Learning Self-Similarity in Space and Time as Generalized Motion for Video Action Recognition

Heeseung Kwon* Manjin Kim* Suha Kwak Minsu Cho
Pohang University of Science and Technology (POSTECH), South Korea
http://cvlab.postech.ac.kr/research/SELFY/

Abstract

Spatio-temporal convolution often fails to learn motion dynamics in videos and thus an effective motion representation is required for video understanding in the wild. In this paper, we propose a rich and robust motion representation based on spatio-temporal self-similarity *(STSS). Given a sequence of frames, STSS represents each local region as similarities to its neighbors in space and time. By converting appearance features into relational values, it enables the learner to better recognize structural patterns in space and time. We leverage the whole volume of STSS and let our model learn to extract an effective motion representation from it. The proposed neural block, dubbed* SELFY*, can be easily inserted into neural architectures and trained end-to-end without additional supervision. With a sufficient volume of the neighborhood in space and time, it effectively captures long-term interaction and fast motion in the video, leading to robust action recognition. Our experimental analysis demonstrates its superiority over previous methods for motion modeling as well as its complementarity to spatio-temporal features from direct convolution. On the standard action recognition benchmarks, Something-Something-V1 & V2, Diving-48, and FineGym, the proposed method achieves the state-of-the-art results.*

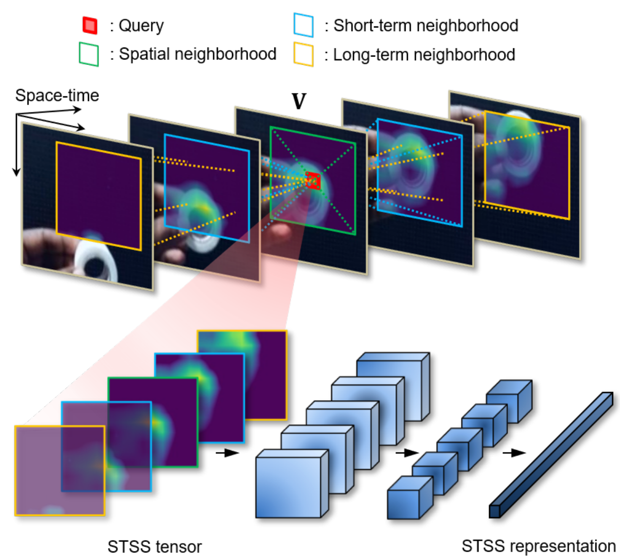


Figure 1: **Spatio-temporal self-similarity (STSS) representation learning.** STSS describes each position (query) by its similarities (STSS tensor) with its neighbors in space and time (neighborhood). It allows to take a generalized, far-sighted view on motion, *i.e.*, both short-term and long-term, both forward and backward, as well as spatial self-motion. Our method learns to extract a rich motion representation from STSS without additional supervision.

1. Introduction

Learning spatio-temporal dynamics is the key to video understanding. While extending standard convolution in space and time has been actively investigated for the purpose in recent years [1, 44, 46], the empirical results so far indicate that spatio-temporal convolution alone is not sufficient for grasping the whole picture; it often learns irrelevant context bias rather than motion information [32, 33] and thus the additional use of optical flow turns out to boost the performance in most cases [1,29]. Motivated by this, recent action recognition methods learn to extract explicit motion, *i.e.*, flow or correspondence, between feature maps of adjacent frames to improve the performance [22, 27]. But, is it essential to extract such an explicit form of flows or correspondences? How can we learn a richer and more robust form of motion information for videos in the wild?

In this paper, we propose to learn *spatio-temporal self-similarity* (STSS) representation for video understanding. Self-similarity is a relational descriptor for an image that effectively captures intra-structures by representing each local region as similarities to its spatial neighbors [37]. As illustrated in Fig. 1, given a sequence of frames, *i.e.*, a video, it

*Equal contribution.

extends along time and thus represents each local region as similarities to its neighbors in space and time. By converting appearance features into relational values, STSS enables a learner to better recognize structural patterns in space and time. For neighbors at the same frame it computes a spatial self-similarity map, while for neighbors at a different frame it extracts a motion likelihood map. Note that if we fix our attention to the similarity map to the very next frame within STSS and attempt to extract a single displacement vector to the most likely position at the frame, the problem reduces to optical flow, which is a limited type of motion information. In contrast, we leverage the whole volume of STSS and let our model learn to extract a generalized motion representation from it in an end-to-end manner. With a sufficient volume of the neighborhood in space and time, it effectively captures long-term interaction and fast motion in the video, leading to robust action recognition.

We introduce a neural block for STSS representation, dubbed *SELFY*, that can be easily inserted into neural architectures and learned end-to-end without additional supervision. Our experimental analysis demonstrates its superiority over previous methods for motion modeling as well as its complementarity to spatio-temporal features from direct convolutions. On the standard benchmarks for action recognition, Something-Something V1&V2 [10], Diving-48 [28], and FineGym [36], the proposed method achieves the state-of-the-art results.

2. Related Work

Video action recognition. Video action recognition aims to categorize videos into pre-defined action classes and one of the main issues in action recognition is to capture temporal dynamics in videos. For modern neural networks, previous methods attempt to learn temporal dynamics in different ways: two-stream networks with external optical flows [38, 49], recurrent networks [3], temporal pooling methods [9, 23], and 3D CNNs [1, 44]. Recent methods have introduced the advanced 3D CNNs [5, 7, 29, 45, 46] and showed the effectiveness of capturing spatio-temporal features, so that 3D CNNs now become a *de facto* approach to learn temporal dynamics in the video. However, spatio-temporal convolution is vulnerable unless relevant features are well aligned across frames within the fixed-sized kernel. To address this issue, a few methods adaptively translate the kernel offsets with deformable convolutions [25, 55], while several methods [8, 26] modulate the other hyper-parameters, *e.g.*, higher frame rate or larger spatial receptive fields. Unlike these methods, we address the problem of the spatio-temporal convolution by a sufficient volume of STSS, capturing far-sighted spatio-temporal relations.

Learning motion features. Since using the external optical flow benefits 3D CNNs to improve the action recognition accuracy [1, 46, 57], several methods propose to learn frame-by-frame motion features from RGB sequences inside neural architectures. Some methods [6, 34] internalize TV-L1 [54] optical flows into the CNN. Frame-wise feature differences [14, 24, 27, 42] are also utilized as the motion features. Recent correlation-based methods [22, 48] adopt a correlation operator [4, 41, 53] to learn motion features between adjacent frames. However, these methods compute frame-by-frame motion features between two adjacent frames and then rely on stacked spatio-temporal convolutions for capturing long-range motion dynamics. In contrast, we propose to learn STSS features, as generalized motion features, that enable to capture both short-term and long-term interactions in the video.

