Lightweight Action Recognition with Sequence-Specific Global Context

1st Yao Chen, Hefei Ling, Jiazhong Chen, Lei Wu, Yuxuan Shi
School of Computer Science and Technology
Huazhong University of Science and Technology
Wuhan, China
{cy423, lhefei, jzchen, leiwu, shiyx}@hust.edu.cn

***Abstract*—With the emergence of a large number of video resources, video action recognition is attracting much attention. Recently, realizing the outstanding performance of three-dimensional (3D) convolutional neural networks (CNNs), many works have began to apply them for action recognition and obtained satisfactory results. However, high computational overheads greatly reduce the efficiency of 3D CNNs. To make up for the shortcoming, in this paper, we first propose two innovations — the Xwise Separable Convolution and the SS block, both of which are lightweight. Then we build an efficient 3D CNN called the XwiseNet based on our innovations. Our work aims to make 3D CNNs lightweight without reducing the recognition accuracy. The key idea of the Xwise Separable Convolution is extremely decoupling the 3D convolution in channel, spatial, and temporal dimensions. The SS block can capture temporal long-range dependencies via aggregating sequence-specific global context to each sequence feature. Experiments have verified that our XwiseNet achieves competitive performance with the least computational overhead.**

***Index Terms*—Action recognition, Three-dimensional convolutional neural networks, Lightweight, Sequence-specific global context**

I. Introduction

Action recognition is attracting more and more attention in the field of computer vision. Due to the spatiotemporal characteristics of videos, spatiotemporal convolutions which we call three-dimensional (3D) convolutions do better than spatial convolutions, and the latter is called two-dimensional (2D) convolutions. C3D is the first model to use 3D convolutions in action recognition [16], which is called 3D convolutional neural network (CNN). Later many variants emerge which dramatically improve the accuracy of action recognition.

3D CNNs are outstanding in action recognition, but with huge computational overheads. 3D CNNs have many parameters, which leads to the need for more computing resources and training data for optimizing. For example, a 2D convolution of size 3 has 9 parameters (we assume that the number of input channels is 1), while a 3D convolution of the same size has 27 parameters. When using a convolution of size 3 to convolve an input of size S, the FLOPs (floating-point operations) of 2D convolution are $S^2 \times 3^2$, while for 3D convolution are $S^3 \times 3^3$.

This work was supported in part by the Natural Science Foundation of China under Grant U1536203 and 61972169, in part by the National key research and development program of China (2016QY01W0200), in part by the Major Scientific and Technological Project of Hubei Province (2018AAA068 and 2019AAA051).

It can be intuitively seen that 3D CNNs have much more parameters and computational requirements when the input and convolution are the same size compared with 2D CNNs. Now important issues follow that we need lots of computing resources and samples to train 3D CNNs. It is an inevitable trend that 3D CNNs are of strong demand for lightweight design. Although decomposed 3D convolutions [17] [19] [23], group convolutions [2], and dual-channel architectures [4] are proposed to construct lightweight 3D CNNs, most of them still remains inefficient in resource-hungry action recognition scenarios.

Let's turn attention to lightweight 2D CNNs where Depthwise Separable Convolution plays an important role in state-of-the-art lightweight image classification networks [7]. Depthwise Separable Convolution contains the depthwise convolution and the pointwise convolution. Different from standard convolution which both filters and combines inputs into a new series of outputs in one step, in Depthwise Separable Convolution, the depthwise convolution first filters each input channel independently, then the pointwise convolution combines the output of the depthwise convolution by filtering with a 1×1 convolution. Essentially, Depthwise Separable Convolution is the factorized convolution. There is a hypothesis behind 2D Depthwise Separable Convolution: channel correlation and spatial correlation can be decoupled. We all know that, in 2D features, each channel represents a class of spatial features. Promoting to 3D features, each channel represents a class of spatiotemporal features. So the idea of Depthwise Separable Convolution can be naturally adapted to 3D convolutions. By adding a temporal dimension, we build a 3D Depthwise Separable Convolution in Fig. 4(a), whose main difference with 2D structure reflects in the convolution dimension — 3D convolution has extra temporal dimension, i.e., 3D convolution extracts spatiotemporal features at the same time. Inspired by Depthwise Separable Convolution's idea of division, can we continue to divide the 3D convolution into a 2D spatial convolution and a 1D temporal convolution? What can we reap from such a decomposition? The first advantage is that 3D CNNs can load the currently trained 2D CNNs' parameters as the initial value of spatial feature extraction, which is equal to providing 3D convolution with a mature prior knowledge. The second potential benefit is from the additional nonlinear mapping between 2D and 1D convolution,

