

ACTION-Net: Multipath Excitation for Action Recognition

Zhengwei Wang¹ Qi She² Aljosa Smolic¹

¹V-SENSE, Trinity College Dublin, Ireland ²ByteDance AI Lab, China

{zhengwei.wang, SMOLICA}@tcd.ie, sheqi1991@gmail.com

Abstract

Spatial-temporal, channel-wise, and motion patterns are three complementary and crucial types of information for video action recognition. Conventional 2D CNNs are computationally cheap but cannot catch temporal relationships; 3D CNNs can achieve good performance but are computationally intensive. In this work, we tackle this dilemma by designing a generic and effective module that can be embedded into 2D CNNs. To this end, we propose a spAtio-temporal, Channel and moTion excitATION (ACTION) module consisting of three paths: Spatio-Temporal Excitation (STE) path, Channel Excitation (CE) path, and Motion Excitation (ME) path. The STE path employs one channel 3D convolution to characterize spatio-temporal representation. The CE path adaptively recalibrates channel-wise feature responses by explicitly modeling interdependencies between channels in terms of the temporal aspect. The ME path calculates feature-level temporal differences, which is then utilized to excite motion-sensitive channels. We equip 2D CNNs with the proposed ACTION module to form a simple yet effective ACTION-Net with very limited extra computational cost. ACTION-Net is demonstrated by consistently outperforming 2D CNN counterparts on three backbones (i.e., ResNet-50, MobileNet V2 and BNInception) employing three datasets (i.e., Something-Something V2, Jester, and EgoGesture). Codes are available at <https://github.com/V-Sense/ACTION-Net>.

1. Introduction

Video understanding has attracted an increasing amount of interest, since it is a crucial step towards real-world applications, such as Virtual Reality/Augmented Reality (VR/AR) and video-sharing social networking services. For instance, millions of videos are uploaded to TikTok, Douyin, and Xigua Video to be processed everyday,

This work is financially supported by Science Foundation Ireland (SFI) under the Grant Number 15/RP/2776. We gratefully acknowledge the support of NVIDIA Corporation with the donation of the NVIDIA DGX station used for this research.

wherein video understanding acts a pivotal part. However, the explosive growth in this video streaming gives rise to challenges on performing video understanding at high accuracy and low computation cost.

Action recognition, a fundamental problem in video understanding, has been a growing demand in video-related applications, such as content moderation (i.e., recognize content in videos that break terms of service) and content recommendations (i.e., videos are ranked by most liked and recommended to similar customers). The complex actions in videos are normally temporal-dependent, which do not only contain spatial information within each frame but also include temporal information over a duration. For example, symmetric action pairs ('opening a box', 'closing a box'), ('rotate fists clockwise', 'rotate fists counterclockwise') contain similar features in spatial domains, but the temporal information is completely reversed. Traditional human action recognition is more *scene-related* [36, 19, 16], wherein actions are not as temporal-dependent e.g., 'apply eye makeup', 'walking', 'running'. With how rapid technology is developing, like VR, which requires employing gestures to interact with environments, *temporal-related* action recognition has recently become a focus for research.

The mainstreams of existing methods are 3D CNN-based frameworks and 2D CNN-based frameworks. 3D CNNs have been shown to be effective in terms of spatio-temporal modeling [39, 4, 37], but spatio-temporal modeling is unable to capture adequate information contained in videos. The two-stream architecture was proposed to take spatio-temporal information and optical flow into account [35, 3, 33], which boosted performance significantly compared to the one-stream architecture. However, computation on optical flow is very expensive, which poses challenges on real-world applications. 3D CNNs suffer from problems including overfitting and slow convergence [8]. With more large-scale datasets being released, such as Kinetics [3], Moments in Time [29] and ActivityNet [2], optimizing 3D CNNs becomes much easier and more popular. However, heavy computations inherent in 3D CNN-based frameworks contribute to slow inferences, which would limit their deployment on real-world applications,

