Human Action Recognition by Discriminative Feature Pooling and Video Segment Attention Model

Md Moniruzzaman, *Student Member, IEEE,* Zhaozheng Yin, *Member, IEEE,* Zhihai He, *Fellow, IEEE,* Ruwen Qin, *Member, IEEE,* Ming C Leu, *Member, IEEE*

***Abstract*—We introduce a simple yet effective network that embeds a novel Discriminative Feature Pooling (DFP) mechanism and a novel Video Segment Attention Model (VSAM), for video-based human action recognition from both trimmed and untrimmed videos. Our DFP module introduces an attentional pooling mechanism for 3D Convolutional Neural Networks that attentionally pools 3D convolutional feature maps to emphasize the most critical spatial, temporal, and channel-wise features related to the actions within a video segment, while our VSAM ensembles these most critical features from all video segments and learns (1) class-specific attention weights to classify the video segments into the corresponding action categories, and (2) class-agnostic attention weights to rank the video segments based on their relevance to the action class. Our action recognition network can be trained from both trimmed videos in a fully-supervised way and untrimmed videos in a weakly-supervised way. For untrimmed videos with weak labels, our network learns attention weights without the requirement of precise temporal annotations of action occurrences in videos. Evaluated on the untrimmed video datasets of THUMOS14 and ActivityNet1.2, and trimmed video datasets of HMDB51, UCF101, and HOLLYWOOD2, our network achieves promising performance, compared to the latest state-of-the-art methods. The implementation code is available at https://github.com/MoniruzzamanMd/DFP-VSAM-Networks.**

***Index Terms*—action recognition, attentional pooling, fully-supervised, weakly-supervised, discriminative features.**

I. Introduction

HUMAN action recognition is a challenging and fundamental problem in computer vision, owing to its applications in many areas such as surveillance systems and human computer interactions [1], [2]. Some human action recognition methods rely on human pose information [3], [4], [5], tracking multiple people as well as recognizing their activities [6], [7], [8], or dense trajectories [9], [10], [11] which extract rich low level descriptors for constructing effective video representations. Recently, human action recognition largely benefited from the advancements in Convolutional Neural Network (CNN) models [12], [13], [14]. For example, 2D CNN models were applied on different input modalities for human action recognition [15], [16], [17], [18], such as RGB images to extract the appearance information and optical flow to extract the motion information, which are two crucial and complementary clues for the action recognition task. Some recent action recognition methods [19], [20], [21] extended the 2D CNN into 3D to effectively learn spatio-temporal features from short video clips. Meanwhile, several works [22], [23], [24], [25] tried to employ recurrent neural networks with the extracted CNN features to capture long-term temporal dynamics for human action recognition.

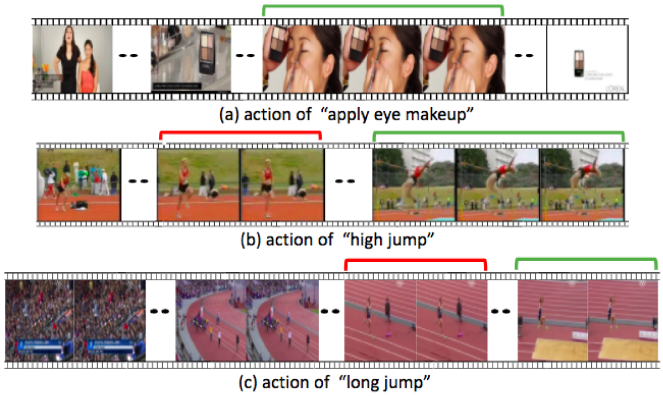

Fig. 1. **Importance of video segment attention.** Frames within the green intervals directly represent the action or differentiate different actions. Although the action of "high jump" and "long jump" are two different classes, frames within the red intervals share the similar motion information (running).

Challenges and Motivations: Despite the recent advance, human action recognition is still challenging from a few aspects:

(1) Usually, an action does not occupy the entire region of a single frame or the entire volume of a short video clip. Some of the pixels are not related to the action class, which may lead to misclassification. Most of the state-of-the-art methods employ deep CNNs (e.g., 2D CNN for a single frame and 3D CNN for a short video clip) over the entire input space to compute feature maps by convolution followed by average pooling or max pooling, without highlighting the most discriminative features. Although some algorithms [26], [27], [28] employed attentional pooling mechanism after the last convolutional layer of 2D CNN to highlight the most discriminative features, it is still unsolved to develop an attentional pooling mechanism

M. Moniruzzaman is with the Department of Computer Science, Stony Brook University (e-mail: mmoniruzzama@cs.stonybrook.edu)

Z. Yin is with the Department of Computer Science and Department of Biomedical Informatics, Stony Brook University (e-mail: zyin@cs.stonybrook.edu)

Z. He is with the Department of Electrical and Computer Engineering, University of Missouri, Columbia (e-mail: hezhi@missouri.edu)

R. Qin is with the Department of Civil Engineering, Stony Brook University (e-mail: ruwen.qin@stonybrook.edu)

M. C. Leu is with the Department of Mechanical and Aerospace Engineering, Missouri University of Science and Technology (e-mail: mleu@mst.edu)

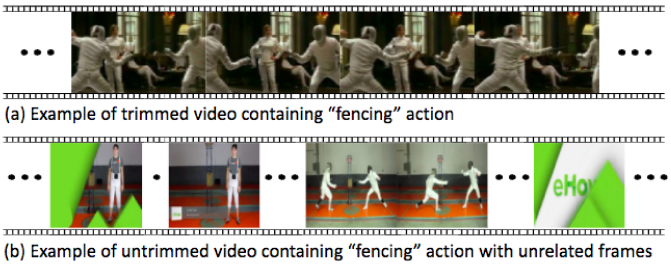

Fig. 2. **Example of trimmed and untrimmed videos containing the fencing action instances.** Trimmed videos do not contain unrelated frames, while untrimmed videos contain many unrelated frames.

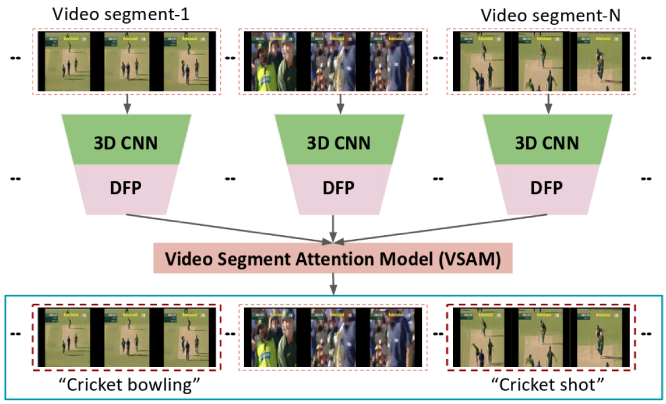

Fig. 3. We propose an end-to-end Discriminative Feature Pooling (DFP) and Video Segment Attention Model (VSAM) for human action recognition from weakly-labeled untrimmed videos (i.e., each video in the training set contains unrelated frames. The training video has action labels but which portion of the video contains the actions is unknown. The video may contain one or multiple action instances from one or multiple action classes). Our DFP attentionally pools the discriminative features of 3D convolutional feature maps, and our VSAM contains class-specific attention weights to classify the video segments into the corresponding action categories (e.g., "cricket bowling" and "cricket shot" in the video) and class-agnostic attention weights to highlight the most representative video segments.

for the 3D CNN models, whose convolutional feature maps contain spatio-temporal and channel-wise feature information. Therefore, the motivated research question is: *from the convolutional feature maps of a 3D CNN, which spatio-temporal and channel-wise features should get more attention to highlight the discriminative features related to the action class?*

(2) Given a sequence of frames in a long video, maybe only a small portion of the video is directly related to the action and the video may contain many unrelated or less-relevant frames, as shown in Fig. 1. Fig. 1(a) shows a video sequence of "apply eye makeup" action. Only a small segment of the video directly belongs to the action and the video contains many unrelated frames. Fig. 1(b) and Fig. 1(c) show the "high jump" and "long jump" actions, respectively. Some frames of the two different actions share similar motion information (frames within red intervals in Fig. 1). Therefore, the frames within the green interval in Fig. 1, which either directly represent the action (Fig. 1(a)) or differentiate different actions (Fig. 1(b) and Fig. 1(c)), deserve more attention for the accurate action recognition. This challenge motivates another research question: *given a video, which video segment should get more attention to highlight the most representative frames related to the action?*

(3) Most of the action recognition methods rely on **trimmed video** datasets (i.e., videos that do not contain unrelated frames, such as those from the datasets of HMDB51 [29] and UCF101 [30], as shown in Fig. 2(a)). But, in practice, it is more common to collect untrimmed videos with weak or noisy labels from the web (e.g., Youtube) than collect precisely annotated videos, thus developing an action recognition method capable of learning from untrimmed videos is in the need. Learning from **untrimmed** videos (i.e., videos that contain irrelevant or less-relevant frames, such as the THUMOS14 dataset [31] as shown in Fig. 2(b)) is a **weakly-supervised learning**. Therefore, the third research question arises: *given a weakly-labeled training dataset of long untrimmed videos (i.e., each video has a label for an action but which portion of the video contains the action is unknown and the video might contains multiple different actions), how can we effectively train an action recognition model without knowing the precise temporal annotations of action instances?*

Our Proposal and Contributions: Motivated by the above challenges, we propose a novel human action recognition network, including a new Discriminative Feature Pooling (DFP) mechanism and a Video Segment Attention Model (DFP-VSAM), to classify human actions from both untrimmed and trimmed videos, as shown in Fig. 3. To summarize, our major contributions are four-fold:

1) We propose a new Discriminative Feature Pooling (DFP) mechanism that integrates spatial, temporal, and channel-wise attentional pooling in a unified network on top of the convolutional feature maps of 3D CNN to highlight the most discriminative features. To the best of our knowledge, this is the first work that applies three different attentional pooling mechanisms on top of the convolutional feature maps of a 3D CNN.
2) We propose a Video Segment Attention Model (VSAM) that first ensembles the most discriminative spatial, temporal, and channel-wise features from all video segments, which are then fed into a *class-specific* attention module to classify the video segments into the corresponding action categories, a *class-agnostic* attention module to emphasize the video segments containing highly representative action-related frames, and a video-level action prediction module to obtain the classification scores on the entire video.
3) Our action recognition network can be directly trained from untrimmed videos in a weakly-supervised way without the requirement of temporal annotations of action occurrences in videos, where the videos may contain one or multiple occurrences of action instances from one or multiple action classes.
4) We conducted experiments on five benchmark datasets, namely THUMOS14, ActivityNet1.2, HMDB51, UCF101 and HOLLYWOOD2 to show the superior performance and generality of the proposed DFP-VSAM human action recognition network. On all the datasets, our network achieves superior or comparable performance compared to the state-of-the-art methods.

