the results are poor. This is because an action is usually consist of several action-stages, and each *Action*S has its own intention and makes a different contribution to the entire action. Thus, we should decompose an action video into *Action*S segments instead of treating the human action as one whole phase or several equal phases that without physical significance.

TABLE I
EVALUATION (ACC.(%)) OF AVFS ON UCF101 SPLIT 1 OF THE SPATIAL-STREAM (*Spa-Stream*).

Methods	AVFS	S=1	S=2	S=3	S=5	S=7	S=9	S=11
Spa-Stream	**86.3**	81.9	84.0	85.6	85.7	85.8	85.6	85.5

2) **Evaluation of ASFS**. Table II shows the results of different sampling strategies. *N* is the number of sampled

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TIP.2018.2890749, IEEE Transactions on Image Processing

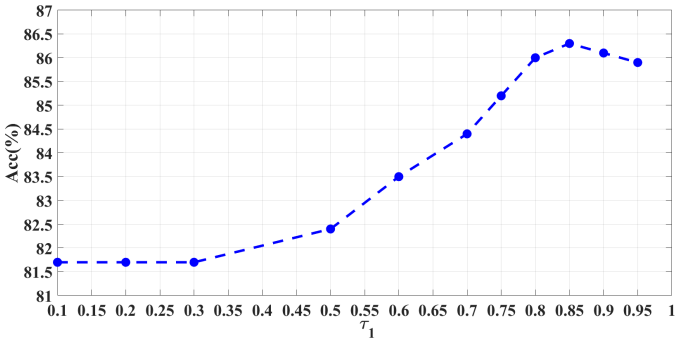

Fig. 6. The performance of AVFS with different τ_1 on UCF101 split 1 of the spatial-stream (*Spa-Stream*).

Fig. 7. The selected key frame of each segment of our AVFS method ($\tau_1 = 0.85$) compare to the equal segmentation (ES) approach of TSN on HMDB51 ("DrawSword" and "Smile") and UCF101 ("Lunges" and "LongJump").

frames. The accuracy of using all the frame features is not much better than only employing uniformly sampled features while the time cost is much higher, which demonstrated that the successive video frames are redundant and most of them are not helpful for action recognition. In contrast, the proposed ASFS is able to select the informative frames and discard useless ones, thus its performance is boosted by 0.6% compared to the widely used sparse sampling – N = 25 [12], [6], and is enhanced by 0.3% compared to the none sampling – No-Sampling [11]. Figure 8 shows that the influence of the threshold parameters τ_2 and τ_3 are insignificant if we set $\tau_2 >= 0.5$ and $\tau_3 <= 0.05$.

TABLE II
EVALUATION OF ASFS ON UCF101 SPLIT 1 (ACC.(%))/ACTIVITYNET V1.2 (MAP(%)) OF THE SPATIAL-STREAM (*Spa-Stream*).

Methods	ASFS	N=25	N=50	N=100	No-Sampling
Spa-Stream	**86.0**	85.5	85.6	85.7	85.7

3) **Evaluation of ASFS-AVFS**. Table III shows the overall

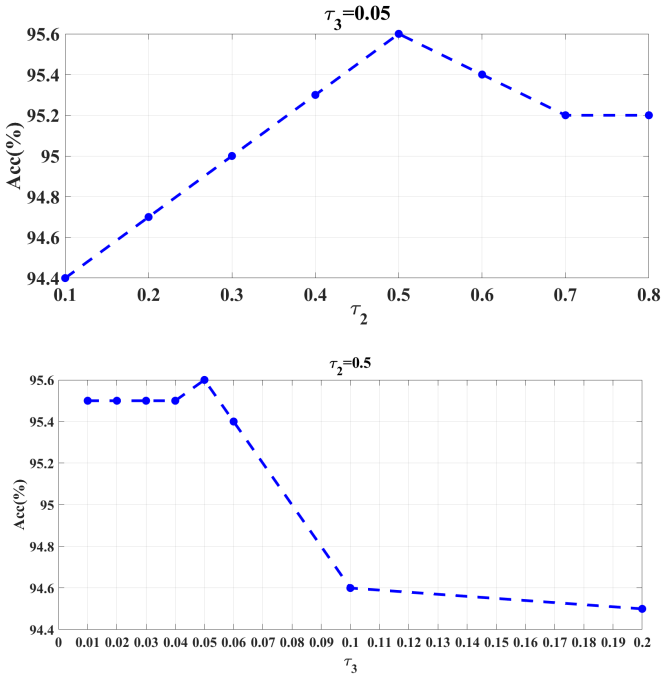

Fig. 8. The influence of the thresholds τ_2 and τ_3 on UCF101 split 1.

TABLE III
EVALUATION OF ASFS-AVFS ON UCF101 SPLIT 1 (ACC.(%))/ACTIVITYNET V1.2 (MAP(%)).

