Design Light-Weight 3D Convolutional Networks for Video Recognition: Temporal Residual, Fully Separable Block, and Fast Algorithm

Haonan Wang, Jun Lin, Zhongfeng Wang
Nanjing University
hnwang@smail.nju.edu.cn, {jlin, zfwang}@nju.edu.cn

Abstract

Deep 3-dimensional (3D) Convolutional Network (ConvNet) has shown promising performance on video recognition tasks because of its powerful spatio-temporal information fusion ability. However, the extremely intensive requirements on memory access and computing power prohibit it from being used in resource-constrained scenarios, such as portable and edge devices. So in this paper, we first propose a two-stage fully separable block to significantly compress the model sizes with little accuracy loss. Then a feature enhancement approach named temporal residual gradient is proposed to improve the compressed model performance on video tasks, which provides higher accuracy, faster convergency and better robustness. Moreover, in order to further decrease the computing workload, we propose a hybrid Fast Algorithm to incredibly reduce the computation complexity of convolutions. These methods are effectively combined to design a light-weight and efficient ConvNet for video recognition tasks. Experiments on the popular dataset report $2.3\times$ *compression rate,* $6.8\times$ *workload reduction, and* 2% *top-1 accuracy gain, over the state-of-the-art SlowFast model, which is already a compact-designed model. The proposed methods also show good adaptability on traditional 3D ConvNet, leading to* $5\times$ *compact model,* $10\times$ *less workload, and* 3% *higher accuracy.*

1. Introduction

Deep Convolutional Network (ConvNet) has demonstrated remarkable performance on visual tasks with still image, such as document recognition [13], object detection [8], and semantic segmentation [3]. It employs 2-dimensional (2D) convolutional layers as basic blocks to effectively learn spatial features. But the direct transplantation of 2D ConvNets are not suitable for video tasks, such as action recognition [1, 2], because of the lack of temporal information fusion. Inspired by the topological structure of 2D ConvNets, many recent works expand them

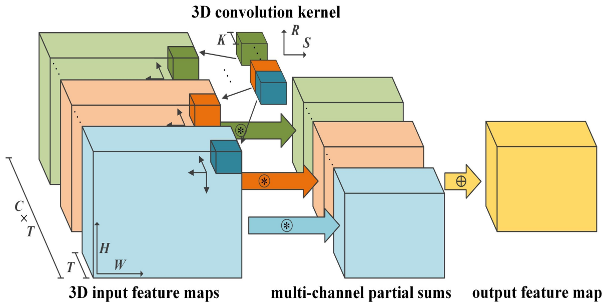


Figure 1. **Illustration of a convolutional layer**, in which multi-channel input feature maps are convolved with one convolution kernel along spatial and temporal dimensions, yielding one output feature map.

to 3-dimensional (3D) ones—an extra temporal dimension is added—to handle with the end-to-end video tasks. As shown in Fig. 1, it illustrates a 3D convolutional layer with multiple input channels and only one output channels. It can be seen that the 3D convolution kernel jointly modeling the spatio-temporal information by drifting along both pixel and frame orientations.

In the work of C3D [21], it explored convolution kernels with size of $3\times3\times3$, which is empirically proved to work best for all layers among limited architecture exploration. However, models that simply stacking 3D convolutional blocks layer by layer cannot reach the similar accuracy with respect to their 2D counterparts for image classification tasks [9], even though the datasets to be trained on for the former are far small than that of the latter (ImageNet). Hence, many works start to rethink of the effectiveness of 3D ConvNet. In [29], it is argued that the high model complexity attributed to massive 3D convolution kernels hinders the model itself from converging to an optimal local minimum. On the other hand, the incredible large model size of 3D ConvNets brings considerable memory access power consumption, and the computation complexity grows exponentially with regard to the model size, which will require more computing power and more training time. Although video recognition applications under resource-restrict scenarios are on a more urgent demand, the above two factors make it more difficult to train a deeper 3D-kernel-stacked ConvNet, in addition, show few possibilities to deploy such models on portable or edge devices. As a result, it is highly desired that a compact and computational efficient 3D ConvNet, which only consumes limited computing resources and reach smaller model size.

In this paper, we propose a two-stage fully separable block (FSB) as a basic block to construct a compact 3D ConvNet. It can be employed to significantly compress the size of naive 3D models and that of already compact-designed ones *e.g.* SlowFast network, while only resulting in slight accuracy loss on various typical video recognition tasks. Nevertheless the FSB can considerably shrink the model size, the massive operations brought by the computationally intensive convolutions is still unacceptable for the deployment of video-tasks-aimed models on budget-restricted platforms. Since the convolution operations dominate the overall computation, we propose a hybrid Fast Algorithm (hFA) to significantly reduce the computation complexity. Moreover, a feature enhancement approach named temporal residual gradient is proposed to improve the model performance on video tasks, providing higher accuracy, faster convergency and better robustness. Based on an effective coupling of these approaches, we design a light-weight and efficient network for video tasks. Experimental results show that the proposed model achieves an overwhelming performance of $2.3\times$ compression rate, $6.8\times$ workload reduction, and 2% top-1 accuracy gain, over the state-of-the-art compact-designed SlowFast network [5]. We also evaluate our methods on the conventional C3D model [21] to verify their adaptability, leading to $5\times$ compression rate, $10\times$ workload decreasing, and 3% accuracy gain.

