Spatio-temporal Filter Analysis Improves 3D-CNN For Action Classification

Takumi Kobayashi, Jiaxing Ye
National Institute of Advanced Industrial Science and Technology
1-1-1 Umezono, Tsukuba, Japan
{takumi.kobayashi,jiaxing.you}@aist.go.jp

Abstract

As 2D-CNNs are growing in image recognition literature, 3D-CNNs are enthusiastically applied to video action recognition. While spatio-temporal (3D) convolution successfully stems from spatial (2D) convolution, it is still unclear how the convolution works for encoding temporal motion patterns in 3D-CNNs. In this paper, we shed light on the mechanism of feature extraction through analyzing the spatio-temporal filters from a temporal viewpoint. The analysis not only describes characteristics of the two action datasets, Something-Something-v2 (SSv2) and Kinetics-400, but also reveals how temporal dynamics are characterized through stacked spatio-temporal convolutions. Based on the analysis, we propose methods to improve temporal feature extraction, covering temporal filter representation and temporal data augmentation. The proposed method contributes to enlarging temporal receptive field of 3D-CNN without touching its fundamental architecture, thus keeping the computation cost. In the experiments on action classification using SSv2 and Kinetics-400, it produces favorable performance improvement of 3D-CNNs.

1. Introduction

Convolutional neural networks (CNNs) produce successful performance on various image recognition tasks [13]. As the CNN techniques become mature, they are extended to 3D-CNNs for analyzing videos [2, 3, 35]. Spatial 2D-convolution is straightforwardly enhanced to 3D-convolution that directly operates on spatio-temporal volume of a video sequence. 3D-CNNs are thus applied to versatile tasks of video recognition including action classification [35], detection [7] and localization [12].

For exploiting temporal dynamics, 2D-CNNs are also applicable to video sequences in conjunction with temporal modeling modules [15, 19, 24, 26, 27, 38] and temporal processing by optical flow and frame/feature difference [31, 37, 38]. On the other hand, 3D-CNNs directly deal with input video sequences in a spatio-temporal manner as is the case with image recognition simply feeding images into 2D-CNNs. Thus, the 3D-CNNs naturally extract spatio-temporal characteristics embedded in video sequences by using deep architectures extended from 2D-CNNs. A seminal work of C3D model [35] simply stacks 3D-convolution layers in a similar fashion to VGG [32]. Then, following the great success of various 2D-CNNs, 3D-CNN models are also constructed based on the established 2D-CNNs as found in I3D [2], X3D [7] and I3D-ResNet [3].

A key process in video recognition is to extract effective features of temporal dynamics patterns involving motions and actions. While context and/or temporal reasoning is required [30] for understanding complex human actions, it is fundamental to encode distinctive temporal patterns by means of stacked spatio-temporal convolutions in 3D-CNNs. The temporal information that 3D-CNNs encode is analyzed through generating video frames preferred by C3D models [14] and visualizing neuron response [9] in a backward fashion. From a performance perspective, various building blocks in 3D-CNNs are analyzed in [3] and architectures of 3D-CNNs are explored by [7]. While the various analyses are conducted in this literature, it is still less clear how spatio-temporal convolution extracts features of temporal dynamics patterns in 3D-CNNs.

This paper delves into the spatio-temporal filters used in 3D-convolution, specifically from the perspective of temporal dynamics patterns. As an orthogonal research direction to the previous analyses [3, 7, 14], we shed light on primary *temporal filter* patterns embedded in optimized 3D-CNNs. Our analysis interestingly clarifies characteristics of Something-Something-v2 (SSv2) [11] and Kinetics-400 (K-400) [20] datasets, two standard benchmark datasets on action classification. More importantly, it provides an insight into the mechanism to encode temporal information through stacked 3D convolutions, which inspires us to propose a method for improving temporal feature extraction. The method involves spatio-temporal filter representation and temporal data augmentation. We leverage multi-branch reparameterization [5] to enhance temporal feature representation of spatio-temporal filters by using a proper reg-

ularization derived from our filter analysis. The enriched form of spatio-temporal filters demands augmentation techniques to endow regularization with training on video sequences for enhancing robustness against temporal perturbations. We analyze the proposed model based on the effective receptive field [28] to attain a better understanding of how the 3D-CNN works in the spatio-temporal domain. The contributions of this paper are summarized as follows.

- We conduct an in-depth analysis of pre-trained spatio-temporal filters to reveal the progressive temporal feature extraction in 3D-CNNs. It also differentiates SSv2 [11] and K-400 [20] based on temporal characteristics.
- The analysis induces the proposed methods to improve temporal feature representation by means of filter reformulation and temporal-enhanced data augmentation.
- The methods are qualitatively analyzed through the lens of effective receptive field [28] and are empirically evaluated on the action classification tasks of SSv2 [11] and K-400 [20] to exhibit favorable performance.

1.1. Related works

3D-CNN. The success of 2D-CNNs in static image recognition has motivated extensive research efforts in developing 3D-CNN models for video action recognition [2, 7, 18, 23, 35, 36]. Modern 3D-CNNs are built upon successful 2D-CNNs such as Inception [34] and ResNet [13]; the 2D-models pretrained on ImageNet [4] are further leveraged to effectively initialize the 3D-models [2, 3]. Toward computational efficiency, an array of research has been conducted to decompose the 3D convolutions into spatial 2D and temporal 1D-convolutions [29, 33, 36, 39]. Efficient 3D architectures are explored in a systematic way through expanding 2D models in [7]. On the other hand, SlowFast model [8] leverages 3D-CNNs to extracting various temporal dynamics especially in terms of motion speed. We cope with such temporal variations in a framework of data augmentation. Various approaches related to 3D-CNNs are thoroughly compared in [3] on action classification.

Analysis of CNN. Spatial and temporal information are both crucial for characterizing actions and are encoded by using 2D and 3D convolutions, respectively. Recent investigations have been conducted to assess the significance of 3D convolution over 2D convolution through empirical comparison experiments [36, 39]. Though they are related to our empirical evaluation in Sec. 2.2, we provide in-depth analysis about spatio-temporal filter weights from a temporal perspective (Sec. 2.1). Our analysis is also contributive to describe characteristics of two benchmark datasets of SSv2 [11] and Kinetics-400 [20]. Bias toward spatial representation in the Kinetics dataset, so-called static bias [9, 14, 25], can be explained by the temporal analysis of spatio-temporal *filters*. In contrast to [36, 39], our analysis covers temporal structure of spatio-temporal filters beyond the simple alternative of 2D or 3D, which leads to the proposed methods to improve temporal feature representation in Sec. 3&4. On the other hand, analyzing convolution filters is found in a framework of 2D-CNNs for understanding feature extraction process [41], exploring filter bases [17, 22] and investigating meta-structures [10]. Our analysis leads to effective filter formulation by identifying primary temporal patterns from the spatio-temporal filters.

