Appearance-and-Relation Networks for Video Classification

Limin Wang[1,2]   Wei Li[3]   Wen Li[2]   Luc Van Gool[2]

[1]State Key Laboratory for Novel Software Technology, Nanjing University, China
[2]Computer Vision Laboratory, ETH Zurich, Switzerland   [3]Google Research

Abstract

Spatiotemporal feature learning in videos is a fundamental problem in computer vision. This paper presents a new architecture, termed as Appearance-and-Relation Network *(ARTNet), to learn video representation in an end-to-end manner. ARTNets are constructed by stacking multiple generic building blocks, called as* SMART, *whose goal is to simultaneously model appearance and relation from RGB input in a separate and explicit manner. Specifically, SMART blocks decouple the spatiotemporal learning module into an appearance branch for spatial modeling and a relation branch for temporal modeling. The appearance branch is implemented based on the linear combination of pixels or filter responses in each frame, while the relation branch is designed based on the multiplicative interactions between pixels or filter responses across multiple frames. We perform experiments on three action recognition benchmarks: Kinetics, UCF101, and HMDB51, demonstrating that SMART blocks obtain an evident improvement over 3D convolutions for spatiotemporal feature learning. Under the same training setting, ARTNets achieve superior performance on these three datasets to the existing state-of-the-art methods.* [1]

1. Introduction

Deep learning has witnessed a series of remarkable successes in computer vision. In particular, Convolutional Neural Networks (CNNs) [25] have turned out to be effective for visual tasks in image domain, such as image classification [21, 13, 36, 39], object detection [12], and semantic segmentation [27]. Deep models have been also introduced into video domain for action recognition [2, 46, 50, 35, 49, 41, 19, 51], and obtain comparable or better recognition accuracy to those traditional methods with hand-crafted representations [23, 44, 47]. However, the progress of architecture design and representation learning in video domain is much slower, partially due to its inherent complexity and

higher dimension. Video could be viewed as the temporal evolution of a sequence of static images. It is generally assumed that two visual cues are crucial for video classification and understanding: (1) *static appearance* in each frame, and (2) *temporal relation* across multiple frames. Therefore, an effective deep architecture should be able to capture both information to achieve excellent classification accuracy.

There are three kinds of successful architectures or frameworks for video classification [2]: (1) two-stream CNNs [35], (2) 3D CNNs [18, 41], and (3) 2D CNNs with temporal models on top such as LSTM [5, 33], temporal convolution [33], sparse sampling and aggregation [49], and attention modeling [48, 28, 9]. Two-stream CNNs capture appearance and motion information with different streams, which turn out to be effective for video classification. Yet, it is time-consuming to train two networks and calculate optical flow in advance. To overcome this limitation, 3D CNNs employ 3D convolutions and 3D pooling operations to directly learn spatiotemporal features from stacked RGB volumes. However, the performance of 3D CNNs is still worse than two-stream CNNs, and it is still unclear whether this straightforward 3D extension over 2D convolution could efficiently model static appearance and temporal relation. 2D CNNs with temporal models usually focus on capturing coarser and long-term temporal structure, but lack capacity of representing finer temporal relation in a local spatiotemporal window.

In this paper, we address the problem of capturing appearance and relation in video domain, by proposing a new architecture unit termed as *SMART* block. Our SMART block aims to Simultaneously Model Appearance and RelaTion from RGB input in a separate and explicit way with a two-branch unit, in contrast to modeling them with two-stream inputs [35] or jointly and implicitly with a 3D convolution [41]. As shown in Figure 1, our SMART block is a multi-branch architecture, which is composed of appearance branch and relation branch, and fuses them with a concatenation and reduction operation. The appearance branch is based on the *linear combination* of pixels or filter responses in each frame to model spatial structure, while the

---

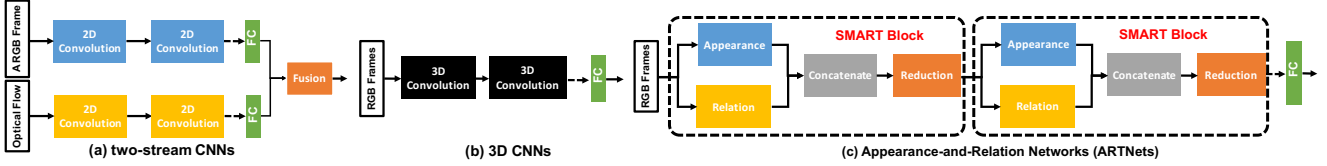[1]The code is at https://github.com/wanglimin/ARTNet.

Figure 1. **Video architecture comparison**: Our Appearance-and-Relation Networks (ARTNets) are constructed based on the SMART building block, which aims to simultaneously model appearance and relation from RGB in a separate and explicit way. In contrast, two-stream CNNs model them with two inputs and 3D CNNs model them jointly and implicitly with a single 3D convolution.

relation branch is based on the *multiplicative interactions* between pixels or filter responses across multiple frames to capture temporal dynamics. Specifically, the appearance branch is implemented with a standard 2D convolution and the relation branch is implemented with a square-pooling structure. The responses from two branches are further concatenated and reduced to a more compact representation.

A SMART block is a basic and generic building module for video architecture design. For video classification, we present an appearance-and-relation network (ARTNet) by stacking a collection of SMART blocks. Essentially, the appearance and relation information in video domain exhibit multi-scale spatiotemporal structure. The ARTNet is able to capture this visual structure in a hierarchical manner, where SMART units in the early layers focus on describe local structure in a short term, while the ones in the later layers can capture increasingly coarser and longer-range visual structure. An ARTNet is a simple and general architecture which offers flexible implementations. In the current implementation of this paper, the ARTNet is instantiated with the network of C3D-ResNet18 [42] for an engineering compromise between accuracy and computation consumption. Moreover, our ARTNet is complementary to those long-term temporal models, which means any of them could be employed to enhance its modeling capacity. As an example, we use the framework of temporal segment network (TSN) [49] to jointly train ARTNets from a set of sparsely sampled snippets and further improve the recognition accuracy.

We test the ARTNet on the task of action recognition in video classification. Particularly, we first study the performance of the ARTNet on the Kinetics dataset [20]. We observe that our ARTNet obtains an evident improvement over C3D, and superior performance to the exiting state-of-the-art methods on this challenging benchmark under the setting of training from scratch with only RGB input. To further demonstrate the generality of ARTNet, we also transfer its learned video representation to other action recognition benchmarks including HMDB51 [22] and UCF101 [37], where performance improvement is also achieved.