II. Related works

Deep learning for human action recognition: With the recent availability of powerful GPUs and after the breakthrough in image classification [32] with Convolutional Neural Networks (CNN) [12], [13], [14], video-based human action recognition recently has achieved significant progresses. Karpathy et al. [33] first designed a multi-resolution CNN architecture and trained the deep networks on a large-scale dataset (Sports-1M). CNN-based models for human action recognition broadly follow three main approaches. (1) ***Multi-stream networks*** [15], [16]: CNNs are trained on multiple input modalities, such as RGB, optical flow, warped flow etc. Given a test video, the predictions from all CNNs are fused to get the final video-level prediction. Simonyan et al. [15] designed two stream CNNs containing spatial and temporal networks by exploiting pre-trained models and optical flow calculation. Wang et al. [16] extended the standard two-stream [15] by using a much deeper base architecture [34]. (2) ***3D CNN*** [19], [20], [35], [36]: the pipelines of 3D CNNs are like those of 2D convolutional networks, but with spatio-temporal filters. Usually, 3D CNNs take short video clips as inputs, perform 3D convolution and 3D pooling to extract spatio-temporal feature maps (e.g., [20], [36]). Tran et al. [19] investigated 3D CNNs on the realistic and large-scale video datasets, which are capable of learning the spatio-temporal information from short video clips. (3) ***CNN + LSTM (Long Short Term Memory)*** [22], [23], [24], [25]: recurrent neural networks are built on top of CNN features to capture the long term dynamics for action recognition. Within the three directions, many algorithms develop techniques to recognize actions based on existing representation methods [37], [38], [39], [40], [41], [42], [43]. These algorithms employ neural networks without using attention mechanisms.

Weakly-supervised learning: Initially, weakly-supervised learning was effectively used in object detection and recognition [44], [45], [46]. Recently, several works [47], [48], [49], [50], [51] tried to adapt weakly-supervised learning methods into the human action recognition task from videos. Laptev et al. [48] tried to learn action models, and Duchennel et al. [47] tried to localize action instances in movies, by leveraging weak labels such as the movie scripts. But, the movie scripts are usually aligned with frames so they can provide approximated temporal annotations of action occurrences, which is not applicable to the general video-based human action recognition task. More recently, several works [52], [53] introduced weakly-supervised action detection and recognition technique called UntrimmedNet and WTALC, respectively, which did not use the temporal annotations during training. However, UntrimmedNet [52] and WTALC [53] are lack of spatial and channel-wise attention modeling.

Attentional pooling: Attention-based models [17], [26], [28], [54], [55], [56], [57], [58] employed attentional pooling operation at the last convolutional layer to dynamically pool convolutional features instead of the conventional average or max pooling operation. Sharma et al. [26] proposed a soft spatial attention-based action recognition model, which learns to focus selectively on parts of the video frames and classifies videos after taking a few glimpses. Several spatio-temporal attention models [54], [55], [56] were proposed for video captioning and human action recognition. Recently, a pose regularized attentional pooling method [28] was proposed, as a plug-in after the last convolutional layer of 2D CNN, for action recognition from still images and videos. Hu et al. [27] introduced squeeze and excitation networks, which put attentions on different feature channels. Most of the previous attention-based works are either trained from trimmed videos in a fully-supervised way or from untrimmed videos in a weakly-supervised way, while our DFP-VSAM can be trained from both trimmed videos in a fully-supervised way and untrimmed videos in a weakly-supervised way. Our DFP-VSAM is different from the previous attention-based works in a few aspects:

- **Applicable for both trimmed and untrimmed videos:** The previous fully-supervised attention-based methods [17], [26], [28], [58] on the trimmed videos applied the attention mechanism to look at the relevant parts in the spatial dimension, which are not applicable for the untrimmed videos since they cannot suppress the unrelated video segments. On the other hand, the previous weakly-supervised attention-based methods [52], [53] applied only the temporal attention to suppress the unrelated video segments, but they lack the spatial and channel-wise attention modeling inside each video segment. Differently, our DFP-VSAM can be trained from both trimmed videos in a fully-supervised way and untrimmed videos in a weakly-supervised way, where the DFP can pool the most discriminative spatial, temporal, and channel-wise features of each video segment, while the VSAM can suppress the unrelated video segments.
- **SOTA fully-supervised attention-based methods vs our method:** Most of the previous SOTA fully-supervised attention-based works [17], [26], [28], [58] only applied the spatial attention, and did not utilize the temporal and channel-wise attentions. On the other hand, we utilize all the three (spatial, temporal, and channel-wise) attentions to design our DFP to emphasize the critical spatial, temporal, and channel-wise features of each video segment. Furthermore, the SOTA on trimmed video datasets used well-tweaked backbone networks for the action recognition, but our method only uses a pre-trained backbone network for feature extraction.
- **SOTA weakly-supervised attention-based methods vs our method:** After our DFP attentionally pools the most discriminative spatial, temporal, and channel-wise features from the feature maps of each video segment, we further utilize a Video Segment Attention Model (VSAM), which is more important for action recognition from untrimmed videos in a weakly-supervised way. The SOTA weakly-supervised attention-based works [52], [53] on untrimmed videos applied the temporal attention on top of the fully-connected layers, while our VSAM first ensembles the attentionally pooled spatial, temporal, and channel-wise feature representations from all video segments and then applies a temporal attention on top

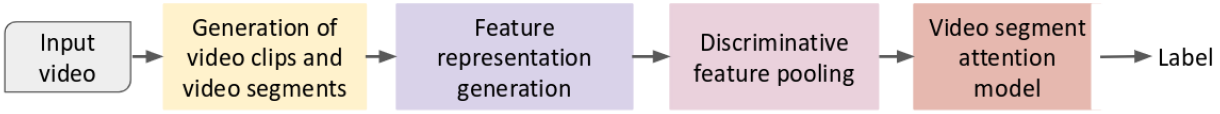

Fig. 4. **Overview of our approach.**

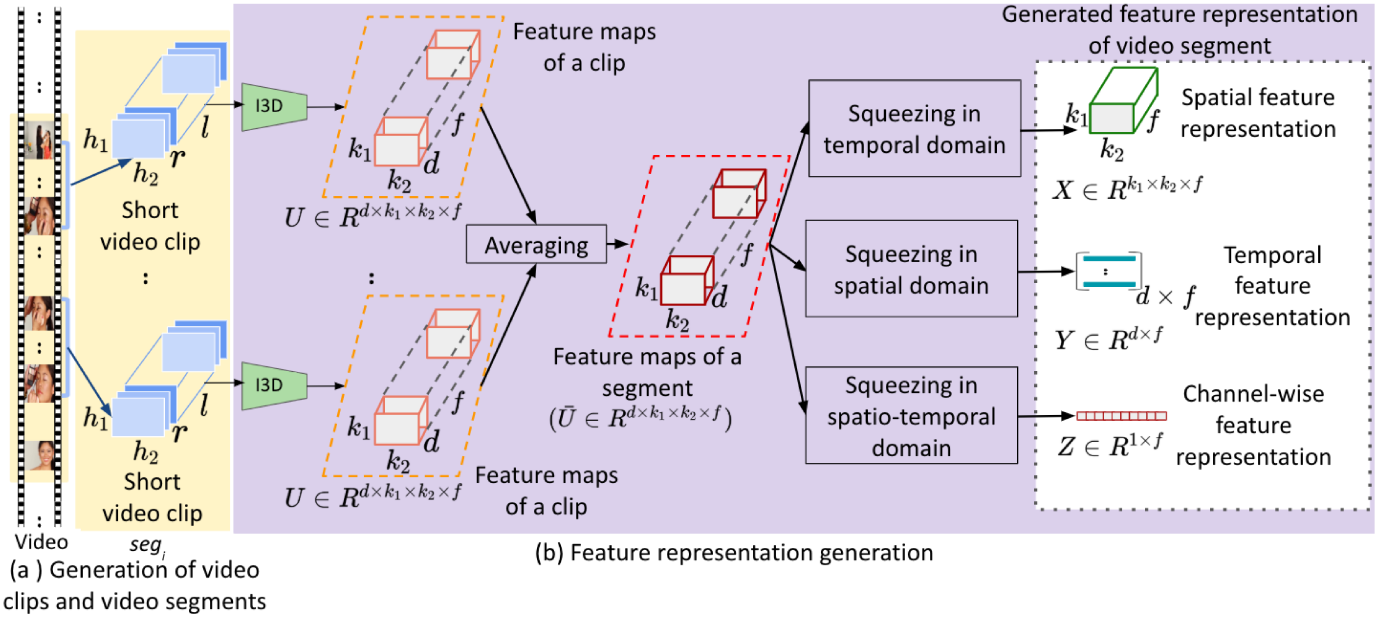

Fig. 5. **Video clips, video segments and feature representation generation.**

of the ensembled features, so that what it learns is more effective for action recognition.

- **Integration of different attentions:** Attention concept has been shown to be effective in many computer vision tasks, but how to design, implement, and integrate different attentions in a unified network is an open research problem. The previous attention-based works [17], [26], [28], [52], [53], [55], [58] either applied the spatial attention on top of the convolutional feature maps of 2D CNN and/or temporal attention on top of the fully-connected layers, while we first generate the spatial, temporal, and channel-wise feature representations from the feature maps of 3D CNN and then apply the corresponding attention on top of the corresponding feature representation to integrate them in a unified network.

III. Approach

In this section, we present our Discriminative Feature Pooling and Video Segment Attention Model. The workflow of our action recognition network is illustrated in Fig. 4.