Table 1. I3D-ResNet architectures. Filter size is denoted as (t, h, w) and an inflated dimension is indicated by t_l. The original models are given by $t_l = 3,\ \forall l \in \{1, \cdots, 5\}$.

Block	output	I3D-ResNet-18	I3D-ResNet-50
conv 1	$32\times112\times112$	$\underline{t_1}\times7\times7, 64$, stride 2	
conv 2	$32\times56\times56$	$1\times3\times3$ max-pool, stride 2	
		$\begin{bmatrix}\underline{t_2}\times3\times3, 64\\ \underline{t_2}\times3\times3, 64\end{bmatrix}\times2$	$\begin{bmatrix}1\times1\times1, 64\\ \underline{t_2}\times3\times3, 64\\ 1\times1\times1, 256\end{bmatrix}\times3$
conv 3	$32\times28\times28$	$\begin{bmatrix}\underline{t_3}\times3\times3, 128\\ \underline{t_3}\times3\times3, 128\end{bmatrix}\times2$	$\begin{bmatrix}1\times1\times1, 128\\ \underline{t_3}\times3\times3, 128\\ 1\times1\times1, 512\end{bmatrix}\times4$
conv 4	$32\times14\times14$	$\begin{bmatrix}\underline{t_4}\times3\times3, 256\\ \underline{t_4}\times3\times3, 256\end{bmatrix}\times2$	$\begin{bmatrix}1\times1\times1, 256\\ \underline{t_4}\times3\times3, 256\\ 1\times1\times1, 1024\end{bmatrix}\times6$
conv 5	$32\times7\times7$	$\begin{bmatrix}\underline{t_5}\times3\times3, 512\\ \underline{t_5}\times3\times3, 512\end{bmatrix}\times2$	$\begin{bmatrix}1\times1\times1, 512\\ \underline{t_5}\times3\times3, 512\\ 1\times1\times1, 2048\end{bmatrix}\times3$
	$1\times1\times1$	avg-pool, 1000-dim. FC, softmax	

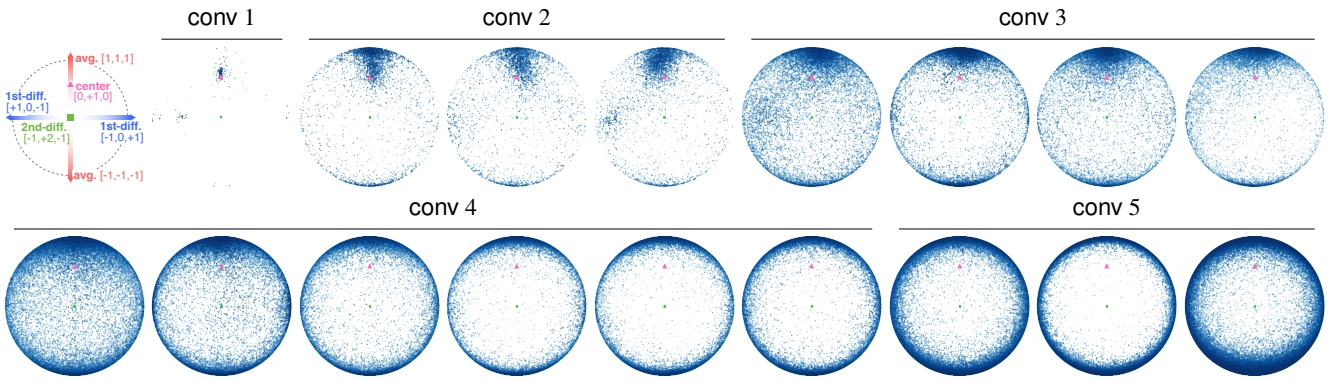
Video augmentation. Data augmentation provides favorable regularization to CNN training for increasing robustness against perturbations of input patterns and thereby enhancing generalization performance. It is studied mainly in a literature of static image recognition, and is also applied to image frames on video action recognition. Recently, some augmentation techniques are proposed by taking into account the three-way tensor structure of video sequences [21, 40]. In this study, we present simple and interpretable augmentation based solely on *temporal dynamics* for enhancing temporal feature representation.

2. Analyzing temporal filters

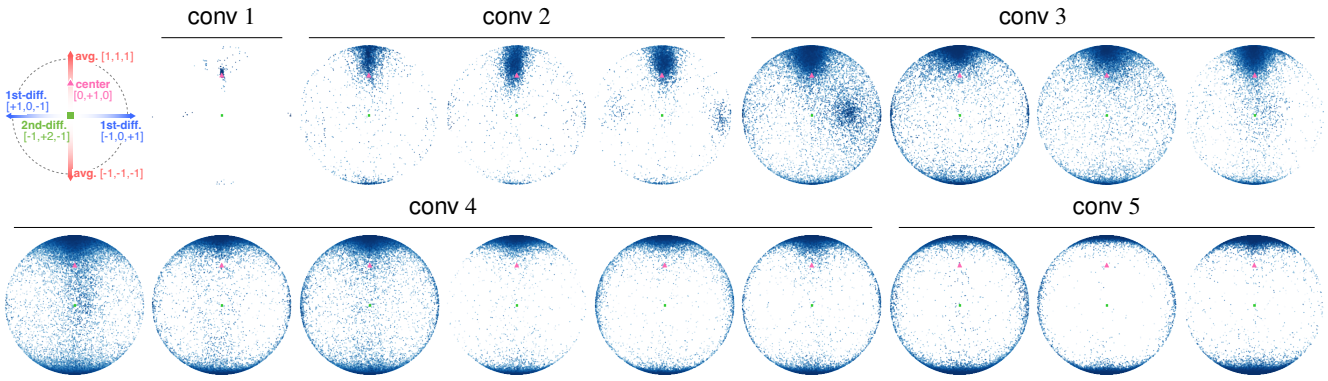
3-D CNNs [18, 35] are composed of spatio-temporal filters which distinctively contain *temporal dimension* in comparison to ordinary 2-D convolution [13]. We analyze the filters from a temporal perspective to explore the mechanism how temporal dynamics are encoded by 3D-CNN and to uncover potential avenues for further improvement.