The main contribution of this paper is three-fold: (1) A SMART block is designed to simultaneously capture appearance and relation in a separate and explicit way. (2) An ARTNet is proposed by stacking multiple SMART blocks

to model appearance and relation information from different scales, which also allows for optimizing the parameters of SMART blocks in an end-to-end way. (3) ARTNets are empirically investigated on the large-scale Kinetics benchmark and state-of-the-art performance on this dataset is obtained under the setting of using only RGB input and training from scratch.

2. Related Work

Deep learning for video classification. Since the breakthrough of Convolutional Neural Networks (CNN) [25] in image classification [21], several works have tried to design effective architectures for video classification and action recognition [19, 35, 41, 49, 33, 5, 2, 34, 38]. Karpathy *et al.* [19] first tested deep networks with different temporal fusion strategies on a large-scale and noisily-labeled dataset (Sports-1M) and achieved lower performance than traditional features [44]. Simonyan *et al.* [35] designed a two-stream architecture containing spatial and temporal nets by explicitly exploiting pre-trained models and optical flow calculation. Tran *et al.* [41] investigated 3D CNNs [18] on realistic and large-scale video datasets and further studied deep ResNet with 3D convolution [42]. Carreira *et al.* proposed a new Two-Stream Inflated 3D CNNs based on 2D CNN inflation, which allows for pre-training with ImageNet models. Meanwhile, several papers [33, 5, 49] tried to model long-term temporal information for action understanding. Ng *et al.* [33] and Donahue *et al.* [5] utilized the LSTM [15] to capture the long range dynamics for action recognition. Wang *et al.* [49] designed a temporal segment network (TSN) to perform sparse sampling and temporal fusion, which aims to learn from the entire video.

Our work focuses on short-term temporal modeling and is most related with 3D CNNs. Our ARTNet mainly differs to 3D CNNs in that we design a new SMART block to model appearance and relation separately and explicitly with a two-branch architecture, while 3D CNNs employ the 3D convolutions to capture appearance and relation jointly and implicitly.

Models based on multiplicative interactions. Modeling or learning correspondence is an important task in computer vision. Typically, these methods are fundamentally based on the multiplicative interactions between pixels or between filter responses [30]. Mapping units [14]

first introduced the idea of multiplicative interactions to model relation between different views. Gated Boltzmann machines [31] were proposed to learn image transformation in unsupervised manner. Energy models [1], which may be viewed as a way to emulate multiplicative interactions by computing squares, were proposed to model motion information in videos. Independent Subspace Analysis (ISA) [16] was designed for invariant feature learning by computing sums over squared. ISA is similar to Energy model but its weights are trained from data. High-order neural networks [11] were proposed to learn invariance based on polynomial expansions of input. Recently, some action recognition methods are based on energy models [4, 45] and feature learning with Gate Boltzmann machines [40] and ISA [24]. Meanwhile, these multiplicative interactions or correlation models were integrated into the CNN architecture for optical flow estimation [6] and person re-identification [26].

Our proposed relation branch is inspired by these early works with multiplicative interactions and in particular it shares a similar *square-pooling* architecture with ISA. Our work differs from them in three important aspects: (1) The weights of relation branch are learned in a supervised manner with standard back propagation, while the previous work manually set model weights or learn them in an unsupervised manner. (2) The relation branch is integrated with an appearance branch to form the SMART block to capture spatiotemporal information, while previous works only has a module focusing on modeling relation. (3) We construct ARTNets by stacking multiple SMART blocks to learn hierarchical spatiotemporal features, while previous work usually has a single layer based on multiplicative interactions.

3. Spatiotemporal Feature Learning

In this section we describe our method for spatiotemporal feature learning. First, we discuss the role of multiplicative interaction in modeling relation across multiple frames. Next, we introduce the design of a SMART block. Finally, we propose the ART-Net by stacking multiple SMART blocks in the architecture of C3D-ResNet18.

3.1. Multiplicative interactions

Suppose we have two patches $\mathbf{x}$ and $\mathbf{y}$ from consecutive frames, we aim to learn the transformation (relation) $\mathbf{z}$ between them. A natural solution to this problem is to perform standard feature learning on the concatenation of these two patches, just like a 3D convolution [18, 41], as follows:

$$z_k = \sum_i w_{ik}^x x_i + \sum_j w_{jk}^y y_j, \qquad (1)$$

where the transformation code z_k is defined as a linear combination of $[\mathbf{x}, \mathbf{y}]$ by the parameters $\mathbf{w} = [\mathbf{w}_k^x, \mathbf{w}_k^y]$. However, in this case, the activation would be dependent on the

appearance (content) of patches not just the transformation between them [30]. In this sense, if both patches change but not transformation between them, the activation value would also change. Thus, this solution couples the information of appearance and relation together, adding the modeling difficulty and increasing the over-fitting risk.

Assuming the independence between appearance and relation, it is reasonable to decouple these two kinds of information when designing learning modules. It is easy to propose an appearance-independent relation detector by using multiplicative interactions between patches $\mathbf{x}$ and $\mathbf{y}$ [30]. Specifically, the transformation code z_k could be defined as follows:

$$z_k = \sum_{ij} w_{ijk} x_i y_j, \qquad (2)$$

where the transformation code z_k is defined as a linear combination of quadratic form from $\mathbf{x}$ and $\mathbf{y}$ by the weight tensor $w_{\cdot\cdot k}$. Essentially, this transformation code z_k pools over the outer product of $\mathbf{x}$ and $\mathbf{y}$, each element of which represents the evidence for a specific type of transformation. Thus, the activation value of z_k is less likely dependent on the patch appearance than the transformation between them.