2. Related Work

3D ConvNets and Compact approaches. Many recent works [11, 22] proposed 3D ConvNet models for human action recognition. Based on a limited exploration of the architecture design space, *Tran et al.* [21] proposed a more powerful network, in which it claimed that the convolution kernels with size of $3 \times 3 \times 3$ are empirically optimal for all layers. Hence, many follow-up works preserved the backbone of the $3 \times 3 \times 3$ kernels with peripheral modifications [I3D]. However, the exponential increasing scale on model complexity made it harder to train a deeper 3D ConvNets, because the optimal converge is less possible to be found. Hence, some recent works started to rethink the necessity of the employment of such computationally intensive operation block. *Zhou et al.* [29] argued that learning spatio-temporal information by simply stacking 3D convolutions layer by layer hindered the SGD optimizer from converging to the global optimum, since the parameter searching space exponentially increased as the model went deeper. So they cascaded 2D convolutions with 3D ones as compact structures to reduce the massive parameters. *Xie et al.* [28] found that separately learning the spatial and temporal features leaded to a better result by replacing the 3D convolutions with 2D/1D cascaded ones, even though the complexity of the latter was lower. He *et al.* [slowfast] also introduced this bottleneck structure to their recent presented work, which is based on the two-stream structure [6, 7, 19], achieving the state-of-the-art performance. It leveraged two path to jointly cope with the video tasks, in which the fast path was designed light-weight to learn movements, while the slow one mainly used for recognizing spatial instances.

Fast Algorithm for low-complexity ConvNets. The application of Fast Algorithms in ConvNets was first proposed by [15, 23]. Then Lavin *et al.* [12] applied another type of fast approach called Winograd Algorithm (WinoA) [27] for 2D convolutions, leading to a great compression rate on the convolutional layers. It have been integrated into the cudnn library [4] as a build-in method, which proved the superior of the WinoA in convolution implementations. Recently the 3D WinoA was also proposed to decrease the computation complexity of 3D ConvNet [18], although it could only be used for 3D kernels with same scales on temporal and spatial orientations.

3. Design Compact and Efficient Networks

In the following sections, we will give a detailed illustration of the proposed FSB structure and its heuristic motivation, as well as how it significantly compresses the state-of-the-art 3D ConvNets. Then the theoretical derivations of the Fast Algorithms are introduced . Moreover, it is presented that how these algorithms are combined to further reduce the computation workload of 3D models to an extremely low level. Due to the main idea of this section is to show the compression ability of the proposed method, more evaluations on the robustness and the performance of our approaches will be illustrated in Section 4.

3.1. Naive 3D Convolution

For the simplicity of description, in this section, we view the convolution kernels with multiple input and output channels as a 5-dimensional tensor. As shown in Fig. 1, in a naive 3D convolutional layer, an extra dimension is expanded along the temporal dimension compared to the 2D ones. It is illustrated that a convolution kernel with only one output channel learns the spatio-temporal features by convolving with the multi-channel input feature maps (ifmaps) along the vertical, horizontal and temporal directions, where $\{T, H, W\}$ and $\{K, R, S\}$ denote the temporal duration and spatial sizes of the ifmaps and kernels, respectively, and C represents the number of input channels. These convolution operations in different input channels are performed independently, then the convolved partial sums need to be accumulated throughout all channels into one output channel. Normally multiple kernels are utilized to yield certain a number of output channels, of which the number is notated as N. And the zero-padding strategy is usually used to maintain the size unchanged between clips in the ifmaps and the ofmaps. The mathematical form of the 3D convolution is as follows:

$$Y_{i,k} = D_i * G_k, \tag{1}$$

$$y_{i,k,t,x,y} = \sum_{c=1}^{C}\sum_{s=1}^{T}\sum_{v=1}^{R}\sum_{u=1}^{S} d_{i,c,t+s,x+u,y+v}\, g_{k,c,s,u,v}, \tag{2}$$

where we denote the i-th ifmap and the k-th kernel as D_i and G_k, respectively. It can be found that the computation complexity exponentially increases as the model size raising.

3.2. Temporal Residual Gradient Module

We propose a heuristic method called temporal residual gradient (TRG) that offers the model more robustness and faster convergency based on a basic priori acknowledge—the gradient between adjacent image frames indicates the motions of the objects. As shown in Fig. 2, the TRG computes the residual value between each two frame on temporal orientation to represent the movement in time domain, yielding $T - 1$ frames of gradient features. And in order to preserve the statistical distributions of the original features, mean values throughout the temporal dimension are calculated and cascaded with the residual gradient features, getting a output tensor with total T frames. Similar to the Histogram of Oriented Gradient (HOG) [14, 17] features in 2D image object detection tasks, which forms the features by computing the spatial gradients of local field, the TRG module can extract the active motion features with an extra frame of the approximate description of the action scenario. It relieves the burden of jointly learning spatio-temporal features with 3D kernels and shrinks the searching space of the optimization operator, offering the model higher performance and faster convergency. More detailed discussions on the TRG method are demonstrated in Section 4. On the other hand, the $T - 1$ frames output from TRG are performed subtractions, and the last one with the mean value can be calculated by addition and bit shifting operations, considering the number of frames it divided by is often the power of 2. So the TRG is efficient in hardware implementation.