2.1. Qualitative analysis

In a manner similar to spatial filter analysis [10, 17, 22], we utilize pretrained 3-D CNN models to provide optimal filter weights which well distinguish temporal pat-



(a) Pretrained on Something-Something-v2 dataset [11]



(b) Pretrained on Kinetics-400 dataset [20]

Figure 1. Distributions of the primary temporal filters embedded in I3D-ResNet-50 (Table 1) which is pretrained on (a) SSv2 [11] and (b) K-400 [20] datasets. The temporal filters are normalized in unit L_2 norm to distribute on a *sphere*. For ease of visualization, the distributions are projected onto a *plane* spanned by the average filter and the 1st-differential filter as shown in the top-left chart; the center filter of $[0, 1, 0]$ is specifically denoted by a triangle maker with magenta color.

terns of actions. Spatio-temporal filters are sampled from I3D-ResNet-50 [2, 3] pretrained on Something-Something-v2 (SSv2) and Kinetics-400 (K-400) datasets. The I3D-ResNet-50 simply inflates spatial convolution filters of ResNet-50 [13] to spatial-temporal ones of $3_{(t)}\times3_{(h)}\times3_{(w)}$; the network architecture is shown in Table 1.

We extract primary temporal filter patterns by means of singular-value decomposition (SVD) as follows. Pretrained filter weight tensor $\mathcal{W} \in \mathbb{R}^{D\times C\times t\times h\times w}$ for C-input and D-output channels is split along channel dimensions into a set of filter matrices $\{\boldsymbol{W}^{c,d} \in \mathbb{R}^{t\times hw}\}_{c,d=1}^{C,D}$. A spatio-temporal filter of $t \times h \times w$ is unfolded into a matrix $\boldsymbol{W}^{c,d}$ of $t \times hw$, to which SVD is applied as

$$\boldsymbol{W}^{c,d} = \boldsymbol{U}^{c,d}\boldsymbol{\Lambda}^{c,d}(\boldsymbol{V}^{c,d})^\top, \tag{1}$$

where $\boldsymbol{U}^{c,d} = [\boldsymbol{u}_1^{c,d}, \cdots, \boldsymbol{u}_t^{c,d}] \in \mathbb{R}^{t\times t}$ denote the (normalized) primary temporal filters embedded in the spatio-temporal filter $\boldsymbol{W}^{c,d}$ with the weights $\boldsymbol{\Lambda}^{c,d} = \mathrm{diag}(\lambda_1^{c,d}, \cdots, \lambda_t^{c,d})$. As the pretrained model is equipped with spatio-temporal filters of $t = 3$ (Table 1), the primary temporal filters $\{\boldsymbol{u}_i^{c,d}\}_{i,c,d}^{3,C,D}$ are distributed on a sphere.

Figure 1 shows the distributions of $\boldsymbol{u}_i^{c,d}$ with weights $\lambda_i^{c,d}$ at respective layers; there are totally 17 layers across the blocks of conv 1∼5. To facilitate the interpretation of temporal patterns conveyed in spherical distributions, we have developed a three-dimensional Cartesian coordinate system comprised of three physically interpretable axes: the average filter $\propto [1, 1, 1]^\top$ (vertical), the 1st differential $\propto [-1, 0, 1]^\top$ (horizontal), and the 2nd differential filter $\propto [-1, 2, -1]^\top$ (center of circle). And, the center filter $[0, 1, 0]^\top$ is specifically shown as a triangle marker. The visualization in Figure 1 leads to the following key findings.

- At the shallower layers of conv 1 and conv 2, distributions are concentrated around the *center filter* $[0, 1, 0]^\top$; particularly at conv 1 a majority of the temporal filters gather at the center filter. It indicates that spatio-temporal filters are degraded into *spatial* filters at these layers. Thus, the shallower layers primarily contribute to extracting spatial features without placing significant emphasis on temporal dynamics. In conv 2, the distributions slightly shift away from the center filter toward the average filter $[1, 1, 1]^\top$.
- The deeper layers at conv 3∼5 exhibit diverse temporal patterns by distributing temporal filters across a sphere

including differential filters. It implies that these deep layers extract a range of temporal dynamics in contrast to the shallow layers. As the temporal filters become complicated for encoding various dynamics, it stands to reason that they require a larger *temporal receptive field*. Actually, the distributions are biased toward the average $[1,1,1]^\top$ and/or 1st differential $[-1,0,1]^\top$ filters which possess larger receptive fields along a time axis. In most layers, the temporal filters are thinly distributed around the 2nd differential (center of circle), implying effectiveness of 1st differential; it interestingly validates the simple feature difference in TAM [27].

• The temporal filters pretrained on the two datasets of SSv2 [11] and K-400 [20] exhibit both similar and distinctive behavior across layers. Specifically, conv 1∼3 show similar distributional patterns, while the deeper layers of conv 4 and conv 5 exhibit different distributions. On SSv2, they are distributed rather *diversely* on a sphere; the distributions are especially found in circumference indicating the combination filters of the average and the 1st differential filters. It suggests that temporal filters are optimized so as to capture various temporal dynamics present in the SSv2 dataset. The SSv2 contains various human-object interection via action, intrinsically demanding for distinguishing detailed differences of temporal dynamics patterns. In contrast, the temporal filters on K-400 are *simply* distriuted around the average filter $[1,1,1]^\top$. The distribution implies that on K-400 encoding spatial characteristics that are consistent across time is more important than capturing temporal dynamics. Such a *static bias* of K-400 is pointed out in [9, 14, 25]. Our analysis reveals these inherent difference between the two benchmark datasets through investigating distributions of *temporal filters* in the pretrained models. It is also noteworthy that the similarity in conv 1∼3 across so different datasets of SSv2 and K-400 can imply the generality of those behaviors at the shallower layers.

2.2. Quantitative analysis

The above analyses suggest that the spatio-temporal filters be reconfigured by equipping the shallower layers with filters of short-time length and the deeper ones with larger-temporal filters. This is empirically evaluated as follows.

Experimental setting. I3D-ResNet-50 (Table 1) is trained on mini-SSv2 dataset [3, 11] which is a subset of SSv2 by randomly picking up half of whole 174 class categories. In the I3D-ResNet, we construct 3D convolution filters of $t_l \times 3 \times 3$ at conv l and initialize them by inflating pretrained spatial filters of 3×3 in ResNet-50 [13] into the spatio-temporal ones of $t_l \times 3 \times 3$. [2]. The original model [3] leverages ImageNet-pretrained ResNet-50 to implement 3-D convolution of $3 \times 3 \times 3$. Input spatio-temporal volume is composed of 32 video frames sampled every two frames through data augmentation of random cropping. Procedures of training and test are shown in a supplementary material.