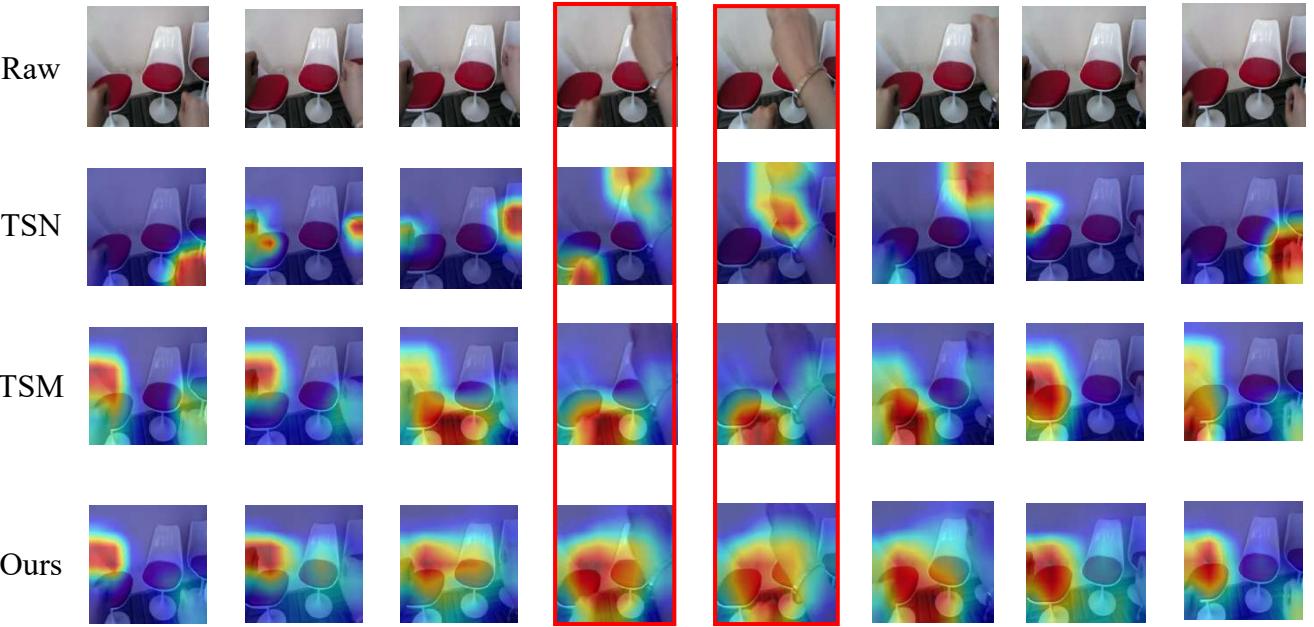


Figure 1: Visualization for significant features extracted by TSN, TSM and our ACTION-Net for the action ‘*Rotate fists counterclockwise*’. Features extracted by each method are visualized by using CAM [48]. Compared to TSN and TSM, it can be noticed that ACTION-Net is able to extract features that are related to movements in an action especially for highlighted frames i.e., 4th and 5th columns. More examples can be referred to *Supplementary Materials*.

such as VR that relies on online video recognition. Current 2D CNN-based frameworks [15, 35, 41, 47, 22] enjoy lightweight and fast inferences. These approaches operated on a sequence of short snippets (known as segments) sparsely sampled from the entire video and were initially introduced in TSN [41]. Original 2D CNNs lack the ability of temporal modeling, which causes losing essential sequential information in some actions e.g., ‘*opening a box*’ vs ‘*closing a box*’. TSM [22] introduced temporal information to 2D CNN-based frameworks by shifting a part of channels on the temporal axis, which significantly improved the baseline for 2D CNN-based frameworks. However, TSM still lacks explicit temporal modeling for an action, such as motion information. Recent works [20, 14, 21, 24, 25] introduced embedded modules into 2D CNNs in terms of ResNet architecture [9], which possessed the capability for motion modeling. In order to capture multi-type information contained by videos, previous works normally operated on input-level frames. For instance, SlowFast networks sampled raw videos at multiple rates to characterize slow and fast actions; two-stream networks utilized pre-computed optical flow for reasoning motion information. This kind of approaches commonly require multi-branch networks, which need expensive computations.