A. Generation of video clips and video segments

Since some of the frames in a video may be not or less relevant to the action class and action instances may occur in various time instants of a video, we propose to divide the video into short clips and group the video clips into segments, so we can learn attention weights to pool the most discriminative features and emphasize the most representative video segments, eventually leading to accurate video-level action recognition by the discriminative feature pooling and video segment attention model.

Formally, for a given video V with T frames, we use a temporal sliding window of l frames (e.g., $l = 64$) with stride e (e.g., $e = 32$) to generate the video clips with the size of $l \times h_1 \times h_2 \times r$, where h_1, h_2, and r are the height, width and the number of color channels of each frame, respectively. Then the generated video clips are grouped into N segments with equal time periods, $\{seg_i\}_{i=1,...N}$, as shown in Fig. 5(a).

B. Feature representation generation

We adopt the I3D network [35] pretrained on the ImageNet and Kinetics dataset to extract features from every video clip, which are the spatio-temporal feature maps of the last 3D convolutional layer, denoted as $U \in R^{d \times k_1 \times k_2 \times f}$, where d, $k_1 \times k_2$, and f denote the temporal dimension, spatial dimension, and the number of feature channels, respectively. Then, we average feature maps of all video clips within the same video segment, as the feature representation $\bar{U} \in R^{d \times k_1 \times k_2 \times f}$ of this segment. As we aim to learn the spatial, temporal, and channel-wise attention weights to highlight the most discriminative features, we propose to generate spatial, temporal, and channel-wise feature representations for the video segment as shown in Fig. 5(b).

Spatial feature representation: We apply the squeezing operation (e.g., average pooling or max pooling) on $\bar{U} \in R^{d \times k_1 \times k_2 \times f}$ in the temporal domain (d-axis) to get the spatial

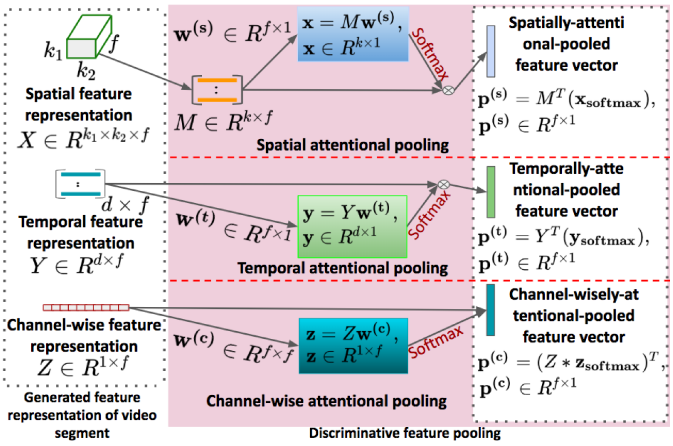

Fig. 6. **Discriminative feature pooling mechanism in a video segment.** Discriminative feature pooling learns attention weights $\mathbf{w}^{(\mathbf{s})}$, $\mathbf{w}^{(\mathbf{t})}$ and $\mathbf{w}^{(\mathbf{c})}$ for spatial, temporal, and channel-wise attentional pooling, respectively, to emphasize the most important features in each video segment. '$\otimes$' and '$*$' denote matrix-vector production and element-wise production, respectively.

feature cube $X \in R^{k_1 \times k_2 \times f}$, which preserves the spatial information of the input video segment.

Temporal feature representation: To get the temporal feature representation from the feature representation of a video segment, we perform the squeezing operation on $\bar{U} \in R^{d \times k_1 \times k_2 \times f}$ in the spatial domain $(k_1 \times k_2)$ to get the temporal feature matrix $Y \in R^{d \times f}$, which represents the temporal feature representation of the input video segment.

Channel-wise feature representation: We perform the squeezing operation on $\bar{U} \in R^{d \times k_1 \times k_2 \times f}$ in the spatio-temporal domain $(d \times k_1 \times k_2)$ to get the channel-wise feature vector $Z \in R^{1 \times f}$, which represents the channel-wise feature representation of the input video segment.

C. Discriminative feature pooling

The spatial, temporal and channel-wise feature representations of a video segment treat every element in the feature maps equally, but some pixels in the video segment may be not or less competent to represent the action and discriminate them from others. Thus, we propose a Discriminative Feature Pooling (DFP) mechanism consisting of spatial, temporal, and channel-wise attentional pooling to gain more attention on those discriminative regions in a video segment. The proposed DFP is a trainable layer, which pools the most discriminative features within a video segment, as shown in Fig. 6.

Spatial attentional pooling: The spatial feature representation of a video segment, $X \in R^{k_1 \times k_2 \times f}$, can be converted to its corresponding Casorati matrix $M \in R^{k \times f}$, where $k = (k_1 \times k_2)$. Each row of M maps to different regions in the input space. From the matrix M, our spatial attentional pooling mechanism learns spatial attention weights $\mathbf{w}^{(\mathbf{s})} \in R^{f \times 1}$ and computes a spatial attention score vector, $\mathbf{x}$, which indicates the feature importance from different spatial regions:

$$\mathbf{x} = M\mathbf{w}^{(\mathbf{s})}, \quad \text{where} \quad \mathbf{x} \in R^{k \times 1} \tag{1}$$

Then, the spatial attention score vector $\mathbf{x}$ is normalized by a *softmax* layer:

$$(x_i)_{softmax} = \frac{exp(x_i)}{\sum_{j=1}^{k} exp(x_j)}, \quad i = 1, ..., k \tag{2}$$

where x_i is the i^{th} dimension of $\mathbf{x}$. We use the notation $\mathbf{x_{softmax}}$ to denote the spatial attention score vector after the *softmax* layer. Finally, the spatially-attentional-pooled feature vector $\mathbf{p}^{(\mathbf{s})}$ of the video segment is computed by

$$\mathbf{p}^{(\mathbf{s})} = M^T(\mathbf{x_{softmax}}), \quad \text{where} \quad \mathbf{p}^{(\mathbf{s})} \in R^{f \times 1} \tag{3}$$

Temporal attentional pooling: Similarly, given the temporal feature representation of a video segment, $Y \in R^{d \times f}$, our temporal attentional pooling mechanism learns temporal attention weights $\mathbf{w}^{(\mathbf{t})} \in R^{f \times 1}$ and computes a temporal attention score vector, $\mathbf{y}$, which indicates the feature importance of different temporal instants within the video segment:

$$\mathbf{y} = Y\mathbf{w}^{(\mathbf{t})}, \quad \text{where} \quad \mathbf{y} \in R^{d \times 1} \tag{4}$$

The temporal attention score vector $\mathbf{y}$ is passed through a *softmax* layer to get $\mathbf{y_{softmax}} \in [0, 1]^d$. Then, the temporally-attentional-pooled feature vector $\mathbf{p}^{(\mathbf{t})}$ of the video segment is computed by

$$\mathbf{p}^{(\mathbf{t})} = Y^T(\mathbf{y_{softmax}}), \quad \text{where} \quad \mathbf{p}^{(\mathbf{t})} \in R^{f \times 1} \tag{5}$$

Channel-wise attentional pooling: From the channel-wise feature representation, $Z \in R^{1 \times f}$, our channel-wise attentional pooling learns channel-wise attention weights $\mathbf{w}^{(\mathbf{c})} \in R^{f \times f}$ and computes a channel-wise attention score, $\mathbf{z}$, indicating the feature importance of different channels:

$$\mathbf{z} = Z\mathbf{w}^{(\mathbf{c})}, \quad \text{where} \quad \mathbf{z} \in R^{1 \times f} \tag{6}$$

The channel-wise attention score vector $\mathbf{z}$ is also passed through a *softmax* layer to get $\mathbf{z_{softmax}} \in [0, 1]^f$. Then, the channel-wisely-attentional-pooled feature vector $\mathbf{p}^{(\mathbf{c})}$ of the video segment is computed by

$$\mathbf{p}^{(\mathbf{c})} = (Z * \mathbf{z_{softmax}})^T, \quad \text{where} \quad \mathbf{p}^{(\mathbf{c})} \in R^{f \times 1} \tag{7}$$

where '$*$' denotes the element-wise multiplication.

The proposed Discriminative Feature Pooling is a trainable layer, where the spatial attention weight vector $\mathbf{w}^{(\mathbf{s})} \in R^{f \times 1}$, temporal attention weight vector $\mathbf{w}^{(\mathbf{t})} \in R^{f \times 1}$, and the channel-wise attention weight matrix $\mathbf{w}^{(\mathbf{c})} \in R^{f \times f}$ are learned during the training time.

D. Video segment attention model

Our DFP generates the most discriminative features ($\mathbf{p}^{(\mathbf{s})}$, $\mathbf{p}^{(\mathbf{t})}$, and $\mathbf{p}^{(\mathbf{c})}$) for each video segment. The next step is to ensemble the spatial, temporal, and channel-wise feature representations from all video segments, with more attention paid to the directly related video segments. To achieve this goal, we propose a Video Segment Attention Model (VSAM) to learn (a) *class-specific* attention weights to classify the video segments into the corresponding action categories and (b) *class-agnostic* attention weights to highlight the most representative video segments without considering the specific

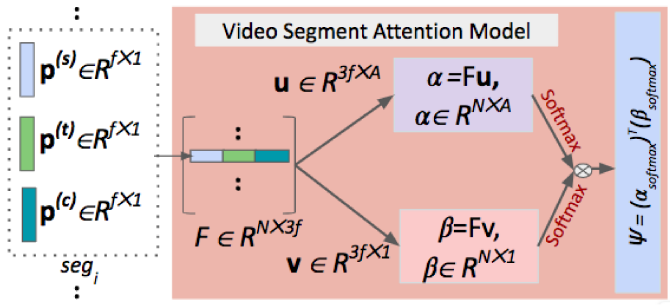

Fig. 7. **Video segment attention model (VSAM).** Our VSAM first ensembles the most discriminative spatial, temporal, and channel-wise features from all video segments in a feature matrix F, which is then used to learn class-specific ($\mathbf{u}$) attention weights to classify each video segment into corresponding action category and class-agnostic ($\mathbf{v}$) attention weights to emphasize the most representative features across the video segments, eventually leading to the accurate video-level classification scores $\psi \in R^{A\times 1}$.

action class information. First, we concatenate all the attentionally pooled feature vectors ($\mathbf{p^{(s)}}$, $\mathbf{p^{(t)}}$, and $\mathbf{p^{(c)}}$), and then the concatenated feature vectors are transposed and stacked in a matrix to generate the feature matrix $F \in R^{N\times 3f}$, as shown in Fig. 7. Each row of the feature matrix represents the most discriminative features of each video segment. Then, we design our VSAM to learn class-specific and class-agnostic weights.