Table 2. Performance results on various temporal-filter length $\{t_l\}_{l=1}^5$ in 3D-convolutions. t_l indicates the temporal length of the spatio-temporal filters at conv l as shown in Table 1.

	t_1	t_2	t_3	t_4	t_5	Acc.	GFLOPs
orig.	3	3	3	3	3	67.54	240
a	1	3	3	3	3	67.71	233
b	1	1	3	3	3	**68.07**	212
c	1	1	1	3	3	66.53	184
d	1	1	1	1	3	63.20	143
2D CNN	1	1	1	1	1	38.39	122
e	3	1	1	1	1	59.70	130
f	3	3	1	1	1	62.37	151
g	3	3	3	1	1	64.08	178
h	3	3	3	3	1	66.51	219
b_1	1	1	3	3	**5**	**68.83**	233
b_2	1	1	3	**5**	**5**	67.62	274
b_3	1	1	**5**	**5**	**5**	67.58	302
c_1	1	1	1	3	**5**	67.80	205
c_2	1	1	1	**5**	**5**	67.18	247

Performance analysis. Table 2 shows performance results over various temporal-filter lengths $\{t_l\}_{l=1}^5$ of 3-D convolution. To validate our analysis in Sec. 2.1 regarding temporal filters at the shallower layers, we degrade the filters of the shallower layers to $1_{(t_l)} \times 3 \times 3$ from the original $3 \times 3 \times 3$. It should be noted that the degraded filter of $t_l = 1$ works as 2-D *spatial* convolution without extracting any temporal dynamics. As shown in Table 2*a*∼*d*, the filter degradation contributes to improving both performance and computation cost until conv 2 ($l = 2$). This is consistent with the analysis of Figure 1 that conv 1&2 exhibit simple distributions of temporal filters being biased toward the center filter $[0,1,0]^\top$. In general, the shallower layers are vulnerable to perturbations of input frames/pixels such as by camera shaking and various noises derived from environments, which make it hard to effectively encode temporal dynamics by using small filters $3 \times 3 \times 3$. Besides, the shallower layers operate on such a high spatial resolution that they are sensitive to small high-frequency movements. Those movements are less relevent to target actions since target actions are mostly composed of low-frequency motions. Thus, it is beneficial to focus on extracting *spatial* characteristics of targets at the first several shallow layers.

The performance, however, is degraded by touching the temporal filters at deeper layers, which ultimately results in simple 2D-CNN (ResNet-50). The filter reduction at the deeper layer significantly drops performance; $t_5 = 1$ deteriorates performance from 63.20% (*d*) to 38.39% (*2D CNN*). These results indicate that the temporal filters at the deeper layers extract temporal dynamics more effectively, as suggested in the analysis of Sec. 2.1.

We then degrade the temporal filters in a manner from deep to shallow layers as shown in Table 2*e*∼*h*. It is less effective in comparison to the above approach from shallow to deep layers (Table 2*a*∼*d*). For example, under the same computation budget (FLOPs), the degradation only at conv 5 (*h*) is outperformed by the shallow one (*b*); 66.51% (219GFLOPs) vs 68.07 (212GFLOPs). These results support our claim that the deeper layers demand sufficient temporal filters to discriminatively encode temporal dynamics.

As shown in Table 2*b*, the best filter configuration is given by $t_1 = t_2 = 1$ to extract *spatial* characteristics at the shallow layers and $t_3 = t_4 = t_5 = 3$ to let the deeper layers encode *temporal* dynamics of so extracted spatial patterns. They respectively contribute to the following two points. (1) The shallow layers focus on spatial feature extraction to enhance robustness against high-frequency input perturbations that are irrelevant to the target action. (2) Target action patterns of low frequency are discriminated at the deep layers by encoding temporal patterns of rather *abstract-level* spatial characteristics, i.e., objects. While the comparison experiments in [36, 39] provide similar results to Table 2*a*∼*h*, our evaluation is distinctively built upon the detailed filter analysis in Sec. 2.1 which further encourages us to conduct the following experiments for the deep layers.

The analysis in Sec. 2.1 also implies that the deeper layers demand larger temporal receptive fields by enlarging temporal filters. The spatio-temporal filters of longer temporal length allows for effective encoding of low-frequency dynamics which exhibit less apparent spatial difference in short-time duration. Besides, it contributes to distinguishing minute difference of those temporal patterns from a frequency viewpoint according to the time-frequency uncertainty principle [1], $\Delta t \Delta \omega \geq \frac{1}{2}$. To empirically validate it, we enlarge temporal length of filters in the favorable configurations of Table 2*bc*. By enlarging temporal length only at conv 5 with $t_5 = 5$, we observe considerable performance improvement as shown in Table $2b_1c_1$. Further enlargement at conv 3 and/or conv 4, however, degrades performance as reported in Table $2b_2b_3c_2$. The longer temporal length requires the larger spatial scale (receptive field) for discriminating detailed spatio-temporal patterns. Thus, only conv 5, which has large spatial receptive fields, takes advantage of the large length $t_t = 5$ for improving performance. From a perspective of training CNNs, the performance drop may also be connected to the increased parameter size especially at conv 4 containing lots of convolution layers (Table 1).

The configuration of Table $2b_1$ that progressively increases temporal length of spatio-temporal filters produces the best performance while keeping the same computation cost as the original one. It is coincident with the filter analysis in Sec. 2.1 with Figure 1 which identifies the progressive mechanism of (temporal) feature extraction at conv 1∼5. We apply this configuration in the subsequent analyses.

3. Effective representation of temporal filter

Next, we technically explore effective representation of spatio-temporal filters in terms of temporal dimension.

3.1. Reparameterization

The filters of longer temporal length would have difficulty in training due to increased number of parameters. The issue is mitigated by means of a reparameterization technique [5] to facilitate the training of those filters while keeping the intrinsic structure. Convolution with a spatio-temporal filter $\boldsymbol{W}$ of $t \times 3 \times 3$ is reformulated via multiple branches (Figure 2a) of various-length filters $\{\hat{\boldsymbol{W}}_\tau\}_\tau$ as

$$\gamma \texttt{conv}(\boldsymbol{X}; \boldsymbol{W}) + \beta = \sum_{\tau \in \{1,3,\cdots,t\}} \gamma_\tau \texttt{conv}(\boldsymbol{X}; \hat{\boldsymbol{W}}_\tau) + \beta_\tau, \tag{2}$$

where γ and β are affine parameters used in BatchNorm [16] and τ indicates a temporal length of the filter $\hat{\boldsymbol{W}}_\tau$ in $\tau \times 3 \times 3$. A single filter $\boldsymbol{W}$ of $t \times 3 \times 3$ can be reconstructed by simply summing up those filters[1] to $\boldsymbol{W} = \sum_\tau \gamma_\tau \hat{\boldsymbol{W}}_\tau$ as shown in Figure 2a. By virtue of multi-branch training path [13], this reparameterization facilitates the CNN learning; while in [5] it is applied to spatial filters of VGG-models [32], we leverage it to reparameterization of *temporal* dimension in spatio-temporal filters.