Inspired by the aforementioned observation, we propose a new *plug-and-play* and *lightweight* spatio-temporal, Channel and motion excitatIOn (ACTION) module to ef-

fectively process the multi-type information on the feature level inside a single network by adopting multipath excitation. The combination of spatio-temporal features and motion features can be understood similarly as the two-stream architecture [35], but we model the motion inside the network based on the feature level rather than generating another type of input (e.g., optical flow [12]) for training the network, which significantly reduces computations. Inspired by SENet, the channel-wise features are extracted based on the temporal domain to characterize the channel interdependencies for the network. Correspondingly, a neural architecture equipped with such a module is dubbed ACTION-Net. ACTION comprises three components for extracting aforementioned features (1) Spatio-Temporal Excitation (STE), (2) Channel Excitation (CE) and (3) Motion Excitation (ME). Figure 1 visualizes features characterized by TSN, TSM, and ACTION-Net for the action ‘*rotate fists counterclockwise*’. It can be observed that both TSN and TSM focus on recognizing objects (two fists) independently instead of reasoning an action. Compared to TSN and TSM, our proposed ACTION-Net better characterizes an action by representing feature maps that cover the two fists, especially for the highlighted 4th and 5th columns. In a nutshell, our contributions are three-fold:

- We propose an ACTION module that works in a *plug-and-play* manner, which is able to extract appropriate spatio-temporal patterns, channel-wise features, and

motion information to recognize actions.

- A simple yet effective neural architecture referred to ACTION-Net, which we demonstrate on three backbones i.e., ResNet-50 [9], BNIception [13] and MobileNet V2 [31].
- We have conducted extensive experiments and shown our superior performances on three datasets Something-Something V2 [7], Jester [27] and EgoGesture [46].

2. Related Works

In this section, we discuss related works by taking 2D and 3D CNN-based frameworks into account, wherein SENet [11] and TEA [21] inspired us to propose the ACTION-Net.

2.1. 3D CNN-based Framework

The 3D CNN-based framework has a spatio-temporal modeling capability, which enhances model performance for video action recognition [39, 8, 39]. I3D [3] inflated the ImageNet pre-trained 2D kernels to 3D kernels for capturing spatio-temporal information. To better represent motion patterns, I3D utilized pre-computed optical flow together with RGB (also known as the two-stream architecture). SlowFast networks [5] were proposed to handle inconsistent speeds of actions in videos e.g., running and walking, which involved a slow branch and a fast branch to model slow actions and fast actions, respectively. Although 3D CNN-based approaches have achieved exciting results on several benchmark datasets, they contain massive parameters. In this case, various problems are caused, such as easily overfitting [8] and difficulty in converging [40], which pose challenges including ineffective inferences for online streaming videos in real-world applications. Although recent works [30, 40] have demonstrated that 3D convolution can be factorized to lessen computations to some extent, the computation is still much more of a burden when compared to 2D CNN-based frameworks.

2.2. 2D CNN-based Framework

TSN [41] was the first proposed framework that applied 2D CNNs for video action recognition, which introduced the concept of ‘*segment*’ to process videos i.e., extract short snippets over a long video sequence with a uniform sparse sampling scheme. However, the direct use of 2D CNNs lacks temporal modeling for video sequences. TSM [22] firstly introduced temporal modeling to 2D CNN-based frameworks, in which a shift operation for a part of channels was embedded into 2D CNNs. However, TSM lacks explicit temporal modeling for actions such as differences among neighbouring frames. Recently, several works

have proposed modules to be embedded into 2D CNNs. These modules are able to model the motion and temporal information. For instance, MFNet [20], TEINet [24] and TEA [21], which introduced this type of modules, were demonstrated to be effective on the ResNet architecture. STM [14] proposed a block for modeling the spatio-temporal and motion information instead of the ordinary residual block. GSM [38] leverages group spatial gating to control interactions in spatial-temporal decomposition.

