Date of publication xxxx 00, 0000, date of current version xxxx 00, 0000.

Digital Object Identifier 10.1109/ACCESS.2017.DOI

Diverse Temporal Aggregation and Depthwise Spatiotemporal Factorization for Efficient Video Classification

YOUNGWAN LEE, HYUNG-IL KIM, (Member, IEEE), KIMIN YUN, JINYOUNG MOON[†]
Electronics and Telecommunications Research Institute (ETRI), Daejeon, Republic of Korea
[†]Corresponding author: Jinyoung Moon, jymoon@etri.re.kr (ORCID:0000-0002-6616-824X)

This work was supported by Institute of Information & Communications Technology Planning & Evaluation (IITP) grant funded by the Korean government (MSIT) (No. 2020-0-00004, Development of Previsional Intelligence based on Long-term Visual Memory Network and No. 2014-3-00123, Development of High Performance Visual BigData Discovery Platform for Large-Scale Realtime Data Analysis).

ABSTRACT

Video classification researches have recently attracted attention in the fields of temporal modeling and efficient 3D convolutional architectures. However, the temporal modeling methods are not efficient, and there is little interest in how to deal with temporal modeling in the 3D efficient architectures. To build an efficient 3D architecture for temporal modeling, we propose a new 3D backbone network, called VoV3D, that consists of a temporal one-shot aggregation (T-OSA) module and a depthwise factorized component, D(2+1)D. The T-OSA is devised to build a feature hierarchy by aggregating spatiotemporal features with different temporal receptive fields. Stacking this T-OSA enables the network itself to model short-range as well as long-range temporal relationships across frames without any external modules. We also design a depthwise spatiotemporal factorization module, D(2+1)D, that decomposes a 3D depthwise convolution into two spatial and temporal depthwise convolutions for efficient architecture. Through the proposed temporal modeling method (T-OSA) and the efficient factorization module (D(2+1)D), we construct two types of VoV3D networks: VoV3D-M and VoV3D-L. Thanks to its efficiency and effectiveness of their temporal modeling, VoV3D-L has $4\times$ fewer model parameters and $14\times$ less computation, surpassing the state-of-the-art TEA model on both Something-Something and Kinetics-400 datasets. We hope that VoV3D can serve as a baseline for efficient temporal modeling architecture.

INDEX TERMS Action recognition, Video classification, Temporal modeling, Efficient 3D CNN architecture, Spatial-temporal feature

I. INTRODUCTION

Recently, many works for video classification [1]–[8] have focused on an ability to model the temporal variation, dynamics of an action (*i.e.*, *visual tempo* [4]), called *temporal modeling* in literature. Unlike 2D image classification, video classification should distinguish visual tempo variation as well as its semantic appearance. In other words, appearance information alone is not sufficient to distinguish between *moving something up* and *down* or between *walking* and *running*, which requires to capture temporal variations. Thus, effectively modeling visual tempo is a key factor for video classification.

Previous works for temporal modeling [1], [2], [6], [9] utilize 2D CNN architecture due to its efficiency rather than 3D CNN architecture. They usually process per-frame inputs and aggregate the per-frame results to produce a final output through the temporal shift module [1] or the motion information embedding module [2], [3]. However, these methods depend heavily on the 2D ResNet backbone [10], which is neither lightweight nor efficient compared to state-of-the-art efficient 2D CNN models [3], [11], [12]. 3D CNN-based temporal modeling methods [4], [5] are also proposed to construct input frame-level pyramid [13] with different input frame rates or feature-level pyramid [4]. However, these methods require extra model capacity by adding a separate network path or a fusion module. In short, since previous works are add-on style modules on top of the backbone network, they are constrained under the backbone network.

Another research that has recently attracted attention for video understanding is to build an efficient 3D convolutional network architecture [14]–[16]. These works exploit 3D depthwise convolution to reduce model parameters and computations like 2D efficient CNN models [3], [11], [12], [17]–[19], which replace a convolution with a combination of depthwise and pointwise convolution. However, these 3D networks simply focused on the efficiency in terms of the computation of the building block and do not consider the efficiency of temporal modeling.

For addressing these issues, in this work, we propose an efficient and effective 3D architecture for temporal modeling, called VoV3D. The proposed VoV3D consists of temporal one-shot aggregation (T-OSA) building blocks, which are made of the proposed depthwise factorization module (*i.e.*, D(2+1)D). The T-OSA is devised to build a temporal feature hierarchy by aggregating features with different temporal receptive fields. As illustrated in Fig. 1, having diverse temporal receptive fields in one feature map is helpful to capture the visual tempo variation of an action. From this perspective, stacking the T-OSA enables the network itself to model short-range as well as long-range temporal relationships across frames without any external modules. Furthermore, inspired by the optimization benefit from kernel factorization [20], [21] and the efficiency of channel factorization [15], [16], we also design a depthwise spatiotemporal factorized module, called D(2+1)D. It decomposes a 3D depthwise convolution into *spatial* and *temporal* depthwise convolutions for making our network more lightweight and efficient. In practice, we have confirmed that combining the two factorization methods achieves better performance and efficiency than each one. Moreover, the efficiency of D(2+1)D allows our network to use more input frames (over 16 frames), which is advantageous for temporal modeling.

By using the proposed temporal modeling method, T-OSA, and the efficient factorized module, D(2+1)D, we construct two types of 3D CNN architectures, VoV3D-M and VoV3D-L models. In order to evaluate the proposed method in terms of modeling temporal variations, we validate VoV3D on the Something-Something dataset [22] which has been well-known to be challenging to classify an action due to the temporal complexity [1], [21], [23]. Moreover, we show the performance on Kinetics-400 dataset [24] to compare the proposed network to the state-of-the-arts. Thanks to its efficiency and effectiveness of the proposed temporal modeling mechanism, VoV3D-L (with 32 frames and Kinetics-400 pretrained) outperforms the state-of-the-art both 2D and 3D temporal modeling methods (*i.e.,* TEA [9] and SlowFast [13]), while having 4× and 5× fewer parameters and 14× and 5× less computation on Something-Something dataset [22]. Furthermore, the proposed VoV3D shows better temporal modeling ability than the state-of-the-art efficient 3D architecture, X3D [16] having comparable model capacity. We hope that the ideas contained within the proposed VoV3D can be widely used for other video architectures.

The main contributions of this work are summarized as below:

- We propose an effective temporal modeling method, Temporal One-Shot Aggregation (T-OSA) that can capture temporal variations by aggregating features having different temporal receptive fields.
- We propose an efficient depthwise factorized module, D(2+1)D that decomposes a 3D convolution into spatial and temporal depthwise convolutions, making T-OSA modules more accurate and efficient.
- We design an efficient 3D CNN architecture, VoV3D, based on the proposed T-OSA and D(2+1)D modules, which outperforms the state-of-the-arts in terms of both temporal modeling and efficiency.

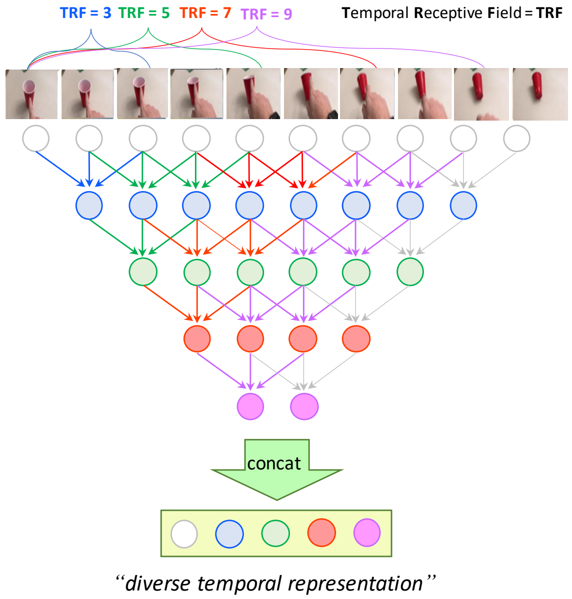

TRF = 3
TRF = 5
TRF = 7
TRF = 9
Temporal Receptive Field = TRF
concat
“diverse temporal representation”

FIGURE 1. **Illustration of the temporal receptive field.** The features having multiple temporal receptive fields are advantageous to capture visual tempo variation of an action.