Class-specific attention: We aim to classify each video segment into its corresponding action class based on the generated feature matrix F. Therefore, we learn class-specific attention weights $\mathbf{u} \in R^{3f\times A}$, where A is the number of action categories, to compute class-specific attention scores, α, for all N video segments:

$$\alpha = F\mathbf{u}, \quad \text{where} \quad \alpha \in R^{N\times A} \tag{8}$$

The class-specific attention score matrix contains the independent action class prediction from every video segment. α is passed through a *softmax* layer to get the normalized class-specific attention score $\alpha_{softmax} \in [0,1]^{N\times A}$.

Class-agnostic attention: Since some of the video segments are not relevant to the action class, we aim to learn attention weights to highlight the most representative video segments and suppress the unrelated or less relevant video segments. Therefore, we learn class-agnostic attention weight $\mathbf{v} \in R^{3f\times 1}$ on feature matrix $F \in R^{N\times 3f}$ to compute the class-agnostic attention scores, β, for all N video segments:

$$\beta = F\mathbf{v}, \quad \text{where} \quad \beta \in R^{N\times 1} \tag{9}$$

The class-agnostic attention score vector represents the importance weight of each video segment without considering the specific action class information. β is passed through a *softmax* layer to get the normalized class-agnostic attention score vector: $\beta_{softmax} \in [0,1]^N$.

It should be noted that, as $\mathbf{u} \in R^{3f\times A}$ is learned depending on the action class information and $\mathbf{v} \in R^{3f\times 1}$ is learned without depending on the action class information, we call $\mathbf{u}$ as class-specific and $\mathbf{v}$ as class-agnostic weights.

Video-level action prediction: So far, based on the class-specific attention scores, we get the classification scores of each video segment belonging to every action category, while based on the class-agnostic attention scores, we get the importance weight of each video segment. The classification score on the entire video is computed by weighting the predictions from all video segments $\alpha_{softmax}$ (normalized class-specific attention scores) with weights $\beta_{softmax}$ (normalized class-agnostic attention scores):

$$\psi = (\alpha_{softmax})^T(\beta_{softmax}), \quad \text{where} \quad \psi \in R^{A\times 1} \tag{10}$$

As the classification score vector ψ is computed from normalized attention scores ($\alpha_{softmax}$ and $\beta_{softmax}$), it is in the range of (0, 1) and no further softmax is required at this stage.

E. Two-stream networks

As multiple streams of information (such as RGB and optical flow (OF)) often provide a significant boost in the action recognition performance [15], [33], [35], we train two DFP-VSAM networks on RGB and optical flow, separately. The video-level action predictions from two stream networks, $\psi^{(RGB)}$ and $\psi^{(OF)}$, are combined to classify the input video:

$$final\_score = \gamma \cdot \psi^{(RGB)} + (1-\gamma) \cdot \psi^{(OF)} \tag{11}$$

where $\gamma \in [0,1]$ is the combination factor. We computed the optical flow by the TV-L1 algorithm [59].

F. Loss function

The loss used to train our DFP-VSAM network is based on the standard multi-label cross-entropy loss between the ground truth and the prediction from our network:

$$loss = \sum_{i=1}^{L}\sum_{j=1}^{A} \xi_{ij} \log \psi(V_i) \tag{12}$$

where ξ_{ij} is set to 1 if video V_i contains action instances of the j-th category, and to 0 otherwise, L is the number of training videos. If a video contains action instances from multiple classes, we first normalize the label vector ξ with its l_1-norm [60], i.e. $\bar{\xi} = \xi/\|\xi\|_1$, which is then used to compute cross-entropy loss.

IV. Experiments

In this section, we present our experimental results, performance comparison with state-of-the-art methods, and ablation studies.

A. Implementation details

We train two DFP-VSAM networks with identical architectures for RGB and optical flow, separately. We generate video clips by sliding a temporal window of 64 frames with stride 32, then they are resized to have a tensor size of $64\times 224\times 224\times 3$ for RGB and $64\times 224\times 224\times 2$ for optical flow. We load the video clips into the pre-trained I3D network to extract the spatio-temporal feature maps of the last mixed

concatenated layer, which produces $8 \times 7 \times 7 \times 1024$ feature maps, to feed into our DFP-VSAM. The parameters of our DFP-VSAM are learned with Adam optimizer [61] with the minibatch size of 32 samples. Our DFP learns spatial attention weights $\mathbf{w}^{(\mathbf{s})}$, temporal attention weights $\mathbf{w}^{(\mathbf{t})}$, and channel-wise $\mathbf{w}^{(\mathbf{c})}$ attention weights to get spatially, temporally, and channel-wisely attentional-pooled feature vectors, respectively, which are further sent to VSAM that learns class-specific attention weights ($\mathbf{u}$) and class-agnostic ($\mathbf{v}$) attention weights to compute the video-level classification scores. The weights of our DFP and VSAM are initialized by Xavier method [62]. Keras with the Tensorflow backend is used to implement our network.

B. Action Recognition from Untrimmed Videos

*1) **Dataset**:* We use THUMOS14 [31] and ActivityNet1.2 [63] datasets to evaluate the performance of our network on the problem of action recognition from untrimmed videos.

THUMOS14 [31]: This dataset has 101 action classes for the action recognition task from untrimmed videos. It is composed of four parts: training data, validation data, testing data, and background data. The entire UCF101 [30] action dataset is used for training, which contains 101 human action categories with 13,320 temporally trimmed videos in total. The validation set has 1010 temporally untrimmed videos. The background set has 2500 untrimmed videos, and the testing set is composed of 1574 temporally untrimmed videos. Some videos in the testing set may contain one or multiple instances from one or multiple action classes, and some videos may not include any actions from the 101 classes. **It should be noted that for the weakly-supervised setting, we do not use the trimmed training dataset in our experiment.** Instead, we use the untrimmed validation set (1010 untrimmed videos) as the training set for weakly-supervised learning, with the same configuration as [52], [53].

ActivityNet1.2 [63]: The ActivityNet1.2 dataset covers 100 action classes, which has temporal boundary annotations for 4819 untrimmed videos for training, 2383 untrimmed videos for validation, and 2480 untrimmed videos for testing. Since the labels of the testing set are withheld, following the rules in the literature [52], [53], **we use the training set without using the temporal annotations to train our network in a weakly-supervised way** and validation set for the evaluation.

Evaluation metrics: For action recognition, we follow the standard evaluation protocol based on Mean Average Precision (mAP). First, we use interpolated Average Precision (AP) as the official measure for evaluating the results on each action class. Then, Mean Average Precision (mAP) is used to evaluate the performance of action recognition on this dataset. The evaluation is conducted using the evaluation code for the action recognition task provided by the corresponding datasets.

*2) **Comparison with the state-of-the-art**:* In this subsection, we compare the performance of our proposed DFP-VSAM with other state-of-the-art methods on THUMOS14 and ActivityNet1.2 datasets.

THUMOS14: We compare the performance of our DFP-VSAM with the performance of other state-of-the-art methods on THUMOS14 dataset. Regarding the level of supervision, we separate the methods into two categories: (i) weak supervision: only use untrimmed validation videos from THUMOS14 for training; and (ii) full supervision: use both untrimmed validation videos from THUMOS14 and trimmed videos from UCF101 for training. We compare with recent successful action recognition methods, which previously achieved the state-of-the-art performance on THUMOS14, including Temporal Segment Networks (TSN) [16], UntrimmedNet [52], W-TALC [53], EMV + RGB [64], IDT + FV [10], Object + Motion [65], and STAN [55]. The methods are also grouped by choice of the feature extractor: BN-Inception [16] and I3D [35]. It should be noted that since the BN-Inception network is a 2D CNN based network, the convolutional feature maps contain spatial and channel-wise information. Therefore, we consider the spatial and channel-wise attentional pooling of our DFP to get the attentionally-pooled feature vectors, which are then passed through the VSAM. The numerical results are summarized in Table I. Our network outperforms all these previous methods and establishes a new state-of-the-art on both weakly-supervised and fully-supervised action recognition on the challenging THUMOS14, regardless of the feature extractor network.

ActivityNet1.2: In Table II, the classification performance (mAP) of our DFP-VSAM on ActivityNet1.2 is reported,

TABLE I
MAP ON THUMOS14. WEAK SUPERVISION INDICATES THAT THE ALGORITHM USES ONLY UNTRIMMED VIDEOS FROM VALIDATION SET OF THUMOS14 FOR TRAINING, WHILE FULL SUPERVISION MEANS THAT THE ALGORITHM USES BOTH UNTRIMMED VIDEOS FROM VALIDATION SET AND TRIMMED VIDEOS FROM TRAINING SET OF THUMOS14 FOR TRAINING.

Supervision	Method	Feature	THUMOS14
Weak	TSN [16]	BN-Inception	68.5
Weak	UntrimmedNet [52]	BN-Inception	74.2
Weak	DFP-VSAM (Ours)	BN-Inception	**75.8**
Weak	W-TALC [53]	I3D	85.6
Weak	DFP-VSAM (Ours)	I3D	**86.9**
Full	EMV + RGB [64]	-	61.5
Full	IDT + FV [10]	-	66.1
Full	Object + Motion [65]	-	71.6
Full	STAN [55]	-	77.3
Full	TSN [16]	BN-Inception	78.5
Full	UntrimmedNet [52]	BN-Inception	82.2
Full	DFP-VSAM (Ours)	BN-Inception	**83.7**
Full	DFP-VSAM (Ours)	I3D	**88.5**

TABLE II
ACTION RECOGNITION PERFORMANCE COMPARISON (MAP) OF OUR DFP-VSAM WITH STATE-OF-THE-ART METHODS ON THE UNTRIMMED DATASET OF ACTIVITYNET1.2.