3.3. Fully Separable Block

According to Section 3.1, the computation complexity and model size of the naive 3D ConvNet are totally unacceptable for deployment under resource-constrain applications. So we propose a compact but effective two-stage fully

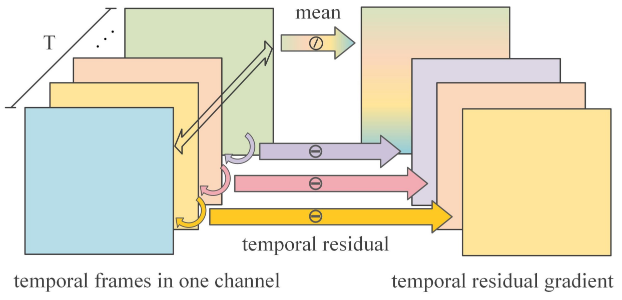


Figure 2. TRG structure

separable block structure, which can significantly slim the ConvNets yet making no harm to the accuracy. As illustrated in Fig. 3, the FSB leverages several separated small steps to gradually fuse the fine-grained spatio-temporal information. At the first stage of the FSB, a temporal bottleneck structure is hired. In this phase, the 3D convolution kernel is separated into two components, of which the first focuses on merge pixels along the temporal orientation and the second employs 2D kernels to learn spatial features of different frames. As for the second stage, we further split the channels apart, each of which convolved with a 2D kernel without inter-channel communication, because we then leave them to the last step, where $1 \times 1 \times 1$ kernels are used to jointly accomplish both spatial and temporal information fusions, as well as the output channel reshaping.

In addition to a heuristic description of the topological structure, we then give a detailed analysis about the considerable compression rate achieved by replacing the naive 3D convolutional layer with the proposed FSB. Considering a convolutional layer that receives C input clips with size of $\{T, H, W\}$ and needs to yield ofmaps consisted of N channels, it rules the shape of kernels as $\{N, C, K, R, S\}$, leading to memory requirement of $N \times C \times K \times R \times S$. As for the FSB, in the first stage, a bottleneck structure—kernel with size of $\{\alpha N, C, K, 1, 1\}$—is employed to focus on the temporal fusion, where α denotes the intermediate expansion rate. Next, a $\{\alpha N, 1, 1, R, S\}$ kernel with αN groups is hired to learn intra-channel spatial features, which means each group only need to be convolved with one corresponding channel. At last, a 3D point-wise convolution is performed via a $\{N, \alpha N, 1, 1, 1\}$ to finally combine the spatio-temporal information separably located in different channels, in addition, to reshape the output tensor to fit the input size constrain of the next layer. As a result, a convolutional layer that leveraging the proposed FSB merely consumes total memory footprint of $\alpha NCK + \alpha NRS + \alpha N^2$. Considering a typical case of $\{N, C, K, R, S\} = \{64, 64, 3, 3, 3\}$, and we set the α to 1 in this paper, the FSB can compress a convolutional layer by $6.5\times$.

As the stacked 3D layers in a naive 3D ConvNet like C3D [21] can be easily replaced, the FSB-based model can

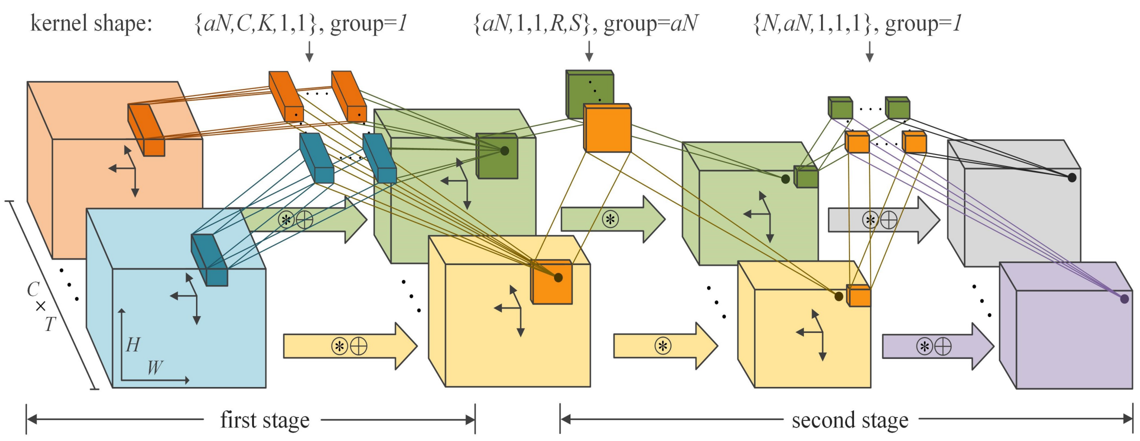


Figure 3. **Illustration of the FSB structure**, where no zero-padding is used for better understanding of the dataflow.

achieve great compression rate. Moreover, even for an already compactly designed network, *e.g.* the state-of-the-art SlowFast network [5], the compression performance can still be improved by replacing the 3D residual blocks with our FSBs. Detailed compression of different 3D ConvNets on model size are listed in 3.

3.4. Efficient Hybrid Fast Algorithm

Even though the proposed FSB leads to considerably small model size, the overall computation complexity of these models still remains at a high level due to its reliability not only on the sizes of weights but also that of activations, the shape of which is determined by the fixed topological architecture and unlikely to be compressed. Hence, a hybrid Fast Algorithm is proposed to directly decrease the computation strength of convolutions in the FSB, as we refer to the number of multiplication operations as an evaluation of the model computation complexity, since the multiplications far more implementation-expensive than additions and they dominate the computational time and power of the deployment of a ConvNet [25]. In the following subsections the Fast Algorithms are theoretically introduced, and we will propose a hybrid dataflow, which couples these methods, to incredibly reduce the computation complexity of a 3D convolutional layer.