II. RELATED WORKS

A. TEMPORAL MODELING FOR VIDEO CLASSIFICATION

Recent attempts for temporal modeling for video classification could be divided into two categories: 2D CNN-based and 3D CNN-based methods. 2D CNN-based methods such as TSN [6], TSM [1], STM [2] and TEA [9] prefer to use 2D CNN, *e.g.*, ResNet-50 as a backbone, due to its efficiency than 3D CNN models. They process per-frame inputs and aggregate these results to produce a final output on top of 2D ResNet. TSN [6] proposes to form a clip by sampling evenly from divided segments and this sparse sampling method becomes a common strategy for many works. TSM [1] is proposed to model temporal motion by utilizing memory shift operation along the temporal dimension. Since motion information is also an important cue for temporal modeling as a short-term temporal relationship, attempts to model feature-level motion features are proposed in STM [2] and TEA [9]. STM and TEA propose to differentiate between adjacent features for representing motion features and then add

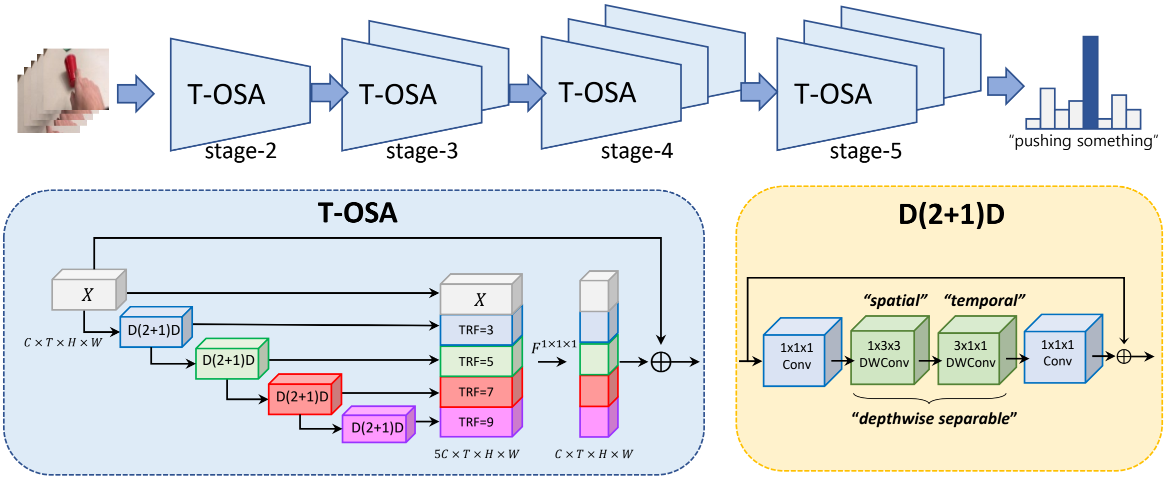


FIGURE 2. **VoV3D** has Temporal One-Shot Aggregation (T-OSA) building blocks. T-OSA consists of depthwise spatiotemporal factorized modules, D(2+1)D. Please refer to the details of the VoV3D architecture in Table. 2.

the spatiotemporal features and motion encoding together. TEA [9] also has a temporal aggregation module to capture long-range temporal dependency. However, TEA is based on 2D CNN features that are not jointly convolved along with spatial and temporal axis. This means that the interaction between spatial and temporal features is limited compared to 3D spatiotemporal methods.

For modeling various visual tempos using spatiotemporal 3D CNN, many works have been proposed by building an input frame-level pyramid [13], [25] or feature-level pyramid [4]. SlowFast [13] has two network inputs with different frame rates to capture different types of visual information, *e.g.*, semantic appearance or motion. DTPN [25] also uses a different sampling rate for arbitrary-length input video, which builds up the input frame-level hierarchy. Unlike these methods, TPN [4] leverages the feature hierarchy on top of the backbone network, instead of the input frame-level hierarchy by building a temporal feature pyramid network. In short, since temporal modeling methods are based on the existing backbone networks, *e.g.*, ResNet-50, they are constrained under the nature of the backbone network.

B. EFFICIENT 3D CNN ARCHITECTURE

Since channelwise separable convolution is densely exploited by efficient 2D CNN models [3], [11], [12], [17]–[19], [19], 3D CNN ones [14]–[16] based on the extended depthwise convolution have been explored. CSN [15] adopts 3D depthwise convolution into the residual bottleneck [10] by replacing the 3 × 3 × 3 convolution and adding a 1 × 1 × 1 convolution in front of the 3D depthwise convolution for interaction between channels. X3D [16] explores 3D CNN architecture along with spatial, temporal, depth, channel axis for maximizing the efficacy of the model. The depthwise bottleneck is also utilized as a key component in X3D, while X3D is progressively expanded from a lightweight to a large-scale model by scale-up all kinds of axes. As a result, X3D achieves state-of-the-art performance with a much smaller model capacity on various video classification datasets. However, this method focuses only on building an efficient network without considering temporal modeling. Therefore, we focus on building an efficient 3D CNN architecture as well as temporal modeling simultaneously.

III. PROPOSED METHOD

Temporal modeling (*i.e.*, capturing visual tempo variation) plays an important role in action recognition [1], [2], [4], [9], [13]. In particular, in the case of a video that lacks appearance variations of the features, video classification networks should rely heavily on temporal variations. Moreover, it is necessary to model long-term as well as short-term temporal relationships because short-term information is not sufficient to distinguish temporal variations such as *walking vs. running*. The conventional temporal modeling methods based on 3D CNN try to model the visual tempo through the input frame-level [13], [25] or feature-level pyramids [4]. However, these methods have to add separate networks on top of the existing 3D backbone as an external (*i.e.*, plug-in) module, which requires more parameters and computations. To address these challenges, in this paper, we aim to propose an efficient video backbone network having temporal modeling ability by itself without external modules. To this end, we design a new 3D CNN architecture inspired by VoVNet [26], [27] that represents hierarchical and diverse spatial features at a small cost.

First, we propose an effective temporal modeling method, named Temporal One-Shot Aggregation (T-OSA). For making a network efficient, we also devise a depthwise spatiotemporal factorization method, D(2+1)D. Lastly, we design a new video classification network, called VoV3D, which consists of the proposed T-OSA and D(2+1)D.

A. TEMPORAL ONE-SHOT AGGREGATION (T-OSA)

VoVNet [26], [27] is a computation and energy-efficient 2D CNN architecture devised to learn diverse feature representations by stacking One-Shot-Aggregation (OSA) modules. The OSA module consists of successive 3×3 convolutions and aggregates those feature maps into one feature map at once in a concatenate manner, followed by a 1×1 convolution. The OSA allows the network to represent diverse features by capturing multiple receptive fields in one feature map, which results in the effect of the feature pyramid. Due to the diverse feature representation power of OSA, VoVNet outperforms ResNet [10] and HRNet [28] in object detection and segmentation tasks that require more complex representation.