Factorization and energy models. The major obstacle to directly deploy Equation (2) is that the number of parameters is roughly cubic in the number of pixels. Factorizing the parameter tensor W into three matrices would be an efficient way to reduce model parameters [30], namely: $w_{ijk} = \sum_{f=1}^F w_{if}^x w_{jf}^y w_{kf}^z$. Thus, the transformation code z_k between patch $\mathbf{x}$ and patch $\mathbf{y}$ in Equation (2) would be rewritten as follows:

$$
\begin{aligned}
z_k &= \sum_f w_{kf}^z \sum_i w_{if}^x x_i \sum_j w_{jf}^y y_j \\
&= \sum_f w_{kf}^z (\mathbf{w}_f^{x\mathrm{T}} \mathbf{x})(\mathbf{w}_f^{y\mathrm{T}} \mathbf{y}).
\end{aligned}
\qquad (3)
$$

This factorization formulation is closely related to energy model [1] and could be implemented with it. Specifically, a hidden unit z_k in the energy model is calculated as follows:

$$
\begin{aligned}
z_k &= \sum_f w_{kf}^z (\mathbf{w}_f^{x\mathrm{T}} \mathbf{x} + \mathbf{w}_f^{y\mathrm{T}} \mathbf{y})^2 \\
&= \sum_f w_{kf}^z [2(\mathbf{w}_f^{x\mathrm{T}} \mathbf{x})(\mathbf{w}_f^{y\mathrm{T}} \mathbf{y}) + (\mathbf{w}_f^{x\mathrm{T}} \mathbf{x})^2 + (\mathbf{w}_f^{y\mathrm{T}} \mathbf{y})^2],
\end{aligned}
$$

where hidden units are the same with Equation (3) except the quadratic terms $(\mathbf{w}_f^{x\mathrm{T}} \mathbf{x})^2$ and $(\mathbf{w}_f^{y\mathrm{T}} \mathbf{y})^2$, which do not have a significant effect on the hidden unit [29]. This energy model could be efficiently implemented with standard operations in 3D CNNs and easily stacked layer-by-layer as introduced in the next subsections.

3.2. SMART blocks

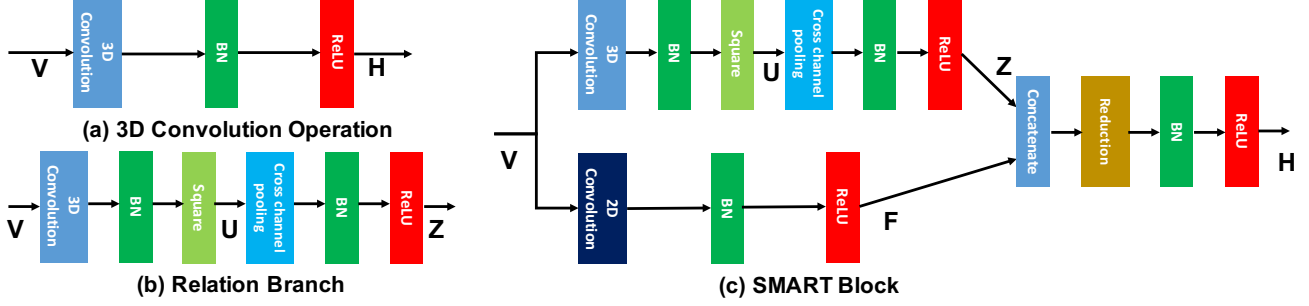As shown in Figure 2, a SMART block is a basic computational unit operating on an input volume $\mathbf{V} \in$

Figure 2. **Building blocks**: (a) the 3D convolution operation learns spatiotemporal features jointly and implicitly. (b) we first propose a *square-pooling* architecture to learn appearance-independent relation between frames. (c) we further construct a SMART block to learn spatiotemporal features separately and explicitly, where appearance branch uses a 2D convolution to capture static structure and relation branch employs a squaring-pooling architecture to model temporal relation.

$\mathbb{R}^{W \times H \times T \times C}$ and producing an output volume $\mathbf{H} \in \mathbb{R}^{W' \times H' \times T' \times C'}$. The motivation of the SMART block is to simultaneously model appearance and relation in a separate and explicit manner. Specifically, it learns spatiotemporal features from volume input with a two-branch architecture: (1) appearance branch for spatial feature learning, and (2) relation branch for temporal feature learning.

Appearance branch operates on individual frames and aims at capturing useful static information for action recognition. Static cues are sometimes important for action recognition as some action classes are strongly associated with certain object or scene categories. Specifically, we employ a *2D convolution* to operate on the video volume $\mathbf{V}$ to capture the spatial structure in each frame. The output of 2D convolution is a volume $\mathbf{F} \in \mathbb{R}^{W_s \times H_s \times T_s \times C_s}$. The response values $\mathbf{F}$ of 2D convolution usually goes into another Batch Normalization (BN) [17] layer and Rectified Linear Unit (ReLU) [32] for non-linearity.

Relation branch operates on stacked consecutive frames and aims to capture the relation among these frames for action recognition. This relation (transformation) information is crucial for action understanding as it contains motion cues. According to the discussion on multiplication interactions in the previous subsection, we design a *square-pooling* architecture to model temporal relation on this volume input. Specifically, we first apply a **3D convolution** to this volume input $\mathbf{V}$, which further goes through a **square function** to obtain hidden units $\mathbf{U} \in \mathbb{R}^{W_t \times H_t \times T_t \times C_t}$. Then, we apply a **cross-channel pooling** to aggregate multiple hidden units in $\mathbf{U}$ into the transformation codes $\mathbf{Z} \in \mathbb{R}^{W_t \times H_t \times T_t \times C'_t}$. This cross-channel is implemented with a $1 \times 1 \times 1$ convolution. In practice, the transformation code $\mathbf{Z}$ would also go through a BN layer and ReLU non-linearity to be consistent with the output of appearance branch. Meanwhile, we also add a BN layer between the 3D convolution and the square non-linearity to improve its stability.

A SMART block combine the output of the appearance and relation branches with a concatenation and reduction operation. Intuitively, the spatial and temporal features are complementary for action recognition and this fusion step aims to compress them into a more compact representation. In particular, we employ a $1 \times 1 \times 1$ convolution on the concatenation volume $[\mathbf{F}, \mathbf{Z}] \in \mathbb{R}^{W' \times H' \times T' \times (C_s + C'_t)}$ to obtain the compressed feature volumes $\mathbf{H} \in \mathbb{R}^{W' \times H' \times T' \times C_f}$. As a common practice, this compressed feature volume $\mathbf{H}$ further goes through a BN layer and ReLU activation function.