3.2. Regularization

Though the multi-branch representation renders effective training, it likely brings up a bias toward the temporal center filter $[0, 1, 0]^\top$ due to overlap of multiple filters at the center position as shown in Figure 2a. Such a *centric* bias degrades a temporal receptive field of the filter, interfering with feature extraction of various temporal patterns. To remedy the bias, we propose a regularization method to suppress the overlap among multiple filters. As analyzed in Figure 1, the filters tend to contain components related to the average filter which causes heavy overlap across different-sized filters. Thus, we impose *high* L_2 regularization on the average filter; the L_2 regularization is usually applied to any filters as *weight decay* of optimizers in deep learning.

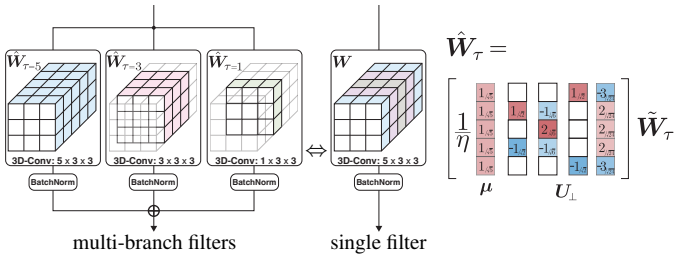
A multi-branch filter weight $\hat{\boldsymbol{W}}_\tau \in \mathbb{R}^{\tau \times hw}$ is decomposed as

$$\hat{\boldsymbol{W}}_\tau = \boldsymbol{\mu} \boldsymbol{A}_\tau + \boldsymbol{U}_\perp \boldsymbol{B}_\tau, \tag{3}$$

where the average filter $\boldsymbol{\mu} \in \mathbb{R}^\tau$ and its complement filters $\boldsymbol{U}_\perp \in \mathbb{R}^{\tau \times \tau - 1}$ are given such as by Figure 2b and

$$\boldsymbol{\mu} = \frac{1}{\sqrt{\tau}} \mathbf{1}_\tau,\ \boldsymbol{U}_\perp^\top \boldsymbol{\mu} = \mathbf{0} \text{ and } \boldsymbol{U}_\perp^\top \boldsymbol{U}_\perp = \boldsymbol{I}. \tag{4}$$

[1] Filters of different sizes are aligned via padding and any bias terms in BatchNorms are merged into a bias of convolution operation.



(a) Reparameterization (b) Regularization

Figure 2. Proposed temporal-filter representation. (a) Reparameterization [5] of spatio-temporal filters of $5_{(t)} \times 3_{(h)} \times 3_{(w)}$ by multiple branches. (b) Regularization on the average filtering.

The matrices $\boldsymbol{A}_\tau \in \mathbb{R}^{1\times hw}$ and $\boldsymbol{B}_\tau \in \mathbb{R}^{\tau-1\times hw}$ are coefficients for $\boldsymbol{\mu}$ and $\boldsymbol{U}_\perp$, respectively. This decomposition reformulates the L_2 regularization of $\|\hat{\boldsymbol{W}}_\tau\|_F^2$ into

$$\ell = \eta^2\|\boldsymbol{A}_\tau\|_F^2 + \|\boldsymbol{B}_\tau\|_F^2 = \|\tilde{\boldsymbol{A}}_\tau\|_F^2 + \|\boldsymbol{B}_\tau\|_F^2, \quad (5)$$

where $\eta\,(\geq 1)$ is a weight parameter. Larger weight η imposes higher regularization on the average filter to suppress it; $\eta = \infty$ completely excludes the average filtering from the spatio-temporal filter. The reparameterization of $\tilde{\boldsymbol{A}}_\tau = \eta\boldsymbol{A}_\tau$ simplifies the filter representation to

$$\hat{\boldsymbol{W}}_\tau = \left[\frac{1}{\eta}\boldsymbol{\mu}, \boldsymbol{U}_\perp\right]\begin{bmatrix}\tilde{\boldsymbol{A}}_\tau \\ \boldsymbol{B}_\tau\end{bmatrix} = \left[\frac{1}{\eta}\boldsymbol{\mu}, \boldsymbol{U}_\perp\right]\tilde{\boldsymbol{W}}_\tau, \quad (6)$$

$$\ell = \|\tilde{\boldsymbol{W}}_\tau\|_F^2. \quad (7)$$

The filter weight $\hat{\boldsymbol{W}}_\tau$ is finally reparameterized by $\tilde{\boldsymbol{W}}_\tau$ and we can simply apply ordinary weight decay to it without touching CNN architecture nor training procedure; the inflation initialization [2] is also applicable to $\tilde{\boldsymbol{W}}_\tau$ and the spatio-temporal convolution filter is reconstructed by (6).

3.3. Empirical evaluation.

The proposed methods are evaluated as in Sec. 2.2. The regularization in Sec. 3.2 is applicable not only to the multi-branch filters $\hat{\boldsymbol{W}}_\tau$ in the reparameterizatoin (2) but also to a (original) single filter $\boldsymbol{W}$. Table 3 shows performance comparison across various regularization weight η . In a case of a single filter, performance is degraded by suppressing the component of average filtering with larger η. The result implies that the average filtering contributes to extract temporal dynamics features by probably enlarging the receptive field. On the other hand, the regularization effectively works on the reparameterized filters by multiple branches (Sec. 3.1). Moderate regularization with $\eta \in \{2, 4\}$ improves performance of $\eta = 1$ (no regularization); the best performance is achieved with $\eta = 2$. While being comparable with the single filter ($\eta = 1$) in this experiment, the proposed temporal-filter representation with regularization is potentially further improved by injecting regularization into the training via *data augmentation* due to its over-parameterized representation.

Table 3. Performance comparison of various temporal filter representation on mini-SSv2. η indicates a regularization weight in (6).

η	single filter	multi-branch filter
1	**68.83**	68.17
2	67.51	**68.82**
4	62.90	68.54
∞	54.83	67.79

4. Temporal-enhanced data augmentation

We consider augmentation processes working on *temporal dynamics*, which are simple and interpretable. Similaly to 2-D CNNs on static images, input video frames are usually subject to *spatial* augmentaion techniques such as random cropping and random color jittering. While spatial augmentation techniques are extended for video sequences [21, 40], our simple augmentations more directly inject perturbation into temporal dynamics (Figure 3) for enhancing the robustness againt temporal variations in video sequences. They are well compatible with the temporal-filter representation in Sec. 3.