Inspired by the spatial feature's hierarchy of OSA in VoVNet, we propose temporal one-shot aggregation, called T-OSA, to capture multiple temporal receptive fields in one 4D feature map, as illustrated in Fig. 2. In detail, the i-th 2D convolution $F_i^{3\times3}$ (3×3 `2DConv`) can be replaced with $F_i^{t\times3\times3}$ ($t \times 3 \times 3$ `3DConv`) for $i \in \{1, 2, ..., n\}$ where t is the temporal kernel size and n is the number of $t \times 3 \times 3$ 3D convolutions in T-OSA. It is noted that we keep temporal dimension T (frames) for feature aggregation. Each feature map $\mathrm{X}_i \in \mathbb{R}^{C\times T\times H\times W}$ that is the result from $F_i^{t\times3\times3}$ has progressively increasing temporal receptive field due to its successive connection. For example, if the temporal receptive field (TRF) of the feature map X_1 is 3 and temporal kernel size t is 3, the TRF of the next X_2 is 5. Thus, once the features are concatenated in channel-axis, the aggregated feature map $\mathrm{X}_{agg} \in \mathbb{R}^{(n+1)C\times T\times H\times W}$ comprised of $\{\mathrm{X}_{in}, \mathrm{X}_1, ..., \mathrm{X}_n\}$ has diverse *temporal* and *spatial* receptive fields in one feature map, where $\mathrm{X}_{in} \in \mathbb{R}^{C\times T\times H\times W}$ is the input feature and n is set to 4 in Fig. 2. Then, a $1 \times 1 \times 1$ 3D convolution is followed for reducing channel size $(n + 1)C$ to C and the residual connection is added to the final feature map. Thus, stacking T-OSA allows the network to have various temporal receptive fields, enabling the model to capture not only short-range but also long-range temporal dependency across frames, which has a similar effect with feature pyramid in the same spatial feature space.

In practice, simply expanding 2D VoVNet to 3D CNN architecture is limited in terms of optimization because 3D CNN models have additional parameter space along with temporal-axis and thus need optimization strategy. Therefore, we elaborate the T-OSA with additional design choices for the adaptation of OSA in 3D temporal feature space. While the OSA module in 2D VoVNet uses only 3×3 `2DConv`, the proposed T-OSA adopts 3D bottleneck architecture [15], [16], [29] (*e.g.*, $1 \times 1 \times 1$ `3DConv`, $3 \times 3 \times 3$ `3DConv`, $1 \times 1 \times 1$ `3DConv`) with more non-linearity operations. As a 3D bottleneck architecture, we propose D(2+1)D in the next section. Also, we add an inner residual connection to facilitate optimization.

B. DEPTHWISE SPTAIOTEMPORAL FACTORIZATION

There are two types of factorization concept on 3D convolution (`3DConv`): 1) Depthwise (or Channelwise) [14]–[16] and 2) Kernelwise [20], [21], [30] methods. Inspired by efficient 2D image classification network [3], [11], [12], [17]–[19], [31], depthwise separable convolution is also mainly used as a key building block for efficient video backbone networks [14]–[16]. 3D depthwise separable convolution (`3DWConv`) is utilized to factorize a `3DConv` into a $t \times k \times k$ depthwise `3DConv` followed by $1 \times 1 \times 1$ pointwise `3DConv`. CSN [15] adds a $1 \times 1 \times 1$ `3DConv` in front of the `3DWConv` for preserving the interaction between channels, which results in improving accuracy. Tran *et.al.* [15] found that the `3DWConv` has two advantages: 1) significant reduction of parameters and computational cost (FLOPs) without sacrificing accuracy, 2) regularization effect. In addition to channel factorization, kernel factorization also has been widely used in [20], [21], [30] for curtailing computation and boosting accuracy. The kernel factorization is also called spatiotemporal factorization as it is decomposed into a $1 \times k \times k$ spatial convolution (space) followed by a $t \times 1 \times 1$ temporal convolution (time) as shown in Fig. 3 (top).

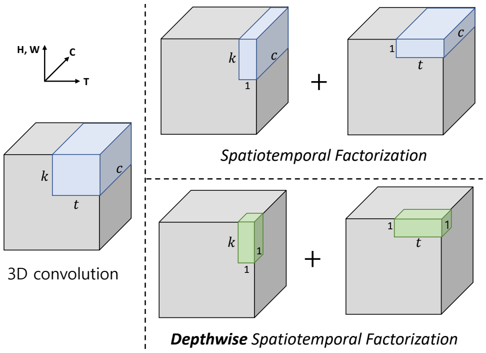

H, W
C
T
k
c
1
+
t
Spatiotemporal Factorization
k
c
t
3D convolution
k
1
1
+
1
1
t
Depthwise Spatiotemporal Factorization

FIGURE 3. **Depthwise spatiotemporal factorization.** k, t, c denote the size of the spatial kernel, temporal kernel, and channel in 3D convolution, respectively. Compared to Spatiotemporal factorization (top) as in R(2+1)D [20], depthwise spatiotemporal factorization (bottom) in the proposed D(2+1)D further decomposes the features along the channel axis, which improves the efficiency of the network.

Our motivation lies in the fusion of these two factorization methods for realizing an efficient video classification network. We design a depthwise spatiotemporal factorization module, D(2+1)D, that decomposes a `3DWConv` into a *spatial* `DWConv` and a *temporal* `DWConv` as shown in Fig. 3 (bottom). We analyze each resource requirement of models in Table. 1 illustrating the number of parameters and computation (FLOPs) of a `3DConv` in the middle of bottleneck architecture. The input tensor of the `3DConv` has $C \times T \times H \times W$ shape, where T and C are the numbers of frames and channels, and H, W is the size of height and width, respectively. Assuming the number of filters (output channel) is the same (C), the 3D filter has $t \times k \times k$ kernel size, where t, k denote temporal and spatial kernel, respectively. As demonstrated in Table. 1, compared to the basic bottleneck `3DConv` in (a), `3DWConv` in (c) is $C\times$ more efficient because it has only one sub-filter for the input tensor as illustrated in Fig. 3. We design two types of factorized modules based on the order of spatial and temporal dimensions: D(1+2)D and D(2+1)D. It is noted that spatial down-sampling is operated in the spatial convolution and the temporal convolution keeps temporal dimension. Compared

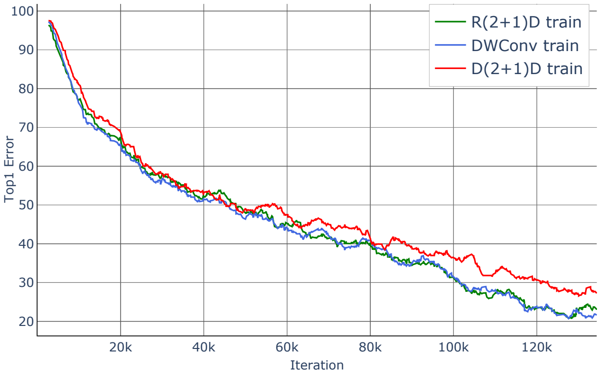


FIGURE 4. Training top-1 error on Something-Something v1. D(2+1)D shows higher training error, but lower testing error (compare validation accuracies in Table. 4). This suggests D(2+1)D yields regularization, preventing overfitting.

Type	Param.	FLOPs
(a) bottleneck	C^2tk^2	$C^2tk^2(HWT)/s^2$
(b) R(2+1)D	$C^2(t+k^2)$	$C^2(t+k^2)(HWT)/s^2$
(c) dw-bottleneck	Ctk^2	$Ctk^2(HWT)/s^2$
(d) D(1+2)D	$C(t+k^2)$	$C(s^2t+k^2)(HWT)/s^2$
(e) D(2+1)D	$C(t+k^2)$	$C(t+k^2)(HWT)/s^2$

TABLE 1. Comparison of parameters and computation. This table considers only a 3D convolution located in the middle of the bottleneck. t, k, and s denote temporal, spatial kernel size, and stride, respectively. C, H, W, T denote channel, height, width, the number of frames in the input 3D feature map, assuming input/output channel size is same.

with 3DWConv in (c), both D(1+2)D and D(2+1)D have about one order of magnitude fewer parameters and computations. In comparison between the two factorized modules, an important difference arises in spatial down-sampling. The number of parameters is the same, while the computation cost is different due to different spatial sizes. Specifically, for D(1+2)D, the temporal DWConv is operated first with $C \times T \times H \times W$ input tensor followed by the spatial DWConv with stride s. It is summarized as:

$$\begin{aligned} \text{FLOPs} &= Ct \times HWT + Ck^2 \times HWT/s^2 \\ &= (s^2t + k^2)CHWT/s^2. \end{aligned} \tag{1}$$

For D(2+1)D, since spatial DWConv with down-sampling goes ahead, the temporal DWConv operates the spatially down-sized input tensor, which results in reducing overall computation. This is summarized as:

$$\begin{aligned} \text{FLOPs} &= Ct \times HWT/s^2 + Ck^2 \times HWT/s^2 \\ &= (t + k^2)CHWT/s^2. \end{aligned} \tag{2}$$

In Fig. 4, we observed that D(2+1)D has higher training error, but better validation accuracy (see Table. 4) than R(2+1)D and DWConv, which is the similar phenomenon as in CSN [15]. Combining the spatiotemporal factorization with depthwise convolution yields better regularization effect, preventing overfitting. Therefore, we expect that the D(2+1)D can be widely used for other 3D CNN architectures to boost their performances. For example, We have confirmed the effect through the combination of the state-of-the-art method (*i.e.*, X3D [16]) and our D(2+1)D, which will be described in the experimental section.