Implementation details. For the design simplicity of SMART block, some default setting is fixed as follows. First, the spatial and temporal dimension of output of two branches are ensured to be the same for concatenation operation, *i.e.*, $W_s = W_t = W'$, $H_s = H_t = H'$, and $T_s = T_t = T'$. In this sense, we let stride of 2D convolution in appearance branch and 3D convolution in relation branch be the same. Meanwhile, the spatial dimension of convolution kernels from two branches are the same as well. Second, the number of 2D convolution kernels in appearance branch is the same with that of 3D convolution kernels in relation branch, *i.e.*, $C_s = C_t$. In cross-channel pooling layer, each transformation code z_k in relation branch is locally connected a group of hidden units instead of using full connectivity. The number of transformation code is set to be half of that of hidden unit u, *i.e.*, $C_t = 2C'_t$, and thereby the group size is set to be 2. The weights in cross-channel pooling are fixed as 0.5. Finally, for the output of SMART block, we set its output number to be equal to that of appearance branch, *i.e.*, $C_f = C_s$. Therefore, the design parameter of a SMART block is the same with that a normal 3D convolution, including kernel size $k \times k \times t$, convolutional stride s_s and s_t, the output number c.

3.3. Exemplars: ARTNet-ResNet18

After introducing the SMART block, we are ready to plug it into the existing network architecture to build the appearance-and-relation network (ARTNet). The flexibility

layer name	output size	C3D-ResNet18	ARTNet-ResNet18 (s)	ARTNet-ResNet18 (d)
conv1	$56 \times 56 \times 8$	3D conv $7 \times 7 \times 3$, stride $2 \times 2 \times 2$	SMART $7 \times 7 \times 3$, stride $2 \times 2 \times 2$	
conv2_x	$56 \times 56 \times 8$	$\begin{bmatrix} \text{3D conv} & 3 \times 3 \times 3 & 64 \\ \text{3D conv} & 3 \times 3 \times 3 & 64 \end{bmatrix} \times 2$	$\begin{bmatrix} \text{3D conv} & 3 \times 3 \times 3 & 64 \\ \text{3D conv} & 3 \times 3 \times 3 & 64 \end{bmatrix} \times 2$	$\begin{bmatrix} \text{3D conv} & 3 \times 3 \times 3 & 64 \\ \text{SMART} & 3 \times 3 \times 3 & 64 \end{bmatrix} \times 2$
conv3_x	$28 \times 28 \times 4$	$\begin{bmatrix} \text{3D conv} & 3 \times 3 \times 3 & 128 \\ \text{3D conv} & 3 \times 3 \times 3 & 128 \end{bmatrix} \times 2$	$\begin{bmatrix} \text{3D conv} & 3 \times 3 \times 3 & 128 \\ \text{3D conv} & 3 \times 3 \times 3 & 128 \end{bmatrix} \times 2$	$\begin{bmatrix} \text{3D conv} & 3 \times 3 \times 3 & 128 \\ \text{SMART} & 3 \times 3 \times 3 & 128 \end{bmatrix} \times 2$
conv4_x	$14 \times 14 \times 2$	$\begin{bmatrix} \text{3D conv} & 3 \times 3 \times 3 & 256 \\ \text{3D conv} & 3 \times 3 \times 3 & 256 \end{bmatrix} \times 2$	$\begin{bmatrix} \text{3D conv} & 3 \times 3 \times 3 & 256 \\ \text{3D conv} & 3 \times 3 \times 3 & 256 \end{bmatrix} \times 2$	$\begin{bmatrix} \text{3D conv} & 3 \times 3 \times 3 & 256 \\ \text{SMART} & 3 \times 3 \times 3 & 256 \end{bmatrix} \times 2$
conv5_x	$7 \times 7 \times 1$	$\begin{bmatrix} \text{3D conv} & 3 \times 3 \times 3 & 512 \\ \text{3D conv} & 3 \times 3 \times 3 & 512 \end{bmatrix} \times 2$	$\begin{bmatrix} \text{3D conv} & 3 \times 3 \times 3 & 512 \\ \text{3D conv} & 3 \times 3 \times 3 & 512 \end{bmatrix} \times 2$	$\begin{bmatrix} \text{3D conv} & 3 \times 3 \times 3 & 512 \\ \text{3D conv} & 3 \times 3 \times 3 & 512 \end{bmatrix} \times 2$
	$1 \times 1 \times 1$	average pool, dropout, 400-d fc, softmax		
params (M)	-	33.37	33.39	35.20
FLOPs (G)	-	19.58	19.97	23.70

Table 1. **Architectures for the Kinetics dataset**: we study three different networks for spatiotemporal feature learning by stacking two types of building blocks from Figure 2: (1) 3D convolutions and (2) SMART blocks. Building blocks are shown in brackets, with the numbers of stacked blocks. The input to these networks is volume of $112 \times 112 \times 16$ and downsample is performed conv3_1, conv4_1, and conv5_1 with a stride of $2 \times 2 \times 2$.

of the SMART block allows it to replace the role of a 3D convolution in learning spatiotemporal feature. In current implementation, we develop an ARTNet by integrating the SMART block into the C3D-ResNet18 architecture [42], and thereby the resulted architecture is coined as **ARTNet-ResNet18**.

We choose the C3D-ResNet18 to instantiate the ART-Net and the architecture details are shown Table 1. These networks take an $112 \times 112 \times 16$ input to keep a balance between model capacity and processing efficiency.To well evaluate the effectiveness of SMART block, we implement two kinds of ARTNet-ResNet18: (1) we only replace the first 3D convolution in C3D-ResNet18 with the SMART block while keep the remaining layers unchanged, denoted as **ARTNet-ResNet18 (s)**. (2) we stack multiple SMART blocks and totally replace seven 3D convolutions, denoted as **ARTNet-ResNet18 (d)**. Stacking multiple SMART blocks allows us to capture appearance and relation information from different scales and further enhance the modeling capacity of ARTNet-ResNet18 (s).

Implementation details. We test these networks on the recently introduced Kinetics dataset [20]. *All these models are trained on the train set of Kinetics dataset from scratch.* We train the C3D-ResNet18 and ARTNet-ResNet18 by following the common practice in [42, 20]. The network parameters are initialized randomly. We use the mini-batch stochastic gradient descent algorithm to learn network parameters, where the batch size is set to 256 and momentum is set to 0.9. The frames are resized to 128×170 and then a volume of $112 \times 112 \times 16$ is randomly trimmed and cropped from each training video. This volume also undergoes a random horizontal flip, with the per-pixel mean subtracted.

The learning rate is initialized as 0.1 and divided by a factor of 10 when validation loss saturates. The total number of iteration is $250,000$ on the Kinetics dataset. To reduce the risk of over-fitting, we add a dropout layer before the final classification layer, where the dropout ratio is set to 0.2.