4.1. Random sampling rate

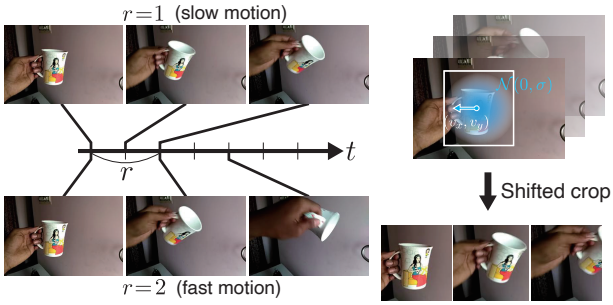
Video sequences are captured at a fixed sampling rate defined by a camera device which is thus variable across video sequences. It is also connected to target motion speed. Even the same action can be performed with different speed by different subjects. In appearance, different motion speed is regarded as different sampling rates as shown in Figure 3a. Thus, by changing sampling rates, we can artificially provide actions of various speed. To enhance robustness against those temporal variations, our data augmentation is formulated as a technique of random sampling rate to sample video frames at random frame intervals for building spatio-temporal volume fed into 3D-CNNs; SlowFast [8] deals with the various sampling rates in an architectural way. Given a video sequence $\mathcal{F} = [F_1, \cdots,]$, we sample T-frame subsequence starting at the s-th frame as

$$\mathcal{I} = [F_s, F_{s+r}, F_{s+2r}, \cdots, F_{s+(T-1)r}], \quad (8)$$

where a parameter r controls the sampling rate (Figure 3a) and is randomly drawn for each video sequence.

4.2. Random camera ego-motion

Movements observed in an video sequence can be categorized into target actions and the other irrelevant background motions caused by a camera *ego-motion*. In order to artificially embed the ego-motion into sequences, spatial random cropping is combined with spatial shift along time



(a) Random sampling rate (b) Random ego-motion

Figure 3. Temporal-enhanced data augmentation.

as shown in Figure 3b. Let (x_0, y_0) denote a position where an image patch is cropped at the first frame I_0, and then image patches are cropped on the subsequent frames at

$$(x_0 + v_x i, y_0 + v_y i) \text{ for } I_i,\ i \in \{0, \cdots, T-1\}, \quad (9)$$

where (v_x, v_y) indicates an ego-motion velocity, which is randomly sampled from $\mathcal{N}(0, \sigma)$ with a hyper-parameter σ indicating strength of the ego-motion. In the video augmentation technique [21], such a spatial shift of cropped image patches can be implicitly implemented through linearly mixing two augmentations of different (random) spatial cropping. Our shifting, however, is limited in the neighborhood of (x_0, y_0) to mimic camera ego-motion, while any spatial interval is employed in [21]. The method is also different from [6] which applies successive homographic transform to generate a pseudo video from a single image.

4.3. Empirical evaluation

As in Sec. 3.3, we apply the temporal-enhanced augmentation techniques to train the models of single filter and regularized multi-branch filter, dubbed as multi-filter.

Table 4a shows performance results by applying augmentation of random sampling rate (Sec. 4.1); it is implemented by uniformly drawing a sampling rate from $r \in \{1, \cdots, 4\}$ during training, and then at inference we apply $r = 2$ as the averaged sampling rate; we used $r = 2$ in the previous experiments. For comparison, we apply fixed sampling rates of $r = 1, 2, 4$. There is no significant performance improvement across those fixed sampling rates. Particularly, the larger rate of $r = 4$ degrades performance as it provides too sparse sampling to capture the motion patterns in the mini-SSv2 dataset. On the other hand, our augmentation considerably improves performance for the multi-filter model, while being less effective for the single-filter one.

Table 4b presents the performance results of the ego-motion augmentation, where we apply various σ in the ego-motion prior $\mathcal{N}(0, \sigma)$. As is the case with Table 4a, the augmentation does not work for the single-filter model but renders performance improvement to the multi-filter model.

Table 4. Performance results on mini-SSv2 by applying the proposed data augmentation to various filter representation (Sec. 3).

(a) Random sampling rate

r	single filter	multi-filter
1	68.62	68.84
2	68.83	68.82
4	65.73	67.06
`Unif`$[1, 4]$	68.81	69.69

(b) Random ego-motion

σ	single filter	multi-filter
0	68.83	68.82
1	68.74	69.00
2	67.93	69.19
4	68.52	68.81

(c) Joint augmentation

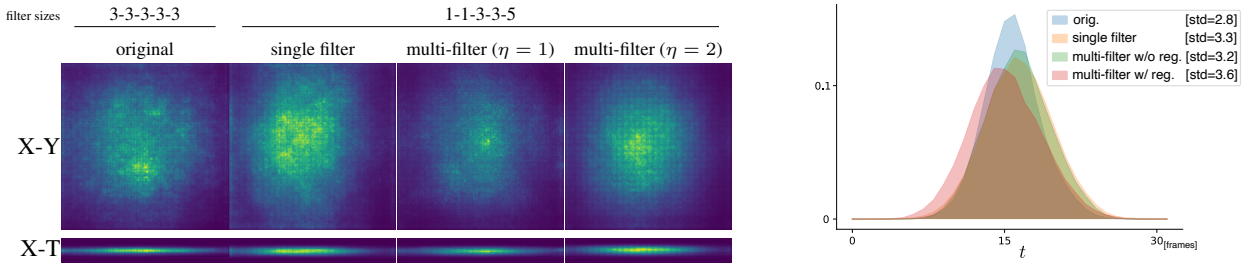
r	σ	original	single filter	multi-filter
2	0	67.54	68.83	68.82
`Unif`$[1, 4]$	2	67.70	68.74	69.98

These results motivate us to apply the two temporal augmentation techniques jointly, as shown in Table 4c. The joint augmentation further improves performance *only* of the multi-filter model, while having no impact on the performance of the single-filter model and the original model (Table 2*orig*). In contrast to the single-filter model, the regularized reparameterization in Sec. 3 facilitates the training of 3D-CNNs with over-parameterization [5] on which the augmentation favorably works for improving performance. These augmentation techniques are simply derived from the common temporal perturbations frequently found in video sequences, and thus are interpretable; the hyper-parameters of η and σ could be tuned based on the prior knowledge about input videos. Besides, they are naturally compatible with the other video augmentation techniques [21, 40].

5. Effective receptive field in temporal domain

As shown in Table 4, the two models of single filter and multi-filter produce different performance. This section analyzes those models in terms of their *effective receptive field* [28] achieved through training in stead of theoretical ones defined by filter sizes. We measure the effective receptive fields of the 3D-CNNs to understand how they work on temporal dimension for extracting temporal features; the computation procedure [28] is detailed in a supplementary material. For comparison, we also apply the original I3D-ResNet-50 (Table 2*orig*) and the simple multi-filter model (Sec. 3.1) without regularization ($\eta = 1$). To ease comparison of 3D receptive fields, we marginalize it into 2D-fields of X-Y and X-T planes in Figure 4a as well as into 1D-field along time axis in Figure 4b.