Stage	VoV3D-M (L)	output size $T \times H \times W$
conv1	$1 \times 3^2, 5 \times 1^2, 24$	$T \times 112 \times 112$
T-OSA2	$\begin{bmatrix} \text{(D(2+1)D, 40(48))}\times 5 \\ 1 \times 1^2, 24 \end{bmatrix} \times 1(1)$	$T \times 56 \times 56$
T-OSA3	$\begin{bmatrix} \text{(D(2+1)D, 80(96))}\times 5 \\ 1 \times 1^2, 48 \end{bmatrix} \times 1(2)$	$T \times 28 \times 28$
T-OSA4	$\begin{bmatrix} \text{(D(2+1)D, 160(192))}\times 5 \\ 1 \times 1^2, 96 \end{bmatrix} \times 2(5)$	$T \times 14 \times 14$
T-OSA5	$\begin{bmatrix} \text{(D(2+1)D, 320(384))}\times 5 \\ 1 \times 1^2, \text{160(192)} \end{bmatrix} \times 2(3)$	$T \times 7 \times 7$
conv5	1×1^2, 320(384)	$T \times 7 \times 7$
pool5	$T \times 7 \times 7$	$1 \times 1 \times 1$
fc1	1×1^2, 2048	$1 \times 1 \times 1$
fc2	1×1^2 #classes	$1 \times 1 \times 1$

TABLE 2. VoV3D architectures: VoV3D-M and VoV3D-L. T denotes the number of input frames. VoV3D has two types of models: VoV3D-M and VoV3D-L. They are comprised of Temporal One-Shot Aggregation (T-OSA) building blocks made of D(2+1)D modules.

Model	T-OSA	D(2+1)D	Top-1	Top-5
Baseline (M)			46.4	75.3
	✓		48.0 +2.4	76.7 +1.4
		✓	48.5 +2.3	76.9 +1.6
	✓	✓	49.0 +2.6	78.2 +2.9
Baseline (L)			47.1	76.5
	✓		48.9 +1.8	77.6 +1.1
		✓	48.8 +1.7	77.4 +0.9
	✓	✓	49.6 +2.5	78.1 +1.6

TABLE 3. Contributions of the proposed components in VoV3D on Something-Something V1.

C. VOV3D ARCHITECTURE

Finally, we construct an efficient 3D CNN architecture, VoV3D, that can model various visual tempos effectively with the proposed T-OSA and D(2+1)D modules. We design two types of lightweight models: VoV3D-M and VoV3D-L which have only 3.3M and 5.8M parameters, respectively. VoV3D is comprised of the proposed T-OSA blocks which have five D(2+1)D modules followed by a $1 \times 1 \times 1$ 3DConv. In the stage level (same spatial resolution), VoV3D has multiple T-OSAs (*e.g.*, 5), in series, which leads to representing diverse temporal features. conv1 is also the (2+1)D style-convolution where 1×3^2 spatial 3DConv is operated and followed by a 5×1^2 temporal 3DConv. Following [16], we also add a channel attention module, SE block [32], into the D(2+1)D with reduction ratio of 1/16. The efficient D(2+1)D allows VoV3D to reduce significant computation cost, so it can use longer frames ($\geq$16) to capture longer visual tempo. The details are illustrated in Table. 2.

IV. EXPERIMENTS

A. DATASETS

We validate the proposed VoV3D on Something-Something (SSv1 & v2) [22] and Kinetics-400 [24]. In contrast to

VoV3D-M	Param.	GFLOPs	Top-1	Top-5
(a) bottleneck [29]	42.9M	103.2 × 6	48.6	76.8
(b) R(2+1)D [20]	20.9M	48.9 × 6	48.6	77.6
(c) dw-bottleneck [15], [16]	3.3M	7.0 × 6	48.0	76.7
(d) D(1+2)D (**ours**)	3.2M	6.5 × 6	48.0	77.2
(e) D(2+1)D (**ours**)	**3.2M**	**6**.4 × 6	**49.0**	**78.2**
X3D-M [16]	3.3M	6.1 × 6	46.4	75.3
X3D-M [16] w/ D(2+1)D	3.2M	5.8 × 6	47.4	75.9

TABLE 4. **Comparison to different bottleneck architectures** on Something-Something V1.

Kinetics-400 [24] that is less sensitive to visual tempo variations, SSv1 & v2 [22] is focused on human-object interaction which requires a more temporal relationship than appearance [1], [21], [23]. Since SSv1 & v2 is widely used as a benchmark for evaluating the effectiveness of temporal modeling, the effectiveness of the proposed VoV3D is mainly investigated for this dataset. SSv1 [22] contains 108k videos with 174 categories, and the second release (v2) of the dataset is increased to 220k videos. Kinetics-400 [24] includes 400 categories and provides download URL links over 240k training and 20k validation videos. Because of the expiration of some YouTube links, we collect 234,619 training and 19,761 validation videos. For a fair comparison with X3D, we train X3D and VoV3D on the same Kinetics-400 collected by ourselves.

B. IMPLEMENTATION DETAILS

Training. Our models are trained *from scratch* without using ImageNet [33] pretrained model unless specified. For SSv1 & v2 [22] dataset, we use segment-based input frame sampling [1], which splits each video into N segments and picks one frame to form a clip (N frames) from each segment. We note that thanks to the memory-efficient VoV3D, our model can be trained with more input frames, *e.g.*, from 16 to 32. For Kinetics-400 [24], we sample 16 frames with a temporal stride of 5 as [16]. We apply the random cropping of 224×224 pixels from a clip and random horizontal flip with a shorter side randomly sampled in [256, 320] pixels [13], [16], [34], [35] for VoV3D-M and VoV3D-L models. In the case of SSv1 & v2, it requires discriminating between directions, so the random flip is not applied. Following [13], [16], we use the same parameters for training SSv1 & v2: SGD optimizer, 100 epochs, mini-batch size 64 (8 clips per a GPU), initial learning rate 0.1, half-period cosine learning rate schedule [36], linear warm-up strategy [37], and weight decay $5{\times}10^{-5}$. Following [1], [38], we also fine-tune VoV3D using Kinetics-400 pretrained model. We use a linear warm-up [37] for 2k iterations from 0.0001 and a weight decay of 5×10^{-5}. We finetune the model for 50 epochs with a base learning rate of 0.05 decreased at 35 and 45 epoch by 0.1 and use sync bathcnorm. For Kinetics-400, we use the same training parameters except for 256 epochs and mini-batch size 128. We train all models using a 8-GPU machine and implementation is based on `PySlowFast` [5].

Model	#F	GFLOPs	From scratch		K-400 finetune	
			Top-1	Top-5	Top-1	Top-5
X-M [16]	16	6.1×6	46.4	75.3	51.2	78.9
V-M	16	6.4×6	**49.0**	**78.2**	**52.4**	**80.3**
X-M [16]	32	12.3×6	48.9	77.6	51.5	79.6
V-M	32	12.8×6	**50.1**	**79.2**	**53.3**	**81.2**
X-L [16]	16	11.9×6	47.1	76.5	50.8	79.3
V-L	16	12.1×6	**49.6**	**78.1**	**53.4**	**81.4**
X-L [16]	32	23.9×6	48.4	77.8	52.6	81.2
V-L	32	24.3×6	**50.7**	**78.8**	**54.7**	**82.0**

TABLE 5. **Comparison to X3D on Something-Something V1.** #F denotes the number of input frames. X and V denote X3D and VoV3D, respectively. For model parameters, X3D-M and VoV3D-M have 3.3M respectively and X3D-L and VoV3D-L have 5.6M and 5.8M, respectively.