Self-similarity. Self-similarity describes a relational structure of individual image features by computing similarities between them [37]. Several methods [15, 16, 37, 43] use the self-similarity as a shallow relational descriptor, which is robust to photometric variations, in fields of template matching [37], capturing view-invariant geometric patterns [15, 16], or finding semantic correspondences [17, 21, 43]. In video understanding, there are a few approaches [30, 50] that use the self-similarity of a video as a form of STSS. These methods, however, use STSS for a subsequent feature aggregation step rather than learn representation from it; non-local operation [50] uses STSS as attention weights in aggregating features [13, 35, 39, 47] and CPNet [30] uses STSS in selecting pairs of appearance features. All these methods lose rich motion information of STSS during aggregation, being not suitable for capturing motion content of videos. In contrast, we advocate using STSS directly for motion representation learning. Our method leverages the full STSS as generalized motion information and learns an effective representation for action recognition within a video-processing architecture. To the best of our knowledge, our work is the first in learning STSS representation using modern neural networks.

The contribution of our paper can be summarized as follows. First, we revisit the notion of self-similarity and propose to learn a generalized, far-sighted motion representation from STSS. Second, we implement STSS representation learning as a neural block, dubbed *SELFY*, that can be integrated into existing neural architectures. Third, we provide comprehensive evaluations on SELFY, achieving the state-of-the-art on benchmarks: Something-Something V1&V2 [10], Diving-48 [28], and FineGym [36].

3. Our approach

In this section, we first revisit the notions of self-similarity and discuss its relation to motion. We then introduce our method for learning effective spatio-temporal self-similarity representation, which can be easily integrated into video-processing architectures and learned end-to-end.

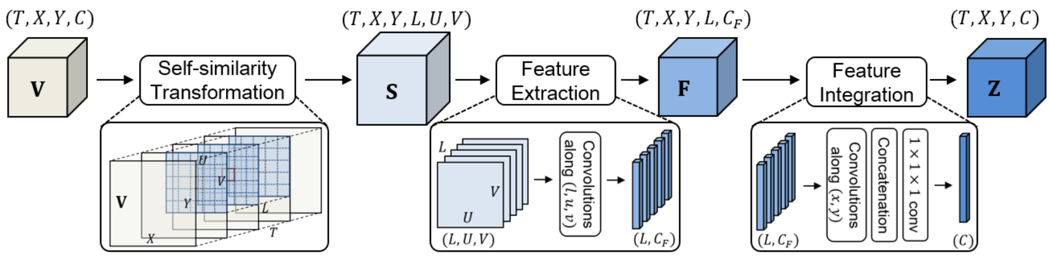


Figure 2: **Overview of our self-similarity representation block (SELFY)**. SELFY block takes as input a video feature tensor $\mathbf{V}$, transforms it to a STSS tensor $\mathbf{S}$, and extracts a feature tensor $\mathbf{F}$ from $\mathbf{S}$. It then produces the final STSS representation $\mathbf{Z}$ via the feature integration, which is the same size as the input $\mathbf{V}$. The resultant representation $\mathbf{Z}$ is fused into the input feature $\mathbf{V}$ by element-wise addition, thus making SELFY act as a residual block. See text for details.

3.1. Self-Similarity Transformation

Self-similarity is a relational descriptor that suppresses variations in appearance and reveals structural patterns [37].

Given an image feature map $\mathbf{I} \in \mathbb{R}^{X \times Y \times C}$, self-similarity transformation of $\mathbf{I}$ results in a 4D tensor $\mathbf{S} \in \mathbb{R}^{X \times Y \times U \times V}$, whose elements are defined as

$$\mathbf{S}_{x,y,u,v} = \mathrm{sim}(\mathbf{I}_{x,y}, \mathbf{I}_{x+u,y+v}),$$

where $\mathrm{sim}(\cdot,\cdot)$ is a similarity function, *e.g.*, cosine similarity. Here, (x, y) is a query coordinate while (u, v) is a spatial offset from it. To impose a locality, the offset is restricted to its neighborhood: $(u, v) \in [-d_{\mathrm{U}}, d_{\mathrm{U}}] \times [-d_{\mathrm{V}}, d_{\mathrm{V}}]$, so that $U = 2d_{\mathrm{U}} + 1$ and $V = 2d_{\mathrm{V}} + 1$, respectively. By converting C-dimensional appearance feature $\mathbf{I}_{x,y}$ into UV-dimensional relational feature $\mathbf{S}_{x,y}$, it suppresses variations in appearance and reveals spatial structures in the image. Note that the self-similarity transformation closely relates to conventional cross-similarity (or correlation) across two different feature maps ($\mathbf{I}, \mathbf{I}' \in \mathbb{R}^{X \times Y \times C}$), which can be defined as

$$\mathbf{S}_{x,y,u,v} = \mathrm{sim}(\mathbf{I}_{x,y}, \mathbf{I}'_{x+u,y+v}).$$

Given a moving object of two images, the cross-similarity transformation effectively captures motion information and thus is commonly used in optical flow and correspondence estimation [4, 41, 53].

For a sequence of frames, *i.e.*, a video, one can naturally extend the spatial self-similarity along the temporal axis. Let $\mathbf{V} \in \mathbb{R}^{T \times X \times Y \times C}$ denote a feature map of the video with T frames. *Spatio-temporal self-similarity* (STSS) transformation of $\mathbf{V}$ results in a 6D tensor $\mathbf{S} \in \mathbb{R}^{T \times X \times Y \times L \times U \times V}$, whose elements are defined as

$$\mathbf{S}_{t,x,y,l,u,v} = \mathrm{sim}(\mathbf{V}_{t,x,y}, \mathbf{V}_{t+l,x+u,y+v}), \tag{1}$$

where (t, x, y) is a query coordinate and (l, u, v) is a spatio-temporal offset from the query. In addition to the locality of spatial offsets above, the temporal offset l is also restricted to its temporal neighborhood: $l \in [-d_{\mathrm{L}}, d_{\mathrm{L}}]$, so that $L = 2d_{\mathrm{L}} + 1$.

What types of information does STSS describe? Interestingly, for each time t, the STSS tensor $\mathbf{S}$ can be decomposed along temporal offset l into a single spatial self-similarity tensor (when $l = 0$) and $2d_{\mathrm{L}}$ spatial cross-similarity tensors (when $l \neq 0$); the partial tensors with a small offset (*e.g.*, $l = -1$ or $+1$) collect motion information from adjacent frames and those with larger offsets capture it from further frames both forward and backward in time. Unlike previous approaches to learn motion [4, 22, 48], which rely on cross-similarity between adjacent frames, STSS allows to take a generalized, far-sighted view on motion, *i.e.*, both short-term and long-term, both forward and backward, as well as spatial self-motion.