Method	Feature	ActivityNet1.2
IDT + FV [10]	-	66.5
Two Stream [15]	-	71.9
C3D [19]	-	74.1
TSN [16]	BN-Inception	88.8
UntrimmedNet [52]	BN-Inception	87.7
DFP-VSAM (Ours)	BN-Inception	**89.9**
W-TALC [53]	I3D	93.2
DFP-VSAM (Ours)	I3D	**94.3**

TABLE III
PERFORMANCE OF OUR NETWORK REGARDING TO THE NUMBER OF VIDEO SEGMENTS (N) ON THUMOS14 AND ACTIVITYNET1.2 DATASETS.

Number of video segments (N)	THUMOS14	ActivityNet1.2
5	85.0	93.2
10	85.6	93.7
15	86.4	94.0
20	**86.9**	**94.3**
25	86.3	94.1
Flexible	86.3	93.9

TABLE IV
PERFORMANCE OF OUR NETWORK FOR DIFFERENT SQUEEZING OPERATIONS ON THUMOS14 AND ACTIVITYNET1.2 DATASETS.

Squeezing operation	THUMOS14	ActivityNet1.2
Max pooling	85.6	93.8
Average pooling	**86.9**	**94.3**

where DFP-VSAM is compared with IDT + FV [10], Two Stream [15], C3D [19], Temporal Segment Networks (TSN) [16], UntrimmedNet [52], and WTALC [53]. As shown in Table II, our algorithm outperforms the other state-of-the-art methods, and establishes a new state-of-the-art on action classification on the challenging ActivityNet1.2.

*3) **Parameter Analysis**:* In this subsection, we perform the parameter analysis to determine the important setups of our approach. We perform the parameter analysis on the number of video segments, squeezing function and combination factor, which are selected based on the cross-validation, where we randomly split the train data into 80:20 for training and validation sets multiple times, and train multiple models and choose the one with the best performance of validation set.

Number of video segments: We perform the experiments for both fixed and flexible number of video segments on THUMOS14 and ActivityNet1.2 datasets, as shown in Table III. We test the number of video segments per video from 5 to 25 for the fixed number of video segments, while we consider each video clip as a segment for the flexible number of video segments. If we consider each video clip as a segment, N becomes flexible since different videos have different number of frames and hence different number of video clips. We achieve the best performance for $N = 20$ on THUMOS14 and ActivityNet1.2, and we use it as the number of video segments for the action recognition from untrimmed videos.

Squeezing function: We compared two squeezing functions to generate spatial ($X \in R^{k_1 \times k_2 \times f}$), temporal ($Y \in R^{d \times f}$), and channel-wise ($Z \in R^{1 \times f}$) feature representations from the feature maps ($\bar{U} \in R^{d \times k_1 \times k_2 \times f}$) of a video segment. As shown in Table IV, we get the best performance from average pooling, and we choose it as the squeezing operation.

TABLE V
PERFORMANCE OF OUR NETWORK REGARDING TO THE COMBINATION FACTOR (γ IN EQ.11) ON THUMOS14 AND ACTIVITYNET1.2 DATASETS.

Combination factor (γ)	THUMOS14	ActivityNet1.2
0.3	84.6	92.8
0.4	86.0	93.5
0.5	**86.9**	**94.3**
0.6	86.3	93.9

Combination factor: In Eq.11, a combination factor (γ) is introduced to combine the results from different input modalities (RGB and OF). The performance of our network for different γ's is summarized in Table V, and we set $\gamma = 0.5$ for the action recognition from untrimmed videos.

*4) **Ablation study**:* We conduct several analytic experiments to investigate the effect of each component of our DFP-VSAM network on THUMOS14 and ActivityNet1.2. As shown in Table VI, we performed the ablation studies on our network by comparing nine configurations on three different inputs (RGB, Optical Flow (OF), and RGB + OF):

(i) Baseline (without any attention): To get a better idea of the performance of our network, we configure the network without any attention pipeline as the baseline approach. For this purpose, first we perform the global average pooling operation on the spatio-temporal feature map of a video segment ($\bar{U} \in R^{d \times k_1 \times k_2 \times f}$) to get a feature vector, which is then processed by a fully-connected layer with softmax activation to generate the classification scores. Finally, the predictions from all the video segments are averaged to get the final video-level label prediction, which gets 77.3% mAP, 70.4% mAP and 80.8% mAP on THUMOS14, and 87.4% mAP, 85.1% mAP and 89.1% mAP on ActivityNet1.2 for RGB, OF and RGB+OF, respectively.

(ii) Spatial attention: We apply the spatial attentional pooling mechanism on top of the spatio-temporal feature maps of video segments to get the spatially-attentional-pooled feature vectors, and then the class-specific attention to get the segment-level classification scores, which are finally averaged to get the video-level classification scores. The spatial attention gets 78.4% mAP, 71.2% mAP and 81.5% mAP on THUMOS14, and 87.9% mAP, 85.5% mAP and 89.7% mAP on ActivityNet1.2 for RGB, OF and RGB+OF, respectively.

(iii) Temporal attention: We apply the temporal attentional pooling mechanism on top of the spatio-temporal feature maps and class-specific attention to get the segment-level classification scores, which are averaged to get the video-level classification scores. The temporal attention achieves 79.1% mAP, 71.7% mAP and 82.2% mAP on THUMOS14, and 88.2% mAP, 85.8% mAP and 90.2% mAP on ActivityNet1.2 for RGB, OF and RGB+OF, respectively.

(iv) Channel-wise attention: We apply the channel-wise attentional pooling mechanism on top of the spatio-temporal feature maps and class-specific attention to get the segment-level classification scores, which are then averaged to get the video-level classification scores. The channel-wise attention gets 79.3% mAP, 72.1% mAP and 82.6% mAP on THUMOS14, and 88.6% mAP, 86.1% mAP and 90.8% mAP on ActivityNet1.2 for RGB, OF and RGB+OF, respectively.

(v) Spatial attention + VSAM: We use spatial attentional pooling on top of the spatio-temporal feature maps and VSAM to get the video-level classification scores, which achieves 81.1% mAP (RGB), 73.2% mAP (OF) and 84.2% mAP (RGB+OF) on THUMOS14, and 89.1% mAP (RGB), 86.4% mAP (OF) and 91.6% mAP (RGB+OF) on ActivityNet1.2.

TABLE VI
ABLATION STUDY OF DIFFERENT ARCHITECTURES ON THE DATASETS OF THUMOS14 AND ACITIVITYNET1.2. DFP: DISCRIMINATIVE FEATURE POOLING (SPATIAL ATTENTION + TEMPORAL ATTENTION + CHANNEL-WISE ATTENTION); VSAM: VIDEO SEGMENT ATTENTION MODEL.

Architecture	THUMOS14			ActivityNet1.2		
	RGB	OF	RGB + OF	RGB	OF	RGB + OF
Baseline (without any attention)	77.3	70.4	80.8	87.4	85.1	89.1
Spatial attention	78.4	71.2	81.5	87.9	85.5	89.7
Temporal attention	79.1	71.7	82.2	88.2	85.8	90.2
Channel-wise attention	79.3	72.1	82.6	88.6	86.1	90.8
Spatial attention + VSAM	81.1	73.2	84.2	89.1	86.4	91.6
Temporal attention + VSAM	81.6	73.7	84.8	89.5	86.7	92.1
Channel-wise attention +VSAM	82.5	74.5	85.7	90.1	87.1	92.9
DFP + VSAM (w/o class-agnostic)	83.1	74.7	86.1	91.1	87.7	93.7
DFP + VSAM	**83.9**	**75.1**	**86.9**	**91.5**	**88.2**	**94.3**

(vi) Temporal attention + VSAM: We use temporal attentional pooling on top of the spatio-temporal feature maps and VSAM to get the video-level classification scores, which gets 81.6% mAP (RGB), 73.7% mAP (OF) and 84.8% mAP (RGB+OF) on THUMOS14, and 89.5% mAP (RGB), 86.7% mAP (OF) and 92.1% mAP (RGB+OF) on ActivityNet1.2.

(vii) Channel-wise attention + VSAM: We apply channel-wise attentional pooling on top of the spatio-temporal feature maps and VSAM to get the classification scores, which gets 82.5% mAP (RGB), 74.5% mAP (OF) and 85.7% mAP (RGB+OF) on THUMOS14, and 90.1% mAP (RGB), 87.1% mAP (OF) and 92.9% mAP (RGB+OF) on ActivityNet1.2.

(viii) DFP + VSAM without class-agnostic: We apply DFP that integrates spatial, temporal, and channel-wise attentional pooling on top of the spatio-temporal feature maps, and then the VSAM without class-agnostic attention, which gets 83.1% mAP, 74.7% mAP and 86.1% mAP on THUMOS14, and 91.1% mAP, 87.7% mAP and 93.7% mAP on ActivityNet1.2 for RGB, OF and RGB+OF, respectively.

(ix) DFP + VSAM: We get the best performance (83.9% mAP (RGB), 75.1% mAP (OF) and 86.9% mAP (RGB+OF) on THUMOS14, and 91.5% mAP (RGB), 88.2% mAP (OF) and 94.3% mAP (RGB+OF) on ActivityNet1.2) by applying the DFP with VSAM that contains both class-agnostic and class-specific attention.

Table VI summarizes the results in four aspects: (1) the spatial/temporal/channel-wise attention improves the performance compared to the baseline approach; (2) the VSAM on top of different pooling further improves the performance; (3) over all the nine configurations, the combination of RGB and OF beats the performance of using a single input modality; and (4) our DFP with VSAM that contains both class-agnostic and class-specific attention achieves the best performance.

C. Action Recognition from Trimmed Videos

We also evaluate our framework on trimmed video datasets to verify its effectiveness.

*1) **Datasets**:* We use HMDB51, UCF101 and HOLLYWOOD2 to demonstrate that our network can recognize actions from trimmed videos.

HMDB51 [29]: HMDB51 (Human Motion Database) contains 6766 realistic videos from 51 action classes. The dataset is challenging, since it has diverse background contexts and variations in motion pattern. HMDB51 provides three train-test splits, each with 3570 train videos (70 per action class) and 1530 test videos (30 per action class). Evaluation is performed using average classification accuracy over three splits.

UCF101 [30]: UCF101 dataset includes 101 action classes. UCF101 is composed of about 13320 trimmed videos downloaded from YouTube which contain challenges such as poor lighting, cluttered background and severe camera motion. UCF101 dataset provides three splits of training/testing data, and the performance is measured by mean classification accuracy across the splits.