To compare VoV3D-M/L to the strong state-of-the-art X3D [16], we also train X3D-M/L having similar parameters and FLOPs with the same training protocols. Note that for X3D-L, unlike origin X3D paper [16], we use the same spatial sample size [256, 320], not [356, 446]. The reason why we invest computation budget to more input frames ($\geq$16) is that the SSv1 & v2 [22] requires more temporal modeling than appearance information.

Inference. Following common practice in [1], [5], [16], [35], we sample multiple clips per video (*e.g.*, 10 for Kinetics and 2 for SSv1 & v2). We scale the shorter spatial side to 256 pixels and take 3 crops of 256×256, as an approximation of fully-convolutional testing [35] called full resolution image testing in TSM [1]. Then, we average the softmax scores for prediction.

C. ABLATION STUDY

In order to verify the effectiveness of the proposed method in terms of temporal modeling, we conduct ablation studies on SSv1 [22] that requires more temporal modeling ability [1], [21], [23] than Kinetics-400.

Component contributions. We study the effect of the individual component of VoV3D and results are shown in Table. 3. We use X3D as a baseline and T-OSA without D(2+1)D consists of the same depthwise bottleneck as X3D. T-OSA boosts performance by large margins in both M and L models, demonstrating the diverse temporal representation of T-OSA improves temporal modeling capability. D(2+1)D also achieves higher accuracy, which suggests that the factorization of spatial and temporal features helps the network to optimize easily.

Comparison with the different bottleneck. We compare the proposed depthwise spatiotemporal factorization module (*i.e.*, D(2+1)D) with other architectures [13], [15], [20] in Table. 4. We alternatively plug the bottleneck architectures into the T-OSA. While R(2+1)D [20] reduces both parameters and GFLOPs with higher accuracy than the standard bottleneck [29] in (a), the depthwise bottleneck [15], [16] in (c) also significantly reduces the computations but obtains lower performance than R(2+1)D. However, both D(1+2)D

Model	Backbone	Pretrain	Frame	Param (M)	GFLOPs	Something V1		SomethingV2	
						Top-1	Top-5	Top-1	Top-5
TSM [1]	ResNet-50	Kinetics-400	16	24.3	33 × 6	47.2	77.0	63.0	88.1
TSM [1]	ResNet-101	Kinetics-400	8	24.3	65 × 6	48.7	77.2	63.2	88.2
TSM+TPN [4]	ResNet-50	ImageNet	8	N/A	N/A	50.7	-	64.7	-
STM [2]	ResNet-50	ImageNet	16	N/A	67 × 30	50.7	80.4	64.2	89.8
TEA [9]	ResNet-50	ImageNet	8	24.4	35 × 30	51.7	80.5	-	-
TEA [9]	ResNet-50	ImageNet	16	24.4	70 × 30	**52.3**	**81.9**	**65.1**	**89.9**
NL-I3D+GCN [39]	3D ResNet-50	Kinetics-400	32	N/A	303 × 6	46.1	76.8	-	-
SlowFast 16 × 8, R50 [13]	-	Kinetics-400	64	34.0	131.4 × 6	-	-	63.9	88.2
ip-CSN-152 [15]	-	-	32	29.7	74.0 × 10	49.3	-	-	-
ViT-B-TimesSformer [40]	ViT-B [41]	ImageNet-21K	8	121.4	1703 × 3	-	-	62.5	-
MViT-B, 32 × 3 [42]	-	Kinetics-400	32	36.6	170.0 × 3	-	-	**67.1**	**90.8**
X3D-M [16]	-	-	16	3.3	6.1 × 6	46.4	75.3	63.1	88.0
VoV3D-M	-	-	16	3.2	6.4 × 6	49.0	78.2	63.6	88.6
VoV3D-M	-	-	32	3.2	12.8 × 6	49.8	78.0	64.3	88.9
VoV3D-M	-	Kinetics-400	32	3.2	12.8 × 6	**53.2**	**81.1**	**65.8**	**89.6**
X3D-L [16]	-	-	16	5.6	12.0 × 6	47.1	76.5	62.7	87.8
VoV3D-L	-	-	16	5.8	12.1 × 6	49.5	78.0	64.5	88.7
VoV3D-L	-	-	32	5.8	24.3 × 6	50.7	78.8	65.9	89.6
VoV3D-L	-	Kinetics-400	32	5.8	24.3 × 6	**54.7**	**82.0**	**67.4**	**90.5**

TABLE 6. **Comparison with the state-of-the-art architectures on Something-Something V1& V2 validation set.** Note that Something-Something dataset requires more temporal relationship than Kinetics-400 [24] (appearance-oriented). For fair comparison, X3D and VoV3D are trained with the same training protocols on `PySlowFast` [5].

and D(2+1)D achieve better accuracy with less computation than dw-bottleneck in (c). In particular, D(2+1)D outperforms all other architectures with a minimum computation and model size. In addition, we also investigate the effect of D(2+1)D by replacing dw-bottleneck with D(2+1)D in X3D. As a result, D(2+1)D improves 1%p Top-1 accuracy gain while reducing model parameters and GFLOPs.

Comparison to X3D under various conditions. We compare VoV3D with X3D under the following conditions: the number of input frames (#F in Table. 5) and whether a backbone is pre-trained with Kinetics-400 or not. We train VoV3D and X3D with 16 and 32 input frames from scratch or using Kinetics-400 pretraining. Table 5 summarizes the results. We can find that using more frames boosts performance in both VoV3D and X3D and VoV3D consistently outperforms X3D. This demonstrates that using more frames helps the networks to capture visual tempo variation and the ability of the proposed T-OSA to represent diverse temporal receptive fields enables VoV3D to yield better temporal modeling than X3D.

D. COMPARISON TO STATE-THE-OF-ART

Results on Something-Something. We validate the efficiency and effectiveness of the proposed VoV3D on SSv1 & v2 requiring more temporal modeling ability than spatial appearance. Table 6 shows the results and resource budgets of other methods: temporal modeling based on 2D CNN methods [1], [2], [9] and 3D CNN architectures [4], [5], [15], [16], [39]. First, under the same input frames (*e.g.*, 16 frames), VoV3D-M/L consistently outperforms X3D-M/L with a comparable model budget on both SSv1 & v2. In particular, the performance gain of 'L' models is bigger than 'M' models. This result demonstrates that stacking the proposed T-OSAs makes it better to model temporal dependency across frames. Qualitative comparison is also illustrated in Fig. 6.

Compared to the representative temporal modeling 2D CNN method, TSM [1] based ResNet-101, VoV3D-M with 16 frames achieves higher accuracy while it requires much fewer parameters (8×) and GFLOPs (10×), even without pretraining. Furthermore, the performance of VoV3D-L with 32 frames pretrained on Kinetics-400 surpasses that of the best model among 2D CNN methods, TEA [9] by a large margin (2.4% / 2.3% @Top-1) on both SSv1 & v2, while having about 14× fewer computation. These results break the prejudice that 3D CNN architectures require an expensive computation budget than 2D CNN. We also note that VoV3D architecture alone shows sufficient performance and efficiency than the add-on style temporal modeling methods on top of 2D backbone networks [1], [2], [4], [9]. It shows that VoV3D can serve as a strong baseline for temporal modeling.

VoV3D is also superior to those 3D CNN-based temporal modeling methods, such as SlowFast [13] and CSN [15]. Even without Kinetics-pretraining, VoV3D-M with 32 frames achieves higher accuracy than SlowFast pretrained on Kinetics-400 with 11× more model parameters. It demonstrates that a 3D single network path is enough to model visual tempo variations. Although CSN [15] contains the depthwise bottleneck architecture, its accuracy is lower than that of VoV3D-M. This result shows that the proposed T-OSA plays an important role in temporal modeling.