3.4.1 Fast Algorithm

Fast Algorithm includes a cluster of computation strength reduction methods, which are initially introduced and exploited in the signal processing field. Several typical algorithms were applied to improve the convolution computation efficiency, such as the Fast Fourier Transform (FFT) [15], the Fast FIR [16, 24], and the WinoA [12], so they could be implemented in the ConvNets naturally. Another work named Strassen Algorithm was introduced to reduce the multiplication operations of the matrix multiplication. Considering the regularity and adaptability of the dataflow, We propose the hybrid Fast Algorithm based on the WinoA in this paper.

1D WinoA. The WinoA is first introduce to reduce the FIR filter complexity. For a r-tap FIR aimed at m outputs, which can be viewed as a 1-dimensional (1D) convolution, The traditional convolution can be written in matrix form as

$$Y = g * d, \tag{3}$$

where the notation $*$ represents convolution operation. The 1D WinoA, which is denoted as $F(m, r)$, re-express the Eq. 3 as follows,

$$Y = A^T[(Gg) \odot (B^T d)], \tag{4}$$

where $\odot$ indicates the element-wise matrix multiplication (EWMM), and the A, G, and B denote the transform matrices. What should be mentioned is that these transform matrix are only consisted of 0, 1, and values of powers of 2, which can be implemented using low-cost bit-shift operations, so multiplications are only performed in the EWMM operations. As the g and d are both transformed into the shape of Y by G and B, separately, the number of multiplications equals to the order of matrix Gg (*i.e.* $m + r - 1$). Hence, for a 1D convolution, the computation complexity can be reduced from $m \times r$ to $m + r - 1$. One can refer to [12] for a more detailed theoretical derivation. In the first stage of the FSB, all the bar stretched along the temporal direction within a channel clip can employ the 1D WinoA.

2D WinoA, denoted as $F(m\times m, r\times r)$, is performed on a local 2D block, which convolves a $(r \times r)$ kernel g with a input tile d, yielding an output tile Y with size of $(m \times m)$. Hence, the input d should be a $(m+r-1)\times(m+r-1)$ tile. By applying the 2D WinoA, the 2D convolution operation can be re-expressed as

$$Y = A^T[(GgG^T) \odot (B^T dB)]A, \tag{5}$$

so the 2D WinoA thus only requires $(m + r - 1)^2$ multiplications, while the traditional convolution needs $m^2 \times r^2$. In the second stage of the FSB, the 2D WinoA is employed to compress the fully separated 2D kernels by iteratively convolving them with input tiles in each temporal frame with $(r - 1, r - 1)$ strides.

3D WinoA. 3D WinoA is recently proposed in [18]. Similarly, it is represented as $F(m \times m \times m, r \times r \times r)$. The theoretical representation of the 3D WinoA can be heuristically derived by expanding the Eq. 5 into 3 dimensions as shown in Eq. 6,

$$Y = \{A^T[(GgG^T)^R G^T \odot (B^T dB)^R B]\}^R A, \tag{6}$$

where operator R means transposition between the 2-nd and 3-rd dimensions. This method only slightly improves the computation complexity reduction compared with its 2D counterpart ($< 15\%$), but it will incredibly increase the transform operations. Otherwise, it can only be used by 3D kernels, which is lack of adaptability.

3.4.2 hybrid Fast Algorithm

The 3D WinoA is introduced to make a comparison of our proposed compression method with the state-of-the-art 3D WinoA. Since a 3D convolution kernel is decomposed into small ones with different in the FSB, we need to exploit hybrid types of the Fast Algorithms to efficiently cope with all kinds of decoupled kernels within the fine-grained stages of the FSB. At the first stage, a temporal kernel with scale of $K \times 1 \times 1$ can utilize the $F(m_1, K)$, so it results in a complexity reduction by $\frac{m_1K - m_1 - K - 1}{m_1 K}$ times. As for the second stage, one can apply the $F(m_2 \times m_2, R \times R)$ to the $1 \times R \times R$ separated kernels to reach $\frac{m_2^2 R^2}{(m_2+R-1)^2}$ times of computation compression. So the hFA can be represented as $F(m_1, K) \otimes F(m_2 \times m_2, R \times R)$, where $\otimes$ denotes concatenation. We evaluate the hFA in Table. 1, in which it can be concluded that this dedicated algorithm is more adaptive for mainstream models and more superior in the computation complexity reduction.

Method	Number of multiplications	
	C3D [21]	SlowFast [5]
Baseline	199.9G	145.5G
3D Winograd []	25.0G	-
FSB-only	27.6G	68.7G
FSB+hFA	**18.1G**	**40.7G**

Table 1. Computation complexity of different models. Evaluations are performed on UCF-101 with input size of $64 \times 112 \times 112$.

4. Experiments

In this section, we first give a brief introduction about the evaluation datasets, and then some experimental settings are discussed. A comprehensive design space exploration is performed to decide the optimal structure of the FSB and to find out the robust strategy of its combination with the mainstream models. We draw a conclusion from the experimental results that with use of the proposed FSB and hFA methods, the 3D ConvNet models can achieve the similar performance on complicated video tasks, while it only costs incredibly low memory footprint and computation complexity, compared with their original editions.