HOLLYWOOD2 [66]: HOLLYWOOD2 dataset has 1707 videos labeled with 12 human action classes collected from movies. These videos are labeled with 12 classes of human actions. Some videos contain multiple action instances. The training set has 823 videos and the testing set has 884 videos. Evaluation is performed using mean Average Precision (mAP).

*2) **Comparison with state-of-the-art**:* In this subsection, we report the performance of our DFP-VSAM on HMDB51, UCF101 and HOLLYWOOD2 datasets, and also compare the results with state-of-the-art methods.

HMDB51: We compare the performance of our network (DFP-VSAM) with the performance of other state-of-the-art methods on HMDB51 dataset. The results are summarized in Table VII, where we compare our method with both traditional approaches and deep learning based approaches. The traditional approaches include Improved Dense Trajectories with Fisher Vector (IDT + FV) [10], VideoDarwin [67], and Rank pooling with trajectory features (RankPool + IDT) [68], while the deep learning based approaches include TSN [16], Two-stream FCAN [17], Attentional pooling [28], Interpretable attention [58], RSTAN + IDT-FV [69], Two-stream I3D [35], SVM pooling with I3D (SVMP + I3D) [70], DTPP [71], LGD-3D Two-stream [41]. As shown in Table VII, our proposed Discriminative Feature Pooling and Video Segment Attention Model (DFP-VSAM) outperforms the other state-of-the-art methods on HMDB51 (over three splits) dataset.

UCF101: In Table VIII, the average classification accuracy on the three testing splits of UCF101 of our DFP-VSAM is reported, where DFP-VSAM is compared with Two-stream networks [15], LRCN [22], ST-ResNet + IDT [37], C3D [19], STAN [55], TSN [16], RSTAN + IDT-FV [69], Two-strem I3D [35], DTTP [71], and LGD-3D Two-stream [41]. Our proposed

TABLE VII
COMPARISON WITH OTHER STATE-OF-THE-ART METHODS ON THE TRIMMED DATASET OF HMDB51.

Method	Pretraining	HMDB51
IDT + FV [10]	-	57.2
VideoDrawin [67]	-	63.7
RankPool + IDT [68]	-	66.9
TSN [16]	ImageNet	69.4
Two-stream FCAN [17]	-	68.2
Attenional pooling [28]	-	52.2
Interpretable attention [58]	-	54.4
RSTAN + IDT-FV [69]	-	79.9
Two-stream I3D [35]	ImageNet+Kinetics-400	80.7
SVMP + I3D [70]	ImageNet+Kinetics-400	81.3
DTPP [71]	-	82.1
LGD-3D Two-stream [41]	ImageNet+Kinetics-600	80.5
DFP-VSAM (Ours)	ImageNet+Kinetics-400	**82.4**

TABLE VIII
COMPARISON WITH OTHER STATE-OF-THE-ART METHODS ON THE TRIMMED DATASET OF UCF101.

Method	pretraining	UCF101
Two-stream networks [15]	ImageNet	88.0
LRCN [22]	-	82.9
ST-ResNet + IDT [37]	-	94.6
Interpretable attention [58]	-	87.1
C3D (3 nets) [19]	Sports-1M	90.4
Two-stream FCAN [17]	-	93.4
STAN [55]	-	93.6
TSN [16]	ImageNet	94.2
RSTAN + IDT-FV [69]	-	95.1
Two-stream I3D [35]	ImageNet+Kinetics-400	98.0
DTPP [71]	-	98.0
LGD-3D Two-stream [41]	ImageNet+Kinetics-600	**98.2**
DFP-VSAM (Ours)	ImageNet+Kinetics-400	98.0

DFP-VSAM outperforms most of the existing methods. Our DFP-VSAM is on par with Two-stream I3D [35] and DTPP [71] on UCF101, but exceeds them on HMDB51 by 1.7% and 0.3%, respectively, as shown in Table VII. On UCF101, the performance of our method (98.0%) is inferior to that of LGD-3D Two-stream [41] (98.2%). However, the LGD-3D Two-stream [41] used the pre-trained network, which is trained on ImageNet+Kinetics-600 dataset and also fine-tuned the pre-trained network on UCF101, while we achieve the comparable performance without fine-tuning the pre-trained network (trained on ImageNet+Kinetics-400 dataset) on UCF101.

HOLLYWOOD2: This dataset is small compared to HMBD51 and UCF101, but it is challenging as some videos contain multiple action instances. We train our model on HOLLYWOOD2, and compare our performance in Table IX. As shown in Table IX, our method achieves a significant boost in performance and establishes a new state-of-the-art on HOLLYWOOD2 dataset.

*3) **Parameter Analysis**:* We perform the parameter analysis to see the performance of our network regarding to the number of video segments (N), different squeezing operations, and different combination factors (γ) on HMDB51, UCF101 and HOLLYWOOD2 datasets.

TABLE IX
COMPARISON WITH OTHER STATE-OF-THE-ART METHODS ON THE TRIMMED DATASET OF HOLLYWOOD2.

Method	HOLLYWOOD2 (mAP)
IDT + FV [10]	64.3
VideoDrawin [67]	73.7
RankPool + IDT [68]	76.7
Two-stream FCAN [17]	78.4
DFP-VSAM (Ours)	**84.8**

TABLE X
PERFORMANCE OF OUR NETWORK REGARDING TO THE NUMBER OF VIDEO SEGMENTS (N) ON HMDB51 (SPLIT-1), UCF101 (SPLIT-1) AND HOLLYWOOD2 DATASETS.

N	HMDB51	UCF101	HOLLYWOOD2
2	81.2	97.4	83.5
3	81.6	97.6	84.1
4	82.1	97.9	84.5
5	**82.6**	**98.0**	**84.8**
6	81.3	97.8	84.4

TABLE XI
PERFORMANCE OF OUR NETWORK FOR DIFFERENT SQUEEZING OPERATIONS ON HMDB51 (SPLIT-1), UCF101 (SPLIT-1) AND HOLLYWOOD2 DATASETS.

Squeezing operation	HMDB51	UCF101	HOLLYWOOD2
Max pooling	81.4	97.8	82.7
Average pooling	**82.6**	**98.0**	**84.8**

Unlike untrimmed videos in THUMOS14 and ActivityNet1.2, trimmed videos in HMDB51, UCF101 and HOLLYWOOD2 are shorter in time duration. Therefore, we test the number of video segments per video from 2 to 6 for HMDB51, UCF101 and HOLLYWOOD2. As shown in Table X, we achieve the best performance for $N = 5$ on HMDB51, UCF101 and HOLLYWOOD2, and we use it as the number of video segments.

The performance of our network for different squeezing operations on HMDB51, UCF101 and HOLLYWOOD2 datasets is summarized in Table XI. We get the best performance from average pooling, and we choose it as the squeezing operation for the action recognition from trimmed videos.

The performance of our network regarding to the combination factors (γ) on HMDB51, UCF101 and HOLLYWOOD2 is summarized in Table XII. We achieve the best performance for $\gamma = 0.5$.

*4) **Ablation Study**:* The performance of our network for different configurations on HMDB51, UCF101 and HOLLYWOOD2 datasets is summarized in Table XIII. The first configuration in Table XIII shows the results of the baseline approach without any attention pipeline. The second set in Table XIII shows the performance of different pooling mechanisms, which includes the performance of individual spatial, temporal, and channel-wise attentional pooling. All the individual spatial, temporal, and channel-wise attentional

TABLE XII
PERFORMANCE OF OUR NETWORK REGARDING TO THE COMBINATION FACTOR (γ IN EQ.11) ON HMDB51 (SPLIT-1), UCF101 (SPLIT-1) AND HOLLYWOOD2 DATASETS.

γ	HMDB51	UCF101	HOLLYWOOD2
0.3	81.3	97.7	83.7
0.4	81.9	97.9	84.2
0.5	**82.6**	**98.0**	**84.8**
0.6	82.1	97.8	84.5

TABLE XIII
ABLATION STUDY OF DIFFERENT ARCHITECTURES ON THE DATASETS OF HMDB51 (SPLIT-1), UCF101 (SPLIT-1) AND HOLLYWOOD2. DFP: DISCRIMINATIVE FEATURE POOLING (SPATIAL ATTENTION+TEMPORAL ATTENTION+CHANNEL-WISE ATTENTION); VSAM: VIDEO SEGMENT ATTENTION MODEL.

Architecture	HMDB51			UCF101			HOLLYWOOD2		
	RGB	OF	RGB + OF	RGB	OF	RGB + OF	RGB	OF	RGB + OF
Baseline (without any attention)	70.4	72.1	75.6	94.2	95.5	96.7	71.6	74.3	78.8
Spatial attention	71.1	72.9	76.8	94.4	95.6	96.9	72.1	75.7	80.1
Temporal attention	71.7	73.5	77.3	94.5	95.8	97.0	72.7	76.4	80.6
Channel-wise attention	72.8	74.6	78.7	94.8	96.0	97.2	73.5	77.6	81.4
Spatial attention + VSAM	74.4	75.9	80.2	95.1	96.1	97.5	74.3	78.1	82.1
Temporal attention + VSAM	74.7	76.3	80.6	95.3	96.4	97.7	75.1	78.8	82.7
Channel-wise attention +VSAM	75.4	77.1	81.1	95.4	96.6	97.8	76.7	80.3	83.4
DPF + VSAM (w/o class-agnostic)	75.6	77.4	81.5	95.6	96.8	97.9	77.1	80.8	83.9
DFP + VSAM	**77.1**	**79.5**	**82.6**	**95.8**	**96.9**	**98.0**	**78.2**	**81.7**	**84.8**