Since attention-based methods [41], [43], [44] have ad-

Method	Pre	#F	P (M)	GFLOPs	Top-1	Top-5
I3D [45]	IN	64	12	108 × N/A	71.1	90.3
Nonlocal R50 [35]	IN	32	35.3	282 × 30	76.5	92.6
TSM [1]	IN	16	24.3	65 × 30	74.7	-
STM [2]	IN	16	N/A	67 × 30	73.7	91.6
TEA R50 [9]	IN	16	24.4	70 × 30	76.1	92.5
R(2+1)D [20]	-	16	63.6	152 × 115	72.0	90.0
SlowFast 4×16, R50 [13]	-	32	34.4	36.1 × 30	75.6	92.1
ip-CSN-152 [15]	-	32	32.8	109 × 30	**77.8**	92.8
X3D-M [16]	-	16	3.8	6.2 × 30	75.1	92.2
X3D-L [16]	-	16	6.1	9.1 × 30	76.1	92.6
VoV3D-M	-	16	3.7	6.4 × 30	74.7	92.1
VoV3D-L	-	16	6.2	9.3 × 30	76.3	**92.9**

TABLE 7. **Comparison with the state-of-the-art architectures on Kinetics-400.** IN, #F, P denote ImageNet pretraining, the number of frames and parameters, in respectively. Note that both VoV3D and X3D are trained with the same training protocols on the same environment such as GPU server, training set, and scale size [256, 320] and implemented on `PySlowFast` [5].

vantages of modeling long-range dependency inherently, We also compare VoV3D with the recent vision transformer based methods such as TimeSformer [40] and MViT [42]. VoV3D outperforms TimeSformer, while having 20× less model parameter and requiring 35× computational cost. The reason TimeSformer has lower performance than VoV3D is that TimeSformer utilizes ViT-B that exploits only single-scale feature map (14×14) instead of multi-scale features as in VoV3D and MViT. Besides, VoV3D achieves similar performance compared to MViT-B (32x3) using 32 input frames, showing better efficiency in terms of model parameters (6×) and GFLOPs (3×), respectively. The reason why VoV3D performs better than the transformer-based methods is that VoV3D can model not only long-range redundancy but also local connectivity. Therefore, these results demonstrate that the capability of modeling both long-range dependency and local connectivity is necessary for temporal modeling.

Results on Kinetics-400. We also compare VoV3D to other state-of-the-art methods on Kinetics-400. VoV3D-L achieves 76.3%/92.9% Top-1/5 accuracy, and it shows better performance than the state-of-the-art temporal modeling 2D method, TEA [9], even without ImageNet pretraining. VoV3D-L also surpasses 3D temporal modeling methods, SlowFast [13] 4 × 16 based on ResNet-50 while having about 5× and 4× fewer model parameters and FLOPs, respectively. Compared to ip-CSN-152 [15] as an efficient 3D CNN, VoV3D-L shows slightly lower Top-1 accuracy, but it achieves higher Top-5 accuracy with much less model capacity. While VoV3D-M shows comparable accuracy with X3D-M, VoV3D-L achieves higher Top-1/Top-5 accuracy.

E. ROBUSTNESS ANALYSIS TO TEMPORAL VARIATION

Inspired by TPN [4], we investigate the robustness to temporal variation of VoV3D and X3D. VoV3D-M and X3D-M are trained with the same sampling rate (temporal stride τ) of 5 on Kinetics-400. At the test phase, we measure the top-1 accuracy drop depending on the change of the sampling rate (*e.g.*, $\tau \in \{5, 8, 10, 12, 14, 16\}$) used for adjusting the visual tempo of a given action instance. The accuracy drop is used for measuring the robustness to temporal variations.

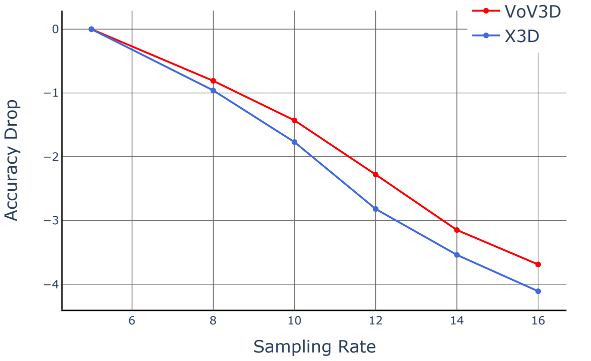


FIGURE 5. **Robustness to temporal variation.** Changing sampling rates (or temporal stride) induces temporal variation. Compared to X3D, VoV3D is more robust to temporal variation due to its temporal modeling ability of T-OSA.

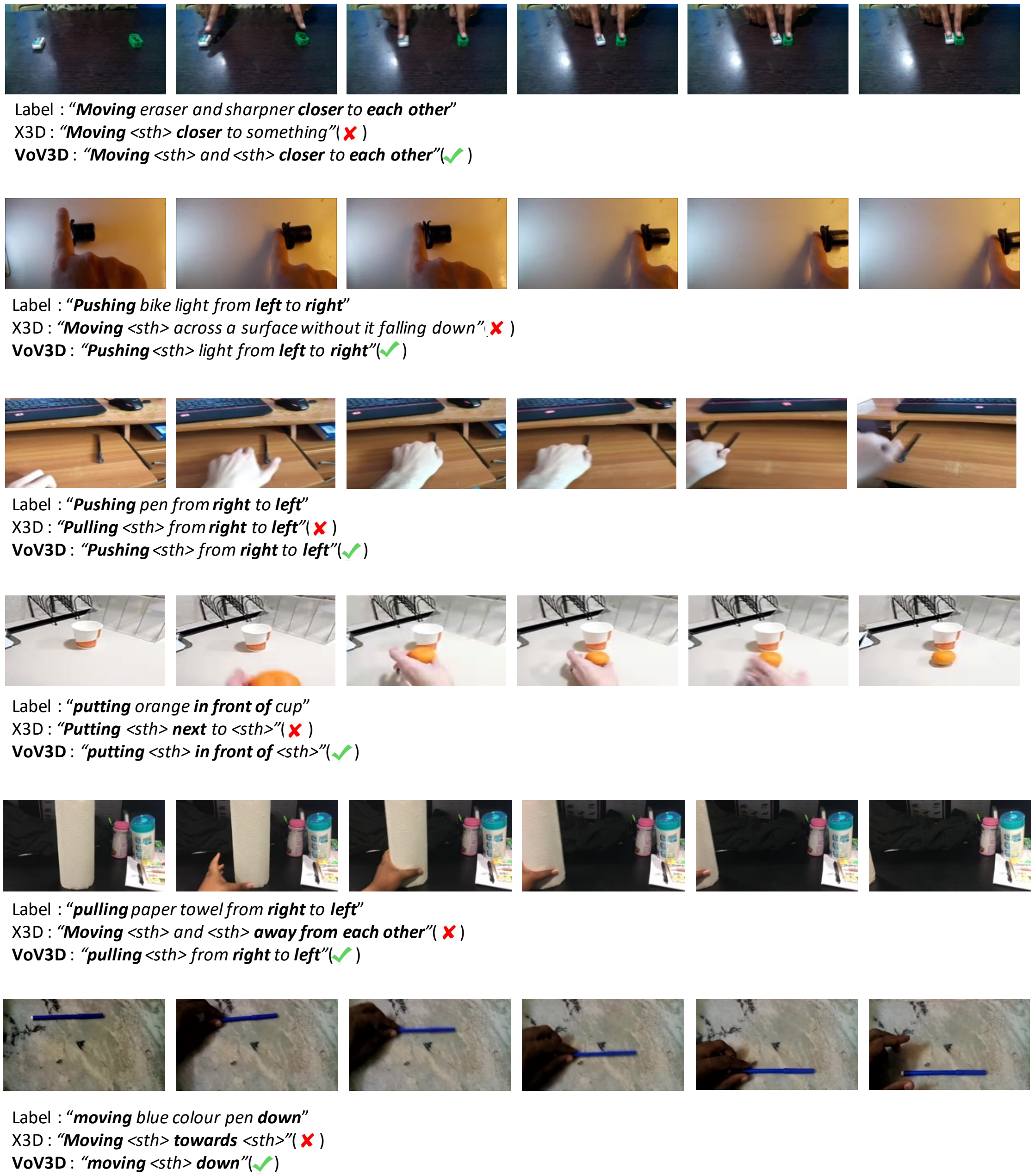
Method	Param.	GPU Memory	sec/video	SSv2 Top-1
R(2+1)D [20]	20.9M	5,952MB	0.030	48.6
dw-bottleneck [16]	3.3M	4,604MB	0.013	48.0
D(2+1)D (**ours**)	3.2M	4,762MB	0.014	49.0
SlowFast-R50 [13]	34.0M	20,284MB	0.088	63.9
X3D-L-32 [16]	5.6M	10,462MB	0.035	62.7
MViT-32 [42]	36.6M	22,038MB	0.055	67.1
VoV3D-L-32 (**ours**)	5.8M	11,974MB	0.039	67.4