4.1. Evaluation Settings

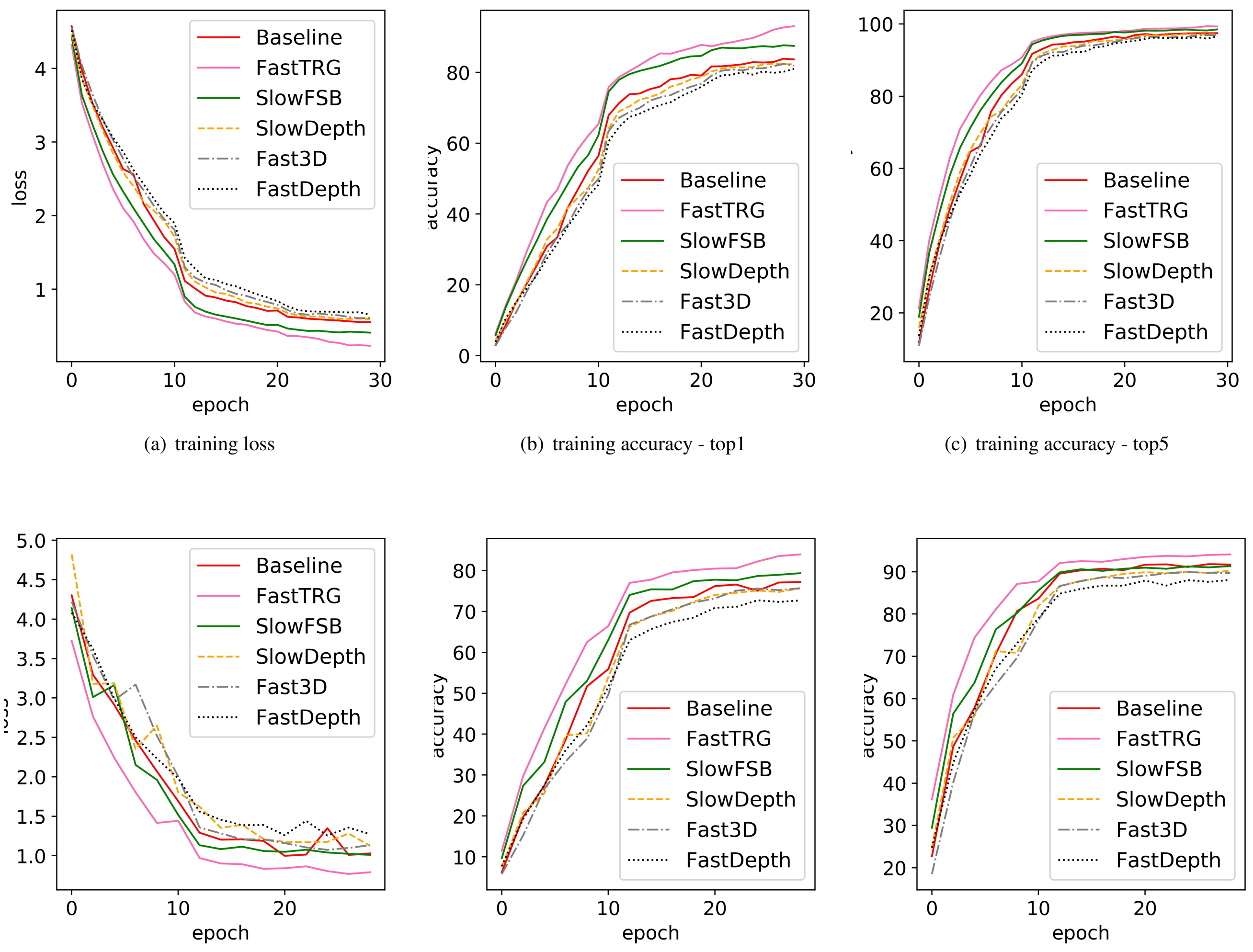
We evaluate the proposed methods on two representative ConvNet prototypes, which are the C3D, a naive 3D-kernel-stacked network, and the SlowFast network, a compact-designed model. In the experiments, all models are training on the widely-used benchmark UCF-101 [20] from scratch without any pre-train. And the optimization phases all use SGD optimizer with momentum on signal GPU for fair comparisons. The UCF-101 is one of the most popular datasets among the video recognition tasks, which contains 13,320 labeled action videos in total with classification of 101 categories. It provide three split lists for dividing the dataset into train/test subsets. In order to evaluate the representative capacity of the proposed model and the baseline, we increase the training and validation sets by ensembling two subsets.

4.2. Design Space Exploration on SlowFast

Our work mainly draw inspirations from the state-of-the-art SlowFast network, so we will analysis the difference in detail and explain why the FSB can further improve the performance. Moreover, several variants of the FSB, based on the SlowFast network, are explored to find the optimal topological structure.

The SlowFast network separates a traditional $3 \times 3 \times 3$ kernels into a 3-step residual bottleneck structure, which can fuse spatio-temporal information step by step, leading to a more accurate learning process and a more precisely route to the optimal minimum, while taking the advantage of low memory consumption of the bottleneck structure. Following this tendency, we believe that further separating the channel merging process may result in a more fine-grained structure suitable for better convergency and, meanwhile, achieve a more compact model. According to this priori hypothesis, we explore the design space of the FSB structure for an optimal performance and then evaluate these variants on UCF-101 dataset, as shown in Fig. 4.

SlowDepth for compression with slight accuracy loss. Considering that the SlowFast network is based on the two-stream method, in which one light-weight path operates



(a) training loss (b) training accuracy - top1 (c) training accuracy - top5

(d) validation loss (e) validation accuracy - top1 (f) validation accuracy - top5

Figure 4. **Design space exploration.** It is strongly recommended to be viewed in color.