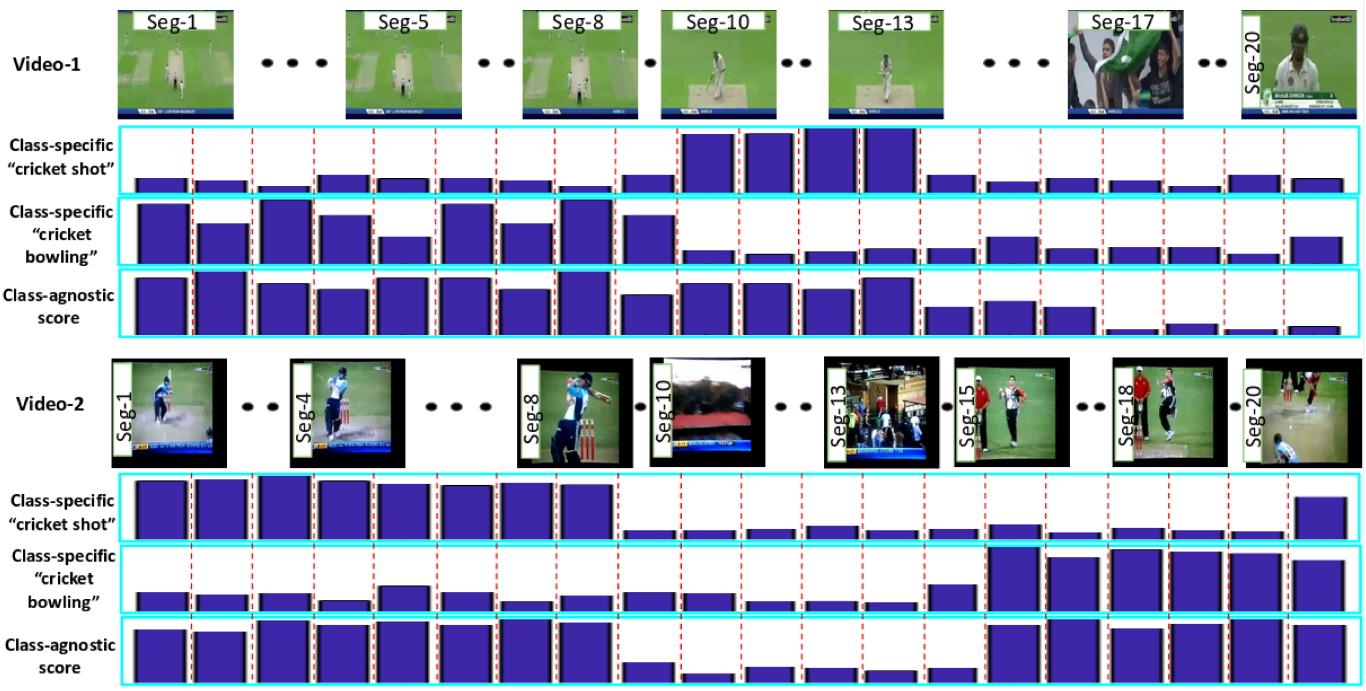

Fig. 8. **Visualization of class-specific and class-agnostic scores on test samples (untrimmed video) of THUMOS14.** The bar graph represents the scores. These videos contain two different action classes "cricket shot" and "cricket bowling". The class-specific scores are high for the video segments that are related to the corresponding action classes. For seg-20 of video-2, both class-specific scores of "cricket shot" and "cricket bowling" are high, as this video segment contains two actions in the same segment. The most representative video segments are with the high class-agnostic scores, while the less or not relevant video segments are with the low class-agnostic scores.

pooling improve the performance, compared to the baseline approach. The third set shows the effect of VSAM on top of different pooling mechanisms, which further improves the performance. The combination of RGB and OF is helpful compared to the single input modality. The last set in Table XIII shows the effect of class-agnostic attention in VSAM, where our DFP with VSAM that contains both class-agnostic and class-specific attention achieves the best performance.

D. Qualitative analysis

Some multi-label untrimmed videos are visualized in Fig. 8, which contain instances of two different actions ("cricket shot" and "cricket bowling") with unrelated video segments. Although "cricket shot" and "cricket bowling" are two different action classes, they share the similar background features and these two actions generally co-occur in videos. Inspite of these challenges, our DFP-VSAM can find the fine details to classify the two actions (shown by class-specific scores), and separate the video segments related to the actions from other unrelated ones (shown by the class-agnostic scores).

We also visualize the spatial attention in the DFP on some randomly selected test samples of the "jump," "hug," and "drink," actions from HMDB51 dataset, which represent actions by human-alone, human-human interaction, and human-object interaction, respectively. From the first two videos (video-1 and video-2) in Fig. 9, we see that the model attempts to focus on the person performing jump to recognize the "jump" action. The second row of Fig. 9 shows the example belonging to the "hug" action from two different videos (video-3 and video-4). It appears that the videos contain multiple persons, and the model correctly predicts that a hug is going to take place and attempts to focus on the region

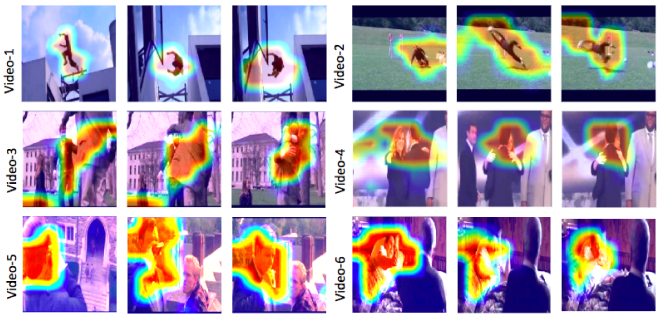

Fig. 9. **Visualization of our spatial attention over time.** Our network learns to look at the relevant parts where the action of interest is being performed. The three rows represent "jump" (human-alone), "hug" (human-human interaction) and "drink" (human-object interaction), respectively.

between two persons, where the action of interest is performed over time. Finally, the third row of Fig. 9 shows the example of the "drink," action from two different videos (video-5 and video-6). Although most of the frames of these two videos contain two persons, our model successfully focuses on the person performing drink with the object (e.g., glass) to correctly recognize the "drink" action.

E. Discussions and future works

For video action recognition, particularly on untrimmed videos, which features of a video segment should get more attention to highlight the discriminative features related to the action class is still an open research question. In this paper, we propose a Discriminative Feature Pooling and a Video Segment Attention Model (DFP-VSAM), to classify human actions from videos. During training, only the video-level labels are given, but which portion of the video contains which actions is unknown. Intuitively, the spatial, temporal, and channel-wise features of a video segment obtained from a 3D CNN contain the vital information to effectively recognize human actions. This motives us to design the Discriminative Feature Pooling (DFP) that contains spatial, temporal, and channel-wise attentional pooling in a unified network. The performance improvements from different attentions in our DFP indicate that the attentional pooling can effectively highlight the most discriminative features inside a video segment. In addition to the discriminative features inside a video segment, which video segment of a long untrimmed video should get more attention is also another open research question. This motivates us to design a Video Segment Attention Model (VSAM) on the whole video. The further improvement in action recognition performance after applying VSAM on top of different attentions and DFP indicates that the VSAM can help highlight the most representative video segments in videos, particularly in untrimmed videos.

The proposed approach achieved superior action recognition performances on large-scale untrimmed datasets and medium-size trimmed datasets. In the future, we will try to test its performance on the large-scale trimmed datasets. Usually, the size of the pre-extracted spatio-temporal feature of 3D CNN is large, which may be problematic for large-scale datasets due to GPU memory constraints. A possible solution to this problem may be to design a data-loader, which is iterable over the dataset with randomly selected number of samples. The number of samples will depend on the available GPU memory. Furthermore, state-of-the-arts on the large-scale trimmed datasets improved the action recognition performance through new designs of the backbone CNN network. On the other hand, since we used a pre-designed backbone CNN network as a feature extractor and applied our proposed model only on top of the last convolutional feature maps, we do not access the lower-level information of the backbone, and without fine-tuning the backbone network, the higher-level feature map may not be the best tuned for the action recognition purpose. Therefore, in the future, we will explore our discriminative feature pooling on top of different convolutional layers of the backbone CNN network and finetune or train the network from the scratch to verify our approach on large-scale trimmed datasets.

V. Conclusions

We have introduced a new video-based human action recognition network that integrates Discriminative Feature Pooling (DFP) with Video Segment Attention Model (VSAM), in an attempt to address the three main challenges: (1) how to attentionally pool 3D convolutional feature maps of a video segment to highlight the most discriminative features to classify actions; (2) which video segment's features should get more attentions to represent an action; and (3) how to train the network from weakly-labeled untrimmed videos. Evaluated on four widely benchmarked datasets, our action recognition network (DFP-VSAM) outperforms the current state-of-the-art action recognition methods, by learning to look at the relevant parts where the action of interest is being performed. The superior performance of the proposed model may be ascribed to its advantages of the joint design of DFP and VSAM modules, which are optimized in an end-to-end manner. The proposed network is also efficient and easy to implement.

Acknowledgment

This research work is supported by the National Science Foundation via CPS Synergy project CMMI-1646162, National Robotics Initiative project NRI-1830479, and Future of Work at the Human-Technology Frontier project ECCS-2025929. Any opinions, findings, and conclusions or recommendations expressed in this material are those of the authors and do not necessarily reflect the views of the National Science Foundation.

References

[1] J. K. Aggarwal and Q. Cai, "Human motion analysis: A review," *CVIU*, 1999.
[2] L. Wang, W. Hu, and T. Tan, "Recent developments in human motion analysis," *Pattern recognition*, 2003.
[3] B. Yao and L. Fei-Fei, "Modeling mutual context of object and human pose in human-object interaction activities," in *CVPR*, 2010.
[4] W. Yang, Y. Wang, and G. Mori, "Recognizing human actions from still images with latent poses," in *CVPR*, 2010.
[5] F. Angelini, Z. Fu, Y. Long, L. Shao, and S. M. Naqvi, "2d pose-based real-time human action recognition with occlusion-handling," *IEEE TMM*, 2019.
[6] Z. Fu, F. Angelini, J. Chambers, and S. M. Naqvi, "Multi-level cooperative fusion of gm-phd filters for online multiple human tracking," *IEEE TMM*, 2019.

[7] I. Haritaoglu, D. Harwood, and L. Davis, "W4: Real-time surveillance of people and their activities." *IEEE TPAMI*, 2000.

[8] M. Hasan and A. K. Roy-Chowdhury, "A continuous learning framework for activity recognition using deep hybrid feature models," *IEEE TMM*, 2015.

[9] H. Wang, A. Klaser, C. Schmid, and C.-L. Liu, "Action recognition by dense trajectories," in *CVPR*, 2011.