978-1-7281-6926-2/20/$31.00 ©2020 IEEE

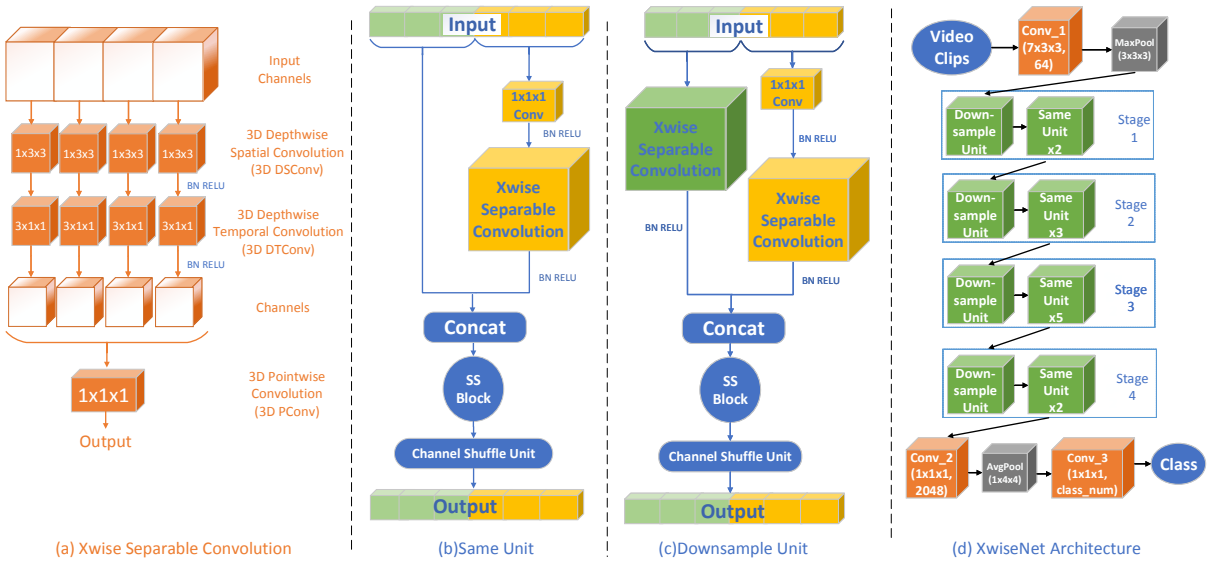

Fig. 1. **Xwise Separable Convolution, Same Unit, Downsample Unit, and XwiseNet architecture.** 3D Depthwise Spatial Convolution and 3D Depthwise Temporal Convolution in the Xwise Separable Convolution mean 3D depthwise convolution that extracts spatial and temporal features. The dimension of convolutions is represented as $\{T \times S \times S\}$ on behalf of the temporal and spatial domain. Same Unit and Downsample Unit are based on the Xwise Separable Convolution. In (d), $(* \ Unit \times Num)$ means the superposition of $* \ Unit$. The dimension of convolutions is represented as $\{T \times S \times S, C\}$ on behalf of the temporal, spatial and channel domain.

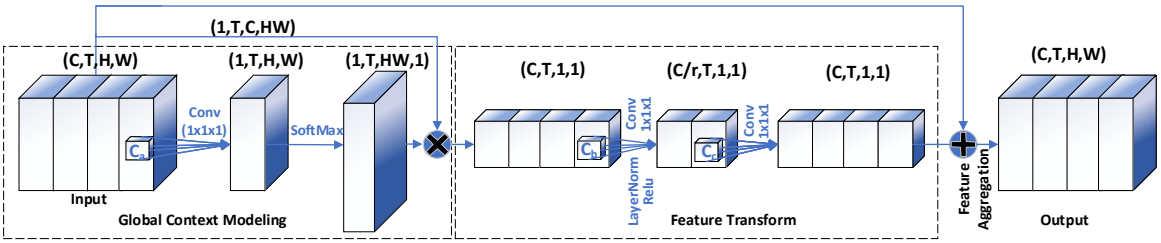

Fig. 2. **SS Block.** (C, T, H, W) means $(channel, frames, height, width)$, r in Feature Transform is set to 8.

which significantly doubles the nonlinear function with fewer parameters, and allows the model to represent more complex mappings compared with the traditional 3D convolution. The third benefit shown in Fig. 6 is that decomposition facilitates optimization, resulting in lower training losses and test losses in the experiment. In other words, we find that the decomposed 3D convolution which independently extracts channel, spatial and temporal features is easier to optimize than the traditional 3D convolution with joint optimization characteristic.

Based on the above mentioned, we propose the Xwise Separable Convolution as shown in Fig. 1(a), which contains the idea of extremely splitting three dimensions. Experiments have shown that the Xwise Separable Convolution achieves higher accuracy on the Part-Kinetics benchmark [1] while more computationally efficient compared with traditional 3D convolution.

Another area of concern is the video action recognition has a high requirement for the overall understanding of the scene. However, convolutional kernels can only handle local features due to their physical design, and the global features modeling relies on the superposition of convolutions. So only the upper layers of CNNs can get the global understanding, which would result in ineffective modeling of global context. Besides, distinguishing discriminative frames is also a key point in action recognition, which needs models to focus on the temporal correlation. To address the above issue, we design a lightweight block as shown in Fig. 2 to capture sequence-specific global context which we call the SS block. The SS Block can be plugged into each layer to handle global features without too much computational burden thanks to its lightweight characteristic. Now we will prove the effectiveness of the SS block based on similarity. We denote T_i as the feature vector for frame i, the average cosine similarity can be expressed as

$$avg\_cos\_sim = \frac{1}{N^2} \sum_{i=1}^{N} \sum_{j=1}^{N} cos(T_i, T_j) \tag{1}$$

where N is the number of input frames. The result are shown in Table I where 'input', 'SS context' and 'output' mean the input, the sequence-specific global context and the output of the SS block. The cosine similarity of the output frames is lower than the input, which indicates that the output can be discriminated across different frames more effectively.

TABLE I
AVERAGE COSINE SIMILARITY

input	SS context	output
0.2826	0.2499	0.2102

The Xwise Separable Convolution and the SS block lead to the design of a new 3D CNN named the XwiseNet. As shown in Table IV and Table V, on the most suitable dataset we can reach, the XwiseNet achieves the competitive accuracy with the lowest computation cost, which shows that our model can efficiently identify action instances. It is worth mentioning that our SS block outperforms other mainstream global context module, e.g. CBAM [34], SE block [28] and GC block [36], showing that the SS block and the Xwise Separable Convolution are complementary effectively in action recognition.

II. RELATED WORK

A. Action recognition

At present, one of the most effective ways of action recognition is two-stream CNNs. Simonyan [15] first proposed the two-stream method whose input is RGB frames and optical flow. In addition to the above input, Wang [18] also tried other inputs — RGB difference and warped optical flow, and RGB + optical flow + warped optical flow has been experimentally proven to be most effective. The two-stream method is limited in practical applications due to the time overhead required to get the optical flow in advance.