2.3. SENet and Beyond

Hu *et al.* [11] introduced a SENet architecture. A squeeze-and-excitation (SE) block was proposed to be embedded into 2D CNNs. In this case, the learning of channel-wise features regarding image recognition tasks was enhanced by explicitly modeling channel interdependencies. To tackle this, the SE block utilized two fully connected (FC) layers in a squeeze-and-unsqueeze manner then applied a Sigmoid activation for exciting essential channel-wise features. However, it processes each image independently without considering critical information such as temporal properties for videos. To tackle this issue, TEA [21] introduces motion excitation (ME) and multiple temporal aggregation (MTA) in tandem to capture short- and long-range temporal evolution. It should be noted that MTA is specifically designed for Res2Net [6], which means TEA can only be embedded into Res2Net. Inspired by these two previous works, we first propose STE and CE modules beyond the SE module by addressing the spatio-temporal perspective and channel interdependencies in temporal dimension. The ACTION module is then constructed by assembling STE, CE and ME in a parallel manner, in which case, multi-type information in videos can be activated.

3. Design of ACTION

In this section, we are going to introduce technical details for our proposed ACTION-Net together with ACTION module. As the ACTION module consists of three sub-modules i.e., Spatio-Temporal Excitation (STE), Channel Excitation (CE) and Motion Excitation (ME), we firstly introduce these three sub-modules respectively and then give an overview on how to integrate them to an ACTION module. Notations used in this section are: N (batch size), T (number of segments), C (channels), H (height), W (width) and r (channel reduce ratio). It should be noticed that all tensors outside the ACTION module are 4D i.e., $(N \times T, C, H, W)$. We first reshape the input 4D tensor to 5D tensor (N, T, C, H, W) before feeding to the ACTION in order to enable the operation on specific dimension inside the ACTION. The 5D output tensor is then reshaped to 4D before being fed to the next 2D convolutional block.

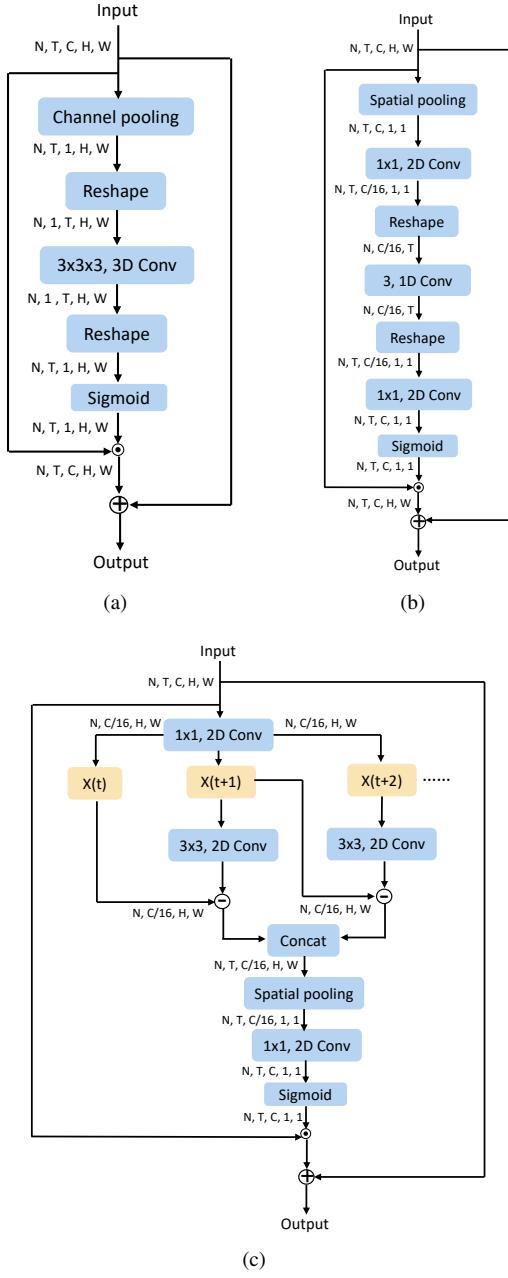


Figure 2: ACTION module consists of three sub-modules (a) Spatio-Temporal Excitation (STE) module, (b) Channel Excitation (CE) module and (c) Motion Excitation (ME) module.