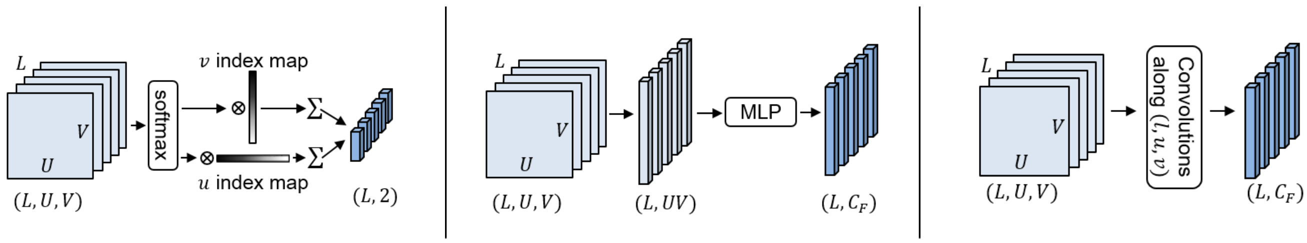
3.2. Spatio-temporal self-similarity representation learning

By leveraging the rich information in STSS, we propose to learn a generalized motion representation for video understanding. To achieve this goal without additional supervision, we design a neural block, dubbed SELFY, which can be inserted into video-processing architectures and learned end-to-end. Figure 2 illustrates the overall structure. It consists of three steps: *self-similarity transformation*, *feature extraction*, and *feature integration*.

Given the input video feature tensor $\mathbf{V}$, the self-similarity transformation step converts it to the STSS tensor $\mathbf{S}$ as in Eq. 1. In the following, we describe feature extraction and integration steps.

3.2.1 Feature extraction

From the STSS tensor $\mathbf{S} \in \mathbb{R}^{T \times X \times Y \times L \times U \times V}$, we extract a C_F-dimensional feature for each spatio-temporal position (t, x, y) and temporal offset l so that the resultant tensor is $\mathbf{F} \in \mathbb{R}^{T \times X \times Y \times L \times C_F}$, which is equivariant to translation in space, time, and temporal offset. The dimension of L is preserved to extract motion information across different



(a) soft-argmax (b) MLP (c) convolution

Figure 3: **Feature extraction from STSS.** For a spatio-temporal position (t, x, y), each method transforms (L, U, V) volume of STSS tensor $\mathbf{S}$ into (L, C_F). See text for details.

temporal offsets in a consistent manner. While there exist many design choices, we introduce three methods for feature extraction in this work.

Soft-argmax. The first method is to compute explicit displacement fields using $\mathbf{S}$, which previous motion learning methods adopt using spatial cross-similarity [4,41,53]. One may extract the displacement field by indexing the positions with the highest similarity value via $\arg\max_{(u,v)}$, but it is not differentiable. We instead use *soft-argmax* [2], which aggregates displacement vectors with softmax weighting (Fig. 3a). The soft-argmax feature extraction can be formulated as

$$\mathbf{F}_{t,x,y,l} = \sum_{u,v} \frac{\exp(\mathbf{S}_{t,x,y,l,u,v}/\tau)}{\sum_{u',v'} \exp(\mathbf{S}_{t,x,y,l,u',v'}/\tau)}[u; v], \tag{2}$$

which results in a feature tensor $\mathbf{F} \in \mathbb{R}^{T \times X \times Y \times L \times 2}$. The temperature factor τ adjusts the softmax distribution, and we set $\tau = 0.01$ in our experiments.

Multi-layer perceptron (MLP). The second method is to learn an MLP that converts self-similarity values into a feature. For this, we flatten the (U, V) volume into UV-dimensional vectors, and apply an MLP to them (Fig. 3b). For the reshaped tensor $\mathbf{S} \in \mathbb{R}^{T \times X \times Y \times L \times UV}$, a perceptron $f(\cdot)$ can be expressed as

$$f(\mathbf{S}) = \mathrm{ReLU}(\mathbf{S} \times_5 \mathbf{W}_\phi), \tag{3}$$

where $\times_n$ denotes the n-mode tensor product, $\mathbf{W}_\phi \in \mathbb{R}^{C' \times UV}$ is the perceptron parameters, and the output is $f(\mathbf{S}) \in \mathbb{R}^{T \times X \times Y \times L \times C'}$. The MLP feature extraction can thus be formulated as

$$\mathbf{F} = (f_n \circ f_{n-1} \circ \cdots \circ f_1)(\mathbf{S}), \tag{4}$$

which produces a feature tensor $\mathbf{F} \in \mathbb{R}^{T \times X \times Y \times L \times C_F}$. This method is more flexible and may also be more effective than the soft-argmax because not only can it encode displacement information but also it can directly access the similarity values, which may be helpful for learning motion distribution.

Convolution. The third method is to learn convolution kernels over (L, U, V) volume of $\mathbf{S}$ (Fig. 3c). When we regard $\mathbf{S}$ as a 7D tensor $\mathbf{S} \in \mathbb{R}^{T \times X \times Y \times L \times U \times V \times C}$ with $C = 1$, the convolution layer $g(\cdot)$ can be expressed as

$$g(\mathbf{S}) = \mathrm{ReLU}(\mathrm{Conv}(\mathbf{S}, \mathbf{K}_e)), \tag{5}$$

where $\mathbf{K}_e \in \mathbb{R}^{1 \times 1 \times 1 \times L_\kappa \times U_\kappa \times V_\kappa \times C \times C'}$ is a multi-channel convolution kernel. Starting from $\mathbb{R}^{T \times X \times Y \times L \times U \times V \times 1}$, we gradually downsample (U,V) and expand channels via multiple convolutions with strides, finally resulting in $\mathbb{R}^{T \times X \times Y \times L \times 1 \times 1 \times C_F}$; we preserve the L dimension, since maintaining fine temporal resolution is shown to be effective for capturing detailed motion information [8, 29]. In practice, we reshape $\mathbf{S}$ and then apply a regular 3D convolution along (l, u, v) dimension of $\mathbf{S}$. The convolutional feature extraction with n layers can thus be formulated as

$$\mathbf{F} = (g_n \circ g_{n-1} \circ \cdots \circ g_1)(\mathbf{S}), \tag{6}$$

which results in a feature tensor $\mathbf{F} \in \mathbb{R}^{T \times X \times Y \times L \times C_F}$. This method is effective in learning structural patterns with their convolution kernels, thus outperforming the former methods as will be seen in our experiments.

3.2.2 Feature integration

In this step, we integrate the extracted STSS features $\mathbf{F} \in \mathbb{R}^{T \times X \times Y \times L \times C_F}$ to feed them back to the original input stream with (T, X, Y, C) volume.