Another important way in action recognition is 3D CNNs. Du [16] first built a 3D network using 3D convolutions and 3D poolings, which they call C3D. However, due to the relatively simple network structure, C3D's accuracy is not competitive with two-stream programs. Joao [1] eliminated the shortcomings of C3D by building 3D CNNs upon state-of-the-art image classification architecture, which they call I3D. Following the idea of I3D to expand 2D CNNs, Diba [3] also proposed Temporal Transition Layer to capture different temporal depths and built a novel model named T3D. Inspired by the human retinal mechanism, Feichtenhofer [4] used a slow high-resolution CNN (Slow Channel) to analyze static content in the video while using a fast low-resolution CNN (Fast Channel) to analyze dynamic content in the video. Different from the accuracy-focused models above, our XwiseNet is a new type of 3D CNN capable of both lightweight and accuracy.

B. Lightweight CNNs

In recent years, deep CNNs have performed well on many tasks. In addition to accuracy, model efficiency is also a factor worthy of attention because it determines whether the model can be applied in the actual scene.

In 2D CNNs, the Depthwise Separable Convolution [14] plays an important role which is firs used by MobileNet [7]. In pursuit of the goal of practical efficiency, ShuffleNet V2 [11] designs a new network structure based on the Depthwise Separable Convolution, while introducing channel shuffle to promote information exchange, which doesn't need extra parameters.

In 3D CNNs, Chen [2] proposed Multi-Fiber Unit drawing on group convolutions to reduce the model size. MiCT [21] integrates 2D CNNs with 3D CNNs to reduce complexity of spatiotemporal networks. Besides modification of the network structure, researchers also focus on innovation of 3D convolutions. S3D [19], R(2+1)D [17] and P3D [23] all use one $1 \times 3 \times 3$ spatial convolution and another $3 \times 1 \times 1$ temporal convolution to approximate the spatiotemporal convolution. Based on the above splitting, Yang [24] further splits $1 \times 3 \times 3$ convolution into $1 \times 1 \times 3$ and $1 \times 3 \times 1$. From the perspective of image transmission frequency, Chen [22] proposed a novel Octave Convolution to store and process low-frequency and high-frequency features separately to improve the model efficiency and reduce spatial redundancy.

Based on the succession and innovation of the previous research, we build a new type of 3D convolution, which greatly reduces parameters and computation cost in the extreme decoupling state while ensuring the stability of performance. In Table IV and Table V, we extensively compare the XwiseNet with earlier state-of-the-art methods and the XwiseNet shows a competitive result on the challenging benchmarks with an extremely lightweight design.

C. Global Context Modeling

Recently, in the field of images, global context modeling has been studied in SENet [28], GENet [29] and PSANet [30], which all focus on recalibrating the channel dependency with global context. In addition to channel dependency, CBAM [34] also explores the dependency among spatial positions. However, for feature fusion, the above methods all apply rescaling which is proved not the most effective way for global context modeling [36]. In videos, making better use of temporal global context helps to improve the effectiveness of action recognition. In [31]–[33], temporal dependency modeling is applied in the motion stream. However, their temporal modeling just focuses on the optical flow information rather than the direct relation among different frames. Moreover, the optical flow information needs extra extraction in advance, which increases resource consumption in the method implementation. An end-to-end spatiotemporal global context modeling is proposed in [35], but additional skeleton data is needed. Reference [37] proposed a Non-local network (NLNet) which models spatiotemporal pixel-level pairwise dependency. Regrettably, the NLNet computes query-independent dependency for each query position, which is proved redundant. Reference [36] removes redundancy from NLNet and they design the Global Context (GC) block to effectively model channel-wise global context via addition fusion as NLNet [37], with the lightweight property as SENet [28]. However, GCblock models sequence-independent global context. In other words, GCblock can't identify the most relevant frames from an input video. As an improved version of the GC block, we introduce a sequence-specific global context modeling block, which can localize

discriminative frames and doesn't require additional motion stream.

III. Method

In this section, we first define a novel 3D convolution as shown in Fig. 1(a). The 3D convolution is split into 3D Depthwise Spatial Convolution, 3D Depthwise Temporal Convolution, and 3D Pointwise Convolution. We follow the hypothesis: channel correlation, spatial correlation and temporal correlation mapping in feature maps of 3D CNNs can be completely decoupled. Because this assumption is a stronger version of the hypothesis behind the Depthwise Separable Convolution, we named our proposed 3D convolution as Xwise Separable Convolution, which means "Extremewise Separable Convolution". Besides, to make up for the bottom-up local operators in CNNs which can't capture long-range contextual interactions, we design the SS block to aggregate contextual information for each sequence adaptively and specifically. Finally, we build an extremely efficient network based on the Xwise Separable Convolutions and the SS blocks, which we call the XwiseNet.

A. Xwise Separable Convolution

Given a video clip, using 3D convolution is the most general way to extract spatiotemporal information [13] [16]. The 3D convolution can construct temporal connections across frames while extracting spatial information. For simplicity, we represent a traditional 3D convolution as (N, C, d, k, k), where C is the number of input channels, d and k are temporal and spatial size of the convolution, N is the number of filters or output channels. As shown in Fig. 1(a), using the idea of Depthwise Separable Convolution extremely, a 3D $d \times k \times k$ convolution can be naturally decoupled into a $1 \times k \times k$ convolution acting in the spatial domain and a $d \times 1 \times 1$ convolution acting in the temporal domain, both of which are depthwise. Then we fuse channel information through N $C \times 1^3$ convolutions. The number of parameters and FLOPs of the Xwise Separable Convolution are about $1/dk^2$ of the traditional 3D convolution. The extremely decoupled 3D convolution not only significantly reduces parameters and FLOPs but also pre-trains 2D convolutions from image data, giving the Xwise Separable Convolution the ability to take more advantage of scene and object knowledge. At the same time, non-linear operations can be added after each convolution, which greatly increases the expressive power of the network. Besides, from the perspective of network optimization, it is easier to optimize in the channel, spatial and temporal dimensions independently.