For testing network, we follow the common evaluation scheme [42, 35], where we sample 250 volumes of $112 \times 112 \times 16$ from the whole video. Specifically, we first uniformly trim 25 clips of $128 \times 170 \times 16$ and then generate 10 crops of $112 \times 112 \times 16$ from each clip (4 corners, 1 center, and their horizontal flipping). The final prediction result is obtained by taking an average over these 250 volumes.

4. Experiments

In this section we describe the experimental results of our method. First, we introduce the action recognition datasets and the evaluation settings. Then, we study different aspects of our proposed ARTNets on the Kinetics dataset and compare with the state-of-the-art methods. Finally, we transfer the learned spatiotemporal representations in ARTNets to the datasets of UCF101 and HMDB51.

4.1. Datasets

We evaluate the performance of ARTNets on three action recognition benchmarks: (1) Kinetics [20], (2) UCF101 [37], and (3) HMDB51 [22]. The Kinetics dataset is the largest well-labeled action recognition dataset. Its current version contains 400 action classes and each category has at least 400 videos. In total, there are around $240,000$ training videos, $20,000$ validation videos, and $40,000$ testing videos. The evaluation metric on the Kinetics dataset is the average of top-1 and top-5 error. As Kinet-

Method	Top-1	Top-5	Avg
C2D-ResNet18	61.2%	82.6%	71.9%
C3D-ResNet18	65.6%	85.7%	75.7%
C3D-ResNet34	67.1%	86.9%	77.0%
Relation-ResNet18 (s)	67.5%	87.0%	77.2%
Relation-ResNet18 (d)	67.1%	86.7%	76.9%
ARTNet-ResNet18 (s)	67.7%	87.1%	77.4%
ARTNet-ResNet18 (d)	**69.2%**	**88.3%**	**78.7%**

Table 2. Comparison of ARTNet and C3D on the validation set of Kinetics dataset. We investigate the performance of basic blocks, including: 2D convolution, 3D convolution, relation branch, and SMART. We also study the effect of the stacking depth of the ART-Net. The performance is measured by **Top-1 and Top-5 accuracy**.

Method	Modality	Top-1	Top-5	Avg
C3D-ResNet18	RGB	65.6%	85.7%	75.7%
C3D-ResNet18	Flow	57.5%	80.6%	69.0%
C3D-ResNet18	Fusion	68.7%	87.8%	78.2%
ARTNet-ResNet18	RGB	69.2%	88.3%	78.7%
ARTNet-ResNet18	Flow	59.8%	82.3%	71.0%
ARTNet-ResNet18	Fusion	**71.3%**	**89.5%**	**80.4%**

Table 3. Comparison of ARTNet and C3D with two stream input, i.e., RGB and Optical Flow. The results are reported on the validation set of Kinetics with the measure of **Top-1 and Top-5 accuracy**.

ics is the largest available dataset, we mainly study different aspects of ARTNets on this dataset **with only RGB input under the setting of training from scratch**.

UCF101 and HMDB51 are another two popular action recognition datasets, whose sizes are relatively small and the performance on them is already very high. The UCF101 has 101 action classes and $13,320$ video clips. We follow the official evaluation scheme and report average accuracy over three training/testing splits. The HMDB51 dataset is a collection of realistic videos from various sources, including movies and web videos. This dataset has $6,766$ videos from 51 action categories. Our experiment follows the original evaluation scheme using three training/testing splits and reports the average accuracy. As these two datasets are relatively small, we cannot train ARTNets from scratch and thereby transfer the video representations learned from the Kinetics dataset to them by fine tuning. The fine-tuning process follows the good practice presented in the temporal segment networks (TSN) [49]. The goal of experiment on UCF101 and HMDB51 is to test the generalization ability of learned spatiotemporal features by the ARTNet.

4.2. Results on the Kinetics dataset

Study on building block. We begin our experiment by studying the performance of four building blocks for spatiotemporal feature learning in videos. These building blocks include: (1) 2D convolution, (2) 3D convolution, (3) Relation branch, and (4) SMART block. We conduct experiments on Kinetics with the ResNet18 architecture as shown in Table 1. For C2D-ResNet18, we replace the 3D convolutions in C3D-ResNet18 with 2D convolutions, and for Relation-ResNet18, we replace the SMART blocks in ARTNet-ResNet18 with relation branch. The results are shown in Table 2. We see that 3D convolutions outperforms 2D convolutions for learning video representations (75.7% vs. 71.9%). Our newly designed relation branch and SMART block both outperform the original 3D convolutions (77.2% vs. 75.7% and 77.4% vs. 75.7%). SMART block obtains the best performance among these four building blocks, demonstrating the effectiveness of modeling ap-

pearance and relation separately and explicitly.

Study on block stacking. We also investigate the effectiveness of stacking multiple Relation branches and SMART blocks. As shown in Table 2, we observe that stacking multiple SMART blocks is able to further boost error rate from 77.4% to 78.7%. This improvement indicates the effectiveness of capturing spatiotemporal features in a hierarchical manner. However, stacking multiple relation branch causes a small performance drop, indicating the importance of modeling spatial structure in higher layers. Remarkably, as stacking SMART blocks would increase the network depth, we also compare the performance with C3D-ResNet34 in Table 2, where ARTNet-ResNet18 even outperforms the deeper C3D-ResNet34 (78.7% vs. 77.0%). This result demonstrates that the performance improvement is brought by the effectiveness of SMART block instead of the increased network depth. In the remaining experiments, we will use the ARTNet-ResNet18 (d) by default.

Study on two-stream inputs. Two stream CNN is a strong baseline for action recognition and its input has two modalities, *i.e.*, RGB and Optical Flow. To further illustrate the effectiveness of SMART block over 3D convolution, we perform experiments with two-stream inputs for both ARTNet-ResNet18 and C3D-ResNet18. The numerical results are reported in Table 3. *First*, we find that two-stream inputs are able to improve the performance of C3D-ResNet18 from 75.7% to 78.2%. This improvement indicates that although 3D convolution aims to directly learn spatiotemporal features from RGB, flow stream is still able to provide complementary information. *Second*, comparing two-stream C3D-ResNet18 with RGB-stream ARTNet-ResNet18, we notice that our proposed ARTNet is still able to yield a slightly better performance (78.7% vs. 78.2%). This better result demonstrates the superiority of SMART block over two stream inputs. *Finally*, we also experiment ARTNet-ResNet18 with two-stream inputs. In flow stream, similar improvement over C3D-ResNet18 is also observed with ARTNet-ResNet18. The two-stream ARTNet-ResNet18 can boost performance to 80.4%. But it is worth noting that the high computational cost of optical flow makes it extremely difficult to apply at large-scale datasets and deploy in real-world applications. Therefore,