We first use spatio-temporal convolution kernels along (t, x, y) dimension of $\mathbf{F}$. The convolution layer $h(\cdot)$ can be expressed as

$$h(\mathbf{F}) = \mathrm{ReLU}(\mathrm{Conv}(\mathbf{F}, \mathbf{K}_i)), \tag{7}$$

where $\mathbf{K}_i \in \mathbb{R}^{T_\kappa \times X_\kappa \times Y_\kappa \times 1 \times C_F \times C'_F}$ is a multi-channel convolution kernel. This type of convolution integrates the extracted STSS features by extending receptive fields along (t, x, y) dimension. In practice, we reshape $\mathbf{F}$ and then apply a regular 3D convolution along (t, x, y) dimension of $\mathbf{F}$.

The resultant features $\mathbf{F}^\star \in \mathbb{R}^{T\times X\times Y\times L\times C_F^\star}$ is defined as

$$\mathbf{F}^\star = (h_n \circ h_{n-1} \circ \cdots \circ h_1)(\mathbf{F}). \quad (8)$$

We then flatten the $(L, C_F^\star)$ volume into $LC_F^\star$-dimensional vectors to obtain $\mathbf{F}^\star \in \mathbb{R}^{T\times X\times Y\times LC_F^\star}$, and apply an $1\times1\times1$ convolution layer to obtain the final output. This convolution layer integrates features from different temporal offsets and also adjusts its channel dimension to fit that of the original input $\mathbf{V}$. The final output tensor $\mathbf{Z}$ is expressed as

$$\mathbf{Z} = \mathrm{ReLU}(\mathbf{F}^\star \times_4 \mathbf{W}_\theta), \quad (9)$$

where $\times_n$ is the n-mode tensor product and $\mathbf{W}_\theta \in \mathbb{R}^{C\times LC_F^\star}$ is the weights of the convolution layer.

Finally, we combine the resultant STSS representation $\mathbf{Z}$ into the input feature $\mathbf{V}$ by element-wise addition, thus making SELFY act as a residual block [11].

4. Experiments

4.1. Implementation details

Action recognition architecture. We employ TSN ResNets [49] as 2D CNN backbones and TSM ResNets [29] as 3D CNN backbones. TSM enables to obtain the effect of spatio-temporal convolutions using spatial convolutions by shifting a part of input channels along the temporal axis before the convolution operation. TSM is inserted into each residual block of the ResNet. We adopt ImageNet pre-trained weights for our backbones. To transform the backbones to the self-similarity network (SELFYNet), we insert a single SELFY block after the third stage in the backbone with additive fusion. For the feature extraction and integration in SELFY block, we use four $1\times3\times3$ convolution layers along (l, u, v) dimensions and four $1\times3\times3$ convolution layers along (t, x, y) dimensions, respectively. For more details, please refer to supplementary material A.

Training & testing. For training, we sample a clip of 8 or 16 frames from each video using segment-based sampling [49]. The spatio-temporal matching region (L, U, V) of SELFY block is set as $(5, 9, 9)$ or $(9, 9, 9)$ when using 8 or 16 frames, respectively. For testing, we sample one or two clips from a video, crop their center, and evaluate the averaged prediction of the sampled clips. For more details, please refer to supplementary material A.

4.2. Datasets

For evaluation, we use benchmarks that contain fine-grained spatio-temporal dynamics in videos.

Something-Something V1 & V2 (SS-V1 & V2) [10], which are both large-scale action recognition datasets, contain ∼108k and ∼220k video clips, respectively. Both datasets share the same 174 action classes that are labeled, *e.g.*, 'pretending to put something next to something.'

Diving-48 [28], which contains ∼18k videos with 48 different diving action classes, is an action recognition dataset that minimizes contextual biases, *i.e.*, scenes or objects.

FineGym [36] is a fine-grained action dataset built on top of gymnastic videos. We adopt the *Gym288* and *Gym99* sets that contain 288 and 99 classes, respectively.

4.3. Comparison with the state-of-the-art methods

For a fair comparison, we compare our model with other models that are not pre-trained on additional large-scale video datasets, *e.g.*, Kinetics [20] or Sports1M [19], in the following experiments.

Table 1 summarizes the results on SS-V1&V2. The first and second compartment of the table shows the results of other 2D CNN and (pseudo-) 3D CNN models, respectively. The last part of each compartment shows the results of SELFYNet. SELFYNet with TSN-ResNet (SELFYNet-TSN-R50) achieves 50.7% and 62.7% at top-1 accuracy, respectively, which outperforms other 2D models using 8 frames only. When we adopt TSM ResNet (TSM-R50) as our backbone and use 16 frames, our method (SELFYNet-TSM-R50) achieves 54.3% and 65.7% at top-1 accuracy, respectively, which is the best among the single models. Compared to TSM-R50, a single SELFY block obtains the significant gains of 7.0%p and 4.5%p at top-1 accuracy, respectively; our method is more accurate than TSM-R50 two-stream on both datasets. Finally, our ensemble model (SELFYNet-TSM-R50$_{EN}$) with 2-clip evaluation sets a new state-of-the-art on both datasets by achieving 56.6% and 67.7% at top-1 accuracy, respectively.

Tables 2 and 3 summarize the results on Diving-48 and FineGym. For Diving-48, TSM-R50 using 16 frames shows 38.8% at top-1 accuracy in our implementation. SELFYNet-TSM-R50 outperforms TSM-R50 by 2.8%p at top-1 accuracy so that it sets a new state-of-the-art top-1 accuracy as 41.6% on Diving-48. For FineGym, SELFYNet-TSM-R50 achieves 49.5% and 87.7% at given 288 and 99 classes, respectively, surpassing all the other models reported in [36].

4.4. Ablation studies

We conduct ablation experiments to demonstrate the effectiveness of the proposed method. All experiments are performed on SS-V1 using 8 frames. Unless specified otherwise, we set ImageNet pre-trained TSM ResNet-18 (TSM-R18) with the single SELFY block of which $(L, U, V) = (5, 9, 9)$, as our default SELFYNet.