3.1. Spatio-Temporal Excitation (STE)

STE is efficiently designed for exciting spatio-temporal information by utilizing 3D convolution. To achieve this, STE generates a spatio-temporal mask $\mathbf{M} \in \mathbb{R}^{N \times T \times 1 \times H \times W}$ that is used for element-wise multiplying the input $\mathbf{X} \in \mathbb{R}^{N \times T \times C \times H \times W}$ across all channels. As illustrated in Fig. 2(a), given an input $\mathbf{X} \in \mathbb{R}^{N \times T \times C \times H \times W}$,

we first average the input tensor across channels in order to get a global spatio-temporal tensor $\mathbf{F} \in \mathbb{R}^{N \times T \times 1 \times H \times W}$ with respect to the channel axis. Then we reshape \mathbf{F} to $\mathbf{F}^* \in \mathbb{R}^{N \times 1 \times T \times H \times W}$ to be fed to a 3D convolutional layer \mathbf{K} with kernel size $3 \times 3 \times 3$, which can be formulated as

$$\mathbf{F}_o^* = \mathbf{K} * \mathbf{F}^* \quad (1)$$

We finally reshape \mathbf{F}_o^* back to $\mathbf{F}_o \in \mathbb{R}^{N \times T \times 1 \times H \times W}$ and feed it to a Sigmoid activation in order to get the mask $\mathbf{M} \in \mathbb{R}^{N \times T \times 1 \times H \times W}$, which can be represented as

$$\mathbf{M} = \delta(\mathbf{F}_o) \quad (2)$$

The final output can be interpreted as

$$\mathbf{Y} = \mathbf{X} + \mathbf{X} \odot \mathbf{M} \quad (3)$$

Compared to the conventional 3D convolutional operation, STE is much more computationally efficient as the input feature \mathbf{F}^* is averaged across channels. Each channel of the input tensor \mathbf{X} can perceive the importance of spatio-temporal information from a refined feature excitation $\delta(\mathbf{F}_o)$.

3.2. Channel Excitation (CE)

CE is designed similarly to SE block [11] as shown in Fig. 2(b). The main difference between CE and SE is that we insert a 1D convolutional layer between two FC layers to characterize temporal information for channel-wise features. Concretely, given an input $\mathbf{X} \in \mathbb{R}^{N \times T \times C \times H \times W}$, we firstly access the global spatial information of the input feature by spatial average pooling the input, which can be represented as

$$\mathbf{F} = \frac{1}{H \times W} \sum_{i=1}^H \sum_{j=1}^W \mathbf{X}[:, :, :, i, j] \quad (4)$$

where $\mathbf{F} \in \mathbb{R}^{N \times T \times C \times 1 \times 1}$. We squeeze the number of channels for \mathbf{F} by a scale ratio r ($r = 16$ in this work), which can be interpreted as

$$\mathbf{F}_r = \mathbf{K}_1 * \mathbf{F} \quad (5)$$

where \mathbf{K}_1 is a 1×1 2D convolutional layer and $\mathbf{F}_r \in \mathbb{R}^{N \times T \times \frac{C}{r} \times 1 \times 1}$. We then reshape \mathbf{F}_r to $\mathbf{F}_r^* \in \mathbb{R}^{N \times \frac{C}{r} \times T \times 1 \times 1}$ to enable the temporal reasoning. A 1D convolutional layer \mathbf{K}_2 with kernel size 3 is utilized to process \mathbf{F}_r^* as

$$\mathbf{F}_{temp}^* = \mathbf{K}_2 * \mathbf{F}_r^* \quad (6)$$

where $\mathbf{F}_{temp}^* \in \mathbb{R}^{N \times \frac{C}{r} \times T \times 1 \times 1}$. \mathbf{F}_{temp}^* is then reshaped to $\mathbf{F}_{temp} \in \mathbb{R}^{N \times T \times \frac{C}{r} \times 1 \times 1}$, which is then unsqueezed by using a 1×1 2D convolutional layer \mathbf{K}_3 and fed to a Sigmoid