TABLE 8. **Model budget comparison.** (b), (c), (e) denote the same models in Table 3. sec/video means GPU runtime. GPU memory and runtime are measured during inference with 16 batch on one V100 GPU (CUDA 10.1 & pytorch 1.6). For a fair comparison, we test all models in the same codebase, PySlowFast.

Fig. 5 shows the accuracy drop curves of varying visual tempos for VoV3D and X3D. When changing the sampling rate, VoV3D shows less accuracy drop than X3D, which supports the fact that VoV3D is more robust to temporal variation and thus has a better ability to model temporal relationships across frames than X3D.

F. MODEL CAPACITY ANALYSIS

In Table 8, we compare model capacity of D(2+1)D with R(2+1)D [20] and depthwise-bottleneck [16] in terms of model parameter, total feature sizes (i.e., Activations), and GPU memory. (b), (c), (e) models are the same ones in Table 3. Compared with R(2+1)D, D(2+1)D consumes less GPU memory even with similar feature sizes due to its much smaller model size, allowing VoV3D and X3D to use longer input frames. Moreover, D(2+1)D shows two times better efficiency than R(2+1)D in terms of inference time. We also compare our VoV3D-L-32 comprised of the efficient D(2+1)D modules with state-of-the-art methods including CNN-based X3D [16] and SlowFast-R50 [13] and the recent Vision transformer-based MViT-32 [42]. Compared to SlowFast-R50, VoV3D consumes two times less GPU memory while running two times faster. Showing faster inference time, VoV3D also requires fewer model parameters and less GPU memory than MViT. These results demonstrate VoV3D based on 3D CNN can be widely used due to its efficacy and effectiveness.



Label : "*Moving eraser and sharpner closer to each other*"
X3D : "*Moving <sth> closer to something*"(✗)
VoV3D : "*Moving <sth> and <sth> closer to each other*"(✓)

Label : "*Pushing bike light from left to right*"
X3D : "*Moving <sth> across a surface without it falling down*"(✗)
VoV3D : "*Pushing <sth> light from left to right*"(✓)

Label : "*Pushing pen from right to left*"
X3D : "*Pulling <sth> from right to left*"(✗)
VoV3D : "*Pushing <sth> from right to left*"(✓)

Label : "*putting orange in front of cup*"
X3D : "*Putting <sth> next to <sth>*"(✗)
VoV3D : "*putting <sth> in front of <sth>*"(✓)

Label : "*pulling paper towel from right to left*"
X3D : "*Moving <sth> and <sth> away from each other*"(✗)
VoV3D : "*pulling <sth> from right to left*"(✓)

Label : "*moving blue colour pen down*"
X3D : "*Moving <sth> towards <sth>*"(✗)
VoV3D : "*moving <sth> down*"(✓)

FIGURE 6. **Qualitative comparison with X3D on Something-Something datasets.** We compare the proposed VoV3D with X3D qualitatively on something-something dataset which requires more temporal modeling ability than spatial appearance (e.g., Kinetics-400). As shown these examples, VoV3D distinguishes directions and interactions between objects and humans rather than X3D, which demonstrates the proposed Temporal One-Shot-Aggregation method effectively models temporal relationships between frames.

V. CONCLUSION

We have proposed a simple yet effective temporal modeling 3D architecture, VoV3D, that consists of Temporal One-Shot Aggregation (T-OSA) and depthwise spatiotemporal factorized module, D(2+1)D. The T-OSA is able to effectively model temporal variations by aggregating features having different temporal receptive fields. The D(2+1D) module decomposes 3D depthwise convolution into a spatial and temporal depthwise convolution, which makes the proposed VoV3D significantly efficient and boosts performance. We expect the proposed VoV3D and its components to be widely used in other video applications.

REFERENCES

[1] J. Lin, C. Gan, and S. Han, "Tsm: Temporal shift module for efficient video understanding," in *Proc. ICCV*, 2019.

[2] B. Jiang, M. Wang, W. Gan, W. Wu, and J. Yan, "Stm: Spatiotemporal and motion encoding for action recognition," in *Proc. ICCV*, 2019.

[3] M. Tan, B. Chen, R. Pang, V. Vasudevan, and Q. V. Le, "Mnasnet: Platform-aware neural architecture search for mobile," *arXiv preprint arXiv:1807.11626*, 2018.

[4] C. Yang, Y. Xu, J. Shi, B. Dai, and B. Zhou, "Temporal pyramid network for action recognition," in *Proc. CVPR*, 2020.

[5] H. Fan, Y. Li, B. Xiong, W.-Y. Lo, and C. Feichtenhofer, “Pyslowfast,” https://github.com/facebookresearch/slowfast, 2020.

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

[7] H. Sang, Z. Zhao, and D. He, “Two-level attention model based video action recognition network,” *IEEE Access*, vol. 7, pp. 118 388–118 401, 2019.

[8] Q. Li, W. Yang, X. Chen, T. Yuan, and Y. Wang, “Temporal segment connection network for action recognition,” *IEEE Access*, vol. 8, pp. 179 118–179 127, 2020.

[9] Y. Li, B. Ji, X. Shi, J. Zhang, B. Kang, and L. Wang, “Tea: Temporal excitation and aggregation for action recognition,” in *Proc. CVPR*, 2020.

[10] K. He, X. Zhang, S. Ren, and J. Sun, “Deep residual learning for image recognition,” in *Proc. CVPR*, 2016.

[11] A. Howard, M. Sandler, G. Chu, L.-C. Chen, B. Chen, M. Tan, W. Wang, Y. Zhu, R. Pang, V. Vasudevan *et al.*, “Searching for mobilenetv3,” in *Proc. ICCV*, 2019.

[12] M. Tan and Q. V. Le, “Efficientnet: Rethinking model scaling for convolutional neural networks,” *arXiv preprint arXiv:1905.11946*, 2019.

[13] C. Feichtenhofer, H. Fan, J. Malik, and K. He, “Slowfast networks for video recognition,” in *Proc. ICCV*, 2019.

[14] O. Köpüklü, N. Kose, A. Gunduz, and G. Rigoll, “Resource efficient 3d convolutional neural networks,” in *Proc. ICCVW*, 2019.

[15] D. Tran, H. Wang, L. Torresani, and M. Feiszli, “Video classification with channel-separated convolutional networks,” in *Proc. ICCV*, 2019.

[16] C. Feichtenhofer, “X3d: Expanding architectures for efficient video recognition,” in *Proc. CVPR*, 2020.

[17] A. G. Howard, M. Zhu, B. Chen, D. Kalenichenko, W. Wang, T. Weyand, M. Andreetto, and H. Adam, “Mobilenets: Efficient convolutional neural networks for mobile vision applications,” *arXiv preprint arXiv:1704.04861*, 2017.

[18] M. Sandler, A. Howard, M. Zhu, A. Zhmoginov, and L.-C. Chen, “Mobilenetv2: Inverted residuals and linear bottlenecks,” in *Proc. CVPR*, 2018.