Types of similarity. In Table 4a, we investigate the effect of different types of similarity by varying the set of temporal offset l on both TSN-ResNet-18 (TSN-R18) and TSM-R18. Interestingly, learning spatial self-similarity ($\{0\}$) im-

model	flow	#frame	FLOPs×clips	SS-V1		SS-V2	
				top-1	top-5	top-1	top-5
TSN-R50 from [29]		8	33 G×1	19.7	46.6	30.0	60.5
TRN-BNIncep [56]		8	16 G×N/A	34.4	-	48.8	-
TRN-BNIncep Two-Stream [56]	✓	8+8	16 G×N/A	42.0	-	55.5	-
MFNet-R50 [24]		10	N/A×10	40.3	70.9	-	-
CPNet-R34 [30]		24	N/A×96	-	-	57.7	84.0
TPN-R50 [52]		8	N/A×10	40.6	-	59.1	-
SELFYNet-R50 (ours)		8	37 G×1	**50.7**	**79.3**	**62.7**	**88.0**
I3D from [51]		32	153 G×2	41.6	72.2	-	-
NL-I3D from [51]		32	168 G×2	44.4	76.0	-	-
TSM-R50 [29]		16	65 G×1	47.3	77.1	61.2	86.9
TSM-R50 Two-Stream from [22]	✓	16+16	129 G×1	52.6	81.9	65.0	89.4
CorrNet-R101 [48]		32	187 G×10	50.9	-	-	-
STM-R50 [14]		16	67 G×30	50.7	80.4	64.2	89.8
TEA-R50 [27]		16	70 G×30	52.3	81.9	-	-
MSNet-TSM-R50 [22]		16	67 G×1	52.1	82.3	64.7	89.4
MSNet-TSM-R50$_{EN}$ [22]		8+16	101 G×10	55.1	84.0	67.1	91.0
SELFYNet-TSM-R50 (ours)		8	37 G×1	52.5	80.8	64.5	89.4
SELFYNet-TSM-R50 (ours)		16	77 G×1	54.3	82.9	65.7	89.8
SELFYNet-TSM-R50$_{EN}$ (ours)		8+16	114 G×1	55.8	83.9	67.4	91.0
SELFYNet-TSM-R50$_{EN}$ (ours)		8+16	114 G×2	**56.6**	**84.4**	**67.7**	**91.1**

Table 1: **Performance comparison on SS-V1&V2**. Top-1, 5 accuracy (%) and FLOPs (G) are shown.

model	#frame	FLOPs ×clips	Top-1
TSN from [28]	-	-	16.8
TRN from [18]	-	-	22.8
Att-LSTM [18]	64	N/A×1	35.6
P3D from [31]	16	N/A×1	32.4
C3D from [31]	16	N/A×1	34.5
GST-R50 [31]	16	59 G×1	38.8
CorrNet-R101 [48]	32	187 G×10	38.2
GSM-IncV3 [40]	16	54 G×2	40.3
TSM-R50 (our impl.)	16	65 G×2	38.8
SELFYNet-TSM-R50 (ours)	16	77 G×2	**41.6**

Table 2: **Performance comparison on Diving-48**. Top-1 accuracy (%) and FLOPs (G) are shown.

model	#frame	Gym288 Mean	Gym99 Mean
TSN [49]	3	26.5	61.4
TRN [56]	3	33.1	68.7
I3D [1]	8	27.9	63.2
NL I3D [50]	8	27.1	62.1
TSM [29]	3	34.8	70.6
TSM Two-Stream [29]	N/A	46.5	81.2
TSM-R50 (our impl.)	3	35.3	73.7
TSM-R50 (our impl.)	8	47.9	86.6
SELFYNet-TSM-R50 (ours)	8	**49.5**	**87.7**

Table 3: **Performance comparison on FineGym**. The averaged per-class accuracy (%) is shown. All results in the upper part are from FineGym paper [36].

proves accuracy on both backbones, which implies that self-similarity features help capture structural patterns of visual features. Learning cross-similarity with a short temporal range ($\{1\}$) shows a noticeable gain at accuracy on both backbones, indicating the significance of motion features. Learning STSS outperforms other types of similarity, and the accuracy of SELFYNet increases as the temporal range becomes longer. When STSS takes a far-sighted view on motion, STSS learns both short-term and long-term interactions in videos, as well as spatial self-similarity.

Feature extraction and integration methods. In Table 4b, we compare the performance of different combinations of feature extraction and integration methods. From the 2$^{\text{nd}}$ to the 4$^{\text{th}}$ rows, different feature extraction methods are compared, fixing the feature integration methods to a single fully-connected (FC) layer. Compared to the baseline, the use of soft-argmax, which extracts spatial displacement features, improves the top-1 accuracy by 1.0%p. Replacing soft-argmax with MLP provides the additional gain of 1.9%p at top-1 accuracy, showing the effectiveness of directly using similarity values. When using the convolution method for feature extraction, we achieve 46.7% at top-1 accuracy; the multi-channel convolution kernel is more effective in capturing structural patterns along (u, v) dimensions than MLP. From the 4$^{\text{th}}$ to the 6$^{\text{th}}$ rows, different feature integration methods are compared, fixing the feature extraction method to convolution. Replacing the single FC layer with MLP improves the top-1 accuracy by 0.6%p. Replacing MLP with convolutional layers further improves and achieves 48.4% at top-1 accuracy. These results demonstrate that our design choice of using convolutions along (u, v) and (h, w) dimensions is the most effective in learning the geometry-aware STSS representation. For more experiments, please refer to supplementary material C.

4.5. Relation with self-attention mechanisms

Note that self-similarity is also used in self-attention mechanisms [13, 35, 39, 47, 50], but both the purpose and the scheme are very different. Self-attention mechanisms aim to perform dynamic feature transformation based on the image context and thus use the self-similarity as attention weights in aggregating individual features. In contrast, our method focuses on learning relational representation from the self-similarity tensor itself. We directly transform the tensor into a relational representation with learnable convolution kernels, where the relational representation of video is interpreted as generalized motion representation.

For an apple-to-apple empirical validation, we compare our method with popular self-attention methods [35,39,50]. We re-implement the local self-attention [35] and Transformer [39] blocks, and extend them to a temporal dimension. For a fair comparison, we insert a single block after res_3 of ResNet-18. All other experimental details are the same as those in supplementary material A. Table 5 summarizes the results. Our method outperforms the self-attention methods at both top-1 and top-5 accuracies with large margins. These results demonstrate that learning the STSS representation effectively leverages motion features, which play a crucial role in action recognition. For more experiments, please refer to supplementary material C.

4.6. Complementarity of STSS features

We conduct experiments for analyzing different meanings of spatio-temporal features and STSS features. We organize two basic blocks for representing two different features: spatio-temporal convolution block (STCB) that consists of several spatial-temporal convolutions (Fig. 4a) and SELFY-s block, light-weighted version of the SELFY block by removing spatial convolution layers (Fig. 4b). Both blocks have the same receptive fields and a similar number of parameters for a fair comparison. Different combinations of the basic blocks are inserted after the third stage of TSN-ResNet-18. Table 6 summarizes the results on SS-V1. STSS features (Figs. 4b and 4d) are more effective than spatio-temporal features (Figs. 4a and 4c) at top-1 and top-5 accuracy when the same number of blocks are inserted. Interestingly, the combination of two different features (Figs. 4e and 4f) shows better results at top-1 and top-5 accuracy compared to the single feature cases (Figs. 4c and 4d), which demonstrate that both features complement each other. We conjecture that this complementarity comes from different characteristics of the two features; while spatio-temporal features are obtained by directly encoding appearance features, STSS features are obtained by suppressing variations in appearance and focusing on the relational features in space and time.