Comparison with other decomposed 3D convolutions. We notice that there have been some works focusing on spatiotemporal factorization of 3D convolutions, e.g. P3D [23], R(2+1)D [17] and S3D [19]. P3D proposed three different blocks according to the order of spatial convolutions and temporal convolutions, which all contain bottlenecks. Our Xwise Separable Convolution only adapts one order without bottlenecks. R(2+1)D and S3D take a similar order to factorize 3D convolutions as us. However, R(2+1)D contains

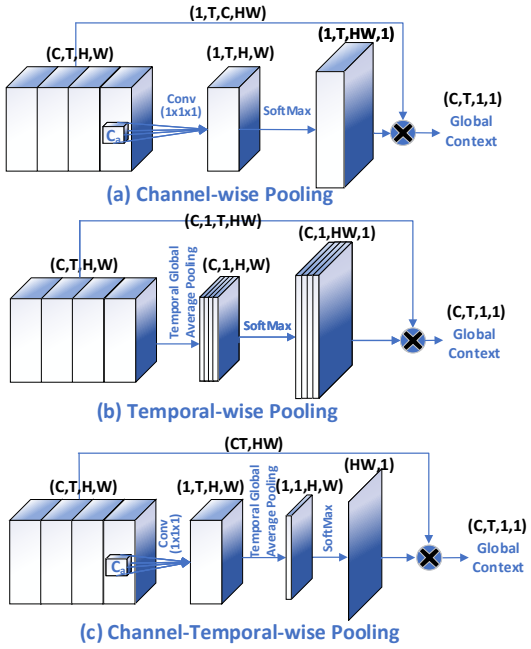

Fig. 3. **Context_Modeling.**

hourglasses (opposite of bottlenecks) and S3D omits activation and batch normalization operations. As can be seen from the following, the way we build the network with the above methods is also very different. Specifically, our network is based on the extremely efficient ShuffleNet V2-50 [11], which further improves the efficiency of the model.

B. Sequence-specific Block

Here we propose the SS Block inspired by the global context (GC) block. The main difference between the SS block and the GC block is that the SS block can capture temporal long-range dependencies via aggregating sequence-specific global context to each sequence feature. Experiments in Table IV prove that the sequence-specific feature makes the SS block better than the GC block in video action recognition. As stated in the GC work, Global Context Modeling Framework can be abstracted into three stages: (1) global context modeling which extracts global context from the input; (2) feature transform which captures temporal-wise dependencies; (3) feature aggregation which aggregates sequence-specific global context to the time dimension of the input. Because of the superiority of the GC block in various tasks, we keep (2) (3) stages and propose a new structure for (1). Above abstraction can be defined as

$$y_i = F(x_i, \lambda(\sum_{j=1}^{N} \omega_j x_j)) \tag{2}$$

where (1) $\sum_{j=1}^{N} \omega_j x_j$ expresses the context modeling module which adopts weighted averaging with weight ω_j to aggregates the features on some dimensions to get the context features (temporal context features in the SS block and channel-wise context features in the GC block); (2) $\lambda(.)$ expresses the feature transform to model relations between channels; (3)

$F(.,.)$ represents the fusion process to fuse the context futures to the original features on a dimension (temporal dimension in the SS block and channel dimension in the GC block). Further, we explore three ways of context modeling: (a) channel-wise pooling; (b) temporal-wise pooling; (c) channel-temporal-wise pooling. Their specific structures are shown in Fig. 3. Results in Table III (a) show the superiority of (a) compared to (b) and (c), so we adopt (a) in the SS block. The detailed SS block is shown in Fig. 2, formulated as

$$y_i = x_i + C_c * RELU(LN(C_b * \sum_{j=1}^{N} \frac{e^{C_a * x_j}}{\sum_{k=1}^{N} e^{C_a * x_k}} x_j)) \quad (3)$$

where C_a represents the weight for context modeling, $C_c * RELU(LN(C_b * (.)))$ represents the bottleneck transform, broadcast element-wise addition is the fusion way.

Since the context feature modeling by our SS block is $C \times T \times 1 \times 1$ which is sequence-specific, it can capture temporal relations and localize discriminative frames. On the other hand, the SS block is flexible and lightweight, which meets our criteria for building lightweight action recognition, so it can be added to multiple layers of the network to better capture the long-term information with acceptable extra computation cost.

C. XwiseNet

In this section, we use the Xwise Separable Convolution and the SS block to design a network called the XwiseNet based on the extremely efficient ShuffleNet V2-50 [11]. The network we design can further verify the generalization ability of the Xwise Separable Convolution and model temporal long-range global context.

We design the Same Unit and Downsample Unit as shown in Fig. 1(b)(c). Same Unit keeps the input and output the same size. Downsample Unit is used to double the number of channels and halve the size of the feature map. Channel Shuffle Unit in both blocks is the key idea of ShuffleNet V2, which can achieve the purpose of information sharing between channels without increasing parameters. To make the SS block fit perfectly into above units, we investigate three positions of the SS block: (a) after Channel Shuffle Unit; (b) before Channel Shuffle Unit; (c) after the last $1 \times 1 \times 1$ convolution in the branch (right branch for Downsample Unit); Results in Tabel III (b) show that (b) performs best. Hence we adopt (b) in the XwiseNet. The complete network structure is shown in Fig. 1(d).