Methods	No AVFS-ASFS	AVFS	ASFS	AVFS-ASFS
Spa-Stream	85.5/82.6	86.3/83.4	86.0/83.1	**87.1/83.9**
Tem-Stream	87.7/73.2	88.3/74.1	88.1/73.8	**88.8/74.7**
Two-Stream	92.3/85.2	92.9/86.4	92.7/86.5	**93.9/88.4**

performance of our ASFS-AVFS in different streams. In either the individual stream or the combined two-stream, our ASFS-AVFS approach obtains the best results, which reveals that splitting actions into different *Action*S segments and adaptive sampling the segments to choose the discriminative deep feature maps can be widely used for different input modality-derived streams. For example, on the trimmed video dataset – UCF101 split 1, the performance gain between our AVFS-ASFS and *No AVFS-ASFS* reaches to 1.6% on the *Two-Stream* network. On the more complicated untrimmed video dataset – ActivityNet V1.2, compared to *No AVFS-ASFS*, the accuracy of our AVFS-ASFS is improved by 3.2%.

In Figure 9, we visualize and compare the learned results for four video actions of Figure 7. From top row to bottom row are the results of "DrawSword" (HMDB51), "Smile" (HMDB51), "Lunges" (UCF101), and "LongJump" (UCF101), respectively. Two problems are generated when without both adaptive deep feature sampling and video segmentation: 1) Some useful features would be eliminated while some useless features would be preserved; 2) The role of action stages, which contain different physical intentions and contribute much to the recognition of the entire human action, is ignored. Specially, for the first setting, in the action video "DrawSword", it is hard to discern the action as some indifferent scenery patterns are extracted, while the significant action features that played by

the man are not learned. In contrast, with our proposed ASFS-AVFS strategy, not only the discriminative features are learned, but also the long-term temporal structure with the concern of *Action*S is modeled. Consequently, it is easy to recognize the DrawSword action in terms of our feature representation. In the action video "Lunges", the result of the first setting is so dim that the action would be easily incorrectly identified. Significantly, in our class representation, the shape of the barbell and the human are obviously shown. In the TSN setting, the action pattern is better modeled than the first setting, but much more irrelevant deep action features are learned than ours. For the other two action classes, our method also performs best. The results demonstrate that adaptively selecting useful deep features and segmenting the action video into *Action*S-based segments are beneficial for improving action recognition.

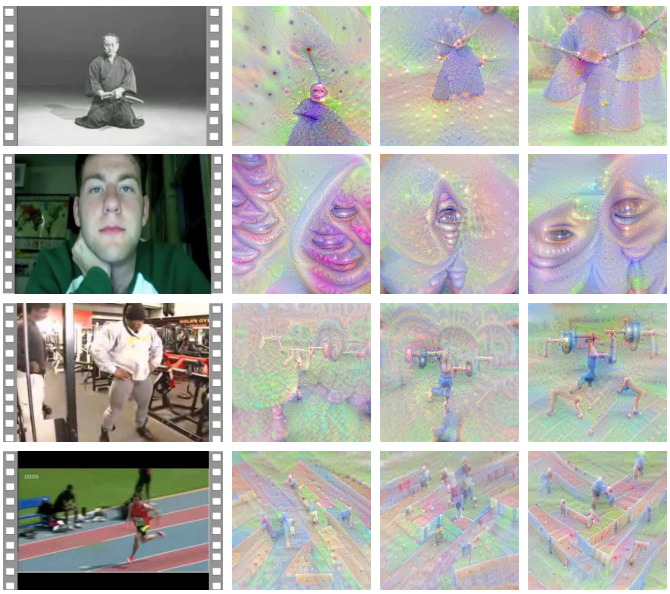

Fig. 9. Visualization of the action class information by using DeepDraw [62] on the spatial-stream. Three settings are compared: 1) Without both randomly sampling and segmentation (second column); 2) Randomly sampling with equal segmentation (TSN [7]) (third column); 3) Our adaptive video feature segmentation and adaptive segment feature sampling (AVFS-ASFS) strategy (fourth column).

C. Performance of the ActionS-ST-VLAD Encoding Method

Accuracy Comparison. In Table IV, we compare the proposed *Action*S-ST-VLAD aggregation method with both the classic frame-level and current video-level aggregation strategies. The traditional max and mean pooling approaches treat all features equally in both space and time, thus they are unable to model the complicated spatio-temporal structure of human actions, and the performances is poor. AdaScan [6] exploited an adaptive scan pooling network to encode informative frames, in which a majority of non-informative frames are discarded. Since the method to predict the discriminative importance of each frame is not good enough, many useful frames are removed which greatly affects its performance. Our *Action*S-ST-VLAD approach avoids this drawback and obtains an accuracy improvement by 1.4% on UCF101 and 2.0% on ActivityNet V1.2 respectively in the TS-Net. The ActionVLAD [12] encoding technique concerns the sub-actions in local clusters spatially. However, it disregards the multiple action-stages temporally, and it also cannot sample discriminative local descriptors in each *Action*S. Our *Action*S-ST-VLAD addresses these drawbacks of ActionVLAD, and a significant gain is achieved. Specially, in the two-stream architecture, in contrast to ActionVLAD, our result is 1.0% more accurate on UCF101 and 1.4% more accurate on ActivityNet V1.2.

TABLE IV
ACCURACY COMPARISON BETWEEN OUR *Action*S-ST-VLAD METHOD AND OTHER POOLING STRATEGIES ON UCF101 SPLIT 1 (ACC.(%))/ACTIVITYNET V1.2 (MAP(%)).