model	range of l	FLOPs	top-1	top-5
TSN-R18	-	14.6 G	16.2	40.8
SELFYNet	$\{0\}$	15.3 G	16.8	42.2
	$\{1\}$	15.3 G	39.7	68.9
	$\{-1, 0, 1\}$	16.3 G	44.7	73.9
	$\{-2, \cdots, 2\}$	17.3 G	**46.9**	75.9
	$\{-3, \cdots, 3\}$	18.3 G	**46.9**	**76.2**
TSM-R18	-	14.6 G	43.0	72.3
SELFYNet	$\{0\}$	15.3 G	45.0	73.4
	$\{1\}$	15.3 G	47.1	76.3
	$\{-1, 0, 1\}$	16.3 G	47.8	76.7
	$\{-2, \cdots, 2\}$	17.3 G	48.4	77.6
	$\{-3, \cdots, 3\}$	18.3 G	**48.6**	**77.7**

(a) **Types of similarity**. Performance comparison with different sets of temporal offset in SELFY block. $\{\cdot\}$ denotes a set of temporal offset l.

model	extraction	integration	top-1	top-5
TSM-R18	-	-	43.0	72.3
SELFYNet	Smax	FC	44.0	72.3
	MLP	FC	45.9	75.1
	Conv	FC	46.7	75.8
	Conv	MLP	47.2	75.9
	Conv	Conv	**48.4**	**77.6**

(b) **Feature extraction and integration methods.** Smax denotes the soft-argmax operation. MLP consist of four FC layers. The $1 \times 1 \times 1$ layer in the feature integration stage is omitted.

Table 4: **Ablations on SS-V1**. Top-1 & 5 accuracy (%) are shown.

model	(L, U, V)	top-1	top-5
TSM-R18	-	43.0	72.3
TSM-R18 + LSA [35]	$(5, 9, 9)$	43.8	72.8
TSM-R18 + NL [50]	global	43.5	73.4
TSM-R18 + MHSA [39]	global	44.0	72.8
SELFYNet	$(5, 9, 9)$	**48.4**	**77.6**

Table 5: **Performance comparison with self-attention methods [35, 39, 50].** LSA, NL, and MHSA denote a local self-attention block [35], non-local block [50], and multi-head self-attention block [39], respectively.

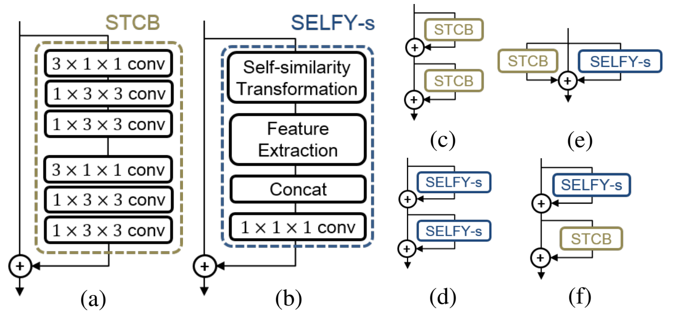
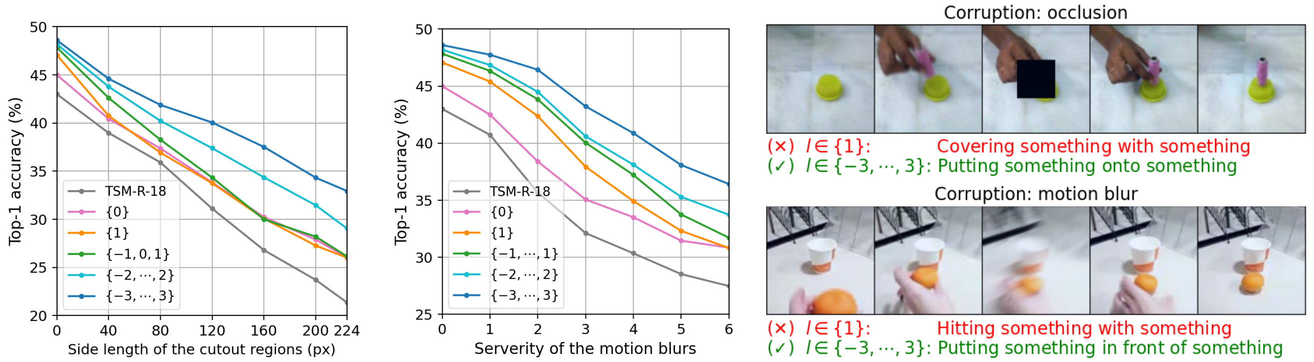


Figure 4: **Basic blocks and their combinations**. (a) spatio-temporal convolution block (STCB), (b) SELFY-s block, and (c-f) their different combinations.

model, TSN-R18	top-1	top-5
baseline	16.2	40.8
(a) STCB	42.4	71.7
(b) SELFY-s	46.3	75.1
(c) STCB + STCB	44.4	73.7
(d) SELFY-s + SELFY-s	46.8	75.9
(e) SELFY-s + STCB (parallel)	46.9	76.5
(f) SELFY-s + STCB (sequential)	**47.6**	**76.6**

Table 6: **Spatio-temporal features v.s. STSS features**. The basic blocks and their different combinations in Fig. 4 are compared on SS-V1.



(a) corruption: occlusion (b) corruption: motion blur (c) qualitative results on corrupted videos

Figure 5: **Robustness experiments**. (a) and (b) show top-1 accuracy of SELFYNet variants (Table 4a) when different degrees of occlusion and motion blur, respectively, are added to input. (c) shows qualitative examples where SELFYNet ($\{-3, \cdots, 3\}$) succeeds while SELFYNet ($\{1\}$) fails.

4.7. Improving robustness with STSS

In this experiment, we demonstrate that STSS representation helps video-processing models to be more robust to video corruptions. We test two types of corruption that are likely to occur in real-world videos: occlusion and motion blur. To induce the corruptions, we either cut out a rectangle patch of a particular frame or generate a motion blur [12]. We corrupt a single center-frame for every clip of SS-V1 at the testing phase and gradually increase the severity of corruption. We compare the results of TSM-R18 and SELFYNet variants of Table 4a. Figures 5a and 5b summarize the results of two corruptions, respectively. Top-1 accuracy of TSM-R18 and SELFYNets with the short temporal range ($\{0\}$, $\{1\}$, and $\{-1, 0, 1\}$) significantly drops as the severity of corruption becomes harder. We conjecture that features of the corrupted frame propagate through the stacked TSMs, confusing the entire network. However, the SELFYNets with the long temporal range ($\{-2, \cdots, 2\}$ and $\{-3, \cdots, 3\}$) show more robust performance than the other models. As shown in Figs. 5a and 5b, the accuracy gap between SELFYNets with the long temporal range and the others increases as the severity of corruptions becomes higher, indicating that the larger size of STSS features can improve the robustness on action recognition. We also present some qualitative results (Fig. 5c) where two SELFYNets with different temporal ranges, $\{1\}$ and $\{-3, \cdots, 3\}$, both answer correctly without corruption, while the SELFYNet with $\{1\}$ fails for the corrupted input.