IV. Experiments

A. Datasets

a) KTH: The KTH dataset [27] covers six human actions: walking, running, jogging, boxing, clapping and waving. Each action is performed several times by 25 subjects in four different scenes: outdoors, outdoors with different clothes, outdoors with scale variation and indoors. In total, KTH contains 2391 video samples. Due to data requirements for network training, we divide the samples into a training set (16 subjects) and a test set (9 subjects).

b) Part-Kinetics: Limited by computation and time resources, as well as for more efficient experimental comparisons, we built a small dataset — Part-Kinetics, which is a 10-classes subset randomly selected from Kinetics [1]. In the training set, Part-Kinetics contains about 500 samples each class, which is larger than some current mainstream datasets like UCF-101 and HMDB-51, so it can avoid overfitting. Part-Kinetics contains 5498 training videos and 459 testing videos. Using Part-Kinetics can obtain more efficient and accurate performance comparison among models.

B. Training

We take the same sampling method of video frames as [5]. First, we select a starting temporal position in the video by uniform sampling to generate a 16-consecutive-frames clip. If the video is shorter than 16 frames, it is populated with existing frames. Next, we randomly select a target location from the center or 4 corners. In addition to the above enhancements, we also perform multi-scale cropping. The scale is selected from $[1, \frac{1}{2^{\frac{1}{4}}}, \frac{1}{\sqrt{2}}, \frac{1}{2^{\frac{3}{4}}}, \frac{1}{2}]$. Note that the aspect ratio of our samples is 1, and the scale 1 indicates that the edge length of the sample is the same as the short edge length of the original video frame, and the scale 0.5 indicates that the sample is half of the short edge length of the frame. After the sample is cropped based on position and scale, we adjust its spatial size to 112×112. The size of each sample is $3\ channels \times 16\ frames \times 112\ pixels \times 112\ pixels$, and each sample flips at 50% probability. Mean subtraction and normalization are also performed.

We optimize all models by backpropagating the gradients of cross-entropy loss from scratch. All models are trained using Adam with a weight decay of 0.001. The learning rate is initialized to 0.001 and decays by a factor of 0.1 according to the accuracy of the validation set.

C. Recognition

We use sliding windows to generate input clips ($i.e.$ each video is divided into non-overlapped 16-frame clips), each of which is spatially cropped at scale 1 in the center. We then use the trained model to predict scores of various classes of each clip. A class with the highest score represents the label for the video, which is averaged on all clips of the video.

D. Results and Analysis

In this section, we first present a comparison of the Xwise Separable Convolution with other forms of 3D convolutions in the accuracy and computation cost for action recognition. Then we conduct ablation experiments to determine a better structure for the SS block. Finally, we show the performance of the XwiseNet compared with state-of-the-art works.

1) Effect of the Xwise Separable Convolution: The Xwise Separable Convolution is primarily proposed to enhance 3D CNNs. To show the effectiveness of the Xwise Separable Convolution, we compare it with other forms of 3D convolutions. The experimental idea is to fix the base network architecture to be the same throughout all the experiments and compare

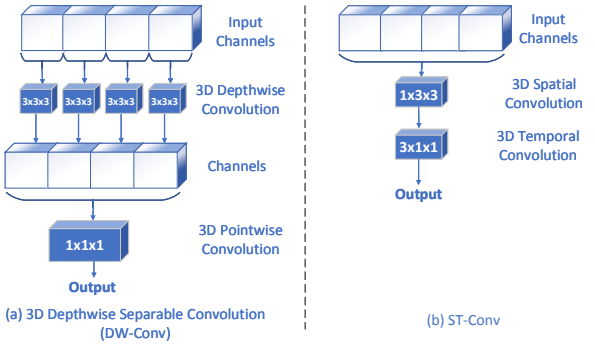

Fig. 4. **3D Depthwise Separable Convolution, ST-Conv** The input is video clips and the dimension of convolutions is represented as $\{T \times S \times S\}$ on behalf of the temporal and spatial domain.

the performance of CNNs with different 3D convolutions. Different 3D convolutions in experiments include Traditional-Conv, ST-Conv, X-Conv, and DW-Conv. Traditional-Conv is the abbreviation of traditional 3D convolutions. ST-Conv is 3D convolutions that are split in time and space dimensions. X-Conv is our proposed the Xwise Separable Convolution. DW-Conv is 3D Depthwise Separable Convolution. DW-Conv and ST-Conv are shown in Fig. 4(a), (b).

In this work, we use deep residual networks (ResNets) [6] as our backbone owing to their good performance and ease of refactoring. Table II reports the performance of different convolutions on Part-Kinetics. It is worth mentioning that X-Conv has the fewest parameters. A more intuitive display can be seen in Fig. 5. X-Conv beats Traditional-Conv in accuracy and efficiency by absolute advantage. Combined with data, the Xwise Separable Convolution can reduce FLOPs by 31% with a loss of less than 1% accuracy compared with ST-Conv. Similarly, compared with DW-Conv, the Xwise Separable Convolution only needs to pay no more than 5% FLOPs to improve the accuracy by 1.74%. This shows that the proposed the Xwise Separable Convolution can achieve a better balance between accuracy and efficiency. From the loss curve in Fig. 6,

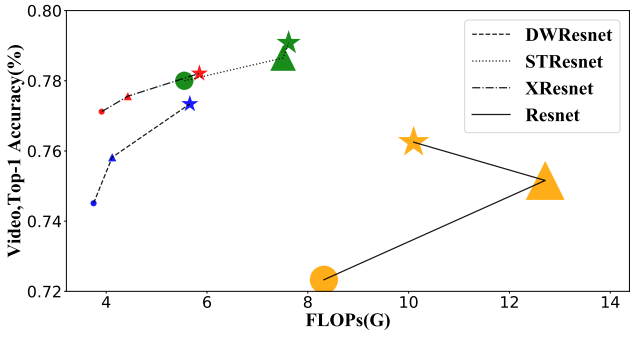

Fig. 5. **Tradeoff comparison between networks based on different 3D convolutions.** The area of each point is proportional to the total parameter number of the model. Circle represents the network whose backbone is Rensnet18, triangle represents the network whose backbone is Rensnet34 and Pentagram represents the network whose backbone is Rensnet50.