Methods	Spa-Stream	Tem-Stream	Two-Stream
Max	84.3/82.6	87.1/69.6	91.9/85.5
Mean	85.8/85.0	88.1/72.3	93.5/88.0
AdaScan	86.4/85.4	88.7/72.4	94.0/88.3
ActionVLAD	87.2/86.4	89.7/73.0	94.4/88.9
*Action*S-ST-VLAD	**88.2/87.9**	**90.5/74.4**	**95.4/90.3**

Efficiency Comparison. In Figure 10, we compare the efficiency by reporting the average number of frames per second (Fr/Sec) on HMDB51 Split 1 on a laptop with an Intel Core (TM) CPU i7-4510U 2.60GHz and 8GB memory. For our AVFS, the most expensive step is using the cosine similarity (CS) measure to obtain the *Action*S, since the CS is very fast, the AVFS does not cost much time. The ASFS contains two main steps: flow guided feature warping and frame discriminative importance evaluation. Since we use the multi-stream network, the optical flow has already been pre-computed, and feature warping can be efficiently conducted by bilinear interpolation. Besides, the similarity measure Eq. (5), which is used to predict the discriminative importance of each frame, is also efficient. Consequently, the proposed AVFS-ASFS approach is not computational expensive. Overall, the encoding speed of our *Action*S-ST-VLAD is about 85% of ActionVLAD. The AdaScan conducts the pooling by recursively calculating two operations, i.e., discriminative importance prediction and weighted mean pooling, with respect to each frame across the whole video, thus its encoding time is high. Compared to our *Action*S-ST-VLAD, the encoding computational cost of AdaScan is about 12% more expensive.

D. Evaluation of the RGBF Modality

Results of different modalities are reported in Table V. With the application of our RGBF modality, the accuracy of action recognition is boosted by 0.2% compared to the baseline two-steam architecture. At least two benefits can be obtained from the proposed RGBF modality: first, the motion salient regions corresponding to the acting parts of human are enhanced. These motion discriminative regions are crucially for recognizing human actions [21]. Second, noises in the background are significantly compressed, reducing the probability to wrongly classify an action.

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TIP.2018.2890749, IEEE Transactions on Image Processing

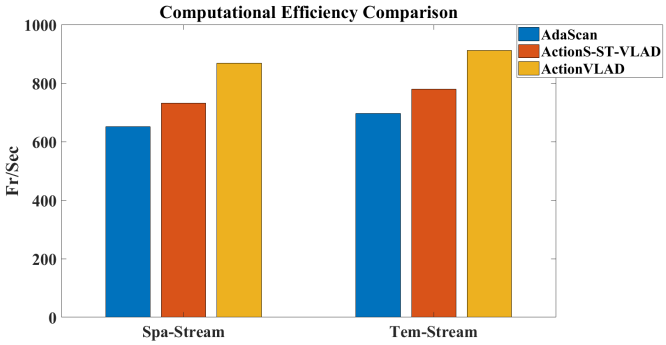

Fig. 10. Computational efficiency comparison (Fr/Sec) between our *Action*S-ST-VLAD pooling method and other pooling strategies on HMDB51 Split 1. The processing speeds – number of frames per second (Fr/Sec) are reported.

TABLE V
EXPLORATION OF OUR RGBF MODALITY ON UCF101 SPLIT 1.

Modalities	Acc.(%)
RGB	87.8
Flow	90.3
RGBF	87.6
RGB+Flow (Two-Stream)	95.4
RGB+Flow+RGBF (Three-Stream)	**95.6**

E. Comparison with State-of-the-Art

Firstly, we compare our method (without pre-trained on the recently Kinetics dataset [50]) to the state-of-the-art approaches over all the three splits on UCF101 and HMDB51 in Table VI. The results of these methods are quoted from the original papers. Our method outperforms all of them. In addition, the performance gain is even more significant when integrating with the complementary video features – iDT [27]. We classify the comparison into 4 groups from top to bottom: 1) The results of three classical hand-crafted methods [27], [57], [58] are presented. Notably, the iDT is one of the most successful hand-crafted features for video-based action recognition at present. Comparing to the most accurate results of [58], the improvements of our method reach to 6.5% and 6.3% on UCF101 and HMDB51 respectively. 2) We select the most recent CNN methods that adopt the two-stream architectures for comparison. These methods do not consider long-term temporal structures of human actions. Our method performs best among them. Especially in HMDB51, the gain is greater than or equal to 2.5% (71.4% vs 68.9%). The results demonstrate that pooling the deep features across the entire video to conduct a video-level classification is very helpful for action recognition. 3) We compare with the deep learning methods that attempt to model the long-range temporal structure. Since the C3D approach is hard to be updated by using new CNN architecture and utilizing the large-size ImageNet [31] for pre-training, its performance is the worst on UCF101. For the most related work – ActionVLAD [12], since it does not concern the temporal coherence of *Action*S, as well as it is unable to select informative feature maps during sampling, our results outperform it by 2.9% on UCF101 and 4.5% on HMDB51. For another related work – TSN, the performance of us is also much better, where the accuracy is improved by 1.4% (95.6% vs 94.2%) on UCF101 and 2.0% (71.4% vs 69.4%) on HMDB51 respectively. One reason is that our proposed *Action*S-ST-VLAD considers both the different action phases and sub-actions spatio-temporally. Secondly, we aggregate the preserved useful features to a single video representation for classification, while the TSN pools randomly-sampled features in three video-snippets and average the classification scores of these snippets. Finally, our framework is end-to-end trainable but the TSN is not as it needs to combine the scores of different streams for final prediction. 4) The CNN methods that combined with iDT are chosen for analysis. The accuracy of our method is further boosted by 0.7% on UCF101 and 1.9% on HMDB51 respectively when integrating with the iDT. In particular, it outperforms any of the algorithm in Table VI by at least 1.7% on UCF101 and 3.0% on HMDB51.