Model	training				validation			
	top-1 acc	top-1 gain	top-5 acc	top-5 gain	top-1 acc	top-1 gain	top-5 acc	top-5 gain
Baseline [5]	84.1%	-	97.8%	-	77.6%	-	91.8%	-
FastTRG	92.7%	**+8.6%**	99.4%	**+1.4%**	83.9%	**+6.3%**	94.1%	**+2.3%**
SlowFSB	87.7%	+3.6%	98.5%	+0.7%	79.4%	+1.8%	91.3%	-0.5%
SlowDepth	83.4%	-0.7%	97.3%	-0.5%	75.8%	-1.8%	90.4%	-1.4%
FastDepth	81.7%	-2.4%	96.7%	-1.1%	73.5%	-4.1%	88.2%	-3.6%
Fast3D	83.4%	-0.7%	97.3%	-0.5%	76.4%	-1.2%	90.0%	-1.8%

Table 2. Model Performance on variants.

at a high frame rate, which means to capture the temporal features (*i.e.* motions), and another path works at a low frame rate to model the spatial information. We suppose that the slow path mainly play a role of capturing features from still images and providing the object instances information to support the final action recognition, but the slow path dominates the overall memory footprint of the network ($\approx$ 98%). According to the general perspective of many works that focus on model compression of 2D ConvNets [10, 26], we believe that there will be considerable redundancy in the slow path, so a channel bottleneck structure are introduced, which is composed of a depth-wise 3×3 kernel and a point-wise 1×1 one, to replace the $1 \times 3 \times 3$ kernel for further reduction of the model size. This variant is denoted as ***SlowDepth*** in Fig. 4. It can be conclude that this method slightly hurt the final model accuracy of around 1% (training(top1/top5): from 84.1%/97.8% to 83.4%/97.3%; validation(top1/top5): from 77.6%/91.8% to 75.8%/90.4%), while it is achieved that a $1.9\times$ compression rate in model size over the original SlowFast network, which is also trained on UCF-101 as a baseline. We have compared six variants with the baseline as listed in Table. 2.

SlowFSB for better accuracy. But we analyse the micro architecture of the SlowDepth and find out that the 2D point-wise convolution 1×1 serves the same purpose of the 3D $1 \times 1 \times 1$ kernel in the second stage as shown in Fig. 3. So we suppose that, not only it is redundant and can be removed, but also may it make the network deeper and optimization searching space wider, which hinder itself from converging to an optimal local minimum. Hence, we design another variant structure named ***SlowFSB*** by resecting the redundant point-wise convolution and merging the duty of reshaping the output channel to the 3D $1 \times 1 \times 1$ kernel at the final step. From the evaluation result, it can be found that, the SlowFSB model is superior to the baseline, especially with respect to the loss and top-1 accuracy of training/validation. It is indicated that our structure is more powerful in fine-grained feature learning, considering that the improvement on top-1 accuracy needs more precise features to distinguish the subtle details. For the partial separated method in SlowFast network, the first stage employ a $3 \times 1 \times 1$ kernel to convolve data along the time direction, so each channel in the ofmap sent to the next step already contains the temporal fusion information. On the other hand, $1 \times 3 \times 3$ kernels at the second stage should be focused on learning the spatial features of image instances. Merging data from different channels here will increase the intrinsic learning workload of the spatial kernels, and disarrange the proper fusion of the spatio-temporal information with $1 \times 1 \times 1$ kernels at the final step. Hence, the proposed fully separable structure can effectively learn the spatial and temporal features step by step without disruptions, thus leading to a higher accuracy on top-1 task (training(top1/top5): from 84.1%/97.8% to 87.7%/98.5%; validation(top1/top5): from 77.6%/91.8% to 79.4%/91.3%). It even achieves a more compact model with a $2.3\times$ compression rate. So we employ this variant prototype as the final FSB structure.

FastTRG for fast and robust learning. Since we have significantly compressed the model size and even achieved a better classification accuracy by introducing the FSB structure to the slow path, it is wondering that if the fast path can be further improved or not. From the experiment results, it is observed that utilizing temporal convolutions in the earlier layers leads to an accuracy degradation, which is argued to be caused by those action cases that moves fast. We suppose it is because in these cases the correlation between adjacent frames is too small, so it makes the fast path harder to learned the action motions precisely. Inspired by the HOG feature [14] extractor widely used in 2D image object detection tasks, which forms the features by computing the spatial gradients of local field, we propose a heuristic method called TRG, shown in Fig. 2, to extract temporal motion features. Intuitively, the gradient between adjacent image frames indicates the motions of the objects. We analyse the feature maps from one middle layer of the fast path, and find out that the intrinsic concept of TRG is still convincible in the early layer. We evaluate the variant, named as ***FastTRG***, which employs the TRG in the fast path and exploits the slow path of the SlowFSB. From the experiment results in Fig. 4, it can be concluded that the FastTRG model shows overall improvements over the SlowFSB and far more superiorities than the baseline, no matter in training or validation. According to the validation curves of loss, top-1 and top-5 accuracy, the TRG significantly improves the performance, which means it has better generalization ability and more robustness, in other words, it learns features closer to the real distribution of data in the video tasks. Moreover, both in training and validation phases, the FastTRG converges faster. In the early stage of learning process, it almost reaches a 10% accuracy gap over the baseline at the same epoch. Meanwhile, considering that the fast path only costs negligible computation operations ($< 3\%$), it can be concluded that the proposed TRG offers more robustness and faster convergency at little cost. We also compared the proposed FastTRG with other popular models, w.r.t. their model size and accuracy, as shown in Table. 3.