TABLE II
ACTION RECOGNITION PERFORMANCE OF DIFFERENT CONVOLUTIONS ON PART-KINETICS TEST SET.

		3D-ResNet-18	3D-ResNet-34	3D-ResNet-50
ST-Conv	Accuracy	**78%**	**78.65%**	**79.08%**
	FLOPs(G)	5.55	7.51	7.62
	Params(M)	14.13	27.62	27.36
DW-Conv	Accuracy	74.51%	75.82%	77.34%
	FLOPs(G)	**3.75**	**4.12**	**5.66**
	Params(M)	1.58	2.82	13.64
Traditional-Conv	Accuracy	72.33%	75.16%	76.25%
	FLOPs(G)	8.32	12.71	10.10
	Params(M)	33.21	63.52	46.22
X-Conv(Ours)	Accuracy	77.12%	77.56%	78.21%
	FLOPs(G)	3.91	4.43	5.85
	Params(M)	**1.53**	**2.72**	**13.59**

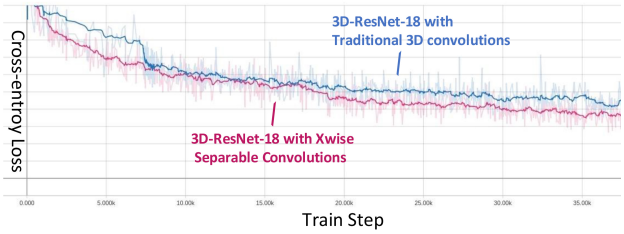

Fig. 6. **3D-ResNet-18 training Loss.** The training loss on 3D-ResNet-18 with the Xwise Separable Convolutions falls faster than 3D-ResNet-18 with traditional 3D Convolutions, as well as converges to a lower value.

we can also find that the Xwise Separable Convolution can be quickly optimized to a better level.

2) Ablation Study for SS Blcok: The ablation study is shown in Table III. **Context modeling:** To model sequence-specific global context efficiently, we compare three methods for context modeling. It shows that channel-wise pooling significantly outperforms the other two methods with a similar number of parameters and FLOPs. This indicates that temporal information is more discriminative in action recognition than channel information, which can't be pooled. Moreover, the above conclusion also verifies the necessity of constructing the sequence-specific block. **Positions:** We investigate three positions of the SS block and inserting the SS block before channel shuffle yields the highest performance. So we adopt before channel shuffle as the default. **Stages:** We compare the results when the SS blocks are inserted at different stages. In the case where the consumption difference is negligible, all stages benefit from the SS block. Inserting the SS block to all stages achieves higher accuracy than inserting to other single stage.

3) Comparison with state-of-the-art methods: In Table IV and Table V, we show the comparison with state-of-the-art methods using only RGB inputs for a fair comparison, *i.e.* no optical flow. It can be observed that on Part-Kinetics, our XwiseNet without any global context modeling blocks (we call it the Simple-XwiseNet) outperforms most methods except R(2+1)D and I3D. In the previous works, the basic

TABLE III
ABLATION STUDY

	Accuracy	Params(M)	FLOPs(G)
(a) Context modeling			
channel-wise pooling	**82.35%**	3.51	1.80
temporal-wise pooling	81.26%	3.50	1.80
channel-temporal-wise pooling	79.74%	3.51	1.80
(b) Positions			
baseline	78.21%	2.90	1.22
after channel shuffle	82.35%	3.51	1.80
before channel shuffle	**83.01%**	3.51	1.80
after 3D PConv	79.52%	3.05	1.80
(c) Stages			
baseline	78.21%	2.90	1.22
stage1	79.52%	3.51	1.80
stage2	79.96%	3.51	1.80
stage3	80.61%	3.51	1.80
stage4	78.65%	3.51	1.80
all stages	**83.01%**	3.51	1.80

TABLE IV
ACTION RECOGNITION PERFORMANCE ON PART-KINETICS TEST SETS.

Model		Accuracy	Params(M)	FLOPs(G)
C3D [16]		70.80%	63.36	38.58
3D-ResNet-18 [5]		72.33%	33.21	8.32
3D-ResNet-34 [5]		75.16%	63.52	12.71
MFNet [2]		76.03%	7.70	2.93
3D-ResNet-50 [5]		76.25%	46.22	10.10
P3D [23]		76.47%	24.95	8.14
ARTNet [26]		77.12%	20.16	14.02
fast-S3D [19]		77.56%	8.28	2.79
I3D [25]		79.30%	12.29	27.82
R(2+1)D [17]		80.17%	63.54	20.7
Backbone(Ours)	Block			
XwiseNet	-	78.21%	**2.90**	**1.22**
XwiseNet	CBAM [34]	78.21%	3.20	7.15
XwiseNet	SE [28]	78.43%	3.20	1.80
XwiseNet	GC [36]	81.05%	3.50	1.80
XwiseNet	SS(Ours)	**83.01%**	3.51	1.80

convolution used by 3D-ResNet-18/34/50, C3D, I3D and ArtNet is all Traditional-Conv, that used by P3D, R(2+1)D and fast-S3D is ST-Conv. In addition, MFNet used the idea of Group Convolution and DW-Conv can be considered as Group Convolution's special version. From Table II, we can know that in terms of accuracy, our proposed X-Conv is better than DW-Conv and slightly lower than ST-Conv. So it's not surprising that R(2+1)D is better than our Simple-XwiseNet in accuracy. But it is worth mentioning that due to the superiority of our network structure design, our Simple-XwiseNet outperforms P3D and fast-S3D by 1.74% and 0.65% with only 15% and 44% FLOPs. It is also not a surprise that the accuracy of the Simple-XwiseNet is still lower than I3D whose input is 16 224 $\times$ 224 frames' clip. In contrast, we take 16 112 $\times$ 112 frames' clip as the input of the Simple-XwiseNet which contains less spatial information but

TABLE V
ACTION RECOGNITION PERFORMANCE ON KTH TEST SETS. 16/9 IS THE NUMBER OF SUBJECTS IN TRAINING AND TEST SET. THE INPUT IS $16\ frames \times 224\ pixels \times 224\ pixels$.