TABLE VI
COMPARISON (ACC.(%)) WITH STATE-OF-THE-ART METHODS ON THE UCF101 AND HMDB51 DATASETS (AVERAGE OVER 3 SPLITS).

Methods	UCF101	HMDB51
iDT+FV [27]	85.9	57.2
iDT+HSV [57]	87.9	61.1
iDT+MIFS [58]	89.1	65.1
TS-Net [2] (VGG_M)	88.0	59.4
TS-Net [60] (VGG16)	91.4	58.5
Two-Stream SR-CNNs [28]	92.6	-
Two-Stream Conv. [4]	92.5	65.4
Multi-Stream Fusion [61]	91.6	61.8
ST-ResNet [56]	93.4	66.4
STCB (BN-Inception) [5]	94.6	68.9
C3D [34]	85.2	-
AdaScan [6]	89.4	54.9
ActionVLAD [12]	92.7	66.9
ST-VLMPF [11]	93.6	69.5
TSN (3 modalities) [7]	94.2	69.4
Ours	**95.6**	**71.4**
TDD+FV [59]	90.3	63.2
C3D+iDT [34]	90.4	-
Two-Stream Conv.+iDT [4]	93.5	69.2
AdaScan+iDT [6]	91.3	61.0
ActionVLAD+iDT [12]	93.6	69.8
ST-ResNet+iDT [56]	94.6	70.3
Ours+iDT	**96.3**	**73.3**

Moreover, we evaluate the performance of our method on the more realistic untrimmed video dataset – ActivityNet V1.2. Similar to [8], we apply the M-TWI strategy and the top-K pooling to extend the action models that learned by our *Action*S-ST-VLAD method in trimmed action videos to action recognition in untrimmed videos. In contrast to [8] which extracts a snippet with the duration of 1 second, e.g., it samples M snippets from a video if it in length of M seconds, we split a video to M' action clips, where for each of the first M'-1 action clips, we equally get it with the duration of 10 seconds. To be fairly compared with ActionVLAD, we replace the encoding method of our AVFS-ASFS based *Action*S-ST-VLAD with ActionVLAD, and subsequently fine-tuning it. As shown in Table VII, the three methods, i.e., TSN [8], ActionVLAD [12] and **Ours**, which are able to model long-

range temporal structures, perform much better than the other four approaches. For these video-level encoding methods, our method outperforms TSN (7 seg) [8] by 1.1%. Noticeably, it significantly boosts the performance of ActionVLAD (90.7% vs 86.2%). The best result of our method illustrates that it is necessary to consider the action-stages both spatially and temporally for complex realistic videos. Particularly for the untrimmed videos, action instance usually spans a small portion of the entire video while the dominating portions are irrelevant background content. Determining the discriminative importance of each frame efficiently is a crucial measure to reduce the interfere of the background for action prediction.

TABLE VII
COMPARISONS WITH STATE-OF-THE-ARTS ON ACTIVITYNET V1.2 DATASET (RESULTS ARE REPORTED AS MAP(%)).

Methods	ActivityNet
iDT+FV [27]	66.5
Depth2Action [55]	78.1
TS-Net [2]	71.9
C3D [34]	74.1
TSN (3 seg) [8]	89.0
TSN (7 seg) [8]	89.6
ActionVLAD [12]	86.2
Ours	**90.7**

Secondly, we test the performance of the proposed method where the ConvNets are pre-trained on the Kinetics dataset. The ConvNet models of[1], which are pre-trained on both ImageNet and Kinetics, are selected as the baseline models of our method, and then we finetune these models with our AVFS-ASFS based *Action*S-ST-VLAD for action recognition in trimmed video on UCF101 and HMDB51, and in untrimmed video on ActivityNet. As shown in Table VIII, our method performs almost equally to I3D-Two-Stream [50], which uses two-stream 3D architectures pre-trained on Kinetics. Because even simple 3D architectures pre-trained on Kinetics performs better than complex 2D architectures to recognize human actions in videos [54]. Consequently, one of our primary work in the near future is to use 3D ConvNets to replace the 2D ConvNets to enhance the performance. In contrast to TSN [8](Inception V3), our method performs 0.6% better on the UCF101 dataset, while performs 0.2% worse on the complicated ActivityNet V1.3 dataset. The results demonstrate that the models of TSN [8](Inception V3), which are pre-trained on Kinetics and finetuned on UCF101 and ActivityNet V1.3, are quite good. After fine-tuning the models of [8] by our encoding method: AVFS-ASFS based *Action*S-ST-VLAD, these ConvNets are able to learn better video-level representations on the simple trimmed videos. For the untrimmed videos in ActivityNet, the action instance may only occupy a small portion of one complex video, learning a whole video-level representation may not effective to recognize the action. In this case, there are generally three important steps: 1) automatically splitting the long-term videos into content-coherent clips, 2) learning a video clip-level representation for each video clip, and 3) adaptively aggregating these video clip-level representations. The 1) and 3) steps are crucial, our method performs slightly worse than TSN [8] on ActivityNet V1.3 due to it focuses on the 2) step. In contrast to the baseline algorithm ActionVLAD [12], our method outperforms it in all the three datasets. The results again show that our AVFS-ASFS based *Action*S-ST-VLAD is better than ActionVLAD to encode deep features, no matter for simple or complicated realistic action videos.