While the spatial receptive fields are similarly depicted like Gaussian shape in Figure 4a, we can find difference in temporal receptive fields in Figure 4b. The original model produces the narrowest temporal field. It is enlarged by the single-filter model which gradually increases temporal filter length (Table 2b_1). The larger temporal receptive field



(a) Marginalized 2-D receptive field

(b) Marginalized 1-D receptive field

Figure 4. Effective receptive fields [28] of 3D-CNNs. (a) 3D receptive field is marginalized into 2-D map. The upper row shows X-Y map while X-T map is shown in the bottom row. (b) The receptive field is marginalized into 1-D temporal axis, followed by normalization into unit sum in a manner of a probabilistic distribution. The standard deviation of the distribution is shown in the legend.

Table 5. Performance results on SSv2 [11] and K-400 [20] datasets by using I3D [2] and S3D [39] architecture.

method	3D-CNN	backbone	temp.-filt. length	GLOPs	Acc@Mini-SSv2	Acc@SSv2	Acc@Mini-K400	Acc@K400
Orignal	I3D	ResNet18	3-3-3-3-3	161	64.01	60.41	71.20	68.84
Ours	I3D	ResNet18	1-1-3-3-5	151	66.03	62.40	73.17	70.56
Original	I3D	ResNet50	3-3-3-3-3	240	67.54	63.55	75.41	73.58
Ours	I3D	ResNet50	1-1-3-3-5	233	69.98	65.52	76.43	73.99
Original	S3D	ResNet50	3-3-3-3-3	148	61.61	63.48	69.94	72.85
Ours	S3D	ResNet50	1-1-3-3-5	140	65.60	66.02	72.08	73.40

contributes to encoding detailed dynamic patterns of motion, which leads to performance improvement in Table 2. As to reparameterization (Sec. 3.1), the naive multi-branch approach [5] without regularization ($\eta = 1$) produces almost the same temporal receptive field as the single-filter model. The regularization with $\eta = 2$ suppresses overlap among multi-branch filters, favorably enlarging the temporal receptive field; the model provides the largest temporal field. It endows the 3D-CNN with discriminative power to well characterize temporal patterns, and works collaborately with the temporal-enhanced augmentation techniques (Sec. 4) for furhter improving performance (Table 4).

6. Performance evaluation

We finally evaluate performance of the proposed model on Something-Something-v2 (SSv2) [11] and Kinetics-400 (K400) [20] datasets on action classification of 174 and 400 categories, respectively. The 3D-CNN models are trained over 100 epochs with mini-batch size of 32 using cosine-scheduled learning rate which starts with 0.01; the detailed training protocol is shown in a supplementary material.

We apply ResNet-18 and ResNet-50 as backbones to construct I3D-CNNs [2], architectures of which are shown in Table 1. Our model is composed of regularized multi-branch filters (Sec. 3) with temporal data augmentations (Sec. 4); it is the same model as that applied in Table 4c. It is also applied to S3D-ResNet-50 [39] which decomposes 3D filters into spatial and temporal ones; our method is applicable to the temporal filters. The performance results are shown in Table 5. The proposed method produces favorable improvement over the original model while retaining the same computation cost. The performance improvements are more apparent on the dataset of SSv2 than K400 and with the backbone of ResNet-18 than ResNet-50. The SSv2 dataset relies on rather pure motion patterns [11] while K400 [20] contains a *static bias* toward static information as analyzed in Sec. 2.1. Thus, the proposed method improving the mechanism of temporal feature extraction works more effectively on the SSv2 for distinguishing detailed motion patterns. As to a backbone model, compared with ResNet-50, ResNet-18 is equipped with larger number of spatio-temporal convolution at `conv 5` to which the longer-length temporal filters ($t_5 = 5$) are applied. They contribute to temporal feature extraction more effectively, leading to significant performance improvement in I3D-ResNet-18. It is noteworthy that our method works for the decomposed temporal filters in S3D [39] which reduces computation cost.

7. Conclusion

We have conducted novel analysis of spatio-temporal filters embedded in pretrained 3D-CNNs from a temporal viewpoint. The analysis reveals the mechanism that temporal dynamics patterns are encoded progressively through numbers of spatio-temporal convolution. Based on the analysis, we propose methods to improve temporal feature extraction in 3D-CNNs without modifying a fundamental architecture nor especially increasing computation cost. The effectiveness of the method is validated through quantitative and qualitative evaluation on SSv2 and K400 datasets.

References

[1] Reza Parhizkar Yann Barbotin and Martin Vetterli. Sequences with minimal time-frequency uncertainty. *Applied and Computational Harmonic Analysis*, 38:452–468, 2015. 5

[2] Joao Carreira and Andrew Zisserman. Quo vadis, action recognition? a new model and the kinetics dataset. In *CVPR*, pages 6299–6308, 2017. 1, 2, 3, 4, 6, 8

[3] Chun-Fu Chen, Rameswar Panda, Kandan Ramakrishnan, Rogerio Feris, John Cohn, Aude Oliva, and Quanfu Fan. Deep analysis of cnn-based spatio-temporal representations for action recognition. In *CVPR*, pages 6165–6175, 2021. 1, 2, 3, 4

[4] Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. Imagenet: A large-scale hierarchical image database. In *CVPR*, pages 248–255, 2009. 2

[5] Xiaohan Ding, Xiangyu Zhang, Ningning Ma, Jungong Han, Guiguang Ding, and Jian Sun. Repvgg: Making vgg-style convnets great again. In *CVPR*, pages 13733–13742, 2021. 1, 5, 6, 7, 8

[6] Haodong Duan, Yue Zhao, Yuanjun Xiong, Wentao Liu, and Dahua Lin. Omni-sourced webly-supervised learning for video recognition. In *ECCV*, page 670–688, 2020. 7

[7] Christoph Feichtenhofer. X3d: Expanding architectures for efficient video recognition. In *CVPR*, pages 203–213, 2020. 1, 2

[8] Christoph Feichtenhofer, Haoqi Fan, Jitendra Malik, and Kaiming He. Slowfast networks for video recognition. In *ICCV*, pages 6202–6211, 2019. 2, 6

[9] Christoph Feichtenhofer, Axel Pinz, Richard P Wildes, and Andrew Zisserman. What have we learned from deep representations for action recognition? In *CVPR*, pages 7844–7853, 2018. 1, 2, 4

[10] Paul Gavrikov and Janis Keuper. Cnn filter db: An empirical investigation of trained convolutional filters. In *CVPR*, pages 19066–19076, 2022. 2