Method	TSN	Modality	Top-1	Top-5	Avg
ARTNet-ResNet18	No	RGB	69.2%	88.3%	78.7%
ARTNet-ResNet18	Yes	RGB	70.7%	89.3%	80.0%
ARTNet-ResNet18	No	Flow	59.8%	82.3%	71.0%
ARTNet-ResNet18	Yes	Flow	60.6%	83.1%	71.9%
ARTNet-ResNet18	No	Fusion	71.3%	89.5%	80.4%
ARTNet-ResNet18	Yes	Fusion	**72.4%**	**90.4%**	**81.4%**

Table 4. Comparison of ARTNet between without TSN and with TSN. ARTNet focuses on short-term spatiotemporal feature learning and is easily combined with the existing long-term modeling architectures. The results are reported on the validation set of Kinetics and measured by **Top-1 and Top-5 accuracy**.

in the remaining experiment, we mainly compare the performance of only using RGB input.

Study on long-term modeling. The proposed SMART block and ARTNet focus on short-term spatiotemporal feature learning and is complementary to the exiting long-term modeling architectures [43, 49, 33]. Temporal segment network (TSN) is a general and flexible video-level framework for learning action models in videos [49]. The simplicity nature of TSN allows us to replace the original 2D CNNs with our proposed ARTNet-ResNet18. Specifically, to keep a balance between modeling capacity and training time, we set the segment number as 2. The experimental results are summarized in Table 4. We see that TSN modeling is helpful to improve the performance of ARTNet-ResNet18. For example, ARTNet-ResNet18 with TSN training can yield the performance of 80.0% with RGB input and 81.4% with two-stream inputs, which is better than the original performance by 1.3% and 1.0%. This improvement demonstrates the complementarity of ARTNet to the TSN framework. In principle, ARTNet is a general short-term video model, that could be explored in any long-term learning framework, such as LSTM [33, 5] and attention modeling [48].

Comparison to the state of the art. We compare the performance of ARTNet-ResNet18 with the state-of-the-art approaches on the validation set and test set of Kinetics. The results are summarized in Table 5. For fair comparison, we consider methods that only use RGB input and learned from scratch on the train set of Kinetics. We also list other important factors such as spatial resolution and backbone architectures.

We *first* compare with three baseline methods: (1) CNN+LSTM [33, 5], (2) Spatial Stream [35], and (3) C3D [41]. Our proposed ARTNets significantly outperform these baselines by around 10%. We *then* compare with deeper C3D architecture [42] such as C3D-ResNet18 and C3D-ResNet34. Our ARTNet is able to yield a better performance (around 3%) than these fairly-comparable models. *Finally*, we compare with the recent state-of-the-art methods, namely temporal segment network (TSN) [49] and Inflated 3D CNN (I3D) [2]. These two architectures employ a deeper backbone architecture (Inception [39])

and larger spatial resolution (224×224). Besides, I3D is also equipped with long-term modeling [43] by stacking 64 frames. Therefore, it is fair for us to use TSN to increase the temporal duration of ARTNet. Our ARTNet with TSN training obtains a slightly better performance than these two very competitive methods (80.0% vs. 77.8% on validation set, and 78.7% vs. 78.2% on test set).

It is worth noting that the current published state-of-the-art performance is 82.7%, that is obtained by two-stream I3D [2] with optical flow input and pre-training on ImageNet. Two-stream I3D is more computational expensive than ARTNet as it uses larger spatial resolution, deeper structure, and two-stream inputs. The winner solution [28] at ActivityNet challenge [10] gets a performance of 87.6% by using more modalities, multi-stage training, and model ensemble. These results are not directly comparable to ours.

4.3. Results on the UCF101 and HMDB51 datasets

In this subsection we study the generalization ability of learned spatiotemporal representations on the Kinetics dataset [20]. Specifically, we transfer the learned models to two popular action recognition benchmarks: UCF101 [37] and HMDB51 [22]. We consider fine tuning three models trained on the Kinetics dataset: C3D-ResNet18, ARTNet-ResNet18 without TSN, ARTNet-ResNet18 with TSN. The fine-tuning process is conducted with the TSN framework and follows the common practice proposed in the original TSN framework [49], where the segment number is set to 2.

The results are summarized in Table 6. *First*, we compare the performance of C3D-ResNet18 and ARTNet-ResNet18 pre-trained on the Kinetics dataset and see that our ARTNet outperform C3D by 3.7% on the UCF101 dataset and by 5.5% on the HMDB51 dataset. This superior performance demonstrates that the spatiotemporal representation learned in ARTNet is more effective than C3D for transfer learning. *Then*, we investigate the ARTNet-ResNet18 models learned under the TSN framework on the Kinetics dataset and these models can yield a slightly better performance (94.3% on UCF101 and 70.9% on HMDB51). This better transfer learning performance on UCF101 and HMDB51 agrees with the original performance improvement on the Kinetics dataset as shown Table 5, which indicates the importance of long-term modeling. *Finally*, we compare with other state-of-the-art methods that pre-train on different datasets. We see that the methods pre-trained on the Kinetics dataset significantly outperform those pre-trained on ImageNet [3] and Sports-1M [19], which might be explained by the better quality of Kinetics. Our ARTNet obtains a comparable performance to the best performer of RGB-3D that is trained at a larger spatial resolution and pre-trained on two datasets (ImageNet and Kinetics).

Method	Spatial resolution	Backbone architecture	Kinetics val set	Kinetics test set
ConvNet+LSTM [5, 33]	299 × 299	ResNet-50	-	68.0%
Two Stream Spatial Networks [35]	299 × 299	ResNet-50	-	66.6%
C3D [41]	112 × 112	VGGNet-11	-	67.8%
C3D [42]	112 × 112	ResNet-18	75.7%	74.4%
C3D [42]	112 × 112	ResNet-34	77.0%	75.3%
TSN Spatial Networks [49]	224 × 224	Inception V2	77.8%	-
RGB-I3D [2]	224 × 224	Inception V1	-	78.2%
ARTNet w/o TSN	112 × 112	ResNet-18	78.7%	77.3%
ARTNet with TSN	112 × 112	ResNet-18	**80.0%**	**78.7%**

Table 5. Comparison with state-of-the-art methods on the validation and test set of Kinetics. The performance is measured by the average of Top-1 and Top-5 accuracy. **For fair comparison, we consider methods that use only RGB input and train from scratch on Kinetics**. Our ARTNets are trained from the spatial resolution of 112 × 112 and can still yield better performance than those trained from the spatial resolution of 224 × 224 or 229 × 229.