[1]http://yjxiong.me/others/kinetics_action/

TABLE VIII
COMPARISONS WITH STATE-OF-THE-ARTS ON UCF101, HMDB51, AND ACTIVITYNET V1.3 DATASETS (PRE-TRAINED ON KINETICS).

Methods	UCF101	HMDB51	ActivityNet
I3D-Two-Stream [50]	**98.0**	80.7	-
T-C3D [53]	92.5	62.4	-
R(2+1)D-TwoStream [52]	97.3	78.7	-
TSN Inception V3 [8]	97.3	-	**90.2**
ActionVLAD [12]	94.8	77.1	86.5
Ours	97.9	**80.9**	90.0

The superior performance of our method reveals that splitting the video into effective *Action*S-based segments, choosing informative features with sampling, exploiting useful modalities, and encoding features to a video (or video clip) level representation are able to greatly enhance the performance of the stream-networks on recognizing human actions in videos.

V. CONCLUSION

We presented a novel *Action*S-ST-VLAD approach to aggregate video features spatio-temporally for action recognition with the consideration of encoding deep features both in sub-actions spatially and in action-stages temporally. An AVFS-ASFS strategy was proposed to split the local deep features into different *Action*S-based segments, and to select the informative features in each segment. This strategy is not only effective in discarding redundant/insignificant/noisy frames that are less helpful or even harmful for the target action, but also more efficient than the state-of-the-art [6] that predicts the discriminative importance of each frame. A video representation was formulated by aggregating the multiple segment-level representations via ST-VLAD for final video-level action classification. An RGBF modality was designed to construct a third stream that attempts to extract motion-salient information. The three video representations were fused at the last convolutional layer. Finally a spatio-temporal loss function was used to optimize our framework end-to-end. The proposed encoding method can be easily applied to different CNN architectures and other video tasks. In future work, we will find a way to learn appropriate hyperparameters, i.e., τ_1, τ_2, and τ_3.

ACKNOWLEDGMENT

The work is supported by the funding CXFW-18-413100063 of Wuhan University. It is also supported by the National Key Research and Development Program of China (No. 2016YFF0103501), the Natural Science Foundation of China (NSFC) (No. 61572012), and the Natural Science Fund of Hubei Province (No. 2017CFB598, No. 2017CFB677).

REFERENCES

[1] S. Herath, M. Harandi, and F. Porikli, "Going deeper into action recognition: A survey", *Image Vis. Comput.*, vol.60, pp.4–21, 2017.

[2] K. Simonyan and A. Zisserman, "Two-stream convolutional networks for action recognition in videos", in *Neural Information Processing Systems*, 2014, pp.568–576.

[3] Z. Tu, W. Xie, Q. Qin, R. Poppe, R. C Veltkamp, B. Li, and J. Yuan, "Multi-Stream CNN: Learning Representations Based on Human-Related Regions for Action Recognition", *Pattern Recognit.*, vol.79, pp.32–43, 2018.

[4] C. Feichtenhofer, A. Pinz, and A. Zisserman, "Convolutional Two-Stream Network Fusion for Video Action Recognition", in *Comput. Vis. Pattern Recognit.*, 2016, pp.1933–1941.

[5] Y. Wang, M. Long, J. Wang, and P. S. Yu, "Spatiotemporal Pyramid Network for Video Action Recognition", in *Comput. Vis. Pattern Recognit.*, 2017, pp.2097–2106.

[6] A. Kar, N. Rai, K. Sikka, and G, Sharma, "AdaScan: Adaptive Scan Pooling in Deep Convolutional Neural Networks for Human Action Recognition in Videos", in *Comput. Vis. Pattern Recognit.*, 2017, pp.5699–5708.

[7] L. Wang, Y. Xiong, Z. Wang, Y. Qiao, D. Lin, X. Tang, and L. Gool, "Temporal Segment Networks: Towards Good Practices for Deep Action Recognition", in *Euro. Conf. Comput. Vis.*, 2016, pp.20–36.

[8] L. Wang, Y. Xiong, Z. Wang, Y. Qiao, D. Lin, X. Tang, and L. Gool, "Temporal Segment Networks for Action Recognition in Videos", *CoRR abs/1705.02953*, 2017.

[9] A. Kar, N. Rai, K. Sikka, and G. Sharma, "Optical Flow Guided Feature: A Fast and Robust Motion Representation for Video Action Recognition", in *Comput. Vis. Pattern Recognit.*, 2018, pp.1390–1399.

[10] T. Yu, Z. Wang, and J. Yuan, "Compressive Quantization for Fast Object Instance Search in Videos", in *Int. Conf. Comput. Vis.*, pp.726–735, 2017.

[11] I. C. Duta, B. Ionescu, K. Aizawa, and N. Sebe, "Spatio-Temporal Vector of Locally Max Pooled Features for Action Recognition in Videos", in *Comput. Vis. Pattern Recognit.*, 2017, pp.3205–3214.

[12] R. Girdhar, D. Ramanan, A. Gupta, J. Sivic, and B. Russell, "ActionVLAD: Learning spatio-temporal aggregation for action classification", in *Comput. Vis. Pattern Recognit.*, 2017, pp.3165–3174.