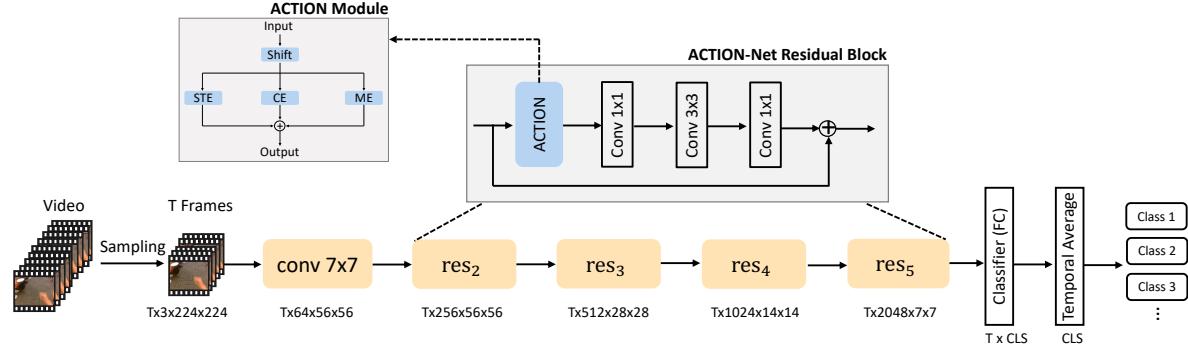


Figure 3: ACTION-Net architecture for ResNet-50 [9]. The size of output feature map is given for each layer (CLS refers to number of classes and T refers to number of segments). The input video is firstly split into T segments equally and then one frame from each segment is randomly sampled [41]. The ACTION module is inserted at the start in each residual block. Performance of different embedded locations can be referred to *Supplementary Materials B*.

activation. These are the last two steps to obtain the channel mask \mathbf{M} , which can be formulated respectively

$$\mathbf{F}_o = \mathbf{K}_3 * \mathbf{F}_{temp} \quad (7)$$

$$\mathbf{M} = \delta(\mathbf{F}_o) \quad (8)$$

where $\mathbf{F}_o \in \mathbb{R}^{N \times T \times C \times 1 \times 1}$ and $\mathbf{M} \in \mathbb{R}^{N \times T \times C \times 1 \times 1}$. Finally, the output of CE is formulated as the same as in equation (3) using the new generated mask.

3.3. Motion Excitation (ME)

ME has been explored by [14, 21] previously, which aims to model motion information based on the feature level instead of the pixel level. Different from previous work [14, 21] that proposed a whole block for extracting motion, we use the ME in parallel with two modules mentioned in previous two sections. As illustrated in Fig. 2(c), the motion information is modeled by adjacent frames. We adopt the same squeeze and unsqueeze strategy as used in the CE sub-module by using two 1×1 2D convolutional layers, which can be referred to equation (5) and equation (7) respectively. Given the feature $\mathbf{F}_r \in \mathbb{R}^{N \times T \times \frac{C}{r} \times H \times W}$ processed by the squeeze operation, motion feature is modeled following the similar operation presented in [14, 21], which can be represented as

$$\mathbf{F}_m = \mathbf{K} * \mathbf{F}_r[:, t+1, :, :, :] - \mathbf{F}_r[:, t, :, :, :] \quad (9)$$

where \mathbf{K} is a 3×3 2D convolutional layer and $\mathbf{F}_m \in \mathbb{R}^{N \times 1 \times \frac{C}{r} \times H \times W}$. The motion feature is then concatenated to each other according to the temporal dimension and 0 is padded to the last element i.e., $\mathbf{F}_M = [\mathbf{F}_m(1), \dots, \mathbf{F}_m(t-1), 0]$, $\mathbf{F}_M \in \mathbb{R}^{N \times T \times \frac{C}{r} \times H \times W}$. The \mathbf{F}_M is then processed by spatial average pooling same as in equation (4). The feature output $\mathbf{F}_o \in \mathbb{R}^{N \times T \times C \times 1 \times 1}$ and the mask $\mathbf{M} \in \mathbb{R}^{N \times T \times C \times 1 \times 1}$ can then be achieved similarly as in CE through equation (7) and equation (8) respectively.

3.4. ACTION-Net

The overall ACTION module takes the element-wise addition of three excited features generated by STE, CE and ME respectively (see ACTION module block in Fig. 3). By doing this, the output of the ACTION module can perceive information from a spatio-temporal perspective, channel interdependencies and motion. Figure 3 shows the ACTION-Net architecture for ResNet-50, wherein the ACTION module is inserted at the beginning in each residual block. It does not require any modification for original components in the block. Details of ACTION-Net architectures for MobileNet V2 and BNInception can be referred to *Supplementary Materials*.