[19] N. Ma, X. Zhang, H.-T. Zheng, and J. Sun, “Shufflenet v2: Practical guidelines for efficient cnn architecture design,” in *Proc. ECCV*, 2018.

[20] D. Tran, H. Wang, L. Torresani, J. Ray, Y. LeCun, and M. Paluri, “A closer look at spatiotemporal convolutions for action recognition,” in *Proc. CVPR*, 2018.

[21] S. Xie, C. Sun, J. Huang, Z. Tu, and K. Murphy, “Rethinking spatiotemporal feature learning: Speed-accuracy trade-offs in video classification,” in *Proc. ECCV*, 2018.

[22] R. Goyal, S. E. Kahou, V. Michalski, J. Materzynska, S. Westphal, H. Kim, V. Haenel, I. Fruend, P. Yianilos, M. Mueller-Freitag *et al.*, “The" something something" video database for learning and evaluating visual common sense.” in *Proc. ICCV*, 2017.

[23] B. Zhou, A. Andonian, A. Oliva, and A. Torralba, “Temporal relational reasoning in videos,” in *Proc. ECCV*, 2018.

[24] W. Kay, J. Carreira, K. Simonyan, B. Zhang, C. Hillier, S. Vijayanarasimhan, F. Viola, T. Green, T. Back, P. Natsev *et al.*, “The kinetics human action video dataset,” *arXiv preprint arXiv:1705.06950*, 2017.

[25] D. Zhang, X. Dai, and Y.-F. Wang, “Dynamic temporal pyramid network: A closer look at multi-scale modeling for activity detection,” in *Proc. ACCV*, 2018.

[26] Y. Lee, J.-w. Hwang, S. Lee, Y. Bae, and J. Park, “An energy and gpu-computation efficient backbone network for real-time object detection,” in *Proc. CVPRW*, 2019, pp. 0–0.

[27] Y. Lee and J. Park, “Centermask: Real-time anchor-free instance segmentation,” in *Proc. CVPR*, 2020.

[28] K. Sun, Y. Zhao, B. Jiang, T. Cheng, B. Xiao, D. Liu, Y. Mu, X. Wang, W. Liu, and J. Wang, “High-resolution representations for labeling pixels and regions,” *arXiv preprint arXiv:1904.04514*, 2019.

[29] K. Hara, H. Kataoka, and Y. Satoh, “Can spatiotemporal 3d cnns retrace the history of 2d cnns and imagenet?” in *Proc. CVPR*, 2018.

[30] Z. Qiu, T. Yao, and T. Mei, “Learning spatio-temporal representation with pseudo-3d residual networks,” in *Proc. ICCV*, 2017.

[31] X. Zhang, X. Zhou, M. Lin, and J. Sun, “Shufflenet: An extremely efficient convolutional neural network for mobile devices,” in *Proc. CVPR*, 2018.

[32] J. Hu, L. Shen, and G. Sun, “Squeeze-and-excitation networks,” in *Proc. CVPR*, 2018, pp. 7132–7141.

[33] O. Russakovsky, J. Deng, H. Su, J. Krause, S. Satheesh, S. Ma, Z. Huang, A. Karpathy, A. Khosla, M. Bernstein *et al.*, “Imagenet large scale visual recognition challenge,” *Int. J. Comput. Vis.*, 2015.

[34] K. Simonyan and A. Zisserman, “Very deep convolutional networks for large-scale image recognition,” *arXiv preprint arXiv:1409.1556*, 2014.

[35] X. Wang, R. Girshick, A. Gupta, and K. He, “Non-local neural networks,” in *Proc. CVPR*, 2018.

[36] I. Loshchilov and F. Hutter, “Sgdr: Stochastic gradient descent with warm restarts,” *arXiv preprint arXiv:1608.03983*, 2016.

[37] P. Goyal, P. Dollár, R. Girshick, P. Noordhuis, L. Wesolowski, A. Kyrola, A. Tulloch, Y. Jia, and K. He, “Accurate, large minibatch sgd: Training imagenet in 1 hour,” *arXiv preprint arXiv:1706.02677*, 2017.

[38] C.-Y. Wu, R. Girshick, K. He, C. Feichtenhofer, and P. Krahenbuhl, “A multigrid method for efficiently training video models,” in *Proc. CVPR*, 2020, pp. 153–162.

[39] X. Wang and A. Gupta, “Videos as space-time region graphs,” in *Proc. ECCV*, 2018.

[40] G. Bertasius, H. Wang, and L. Torresani, “Is space-time attention all you need for video understanding?” in *Proc. ICML*, 2021.

[41] A. Dosovitskiy, L. Beyer, A. Kolesnikov, D. Weissenborn, X. Zhai, T. Unterthiner, M. Dehghani, M. Minderer, G. Heigold, S. Gelly *et al.*, “An image is worth 16x16 words: Transformers for image recognition at scale,” in *Proc. ICLR*, 2021.

[42] H. Fan, B. Xiong, K. Mangalam, Y. Li, Z. Yan, J. Malik, and C. Feichtenhofer, “Multiscale vision transformers,” in *Proc. ICCV*, 2021.

[43] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, Ł. Kaiser, and I. Polosukhin, “Attention is all you need,” in *Proc. NeurIPS*, 2017.

[44] J. Devlin, M.-W. Chang, K. Lee, and K. Toutanova, “Bert: Pre-training of deep bidirectional transformers for language understanding,” in *Proc. NACCL*, 2019.

[45] J. Carreira and A. Zisserman, “Quo vadis, action recognition? a new model and the kinetics dataset,” in *Proc. CVPR*, 2017, pp. 6299–6308.

YOUNGWAN LEE received the B.S. and M.S. degree in Information and Communication Engineering from Inha University, Incheon, Republic of Korea, in 2015 and 2017, respectively. He is now Ph.D student in graduate school of AI from Korea Advanced Institute of Science and Technology (KAIST), Daejeon, Republic of Korea. Since 2017, he has been working with the Visual Intelligence Research Section, the Artificial Intelligence Research Laboratory, the Electronics and Telecommunications Research Institute (ETRI), Daejeon, Republic of Korea. His research interests include object detection, instance segmentation, video recognition, and vision transformer.

HYUNG-IL KIM (M’21) received the B.S. degree (*Summa Cum Laude*) in Semiconductor Science from Dongguk University, Seoul, South Korea, and the M.S. and Ph.D. degrees in Electrical Engineering from the Korea Advanced Institute of Science and Technology (KAIST), Daejeon, South Korea. He was a researcher in the Agency for Defense Development, Taean, South Korea. Since 2017, he has been working with the Visual Intelligence Research Section in the Artificial Intelligence Research Laboratory of the Electronics and Telecommunications Research Institute (ETRI), Daejeon, South Korea. His research interests include visual recognition, face recognition, computer vision, and deep learning.

KIMIN YUN received the B.S and the Ph.D degrees at the Department of Electrical and Computer Engineering from Seoul National University, Seoul, Rep. of Korea, in 2010 and 2017, respectively. Since 2017, he has been working with the Visual Intelligence Research Section in the Artificial Intelligence Research Laboratory at the Electronics and Telecommunications Research Institute (ETRI), Daejeon, Republic of Korea. His current research interests include machine learning, computer vision, visual event detection, moving object detection, and video analysis.

JINYOUNG MOON received her B.S. degree in Computer Engineering from the Kyungpook National University (KNU), Daegu, Republic of Korea, in 2000. She received her M.S. degree in Computer Science and Ph.D. in Industrial Systems Engineering from the Korea Advanced Institute of Science and Technology (KAIST), Daejeon, Republic of Korea, in 2002 and 2018, respectively. Since 2002, she has been working with the Visual Intelligence Research Section, the Artificial Intelligence Research Laboratory, the Electronics and Telecommunications Research Institute (ETRI), Daejeon, Republic of Korea. Since 2019, she has also been with the ICT department, the University of Science and Technology (UST), where she is currently an Assistant Professor. Her research interests include action recognition, online and offline action detection, temporal moment localization, and video QA.