[13] I. C. Duta, B. Ionescu, K. Aizawa, and N. Sebe, "Spatio-Temporal VLAD encoding for human action recognition in videos", in *Int. Conf. Multi. Model.*, 2017.

[14] Z. Xu, Y. Yang, and A. G. Hauptmann, "A discriminative CNN video representation for event detection", in *Comput. Vis. Pattern Recognit.*, 2015, pp.1798–1807.

[15] S. Hong. J. Ryu, and H. Yang, "Not all frames are equal: aggregating salient features for dynamic texture classification", *Multidimensional Systems and Signal Processing*, vol.29, no.1, pp.279–298, 2018.

[16] S. Hong, J. Ryu, W. Im, and H. Yang, "D3: Recognizing dynamic scenes with deep dual descriptor based on key frames and key segments", *Neurocomputing*, vol.273, pp.611–621, 2018.

[17] J. Li, B. Yang, C. Chen, R. Huang, Z. Dong, and W. Xiao, "Automatic registration of panoramic image sequence and mobile laser scanning data using semantic features", *ISPRS Journal of Photogrammetry and Remote Sensing*, 2018, vol.136, pp.41–57.

[18] Y. Tang, Y. Tian, J. Lu, P. Li, and J. Zhou, "Deep Progressive Reinforcement Learning for Skeleton-Based Action Recognition", in *Comput. Vis. Pattern Recognit.*, 2018, pp.5323–5332.

[19] L. Chen, J. Lu, Z. Song, and J. Zhou, "Part-Activated Deep Reinforcement Learning for Action Prediction", in *Euro. Conf. Comput. Vis.*, 2018, pp.421–436.

[20] V. Choutas, P. Weinzaepfel, J. Revaud, and C. Schmid, "PoTion: Pose MoTion Representation for Action Recognition", in *Comput. Vis. Pattern Recognit.*, 2018, pp.7024–7033.

[21] Z. Tu, W. Xie, J. Dauwels, B. Li, and J. Yuan, "Semantic Cues Enhanced Multi-modality Multi-Stream CNN for Action Recognition", *IEEE Trans. Cir. and Systems for Video Tech.*, doi:10.1109/TCSVT.2018.2830102, 2018.

[22] Y. Wang and M. Hoai, "Improving Human Action Recognition by Non-action Classification", in *Comput. Vis. Pattern Recognit.*, 2016, pp.2698–2707.

[23] H. Bay, T. Tuytelaars, and L. V. Gool, "SURF: Speeded Up Robust Features", in *Euro. Conf. Comput. Vis.*, 2006, pp.404–417.

[24] B. Li, R. Chellappa, Q. Zheng, and S. Der, "Model-based temporal object verification using video", *IEEE Trans. Image Processing*, vol.10, no.6, pp.897-908, 2001.

[25] N. Dalal and B. Triggs, "Histograms of oriented gradients for human detection", in *Comput. Vis. Pattern Recognit.*, 2005, pp.886–893.

[26] H. Wang, A. Klaser, C. Schmid, and C. Liu, "Dense Trajectories and Motion Boundary Descriptors for Action Recognition", *Int. J. Comput. Vis.*, vol.103, pp.60–79, 2013.

[27] H. Wang and C. Schmid, "Action recognition with improved trajectories", in *Int. Conf. Comput. Vis.*, pp.3551–3558, 2013.

[28] Y. Wang, J. Song, L. Wang, O. Hilliges, and L. Van Gool, "Two-Stream SR-CNNs for Action Recognition in Videos", in *BMVC*, 2016.

[29] C. Feichtenhofer, A. Pinz, and R. P. Wildes, "Spatiotemporal Multiplier Networks for Video Action Recognition", in *Comput. Vis. Pattern Recognit.*, 2017, pp.7445–7454.

[30] L. Wang, Y. Xiong, D. Lin, and L. Gool, "UntrimmedNets for Weakly Supervised Action Recognition and Detection", in *Comput. Vis. Pattern Recognit.*, 2017, pp.6402–6411.

[31] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "ImageNet classification with deep convolutional neural networks", in *Neural Information Processing Systems*, 2012, pp.1106–1114.

[32] K. Simonyan and A. Zisserman, "Very deep convolutional networks for large-scale image recognition", in *Int. Conf. Learning Representations*, 2015, pp.1–14.

[33] J. Lu, J. Hu, and J. Zhou, "Deep Metric Learning for Visual Understanding: An Overview of Recent Advances", *IEEE Signal Proc. Magazine*, vol.34, no.6, pp.76–84, 2017.

[34] D. Tran, L. Bourdev, R. Fergus, L. Torresani, and M. Paluri, "Learning spatiotemporal features with 3D convolutional networks", in *Int. Conf. Comput. Vis.*, pp.4489–4497, 2015.

[35] Z. Tu, N. Aa, C. V. Gemeren, and R. C. Veltkamp, "A combined post-filtering method to improve accuracy of variational optical flow estimation", *Pattern Recognit.*, vol.47, no.5, pp.1926–1940, 2014.

[36] S. Ioffe and C.Szegedy, "Batch normalization: Accelerating deep network training by reducing internal covariate shift", in *Int. Conf. Machine Learning*, pp.448–456, 2015.