5. Conclusion

We have proposed to learn a generalized, far-sighted motion representation from STSS for video understanding. The comprehensive analyses on the STSS demonstrate that STSS features effectively capture both short-term and long-term interactions, complement spatio-temporal features, and improve the robustness of video-processing models. Our method outperforms other state-of-the-art methods on the three benchmarks for video action recognition.

Acknowledgements. This work is supported by Samsung Advanced Institute of Technology (SAIT), the NRF grant (NRF-2021R1A2C3012728), and the IITP grant (No.2019-0-01906, AI Graduate School Program - POSTECH) funded by Ministry of Science and ICT, Korea.

References

[1] Joao Carreira and Andrew Zisserman. Quo vadis, action recognition? a new model and the kinetics dataset. In *Proc. IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2017. 1, 2, 6

[2] Olivier Chapelle and Mingrui Wu. Gradient descent optimization of smoothed information retrieval metrics. *Information retrieval*, 13(3):216–235, 2010. 4

[3] Jeffrey Donahue, Lisa Anne Hendricks, Sergio Guadarrama, Marcus Rohrbach, Subhashini Venugopalan, Kate Saenko, and Trevor Darrell. Long-term recurrent convolutional networks for visual recognition and description. In *Proc. IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2015. 2

[4] Alexey Dosovitskiy, Philipp Fischer, Eddy Ilg, Philip Hausser, Caner Hazirbas, Vladimir Golkov, Patrick Van Der Smagt, Daniel Cremers, and Thomas Brox. Flownet: Learning optical flow with convolutional networks. In *Proc. IEEE International Conference on Computer Vision (ICCV)*, 2015. 2, 3, 4

[5] Linxi Fan, Shyamal Buch, Guanzhi Wang, Ryan Cao, Yuke Zhu, Juan Carlos Niebles, and Li Fei-Fei. Rubiksnet: Learnable 3d-shift for efficient video action recognition. In *Proc. European Conference on Computer Vision (ECCV)*, 2020. 2

[6] Lijie Fan, Wenbing Huang, Chuang Gan, Stefano Ermon, Boqing Gong, and Junzhou Huang. End-to-end learning of motion representation for video understanding. In *Proc. IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2018. 2

[7] Christoph Feichtenhofer. X3d: Expanding architectures for efficient video recognition. In *Proc. IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2020. 2

[8] Christoph Feichtenhofer, Haoqi Fan, Jitendra Malik, and Kaiming He. Slowfast networks for video recognition. In *Proc. IEEE International Conference on Computer Vision (ICCV)*, 2019. 2, 4

[9] Rohit Girdhar and Deva Ramanan. Attentional pooling for action recognition. *arXiv preprint arXiv:1711.01467*, 2017. 2

[10] Raghav Goyal, Samira Ebrahimi Kahou, Vincent Michalski, Joanna Materzynska, Susanne Westphal, Heuna Kim, Valentin Haenel, Ingo Fruend, Peter Yianilos, Moritz Mueller-Freitag, et al. The" something something" video database for learning and evaluating visual common sense. In *Proc. IEEE International Conference on Computer Vision (ICCV)*, 2017. 2, 5

[11] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proc. IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016. 5

[12] Dan Hendrycks and Thomas Dietterich. Benchmarking neural network robustness to common corruptions and perturbations. *arXiv preprint arXiv:1903.12261*, 2019. 8

[13] Han Hu, Zheng Zhang, Zhenda Xie, and Stephen Lin. Local relation networks for image recognition. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 3464–3473, 2019. 2, 7

[14] Boyuan Jiang, Mengmeng Wang, Weihao Gan, Wei Wu, and Junjie Yan. Stm: Spatiotemporal and motion encoding for action recognition. In *Proc. IEEE International Conference on Computer Vision (ICCV)*, 2019. 2, 6

[15] Imran N Junejo, Emilie Dexter, Ivan Laptev, and Patrick Perez. View-independent action recognition from temporal self-similarities. *IEEE Transactions on Pattern Analysis and Machine Intelligence (TPAMI)*, 2010. 2

[16] Imran N Junejo, Emilie Dexter, Ivan Laptev, and Patrick PÚrez. Cross-view action recognition from temporal self-similarities. In *Proc. European Conference on Computer Vision (ECCV)*, 2008. 2

[17] Dahyun Kang, Heeseung Kwon, Juhong Min, and Minsu Cho. Relational embedding for few-shot classification. In *Proc. IEEE International Conference on Computer Vision (ICCV)*, 2021. 2

[18] Gagan Kanojia, Sudhakar Kumawat, and Shanmuganathan Raman. Attentive spatio-temporal representation learning for diving classification. In *Proc. IEEE Conference on Computer Vision and Pattern Recognition Workshops (CVPRW)*, 2019. 6

[19] Andrej Karpathy, George Toderici, Sanketh Shetty, Thomas Leung, Rahul Sukthankar, and Li Fei-Fei. Large-scale video classification with convolutional neural networks. In *Proc. IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2014. 5

[20] Will Kay, Joao Carreira, Karen Simonyan, Brian Zhang, Chloe Hillier, Sudheendra Vijayanarasimhan, Fabio Viola, Tim Green, Trevor Back, Paul Natsev, et al. The kinetics human action video dataset. *arXiv preprint arXiv:1705.06950*, 2017. 5

[21] Seungryong Kim, Dongbo Min, Bumsub Ham, Sangryul Jeon, Stephen Lin, and Kwanghoon Sohn. Fcss: Fully convolutional self-similarity for dense semantic correspondence. In *Proc. IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2017. 2

[22] Heeseung Kwon, Manjin Kim, Suha Kwak, and Minsu Cho. Motionsqueeze: Neural motion feature learning for video understanding. *arXiv preprint arXiv:2007.09933*, 2020. 1, 2, 3, 6

[23] Heeseung Kwon, Yeonho Kim, Jin S Lee, and Minsu Cho. First person action recognition via two-stream convnet with long-term fusion pooling. *Pattern Recognition Letters*, 112:161–167, 2018. 2

[24] Myunggi Lee, Seungeui Lee, Sungjoon Son, Gyutae Park, and Nojun Kwak. Motion feature network: Fixed motion filter for action recognition. In *Proc. European Conference on Computer Vision (ECCV)*, 2018. 2, 6

[25] Jun Li, Xianglong Liu, Mingyuan Zhang, and Deqing Wang. Spatio-temporal deformable 3d convnets with attention for action recognition. *Pattern Recognition*, 98:107037, 2020. 2