Method	Pre-train dataset	Spatial resolution	Backbone architecture	UCF101	HMDB51
HOG [44]	None	240 × 320	None	72.4%	40.2%
ConvNet+LSTM [5]	ImageNet	224 × 224	AlexNet	68.2%	-
Two Stream Spatial Network [35]	ImageNet	224 × 224	VGG-M	73.0%	40.5%
Conv Pooling Spatial Network [8]	ImageNet	224 × 224	VGGNet-16	82.6%	-
Spatial Stream ResNet [7]	ImageNet	224 × 224	ResNet-50	82.3%	43.4%
Spatial TDD [46]	ImageNet	224 × 224	VGG-M	82.8%	50.0%
RGB-I3D [2]	ImageNet	224 × 224	Inception V1	84.5%	49.8%
TSN Spatial Network [49]	ImageNet	224 × 224	Inception V2	**86.4%**	**53.7%**
Slow Fusion [19]	Sports-1M	170 × 170	AlexNet	65.4%	-
C3D [41]	Sports-1M	112 × 112	VGGNet-11	82.3%	51.6%
LTC [43]	Sports-1M	71 × 71	VGGNet-11	82.4%	48.7%
C3D [42]	Sports-1M	112 × 112	ResNet-18	**85.8%**	**54.9%**
TSN Spatial Network [49]	ImageNet+Kinetics	224 × 224	Inception V2	91.1%	-
TSN Spatial Network [49]	ImageNet+Kinetics	229 × 229	Inception V3	93.2%	-
RGB-I3D [2]	ImageNet+Kinetics	224 × 224	Inception V1	**95.6%**	**74.8%**
C3D	Kinetics	112 × 112	ResNet-18	89.8%	62.1%
ARTNet w/o TSN	Kinetics	112 × 112	ResNet-18	93.5%	67.6%
ARTNet with TSN	Kinetics	112 × 112	ResNet-18	**94.3%**	**70.9%**

Table 6. Comparison with state-of-the-art methods on the UCF101 and HMDB51 datasets. The accuracy is reported as average over three splits. **For fair comparison, we consider methods that use only RGB input and pre-train on different datasets**. The performance is grouped according to its pre-training dataset. Our ARTNet obtains the best performance under the setting of pre-training only on the Kinetics dataset, and a comparable performance to the RGB-I3D pre-trained on the datasets of ImageNet+Kinetics.

5. Conclusion and Future Work

In this paper we have presented a new architecture, coined as *ARTNet*, for spatiotemporal feature learning in videos. The construction of ARTNet is based on a generic building block, termed as *SMART*, which aims to model appearance and relation separately and explicitly with a two-branch unit. As demonstrated on the Kinetics dataset, SMART block is able to yield better performance than the 3D convolution, and ARTNet with a single RGB input even outperforms the C3D with two-stream inputs. For representation transfer from Kinetics to datasets of UCF101 and HMDB51, ARTNet also achieves superior performance to the original C3D. This performance improvement may be ascribed to the fact that we separately model appearance and relation, by using the linear combination of filter responses in each frame and the multiplicative interactions between filter responses across frames, respectively.

For ARTNet, augmenting RGB input with optical flow also helps to improve performance. This improvement indicates optical flow modality is still able to provide complementary information. However, the high computational cost of optical flow prohibits its application in real-world systems. In the future we plan to further improve the ARTNet architecture to overcome the performance gap between single-stream and two-stream inputs. Meanwhile, we will try to instantiate the ARTNets with more deeper structures such as ResNet101 and train them on more larger spatial resolutions.

References

[1] E. H. Adelson and J. R. Bergen. Spatiotemporal energy models for the perception of motion. *J. Opt. Soc. Am. A*, 2(2):284–299, 1985.

[2] J. Carreira and A. Zisserman. Quo vadis, action recognition? A new model and the kinetics dataset. In *CVPR*, pages 6299–6308, 2017.

[3] J. Deng, W. Dong, R. Socher, L. Li, K. Li, and F. Li. Imagenet: A large-scale hierarchical image database. In *CVPR*, pages 248–255, 2009.

[4] K. G. Derpanis, M. Sizintsev, K. J. Cannons, and R. P. Wildes. Efficient action spotting based on a spacetime oriented structure representation. In *CVPR*, pages 1990–1997, 2010.

[5] J. Donahue, L. A. Hendricks, S. Guadarrama, M. Rohrbach, S. Venugopalan, T. Darrell, and K. Saenko. Long-term recurrent convolutional networks for visual recognition and description. In *CVPR*, pages 2625–2634, 2015.

[6] A. Dosovitskiy, P. Fischer, E. Ilg, P. Häusser, C. Hazirbas, V. Golkov, P. van der Smagt, D. Cremers, and T. Brox. Flownet: Learning optical flow with convolutional networks. In *ICCV*, pages 2758–2766, 2015.

[7] C. Feichtenhofer, A. Pinz, and R. P. Wildes. Spatiotemporal residual networks for video action recognition. In *NIPS*, pages 3468–3476, 2016.

[8] C. Feichtenhofer, A. Pinz, and A. Zisserman. Convolutional two-stream network fusion for video action recognition. In *CVPR*, pages 1933–1941, 2016.

[9] C. Gan, N. Wang, Y. Yang, D. Yeung, and A. G. Hauptmann. DevNet: A deep event network for multimedia event detection and evidence recounting. In *CVPR*, pages 2568–2577, 2015.

[10] B. Ghanem, J. C. Niebles, C. Snoek, F. C. Heilbron, H. Alwassel, R. Krishna, V. Escorcia, K. Hata, and S. Buch. Activitynet challenge 2017 summary. *CoRR*, abs/1710.08011, 2017.

[11] C. L. Giles and T. Maxwell. Learning, invariance, and generalization in high-order neural networks. *Applied optics*, 26(23):4972–4978, 1987.

[12] R. B. Girshick, J. Donahue, T. Darrell, and J. Malik. Rich feature hierarchies for accurate object detection and semantic segmentation. In *CVPR*, pages 580–587, 2014.

[13] K. He, X. Zhang, S. Ren, and J. Sun. Deep residual learning for image recognition. In *CVPR*, pages 770–778, 2016.