[37] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition", in *Comput. Vis. Pattern Recognit.*, 2016, pp.770–778.

[38] X. Zhu, Y. Xiong, J. Dai, L. Yuan, and Y. Wei, "Deep Feature Flow for Video Recognition", in *Comput. Vis. Pattern Recognit.*, 2017, pp.2349–2358.

[39] J. Donahue, L. Hendricks, M. Rohrbach, S. Venugopalan, S. Guadarrama, K. Saenko, and T. Darrell, "Long-Term Recurrent Convolutional Networks for Visual Recognition and Description", *IEEE Trans. Pattern Anal. Mach. Intell.*, vol.39, no.4, pp.677–691, 2017.

[40] G. Cheron, I. Laptev, and C. Schmid, "P-CNN: Pose-based CNN features for action recognition", in *Int. Conf. Comput. Vis.*, pp.3218–3226, 2015.

[41] S. Ren, K. He, R. Girshick, and J. Sun, "Faster R-CNN: Towards real-time object detection with region proposal networks", in *Neural Information Processing Systems*, 2015, pp.91–99.

[42] B. Singh, T. Marks, M. Jones, O. Tuzel, and M. Shao, "A Multi-Stream Bi-Directional Recurrent Neural Network for Fine-Grained Action Detection", in *Comput. Vis. Pattern Recognit.*, 2016, pp.1961–1970.

[43] R. Arandjelovic, P. Gronat, A. Torii, T. Pajdla, and J. Sivic, "NetVLAD: CNN architecture for weakly supervised place recognition", in *Comput. Vis. Pattern Recognit.*, 2016, pp.5297–5307.

[44] H. Jegou, M. Douze, C. Schmid, and P. Perez, "Aggregating local descriptors into a compact image representation", in *Comput. Vis. Pattern Recognit.*, 2010, pp.3304–3311.

[45] X. Zhu, Y. Wang, J. Dai, L. Yuan, and Y. Wei, "Flow-Guided Feature Aggregation for Video Object Detection", in *Int. Conf. Comput. Vis.*, pp.408–417, 2017.

[46] Z. Tu, R. Poppe, R. C. Veltkamp, "Weighted local intensity fusion method for variational optical flow estimation", *Pattern Recognit.*, vol.50, pp.223–232, 2016.

[47] E. Ilg, N. Mayer, T. Saikia, M. Keuper, A. Dosovitskiy, and T. Brox, "FlowNet 2.0: Evolution of Optical Flow Estimation with Deep Networks", in *Comput. Vis. Pattern Recognit.*, 2017, pp.1647–1655.

[48] K. Soomro, A. R. Zamir, and M. Shah, "UCF101: A dataset of 101 human action classes from videos in the wild", *CoRR, abs/1212.0402*, 2012.

[49] H. Kuehne, H. Jhuang, E. Garrote, T. Poggio, and T. Serre, "HMDB: a large video database for human motion recognition", in *Int. Conf. Comput. Vis.*, pp.2556–2563, 2011.

[50] J. Carreira and A. Zisserman, "Quo vadis, action recognition? a new model and the kinetics dataset", in *Comput. Vis. Pattern Recognit.*, pp.4724–4733, 2017.

[51] F. C. Heilbron, V. Escorcia, B. Ghanem, and J. C. Niebles, "Activitynet: A large-scale video benchmark for human activity understanding", in *Comput. Vis. Pattern Recognit.*, pp.961–970, 2015.

[52] D. Tran, H. Wang, L. Torresani, J. Ray, Y. LeCun, and M. Paluri, "A Closer Look at Spatiotemporal Convolutions for Action Recognition", in *Comput. Vis. Pattern Recognit.*, 2018.
[53] K. Liu, W. Liu, C. Gan, M. Tan, and H. Ma, "T-C3D: Temporal Convolutional 3D Network for Real-Time Action Recognition", in *AAAI Conf. Artificial Intell.*, 2018.
[54] K. Hara, H. Kataoka, and Y. Satoh, "Can Spatiotemporal 3D CNNs Retrace the History of 2D CNNs and ImageNet?", in *Comput. Vis. Pattern Recognit.*, 2018.
[55] Y. Zhu and S. Newsam, "Depth2action: Exploring embedded depth for large-scale action recognition", in *Euro. Conf. Comput. Vis.*, 2016, pp.668–684.
[56] C. Feichtenhofer, A. Pinz, and R. P. Wildes, "Spatiotemporal Residual Networks for Video Action Recognition", in *Neural Information Processing Systems*, 2016, pp.3468–3476.
[57] X. Peng, L. Wang, X. Wang, and Y. Qiao, "Bag of visual words and fusion methods for action recognition: Comprehensive study and good practice", *Comput. Vis. Image Understand.*, vol.150, pp.109–125, 2016.
[58] Z. Lan, M. Lin, X. Li, A. Hauptmann, and B. Raj, "Beyond gaussian pyramid: Multi-skip feature stacking for action recognition", in *Comput. Vis. Pattern Recognit.*, 2015, pp.204–212.
[59] L. Wang, Y. Qiao, and X. Tang, "Action Recognition with Trajectory-Pooled Deep-Convolutional Descriptors", in *Comput. Vis. Pattern Recognit.*, 2015, pp.4305–4314.
[60] X. Wang, A. Farhadi, and A. Gupta, "Actions transformations", in *Comput. Vis. Pattern Recognit.*, 2016, pp.2658–2667.
[61] X. Yang, P. Molchanov, and J. Kautz, "Multilayer and Multimodal Fusion of Deep Neural Networks for Video Classification", in *ACM Multimedia*, 2016, pp.978–987.
[62] "Deep draw", https://github.com/auduno/deepdraw.