[26] Xianhang Li, Yali Wang, Zhipeng Zhou, and Yu Qiao. Smallbignet: Integrating core and contextual views for video classification. In *Proc. IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2020. 2

[27] Yan Li, Bin Ji, Xintian Shi, Jianguo Zhang, Bin Kang, and Limin Wang. Tea: Temporal excitation and aggregation for

action recognition. In *Proc. IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2020. 1, 2, 6

[28] Yingwei Li, Yi Li, and Nuno Vasconcelos. Resound: Towards action recognition without representation bias. In *Proc. European Conference on Computer Vision (ECCV)*, 2018. 2, 5, 6

[29] Ji Lin, Chuang Gan, and Song Han. Tsm: Temporal shift module for efficient video understanding. In *Proc. IEEE International Conference on Computer Vision (ICCV)*, 2019. 1, 2, 4, 5, 6

[30] Xingyu Liu, Joon-Young Lee, and Hailin Jin. Learning video representations from correspondence proposals. In *Proc. IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2019. 2, 6

[31] Chenxu Luo and Alan L Yuille. Grouped spatial-temporal aggregation for efficient action recognition. In *Proc. IEEE International Conference on Computer Vision (ICCV)*, 2019. 6

[32] Joanna Materzynska, Tete Xiao, Roei Herzig, Huijuan Xu, Xiaolong Wang, and Trevor Darrell. Something-else: Compositional action recognition with spatial-temporal interaction networks. In *Proc. IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2020. 1

[33] Gyeongsik Moon, Heeseung Kwon, Kyoung Mu Lee, and Minsu Cho. Integralaction: Pose-driven feature integration for robust human action recognition in videos. In *Proc. IEEE Conference on Computer Vision and Pattern Recognition Workshops (CVPRW)*, 2021. 1

[34] AJ Piergiovanni and Michael S Ryoo. Representation flow for action recognition. In *Proc. IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2019. 2

[35] Prajit Ramachandran, Niki Parmar, Ashish Vaswani, Irwan Bello, Anselm Levskaya, and Jonathon Shlens. Stand-alone self-attention in vision models. *arXiv preprint arXiv:1906.05909*, 2019. 2, 7

[36] Dian Shao, Yue Zhao, Bo Dai, and Dahua Lin. Finegym: A hierarchical video dataset for fine-grained action understanding. In *Proc. IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2020. 2, 5, 6

[37] Eli Shechtman and Michal Irani. Matching local self-similarities across images and videos. In *Proc. IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2007. 1, 2, 3

[38] Karen Simonyan and Andrew Zisserman. Two-stream convolutional networks for action recognition in videos. In *Proc. Neural Information Processing Systems (NeurIPS)*, 2014. 2

[39] Aravind Srinivas, Tsung-Yi Lin, Niki Parmar, Jonathon Shlens, Pieter Abbeel, and Ashish Vaswani. Bottleneck transformers for visual recognition. *arXiv preprint arXiv:2101.11605*, 2021. 2, 7

[40] Swathikiran Sudhakaran, Sergio Escalera, and Oswald Lanz. Gate-shift networks for video action recognition. In *Proc. IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2020. 6

[41] Deqing Sun, Xiaodong Yang, Ming-Yu Liu, and Jan Kautz. Pwc-net: Cnns for optical flow using pyramid, warping, and cost volume. In *Proc. IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2018. 2, 3, 4

[42] Shuyang Sun, Zhanghui Kuang, Lu Sheng, Wanli Ouyang, and Wei Zhang. Optical flow guided feature: A fast and robust motion representation for video action recognition. In *Proc. IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2018. 2

[43] Atousa Torabi and Guillaume-Alexandre Bilodeau. Local self-similarity-based registration of human rois in pairs of stereo thermal-visible videos. *Pattern Recognition*, 46(2):578–589, 2013. 2

[44] Du Tran, Lubomir Bourdev, Rob Fergus, Lorenzo Torresani, and Manohar Paluri. Learning spatiotemporal features with 3d convolutional networks. In *Proc. IEEE International Conference on Computer Vision (ICCV)*, 2015. 1, 2

[45] Du Tran, Heng Wang, Lorenzo Torresani, and Matt Feiszli. Video classification with channel-separated convolutional networks. In *Proc. IEEE International Conference on Computer Vision (ICCV)*, 2019. 2

[46] Du Tran, Heng Wang, Lorenzo Torresani, Jamie Ray, Yann LeCun, and Manohar Paluri. A closer look at spatiotemporal convolutions for action recognition. In *Proc. IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2018. 1, 2

[47] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Lukasz Kaiser, and Illia Polosukhin. Attention is all you need. *arXiv preprint arXiv:1706.03762*, 2017. 2, 7

[48] Heng Wang, Du Tran, Lorenzo Torresani, and Matt Feiszli. Video modeling with correlation networks. In *Proc. IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2020. 2, 3, 6

[49] Limin Wang, Yuanjun Xiong, Zhe Wang, Yu Qiao, Dahua Lin, Xiaoou Tang, and Luc Van Gool. Temporal segment networks: Towards good practices for deep action recognition. In *Proc. European Conference on Computer Vision (ECCV)*, 2016. 2, 5, 6

[50] Xiaolong Wang, Ross Girshick, Abhinav Gupta, and Kaiming He. Non-local neural networks. In *Proc. IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2018. 2, 6, 7

[51] Xiaolong Wang and Abhinav Gupta. Videos as space-time region graphs. In *Proc. European Conference on Computer Vision (ECCV)*, pages 399–417, 2018. 6

[52] Ceyuan Yang, Yinghao Xu, Jianping Shi, Bo Dai, and Bolei Zhou. Temporal pyramid network for action recognition. In *Proc. IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2020. 6

[53] Gengshan Yang and Deva Ramanan. Volumetric correspondence networks for optical flow. In *Proc. Neural Information Processing Systems (NeurIPS)*, 2019. 2, 3, 4

[54] Christopher Zach, Thomas Pock, and Horst Bischof. A duality based approach for realtime tv-l 1 optical flow. *Pattern Recognition*, pages 214–223, 2007. 2

[55] Yue Zhao, Yuanjun Xiong, and Dahua Lin. Trajectory convolution for action recognition. In *Proc. Neural Information Processing Systems (NeurIPS)*, 2018. 2

[56] Bolei Zhou, Alex Andonian, Aude Oliva, and Antonio Torralba. Temporal relational reasoning in videos. In *Proc. European Conference on Computer Vision (ECCV)*, 2018. 6

[57] Mohammadreza Zolfaghari, Kamaljeet Singh, and Thomas Brox. Eco: Efficient convolutional network for online video understanding. In *Proc. European Conference on Computer Vision (ECCV)*, 2018. 2