4. Experiments

We first show that ACTION-Net is able to consistently improve the performance for 2D CNNs compared to previous two fundamental works TSN [41] and TSM [22] on three datasets i.e., EgoGesture [46], Something-Something V2 [7] and Jester [27]. We then perform extensive experiments for comparing ACTION-Net with state-of-the-arts on these three datasets. Abundant ablation studies are conducted to analyze the efficacy for each excitation path in ACTION-Net. Finally, we further compare ACTION-Net with TSM on three backbones i.e., ResNet-50, MobileNet V2 and BNInception by considering performance and efficiency (η).

4.1. Datasets

We evaluated the performance for the proposed ACTION-Net on three large-scale and widely used action recognition datasets i.e., Something-Something V2 [7], Jester [27] and EgoGesture [46]. The Something-Something V2 dataset [20, 24, 44, 49] is a large collection of humans performing actions with everyday ob-

jects. It includes 174 categories with 168,913 training videos, 24,777 validation videos and 27,157 testing videos. Jester [14, 17, 45, 28, 18] is a third-person view gesture dataset, which has a potential usage for human computer interaction. It has 27 categories with 118,562 training videos, 14,787 validation videos and 14,743 testing videos. EgoGesture [17, 43, 33, 32, 1, 34] is a large-scale dataset for egocentric hand gesture recognition recorded by a head-mounted camera, which is designed for VR/AR use cases. It involves 83 classes of gestures with 14,416 training samples, 4,768 validation samples and 4,977 testing samples.

4.2. Implementation Details

Training. We conducted our experiments on video action recognition tasks by following the same strategy mentioned in TSN [41]. Given an input video, we firstly divided it into T segments of equal duration. Then we randomly selected one frame from each segment to obtain a clip with T frames. The size of the shorter side of these frames is fixed to 256 and corner cropping and scale-jittering were utilized for data augmentation. Each cropped frame was finally resized to 224×224 , which was used for training the model. The input fed to the model is of the size $N \times T \times 3 \times 224 \times 224$, in which N is the batch size, T is the number of segments.

The models were trained on a NVIDIA DGX station with four Tesla V100 GPUs. We adopted SGD as optimizer with a momentum of 0.9 and a weight decay of 5×10^{-4} . Batch size was set as $N = 64$ when $T = 8$ and $N = 48$ when $T = 16$. Network weights were initialized using ImageNet pretrained weights. For Something-Something V2, we started with a learning rate of 0.01 and reduced it by a factor of 10 at 30, 40, 45 epochs and stopped at 50 epochs. For Jester dataset, we started with a learning rate of 0.01 and reduced it by a factor of 10 at 10, 20, 25 epochs and stopped at 30 epochs. For EgoGesture dataset, we started with a learning rate of 0.01 and reduced it by a factor of 10 at 5, 10, 15 epochs and stopped at 25 epochs.

Inference. We utilized the three-crop strategy following [14, 42, 5] for inference. We firstly scaled the shorter side to 256 for each frame and took three crops of 256×256 from scaled frames. We randomly sampled from the full-length video for 10 times. The final prediction was the averaged Softmax score for all clips.

4.3. Improving Performance of 2D CNNs

We compare ACTION-Net to two fundamental 2D CNN counterparts TSN and TSM. As illustrated in Table 1, ACTION-Net consistently outperforms these two 2D CNN counterparts on all three datasets. It is worth nothing that TSN does not contain any component that is able to model the temporal information. By employing a temporal shift operation to a part of channels, TSM introduces some temporal information to the network, which significantly im-

Table 1: ACTION-Net consistently outperforms 2D counterparts on three representative datasets. All methods use ResNet-50 as backbone and 8 frames input for the fair comparison.