Model	Pre-train	Accuracy	Model Size
C3D [21]	from scratch	47.9%	27.7M
FSB-C3D	from scratch	47.5%	**3.7M**
MiCT [29]	from scratch	56.5%	19.1M
MiCT	ImageNet	85.1%	19.1M
SlowFast [5]	from scratch	91.5%	22.46M
FastTRG	from scratch	**94.1**%	**9.9M**

Table 3. **Comparisons of model size and accuracy between the proposed FastTRG and other baselines on UCF-101.** The accuracy data of C3D and MiCT refer to [29]

Although the TRG method is based on the priori acknowledge of the video tasks and forces the model to learn some heuristic manual-designed features, it unveils some intrinsic concepts of how the video-aimed 3D ConvNet actually learns and works. It also provides a instructive guideline for future works on visual tasks that a manual feature method can be applied to the end-to-end ConvNet approach and used to improve its performance. On the other hand, from the view of model compression, the TRG will introduce more sparsity to the dataflow, which can be leveraged for accelerating the inferences of video applications by sparse-dedicated hardware accelerator.

Other variants. There are two more variants for comparisons. The first one, noted as ***Fast3D***, shows slightly low accuracy than the baseline. It replaces the residual bottleneck structure in the fast path with traditional 3D $3 \times 3 \times 3$ kernels. So it is concluded that simply increasing the model capacity of the fast path can not improve the model accuracy, on the contrary, it proves that the TRG method in the FastTRG variant actually helps the model to learn some useful features. As for the second variant, named as ***FastDepth***, it only applies the FSB structure to the fast path. It can be seen that the compressed fast path results in considerable accuracy loss, despite of its incredibly low proportion of the entire model ($< 2\%$). It demonstrates that the fast path plays an important role in the final model accuracy and a little wrong modification on its structure may cause great accuracy loss. On the other hand, the curve further proves the efficiency of the TRG method, and shows the correctness of the efforts we made on compressing the slow path, which indeed has sufficient redundancy and low correlation with the final accuracy.

4.3. Validation on C3D Network

We also evaluate the C3D ConvNet modified by the proposed compression methods on the UCF-101 to prove their adaptivity. The architecture is modified by replacing all $3 \times 3 \times 3$ convolution kernels with our FSB structures, as shown in Table. 4. Because the C3D network is simply constructed by stacking convolution layers, in which batch normalization layers are not utilized. However, intensive residual operations in the TRG will result in significant network degradation when coupled with the ReLU function, *i.e.* gradient vanishing problem. So we only employ the TRG modules in the first two layers, without batch normalization operation between blocks for fair comparison. We denote the model modified with our methods as ***FSB-C3D***. The evaluation results illustrated in Fig. 5 shows that the FSB-C3D reaches a better validation accuracy as well as $5\times$ compression rate on convolutional layers. We argue that the performance gap between the training curves results from the severe overfitting of the C3D baseline, on the contrary, the FSB-C3D imposes strong restriction onto the architecture, forcing the model to learn intrinsic data distribution of video tasks instead of that of the training set.

5. Conclusion

In this work, we first propose a light-weight convolutional block called FSB to efficiently extract the spatio-temporal features, which can be employed to construct a extremely compact 3D ConvNet for video tasks. Evaluations on the C3D model demonstrates the effectiveness of the FSB in significantly compressing the model size by $5\times$ with even better validation accuracy, which indicates the superior modeling capacity of the proposed FSB. Furthermore, inspired by the HOG method in image feature extraction, we propose a temporal feature enhancement approach termed TRG, which remarkably improves the model performance. It is shown that the TRG offers higher accuracy, faster convergency and better robustness, while only costs negligible addition and shifting operations, which are hardware-efficient. On the other hand, we also propose a computation complexity reduction method named hFA to cooperate with the FSB for further decreasing the multiplications in a 3D ConvNet. Based on these comprehensive optimization methods, we design a light-weight network for video tasks, which shows overwhelming performance above the state-of-the-art networks. Our work indicates that there is still considerable redundancy even in the current compact-designed models, meanwhile, it also illuminates that heuristic methods like traditional feature engineering can be embedded in the popular end-to-end ConvNet models and improve the performance on specific task.

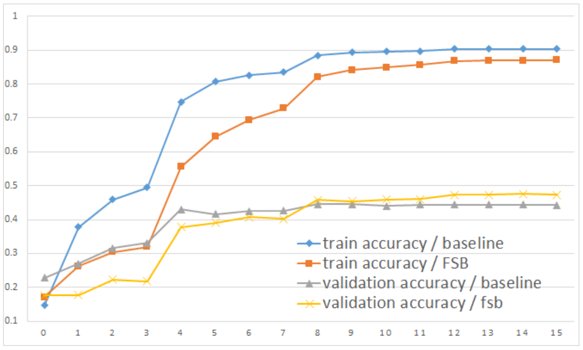


Figure 5. Verify the proposed methods on C3D.

Type	FSB-C3D	C3D [21]	Rate
$conv_1$	FSB$_{group=64}$	$[64, 3]$	$1.0\times$
$conv_2$	+TRG	$[128, 64]$	$10.5\times$
$conv_3$	FSB$_{group=128}$	$[256, 128]$	$10.7\times$
$conv_4$	FSB$_{group=256}$	$[256, 256]$	$13.3\times$
$conv_5$	FSB$_{group=256}$	$[512, 256]$	$10.7\times$
$conv_6$	FSB$_{group=512}$	$[512, 512] \times 3$	$6.7\times$
Total	27.7M	**3.7M**	**7.4**$\times$

Table 4. Architecture of FSB-C3D and compression rate of each layer over the C3D baseline.