Method	cross-validation	Accuracy	Params(M)	FLOPs(G)
TCCA [38]	Leave-one-out	95.33%	-	-
pLSA-ISM [39]	Leave-one-out	91.60%	-	-
Dollar et al. [40]	Leave-one-out	80.00%	-	-
Klaser et al. [41]	Leave-one-out	91.40%	-	-
Ikizler et al. [8]	16/9	94.00%	-	-
Jhuang et al. [10]	16/9	91.70%	-	-
Niebles et al. [20]	16/9	81.50%	-	-
Schuldt et al. [27]	16/9	71.72%	-	-
P3D	16/9	91.54%	24.95	15.95
ARTNet	16/9	94.09%	20.16	56.09
fast-S3D	16/9	95.37%	8.28	11.26
MFNet	16/9	95.71%	7.70	11.16
I3D	16/9	95.78%	12.29	27.82
XwiseNet(Ours)	16/9	94.79%	**3.51**	**7.11**

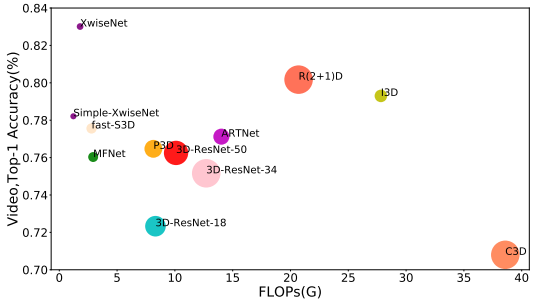

Fig. 7. **Tradeoff comparison between different 3D CNNs on Part-Kinetics.** The area of each circle is proportional to the total parameter number of the model.

take up less computational overhead, making our model more efficiency. Regarding computation and parameter efficiency, it can be seens intuitively from Fig. 7 that our proposed Simple-XwiseNet is with the fewest FLOPs and parameters, yet achieves 78.21% Top-1 accuracy. Based on Table II, we think that it is mainly due to the Xwise Separable Convolutions's competitiveness in the trade-offs of lightweight and accuracy in action recognition. With the SS block, our XwiseNet yields the best performance among all networks and global context modeling blocks. We also observe that the XwiseNet costs the lowest GPU memory for both training and testing benefiting from the Xwise Separable Convolution. On the KTH dataset as shown in Table V, in the 16/9-based cross-validation, the training set and test set of each model are the same except for Jhuang [10], whose subjects are randomly selected. The XwiseNet achieves 94.79% accuracy which is comparable with the sate-of-the-arts, and it is more resource friendly. For example, the XwiseNet can reduce FLOPs by 74% with a loss of 1% accuracy compared with I3D. Even compared with the methods that using Leave-one-out, our XwiseNet still better than most except TCCA.

V. Conclusion

In this work, we focus on lightweight action recognition. We first propose the Xwise Separable Convolution, which beats the traditional 3D convolution in lightweight and performance. Then we build a lightweight SS block modeling sequence-specific global context to further improve the performance of 3D CNN. Our XwiseNet is based on the Xwise Separable Convolutions and the SS blocks, which significantly achieves competitive performance with the least computation cost on two benchmarks. Although our experiments are based on video action recognition, we can extend this idea to other similar video tasks, such as object detection, object tracking to achieve the goal of balancing lightweight and accuracy.

References

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

[2] Yunpeng Chen, Yannis Kalantidis, Jianshu Li, Shuicheng Yan, and Jiashi Feng. Multi-fiber networks for video recognition. CoRR, abs/1807.11195, 2018.

[3] Ali Diba, Mohsen Fayyaz, Vivek Sharma, Amir Hossein Karami, Mohammad Mahdi Arzani, Rahman Yousefzadeh, and Luc Van Gool. Temporal 3d convnets: New architecture and transfer learning for video classification. CoRR, abs/1711.08200, 2017.

[4] Feichtenhofer C, Fan H, Malik J, et al. Slowfast networks for video recognition[C]//Proceedings of the IEEE International Conference on Computer Vision. 2019: 6202-6211.

[5] Kensho Hara, Hirokatsu Kataoka, and Yutaka Satoh. Can spatiotemporal 3d cnns retrace the history of 2d cnns and imagenet? In The IEEE Conference on Computer Vision and Pattern Recognition (CVPR), June 2018.

[6] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. volume abs/1512.03385, 2015.

[7] Andrew G. Howard, Menglong Zhu, Bo Chen, Dmitry Kalenichenko, Weijun Wang, Tobias Weyand, Marco Andreetto, and Hartwig Adam. Mobilenets: Efficient convolutional neural networks for mobile vision applications. CoRR, abs/1704.04861, 2017.

[8] Ikizler N, Cinbis R G, Duygulu P. Human action recognition with line and flow histograms[C]//2008 19th International Conference on Pattern Recognition. IEEE, 2008: 1-4.

[9] Sergey Ioffe and Christian Szegedy. Batch normalization: Accelerating deep network training by reducing internal covariate shift. CoRR, abs/1502.03167, 2015.

[10] Jhuang H, Serre T, Wolf L, et al. A biologically inspired system for action recognition[C]//2007 IEEE 11th International Conference on Computer Vision. Ieee, 2007: 1-8.

[11] Ningning Ma, Xiangyu Zhang, Hai-Tao Zheng, and Jian Sun. Shufflenet v2: Practical guidelines for efficient cnn architecture design. In Proceedings of the European Conference on Computer Vision (ECCV), pages 116–131, 2018.