Zhigang Tu started his Master Degree in image processing at the School of Electronic Information, Wuhan University, China, 2008. In 2015, he received the Ph.D. degree in Computer Science from Utrecht University, Netherlands. From 2015 to 2016, he was a postdoctoral researcher at Arizona State University, US. Then from 2016 to 2018, he was a research fellow at the School of Electrical and Electronic Engineering, Nanyang Technological University, Singapore. He is currently a professor at the State Key Laboratory of Information Engineering in Surveying, Mapping and Remote sensing, Wuhan University. His research interests include computer vision, image processing, video analytics, and machine learning. Specially for motion estimation, object segmentation, object tracking, action recognition and localization, and anomaly detection.

Hongyan Li started her Master Degree in computer science at the Central China Normal University, China, 2005. In 2016, she received the Ph.D. degree in Computer Architecture from Huazhong University of Science and Technology, China. She is currently an associate professor at Hubei University of Economics. From 2017 until now, she is also a postdoctoral researcher at the State Key Laboratory of Information Engineering in Surveying, Mapping and Remote Sensing, Wuhan University. Her research interests include computer vision, big data, and machine learning.

Dejun Zhang received the bachelor in communication engineering at the School of Information Science and Technology, Southwest Jiaotong University, China, 2006. In 2011, he received the Master degree in electronic engineering at the School of Manufacturing Science and Engineering, Southwest University of Science and Technology, China. In 2015, he received the Ph.D. degree in Computer Science from Wuhan Univeristy, China. He is currently a lecturer with the faculty of School of Information Engineering, China University of Geosciences, Wuhan, China. His research areas include machine learning, bioinformatics and computer graphics. His research interests include digital geometric processing, computer graphic, action recognition and localization.

Justin Dauwels is an Associate Professor of the School of Electrical and Electronic Engineering at the Nanyang Technological University (NTU) in Singapore. He also serves as Deputy Director of the ST Engineering NTU corporate lab, which comprises 100+ PhD students, research staff and engineers, developing novel autonomous systems for airport operations and transportation. His research interests are in data analytics with applications to intelligent transportation systems, autonomous systems, and analysis of human behaviour and physiology. He obtained his PhD degree in electrical engineering at the Swiss Polytechnical Institute of Technology (ETH) in Zurich in December 2005. Moreover, he was a postdoctoral fellow at the RIKEN Brain Science Institute (2006-2007) and a research scientist at the Massachusetts Institute of Technology (2008-2010). His research on intelligent transportation systems has been featured by the BBC, Straits Times, Lianhe Zaobao, Channel 5, and numerous technology websites. Besides his academic efforts, the team of Dr. Justin Dauwels also collaborates intensely with local start-ups, SMEs, and agencies, in addition to MNCs, in the field of data-driven transportation, logistics, and medical data analytics.

Baoxin Li received the PhD degree in electrical engineering from the University of Maryland, College Park, in 2000. He is currently a full professor and the Chair of computer science and engineering with Arizona State University, Phoenix, US. From 2000 to 2004, he was a senior researcher with SHARP Laboratories of America, Camas, WA, where he was the technical Lead in developing SHARP's HiIMPACT Sports technologies. From 2003 to 2004, he was also an adjunct professor with the Portland State University, Portland, OR. He holds nine issued US patents. His current research interests include computer vision and pattern recognition, image/video processing, multimedia, medical image processing, and statistical methods in visual computing. He won the SHARP Laboratories' President Award twice, in 2001 and 2004. He also received the SHARP Laboratories' Inventor of the Year Award in 2002. He received the National Science Foundation's CAREER Award from 2008 to 2009. He is a senior member of the IEEE.

Junsong Yuan (M'08–SM'14) received his Ph.D. from Northwestern University and M.Eng. from National University of Singapore. Before that, he graduated from the Special Class for the Gifted Young of Huazhong University of Science and Technology, Wuhan, China, in 2002. He is currently an associate professor at Computer Science and Engineering department of State University of New York at Buffalo. Before that, he was an associate professor at Nanyang Technological University (NTU), Singapore. His research interests include computer vision, video analytics, gesture and action analysis, large-scale visual search and mining. He received best paper award from Intl. Conf. on Advanced Robotics (ICAR'17), 2016 Best Paper Award from IEEE Trans. on Multimedia, Doctoral Spotlight Award from IEEE Conf. on Computer Vision and Pattern Recognition (CVPR'09), Nanyang Assistant Professorship from NTU, and Outstanding EECS Ph.D. Thesis award from Northwestern University.

He is currently Senior Area Editor of Journal of Visual Communications and Image Representations (JVCI), Associate Editor of IEEE Trans. on Image Processing (T-IP), IEEE Trans. on Circuits and Systems for Video Technology (T-CSVT), and served as Guest Editor of International Journal of Computer Vision (IJCV). He is Program Co-chair of ICME'18 and VCIP'15, and Area Chair of ACM MM'18, ICPR'18, CVPR'17, ICIP'18'17, ACCV'18'14 etc.