[10] H. Wang and C. Schmid, "Action recognition with improved trajectories," in *ICCV*, 2013.

[11] Y. Shi, Y. Tian, Y. Wang, and T. Huang, "Sequential deep trajectory descriptor for action recognition with three-stream cnn," *IEEE TMM*, 2017.

[12] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *CVPR*, 2016.

[13] K. Simonyan and A. Zisserman, "Very deep convolutional networks for large-scale image recognition," in *ICLR*, 2014.

[14] C. Szegedy, S. Ioffe, V. Vanhoucke, and A. Alemi, "Inception-v4, inception-resnet and the impact of residual connections on learning," in *AAAI*, 2017.

[15] K. Simonyan and A. Zisserman, "Two-stream convolutional networks for action recognition in videos," in *NIPS*, 2014.

[16] L. Wang, Y. Xiong, Z. Wang, Y. Qiao, D. Lin, X. Tang, and L. Van Gool, "Temporal segment networks: Towards good practices for deep action recognition," in *ECCV*, 2016.

[17] A. Tran and L.-F. Cheong, "Two-stream flow-guided convolutional attention networks for action recognition," in *ICCV Workshops*, 2017.

[18] S. Sun, Z. Kuang, L. Sheng, W. Ouyang, and W. Zhang, "Optical flow guided feature: A fast and robust motion representation for video action recognition," in *CVPR*, 2018.

[19] D. Tran, L. Bourdev, R. Fergus, L. Torresani, and M. Paluri, "Learning spatiotemporal features with 3d convolutional networks," in *ICCV*, 2015.

[20] S. Ji, W. Xu, M. Yang, and K. Yu, "3d convolutional neural networks for human action recognition," *IEEE TPAMI*, 2012.

[21] X. Wang, L. Gao, P. Wang, X. Sun, and X. Liu, "Two-stream 3-d convnet fusion for action recognition in videos with arbitrary size and length," *IEEE TMM*, 2017.

[22] J. Donahue, L. Anne Hendricks, S. Guadarrama, M. Rohrbach, S. Venugopalan, K. Saenko, and T. Darrell, "Long-term recurrent convolutional networks for visual recognition and description," in *CVPR*, 2015.

[23] G. Varol, I. Laptev, and C. Schmid, "Long-term temporal convolutions for action recognition," *IEEE TPAMI*, 2017.

[24] S. Zhang, Y. Yang, J. Xiao, X. Liu, Y. Yang, D. Xie, and Y. Zhuang, "Fusing geometric features for skeleton-based action recognition using multilayer lstm networks," *IEEE TMM*, 2018.

[25] L. Yao, A. Torabi, K. Cho, N. Ballas, C. Pal, H. Larochelle, and A. Courville, "Describing videos by exploiting temporal structure," in *ICCV*, 2015.

[26] S. Sharma, R. Kiros, and R. Salakhutdinov, "Action recognition using visual attention," in *ICLR Workshops*, 2015.

[27] J. Hu, L. Shen, and G. Sun, "Squeeze-and-excitation networks," in *CVPR*, 2018.

[28] R. Girdhar and D. Ramanan, "Attentional pooling for action recognition," in *NIPS*, 2017.

[29] H. Kuehne, H. Jhuang, E. Garrote, T. Poggio, and T. Serre, "Hmdb: a large video database for human motion recognition," in *ICCV*, 2011.

[30] K. Soomro, A. R. Zamir, and M. Shah, "Ucf101: A dataset of 101 human actions classes from videos in the wild," *arXiv preprint arXiv:1212.0402*, 2012.

[31] H. Idrees, A. R. Zamir, Y.-G. Jiang, A. Gorban, I. Laptev, R. Sukthankar, and M. Shah, "The thumos challenge on action recognition for videos "in the wild"," *CVIU*, 2017.

[32] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "Imagenet classification with deep convolutional neural networks," in *NIPS*, 2012.

[33] A. Karpathy, G. Toderici, S. Shetty, T. Leung, R. Sukthankar, and L. Fei-Fei, "Large-scale video classification with convolutional neural networks," in *CVPR*, 2014.

[34] S. Ioffe and C. Szegedy, "Batch normalization: Accelerating deep network training by reducing internal covariate shift," in *ICML*, 2015.

[35] J. Carreira and A. Zisserman, "Quo vadis, action recognition? a new model and the kinetics dataset," in *CVPR*, 2017.

[36] G. W. Taylor, R. Fergus, Y. LeCun, and C. Bregler, "Convolutional learning of spatio-temporal features," in *ECCV*, 2010.

[37] R. Christoph and F. A. Pinz, "Spatiotemporal residual networks for video action recognition," in *NIPS*, 2016.

[38] C. Feichtenhofer, A. Pinz, and A. Zisserman, "Convolutional two-stream network fusion for video action recognition," in *CVPR*, 2016.

[39] Y. Yang, C. Deng, S. Gao, W. Liu, D. Tao, and X. Gao, "Discriminative multi-instance multitask learning for 3d action recognition," *IEEE TMM*, 2016.

[40] Y. Wang, M. Long, J. Wang, and P. S. Yu, "Spatiotemporal pyramid network for video action recognition," in *CVPR*, 2017.

[41] Z. Qiu, T. Yao, C.-W. Ngo, X. Tian, and T. Mei, "Learning spatio-temporal representation with local and global diffusion," in *CVPR*, 2019.

[42] C. Feichtenhofer, H. Fan, J. Malik, and K. He, "Slowfast networks for video recognition," in *ICCV*, 2019.

[43] C. Li, Q. Zhong, D. Xie, and S. Pu, "Collaborative spatiotemporal feature learning for video action recognition," in *CVPR*, 2019.

[44] H. Bilen and A. Vedaldi, "Weakly supervised deep detection networks," in *CVPR*, 2016.

[45] Y. Zhang, K. Jia, and Z. Wang, "Part-aware fine-grained object categorization using weakly supervised part detection network," *IEEE TMM*, 2019.

[46] Q. Tao, H. Yang, and J. Cai, "Exploiting web images for weakly supervised object detection," *IEEE TMM*, 2018.

[47] P. Bojanowski, F. Bach, I. Laptev, J. Ponce, C. Schmid, and J. Sivic, "Finding actors and actions in movies," in *ICCV*, 2013.

[48] I. Laptev, M. Marszalek, C. Schmid, and B. Rozenfeld, "Learning realistic human actions from movies," in *CVPR*, 2008.

[49] D.-A. Huang, L. Fei-Fei, and J. C. Niebles, "Connectionist temporal modeling for weakly supervised action labeling," in *ECCV*, 2016.

[50] C. Gan, T. Yao, K. Yang, Y. Yang, and T. Mei, "You lead, we exceed: Labor-free video concept learning by jointly exploiting web videos and images," in *CVPR*, 2016.

[51] T. Yu, L. Wang, C. Da, H. Gu, S. Xiang, and C. Pan, "Weakly semantic guided action recognition," *IEEE TMM*, 2019.

[52] L. Wang, Y. Xiong, D. Lin, and L. Van Gool, "Untrimmednets for weakly supervised action recognition and detection," in *CVPR*, 2017.

[53] S. Paul, S. Roy, and A. K. Roy-Chowdhury, "W-talc: Weakly-supervised temporal activity localization and classification," in *ECCV*, 2018.

[54] C. Yan, Y. Tu, X. Wang, Y. Zhang, X. Hao, Y. Zhang, and Q. Dai, "Stat: Spatial-temporal attention mechanism for video captioning," *IEEE TMM*, 2019.

[55] D. Li, T. Yao, L.-Y. Duan, T. Mei, and Y. Rui, "Unified spatio-temporal attention networks for action recognition in videos," *IEEE TMM*, 2018.

[56] S. Song, C. Lan, J. Xing, W. Zeng, and J. Liu, "An end-to-end spatio-temporal attention model for human action recognition from skeleton data," in *AAAI*, 2017.

[57] Q. Wang, C. Yuan, J. Wang, and W. Zeng, "Learning attentional recurrent neural network for visual tracking," *IEEE TMM*, 2018.

[58] L. Meng, B. Zhao, B. Chang, G. Huang, W. Sun, F. Tung, and L. Sigal, "Interpretable spatio-temporal attention for video action recognition," in *CVPR Workshops*, 2019.

[59] C. Zach, T. Pock, and H. Bischof, "A duality based approach for realtime tv-l 1 optical flow," in *Pattern Recognition*, 2007.

[60] Y. Wei, W. Xia, J. Huang, B. Ni, J. Dong, Y. Zhao, and S. Yan, "Cnn: Single-label to multi-label," *arXiv preprint arXiv:1406.5726*, 2014.

[61] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," *arXiv preprint arXiv:1412.6980*, 2014.

[62] X. Glorot and Y. Bengio, "Understanding the difficulty of training deep feedforward neural networks," in *AISTATS*, 2010.

[63] F. Caba Heilbron, V. Escorcia, B. Ghanem, and J. Carlos Niebles, "Activitynet: A large-scale video benchmark for human activity understanding," in *CVPR*, 2015.

[64] B. Zhang, L. Wang, Z. Wang, Y. Qiao, and H. Wang, "Real-time action recognition with enhanced motion vector cnns," in *CVPR*, 2016.

[65] M. Jain, J. C. Van Gemert, and C. G. Snoek, "What do 15,000 object categories tell us about classifying and localizing actions?" in *CVPR*, 2015.

[66] M. Marszalek, I. Laptev, and C. Schmid, "Actions in context," in *CVPR*, 2009.

[67] B. Fernando, E. Gavves, J. M. Oramas, A. Ghodrati, and T. Tuytelaars, "Modeling video evolution for action recognition," in *CVPR*, 2015.

[68] B. Fernando, P. Anderson, M. Hutter, and S. Gould, "Discriminative hierarchical rank pooling for activity recognition," in *CVPR*, 2016.

[69] W. Du, Y. Wang, and Y. Qiao, "Recurrent spatial-temporal attention network for action recognition in videos," *IEEE TIP*, 2017.

[70] J. Wang, A. Cherian, F. Porikli, and S. Gould, "Video representation learning using discriminative pooling," in *CVPR*, 2018.

[71] J. Zhu, Z. Zhu, and W. Zou, "End-to-end video-level representation learning for action recognition," in *ICPR*, 2018.