References

[1] M. Baccouche, F. Mamalet, C. Wolf, C. Garcia, and A. Baskurt. Action classification in soccer videos with long short-term memory recurrent neural networks. In *International Conference on Artificial Neural Networks*, pages 154–159. Springer, 2010.

[2] M. Baccouche, F. Mamalet, C. Wolf, C. Garcia, and A. Baskurt. Sequential deep learning for human action recognition. In *International workshop on human behavior understanding*, pages 29–39. Springer, 2011.

[3] L.-C. Chen, G. Papandreou, I. Kokkinos, K. Murphy, and A. L. Yuille. Semantic image segmentation with deep convolutional nets and fully connected crfs. *arXiv preprint arXiv:1412.7062*, 2014.

[4] S. Chetlur, C. Woolley, P. Vandermersch, J. Cohen, J. Tran, B. Catanzaro, and E. Shelhamer. cudnn: Efficient primitives for deep learning. *arXiv preprint arXiv:1410.0759*, 2014.

[5] C. Feichtenhofer, H. Fan, J. Malik, and K. He. Slowfast networks for video recognition. *arXiv preprint arXiv:1812.03982*, 2018.

[6] C. Feichtenhofer, A. Pinz, and R. Wildes. Spatiotemporal residual networks for video action recognition. In *Advances in neural information processing systems*, pages 3468–3476, 2016.

[7] C. Feichtenhofer, A. Pinz, and A. Zisserman. Convolutional two-stream network fusion for video action recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 1933–1941, 2016.

[8] R. Girshick. Fast r-cnn. In *Proceedings of the IEEE international conference on computer vision*, pages 1440–1448, 2015.

[9] K. He, X. Zhang, S. Ren, and J. Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 770–778, 2016.

[10] F. N. Iandola, S. Han, M. W. Moskewicz, K. Ashraf, W. J. Dally, and K. Keutzer. Squeezenet: Alexnet-level accuracy with 50x fewer parameters and¡ 0.5 mb model size. *arXiv preprint arXiv:1602.07360*, 2016.

[11] S. Ji, W. Xu, M. Yang, and K. Yu. 3d convolutional neural networks for human action recognition. *IEEE transactions on pattern analysis and machine intelligence*, 35(1):221–231, 2013.

[12] A. Lavin and S. Gray. Fast algorithms for convolutional neural networks. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 4013–4021, 2016.

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

[14] D. F. Llorca, R. Arroyo, and M. A. Sotelo. Vehicle logo recognition in traffic images using hog features and svm. In *16th International IEEE Conference on Intelligent Transportation Systems (ITSC 2013)*, pages 2229–2234. IEEE, 2013.

[15] M. Mathieu, M. Henaff, and Y. Lecun. Fast training of convolutional networks through ffts. In *International Conference on Learning Representations (ICLR2014), CBLS, April 2014*, 2014.

[16] Z.-J. Mou and P. Duhamel. Fast fir filtering: Algorithms and implementations. *Signal Processing*, 13(4):377–384, 1987.

[17] Y. Pang, Y. Yuan, X. Li, and J. Pan. Efficient hog human detection. *Signal Processing*, 91(4):773–781, 2011.

[18] J. Shen, Y. Huang, Z. Wang, Y. Qiao, M. Wen, and C. Zhang. Towards a uniform template-based architecture for accelerating 2d and 3d cnns on fpga. In *Proceedings of the 2018 ACM/SIGDA International Symposium on Field-Programmable Gate Arrays*, pages 97–106. ACM, 2018.

[19] K. Simonyan and A. Zisserman. Two-stream convolutional networks for action recognition in videos. In *Advances in neural information processing systems*, pages 568–576, 2014.

[20] K. Soomro, A. R. Zamir, and M. Shah. Ucf101: A dataset of 101 human actions classes from videos in the wild. *arXiv preprint arXiv:1212.0402*, 2012.

[21] D. Tran, L. Bourdev, R. Fergus, L. Torresani, and M. Paluri. Learning spatiotemporal features with 3d convolutional networks. In *Proceedings of the IEEE international conference on computer vision*, pages 4489–4497, 2015.

[22] G. Varol, I. Laptev, and C. Schmid. Long-term temporal convolutions for action recognition. *IEEE transactions on pattern analysis and machine intelligence*, 40(6):1510–1517, 2018.

[23] N. Vasilache, J. Johnson, M. Mathieu, S. Chintala, S. Piantino, and Y. LeCun. Fast convolutional nets with fbfft: A gpu performance evaluation. *arXiv preprint arXiv:1412.7580*, 2014.

[24] H. Wang, J. Lin, Y. Xie, B. Yuan, and Z. Wang. Efficient reconfigurable hardware core for convolutional neural networks. In *2018 52nd Asilomar Conference on Signals, Systems, and Computers*, pages 777–781. IEEE, 2018.

[25] J. Wang, J. Lin, and Z. Wang. Efficient convolution architectures for convolutional neural network. In *2016 8th International Conference on Wireless Communications & Signal Processing (WCSP)*, pages 1–5. IEEE, 2016.

[26] Z. Wang, F. Sun, J. Lin, Z. Wang, and B. Yuan. Sgad: Soft-guided adaptively-dropped neural network. *arXiv preprint arXiv:1807.01430*, 2018.

[27] S. Winograd. *Arithmetic complexity of computations*, volume 33. Siam, 1980.

[28] S. Xie, C. Sun, J. Huang, Z. Tu, and K. Murphy. Rethinking spatiotemporal feature learning: Speed-accuracy trade-offs in video classification. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 305–321, 2018.

[29] 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.