[11] Raghav Goyal, Samira Ebrahimi Kahou, Vincent Michalski, Joanna Materzyńska, Susanne Westphal, Heuna Kim, Valentin Haenel, Ingo Fruend, Peter Yianilos, Moritz Mueller-Freitag, Florian Hoppe, Christian Thurau, Ingo Bax, and Roland Memisevic. The 'something something' video database for learning and evaluating visual common sense. In *ICCV*, pages 5843–5851, 2017. 1, 2, 3, 4, 8

[12] Bo He, Xitong Yang, Le Kang, Zhiyu Cheng, Xin Zhou, and Abhinav Shrivastava. Asm-loc: Action-aware segment modeling for weakly-supervised temporal action localization. In *CVPR*, pages 13925–13935, 2022. 1

[13] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *CVPR*, pages 770–778, 2016. 1, 2, 3, 4, 5

[14] De-An Huang, Vignesh Ramanathan, Dhruv Mahajan, Lorenzo Torresani, Manohar Paluri, Li Fei-Fei, and Juan Carlos Niebles. What makes a video a video: Analyzing temporal information in video understanding models and datasets. In *CVPR*, pages 7366–7375, 2018. 1, 2, 4

[15] Noureldien Hussein, Efstratios Gavves, and Arnold W.M. Smeulders. Timeception for complex action recognition. In *CVPR*, pages 254–263, 2019. 1

[16] Sergey Ioffe and Christian Szegedy. Batch normalization: Accelerating deep network training by reducing internal covariate shift. *Journal of Machine Learning Research*, 37:448–456, 2015. 5

[17] Jörn-Henrik Jacobsen, Jan van Gemert, Zhongyou Lou, and Arnold W. M. Smeulders. Structured receptive fields in cnns. In *CVPR*, pages 2610–2619, 2016. 2

[18] Shuiwang Ji, Wei Xu, Ming Yang, and Kai Yu. 3d convolutional neural networks for human action recognition. In *ICML*, 2010. 2

[19] Boyuan Jiang, MengMeng Wang, Weihao Gan, Wei Wu, and Junjie Yan. Stm: Spatiotemporal and motion encoding for action recognition. In *ICCV*, pages 2000–2009, 2019. 1

[20] Will Kay, Joao Carreira, Karen Simonyan, Brian Zhang, Chloe Hillier, Sudheendra Vijayanarasimhan, Fabio Viola, Tim Green, Trevor Back, Paul Natsev, Mustafa Suleyman, and Andrew Zisserman. The kinetics human action video dataset. *arXiv,* 1705.06950, 2017. 1, 2, 3, 4, 8

[21] Taeoh Kim, Jinhyung Kim, Minho Shim, Sangdoo Yun, Myunggu Kang, Dongyoon Wee, and Sangyoun Lee. Exploring temporally dynamic data augmentation for video recognition. *arXiv,* 2206.15015, 2022. 2, 6, 7

[22] Takumi Kobayashi. Analyzing filters toward efficient convnets. In *CVPR*, pages 5619–5628, 2018. 2

[23] Okan Köpüklü, Neslihan Kose, Ahmet Gunduz, and Gerhard Rigoll. Resource efficient 3d convolutional neural networks. In *ICCVW*, 2019. 2

[24] Yan Li, Bin Ji, Xintian Shi, Jianguo Zhang, Bin Kang, and Limin Wang. Tea: Temporal excitation and aggregation for action recognition. In *CVPR*, pages 909–918, 2020. 1

[25] Yi Li and Nuno Vasconcelos. Repair: Removing representation bias by dataset resampling. In *CVPR*, pages 9572–9581, 2019. 2, 4

[26] Ji Lin, Chuang Gan, and Song Han. Tsm: Temporal shift module for efficient video understanding. In *ICCV*, pages 7083–7093, 2019. 1

[27] Zhaoyang Liu, Limin Wang, Wayne Wu, Chen Qian, and Tong Lu. Tam: Temporal adaptive module for video recognition. In *ICCV*, pages 13708–13718, 2021. 1, 4

[28] Wenjie Luo, Yujia Li, Raquel Urtasun, and Richard Zemel. Understanding the effective receptive field in deep convolutional neural networks. In *NeurIPS*, pages 9446–9454, 2016. 2, 7, 8

[29] Zhaofan Qiu, Ting Yao, and Tao Mei. Learning spatio-temporal representation with pseudo-3d residual networks. In *ICCV*, pages 5534–5542, 2017. 2

[30] Gunnar A Sigurdsson, Olga Russakovsky, and Abhinav Gupta. What actions are needed for understanding human actions in videos? In *ICCV*, pages 2137–2146, 2017. 1

[31] Karen Simonyan and Andrew Zisserman. Two-stream convolutional networks for action recognition in videos. In *NeurIPS*, 2014. 1

[32] Karen Simonyan and Andrew Zisserman. Very deep convolutional networks for large-scale image recognition. *CoRR*, abs/1409.1556, 2014. 1, 5

[33] Lin Sun, Kui Jia, Dit-Yan Yeung, and Bertram E. Shi. Human action recognition using factorized spatio-temporal convolutional networks. In *ICCV*, pages 4597–4605, 2015. 2

[34] Christian Szegedy, Vincent Vanhoucke, Sergey Ioffe, and Jonathon Shlens. Rethinking the inception architecture for computer vision. In *CVPR*, pages 2818–2826, 2016. 2

[35] Du Tran, Lubomir Bourdev, Rob Fergus, Lorenzo Torresani, and Manohar Paluri. Learning spatiotemporal features with 3d convolutional networks. In *ICCV*, pages 4489–4497, 2015. 1, 2

[36] Du Tran, Heng Wang, Lorenzo Torresani, Jamie Ray1, Yann LeCun, and Manohar Paluri. A closer look at spatiotemporal convolutions for action recognition. In *CVPR*, pages 6450–6459, 2018. 2, 5

[37] Limin Wang, Zhan Tong, Bin Ji, and Gangshan Wu. Tdn: Temporal difference networks for efficient action recognition. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 1895–1904, 2021. 1

[38] 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 *ECCV*, pages 20–36, 2016. 1

[39] Saining Xie, Chen Sun, Jonathan Huang, Zhuowen Tu, and Kevin Murphy. Rethinking spatiotemporal feature learning: Speed-accuracy trade-offs in video classification. In *ECCV*, 2018. 2, 5, 8

[40] Sangdoo Yun, Seong Joon Oh, Byeongho Heo, Dongyoon Han, and Jinhyung Kim. Videomix: Rethinking data augmentation for video classification. *arXiv,* 2012.03457, 2020. 2, 6, 7

[41] Matthew D. Zeiler and Rob Fergus. Visualizing and understanding convolutional networks. In *ECCV*, pages 818–833, 2014. 2