[14] G. E. Hinton. A parallel computation that assigns canonical object-based frames of reference. In *IJCAI*, pages 683–685, 1981.

[15] S. Hochreiter and J. Schmidhuber. Long short-term memory. *Neural Computation*, 9(8):1735–1780, 1997.

[16] A. Hyvärinen and P. O. Hoyer. Emergence of phase- and shift-invariant features by decomposition of natural images into independent feature subspaces. *Neural Computation*, 12(7):1705–1720, 2000.

[17] S. Ioffe and C. Szegedy. Batch normalization: Accelerating deep network training by reducing internal covariate shift. In *ICML*, pages 448–456, 2015.

[18] S. Ji, W. Xu, M. Yang, and K. Yu. 3D convolutional neural networks for human action recognition. *IEEE Trans. Pattern Anal. Mach. Intell.*, 35(1):221–231, 2013.

[19] A. Karpathy, G. Toderici, S. Shetty, T. Leung, R. Sukthankar, and F. Li. Large-scale video classification with convolutional neural networks. In *CVPR*, pages 1725–1732, 2014.

[20] W. Kay, J. Carreira, K. Simonyan, B. Zhang, C. Hillier, S. Vijayanarasimhan, F. Viola, T. Green, T. Back, P. Natsev, M. Suleyman, and A. Zisserman. The kinetics human action video dataset. *CoRR*, abs/1705.06950, 2017.

[21] A. Krizhevsky, I. Sutskever, and G. E. Hinton. ImageNet classification with deep convolutional neural networks. In *NIPS*, pages 1106–1114, 2012.

[22] H. Kuehne, H. Jhuang, E. Garrote, T. A. Poggio, and T. Serre. HMDB: A large video database for human motion recognition. In *ICCV*, pages 2556–2563, 2011.

[23] I. Laptev. On space-time interest points. *International Journal of Computer Vision*, 64(2-3):107–123, 2005.

[24] Q. V. Le, W. Y. Zou, S. Y. Yeung, and A. Y. Ng. Learning hierarchical invariant spatio-temporal features for action recognition with independent subspace analysis. In *CVPR*, pages 3361–3368, 2011.

[25] Y. LeCun, L. Bottou, Y. Bengio, and P. Haffner. Gradient-based learning applied to document recognition. *Proceedings of the IEEE*, 86(11):2278–2324, 1998.

[26] W. Li, R. Zhao, T. Xiao, and X. Wang. Deepreid: Deep filter pairing neural network for person re-identification. In *CVPR*, pages 152–159, 2014.

[27] J. Long, E. Shelhamer, and T. Darrell. Fully convolutional networks for semantic segmentation. In *CVPR*, pages 3431–3440, 2015.

[28] X. Long, C. Gan, G. de Melo, J. Wu, X. Liu, and S. Wen. Attention clusters: Purely attention based local feature integration for video classification. *CVPR*, 2018.

[29] R. Memisevic. On multi-view feature learning. In *ICML*, 2012.

[30] R. Memisevic. Learning to relate images. *IEEE Trans. Pattern Anal. Mach. Intell.*, 35(8):1829–1846, 2013.

[31] R. Memisevic and G. E. Hinton. Unsupervised learning of image transformations. In *CVPR*, 2007.

[32] V. Nair and G. E. Hinton. Rectified linear units improve restricted boltzmann machines. In *ICML*, pages 807–814, 2010.

[33] J. Y. Ng, M. J. Hausknecht, S. Vijayanarasimhan, O. Vinyals, R. Monga, and G. Toderici. Beyond short snippets: Deep networks for video classification. In *CVPR*, pages 4694–4702, 2015.

[34] Z. Qiu, T. Yao, and T. Mei. Learning spatio-temporal representation with pseudo-3d residual networks. In *ICCV*, pages 5534–5542, 2017.

[35] K. Simonyan and A. Zisserman. Two-stream convolutional networks for action recognition in videos. In *NIPS*, pages 568–576, 2014.

[36] K. Simonyan and A. Zisserman. Very deep convolutional networks for large-scale image recognition. *CoRR*, abs/1409.1556, 2014.

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

[38] L. Sun, K. Jia, D. Yeung, and B. E. Shi. Human action recognition using factorized spatio-temporal convolutional networks. In *ICCV*, pages 4597–4605, 2015.

[39] C. Szegedy, W. Liu, Y. Jia, P. Sermanet, S. E. Reed, D. Anguelov, D. Erhan, V. Vanhoucke, and A. Rabinovich. Going deeper with convolutions. In *CVPR*, pages 1–9.

[40] G. W. Taylor, R. Fergus, Y. LeCun, and C. Bregler. Convolutional learning of spatio-temporal features. In *ECCV*, pages 140–153, 2010.

[41] D. Tran, L. D. Bourdev, R. Fergus, L. Torresani, and M. Paluri. Learning spatiotemporal features with 3D convolutional networks. In *ICCV*, pages 4489–4497, 2015.

[42] D. Tran, J. Ray, Z. Shou, S. Chang, and M. Paluri. ConvNet architecture search for spatiotemporal feature learning. *CoRR*, abs/1708.05038, 2017.

[43] G. Varol, I. Laptev, and C. Schmid. Long-term temporal convolutions for action recognition. *CoRR*, abs/1604.04494, 2016.

[44] H. Wang and C. Schmid. Action recognition with improved trajectories. In *ICCV*, pages 3551–3558, 2013.

[45] L. Wang, Y. Qiao, and X. Tang. Motionlets: Mid-level 3D parts for human motion recognition. In *CVPR*, pages 2674–2681, 2013.

[46] L. Wang, Y. Qiao, and X. Tang. Action recognition with trajectory-pooled deep-convolutional descriptors. In *CVPR*, pages 4305–4314, 2015.

[47] L. Wang, Y. Qiao, and X. Tang. MoFAP: A multi-level representation for action recognition. *International Journal of Computer Vision*, 119(3):254–271, 2016.

[48] L. Wang, Y. Xiong, D. Lin, and L. Van Gool. UntrimmedNets for weakly supervised action recognition and detection. In *CVPR*, pages 4325–4334, 2017.

[49] 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*, pages 20–36, 2016.

[50] X. Wang, A. Farhadi, and A. Gupta. Actions ˜ transformations. In *CVPR*, pages 2658–2667, 2016.

[51] B. Zhang, L. Wang, Z. Wang, Y. Qiao, and H. Wang. Real-time action recognition with enhanced motion vector CNNs. In *CVPR*, pages 2718–2726, 2016.