[12] Vinod Nair and Geoffrey E. Hinton. Rectified linear units improve restricted boltzmann machines. In International Conference on International Conference on Machine Learning, 2010.

[13] Ji Shuiwang, Yang Ming, and Yu Kai. 3d convolutional neural networks for human action recognition. IEEE Transactions on Pattern Analysis and Machine Intelligence, 35(1):221–231, 2013.

[14] Laurent Sifre and Stéphane Mallat. Rigid-motion scattering for image classification. PhD thesis, Ph. D. thesis, 1:3, 2014.

[15] Karen Simonyan and Andrew Zisserman. Two-stream convolutional networks for action recognition in videos. CoRR, abs/1406.2199, 2014.

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

[17] Du Tran, Heng Wang, Lorenzo Torresani, Jamie Ray, Yann LeCun, and Manohar Paluri. A closer look at spatiotemporal convolutions for action recognition. CoRR, abs/1711.11248, 2017.

[18] 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 Bastian Leibe, Jiri Matas, Nicu Sebe, and Max Welling, editors, Computer Vision – ECCV 2016, pages 20–36, Cham, 2016. Springer International Publishing.

[19] Xie S, Sun C, Huang J, et al. Rethinking spatiotemporal feature learning: Speed-accuracy trade-offs in video classification[C]//Proceedings of the European Conference on Computer Vision (ECCV). 2018: 305-321.

[20] Niebles J C, Wang H, Fei-Fei L. Unsupervised learning of human action categories using spatial-temporal words[J]. International journal of computer vision, 2008, 79(3): 299-318.

[21] Y. Zhou, X. Sun, Z.-J. Zha, and W. Zeng. MiCT: Mixed 3D/2D Convolutional Tube for Human Action Recognition. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pages 449–458, 2018.

[22] Yunpeng Chen, Haoqi Fang, Bing Xu, Zhicheng Yan, Yannis Kalantidis, Marcus Rohrbach, Shuicheng Yan, and Jiashi Feng. Drop an Octave: Reducing Spatial Redundancy in Convolutional Neural Networks with Octave Convolution. arXiv preprint arXiv:1904.05049, 2019.

[23] Z. Qiu, T. Yao, and T. Mei. Learning Spatio-Temporal Representation with Pseudo-3D Residual Networks. In Proceedings of the International Conference on Computer Vision (ICCV), 2017. 8

[24] H. Yang, C. Yuan, B. Li, Y. Du, J. Xing, W. Hu, S.J. Maybank, Asymmetric 3D Convolutional Neural Networks for action recognition, Pattern Recognit. 85 (2019) 1–12.

[25] Carreira J, Zisserman A. Quo Vadis, Action Recognition? A New Model and the Kinetics Dataset[J]. 2017. 1–12.

[26] Wang L , Li W , Li W , et al. Appearance-and-Relation Networks for Video Classification[J]. 2017. 1–12.

[27] Schuldt C , Laptev I , Caputo B . Recognizing human actions: a local SVM approach[C]// Proceedings of the 17th International Conference on Pattern Recognition, 2004. ICPR 2004. IEEE, 2004.

[28] Hu J, She L, Sun G. Squeeze-and-excitation networks[C]//Proceedings of the IEEE conference on computer vision and pattern recognition. 2018: 7132-7141.

[29] Hu J, Shen L, Albanie S, et al. Gather-excite: Exploiting feature context in convolutional neural networks[C]//Advances in Neural Information Processing Systems. 2018: 9401-9411.

[30] Zhao H, Zhang Y, Liu S, et al. Psanet: Point-wise spatial attention network for scene parsing[C]//Proceedings of the European Conference on Computer Vision (ECCV). 2018: 267-283.

[31] W. Du, Y. Wang, and Y. Qiao. Recurrent spatial-temporal attention network for action recognition in videos. IEEE Transactions on Image Processing (ICIP), 2018. 2

[32] Z. Li, K. Gavrilyuk, E. Gavves, M. Jain, and C. G. Snoek. Videolstm convolves, attends and flows for action recognition. Computer Vision and Image Understanding (CVIU), 2018. 1, 2, 5, 6

[33] Y. Wang, S. Wang, J. Tang, N. O'Hare, Y. Chang, and B. Li. Hierarchical attention network for action recognition in videos. arXiv preprint arXiv:1607.06416, 2016. 2

[34] S. Woo, J. Park, J.-Y. Lee, and I. So Kweon. Cbam: Convolutional block attention module. In Proceedings of the European Conference on Computer Vision(ECCV), pages 3–19, 2018. 2

[35] S. Song, C. Lan, J. Xing, W. Zeng, and J. Liu. An end-to-end spatio-temporal attention model for human action recognition from skeleton data. In AAAI, 2017. 2, 3

[36] Cao Y, Xu J, Lin S, et al. GCNet: Non-local Networks Meet Squeeze-Excitation Networks and Beyond[J]. arXiv preprint arXiv:1904.11492, 2019.

[37] Wang X, Girshick R, Gupta A, et al. Non-local neural networks[C]//Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. 2018: 7794-7803.

[38] Kim T K, Wong S F, Cipolla R. Tensor canonical correlation analysis for action classification[C]//2007 IEEE Conference on Computer Vision and Pattern Recognition. IEEE, 2007: 1-8.

[39] Wong S F, Kim T K, Cipolla R. Learning motion categories using both semantic and structural information[C]//2007 IEEE Conference on Computer Vision and Pattern Recognition. IEEE, 2007: 1-6.

[40] Dollár P, Rabaud V, Cottrell G, et al. Behavior recognition via sparse spatio-temporal features[C]. Beijing, China: VS-PETS, 2005.

[41] Klaser A, Marszałek M, Schmid C. A spatio-temporal descriptor based on 3d-gradients[C]. 2008.