Dataset	Model	Top-1	Δ Top-1	Top-5	Δ Top-5
EgoGesture*	TSN	83.1	-	97.3	-
	TSM	92.1	+ 9.0	98.3	+ 1.0
	ACTION-Net	94.2	+ 11.1	98.7	+ 1.4
SomethingV2	TSN	27.8	-	57.6	-
	TSM	58.7	+ 30.9	84.8	+ 27.2
	ACTION-Net	62.5	+ 34.7	87.3	+ 29.7
Jester	TSN	81.0	-	99.0	-
	TSM	94.4	+ 13.4	99.7	+ 0.7
	ACTION-Net	97.1	+ 16.1	99.8	+ 0.8

* We re-implement TSN and TSM using the official public code in [22].

proves the 2D CNN baseline compared to TSN. However, TSM still lacks explicit temporal modeling. By adding the ACTION module to TSN, ACTION-Net takes spatio-temporal modeling, channel interdependencies modeling and motion modeling into account. It can be noticed that the Top-1 accuracy of ACTION-Net is improved by 2%, 3.8% and 2.7% compared to TSM with respect to EgoGesture, Something-Something V2 and Jester datasets.

4.4. Comparisons with the State-of-the-Art

We compare our approach with the state-of-the-art on Something-Something V2, Jester and EgoGesture, which is summarized in Table 2. We mainly compare our approach with 2D CNN counterparts as 3D CNN-based frameworks are more favored in *scene-related* datasets e.g., Kinetics [30, 5]. The superiority of ACTION-Net on Jester and EgoGesture is quite impressive. It is clear that even ACTION-Net with 8 RGB frames as input achieves the state-of-the-art performance compared to other methods, which confirms the remarkable ability of ACTION-Net for integrating useful information from three excitation paths. In terms of Something-Something V2, ACTION-Net also achieves competitive results compared to STM and TEA. It should be noted that both STM and TEA are specifically designed for ResNet and Res2Net respectively, while our ACTION enjoys being easily equipped by other architectures i.e., MobileNet V2 and BNInception investigated in this work.

4.5. Ablation Study

In this section, we investigate the design of our ACTION-Net with respect to (1) efficacy of each excitation and (2) impact of the number of ACTION modules in ACTION-Net regarding the ResNet-50 architecture. We carry out ablation experiments using 8 frames as the input

Table 2: Comparisons with the state-of-the-arts on Something-Something V2, Jester and EgoGesture datasets.

Method	Backbone	Plug-and-play	Pretrain	Frame	Something V2		Jester		EgoGesture	
					Top-1	Top-5	Top-1	Top-5	Top-1	Top-5
C3D + RSTTM [46]	ResNet-50	✓	Scratch	16	-	-	-	-	89.3	-
C3D [17]			Kinetics	16	-	-	95.9	-	90.9	-
TRN Multiscale [22]			BNInception	ImageNet	8	48.8	77.6	95.3	-	-
MFNet-C50 [20]			ImageNet	7	-	-	96.1	99.7	-	-
TSN [22]			Kinetics	8	27.8	57.6	81.0	99.0	83.1	98.3
				16	30.0	60.5	82.3	99.2	-	-
TSM [14]			Kinetics	8	58.7	84.8	94.4	99.7	92.1	98.3
				16	61.2	86.9	95.3	99.8	-	-
GST [26]			ImageNet	8	61.6	87.2	-	-	-	-
				16	62.6	87.9	-	-	-	-
bLVNet-TAM [24]			ImageNet	32	61.7	88.1	-	-	-	-
CPNet [23]			ImageNet	24	57.7	84.0	-	-	-	-
TEINet [24]			ImageNet	8	62.7	-	-	-	-	-
				16	63.0	-	-	-	-	-
STM [14]			ImageNet	8	62.3	88.8	96.6	99.9	-	-
				16	64.2	89.8	96.7	99.9	-	-
TEA [21]			ImageNet	8	-	-	96.5	99.8	92.3	98.3
				16	64.5	89.8	96.7	99.8	92.5	98.5
ACTION-Net ¹			ImageNet	8	62.5	87.3	97.1	99.8	94.2	98.7
				16	64.0	89.3	97.1	99.9	94.4	98.8

¹ ACTION is inserted into each residual block in this experiment.

Table 3: Accuracy and model complexity on EgoGesture dataset. Three excitation types are compared to TSM and TSN. All methods use ResNet-50 as backbone and 8 frames input for fair comparison. The least FLOPs/ Δ FLOPs/Param. and the best performance for ACTION-Net and sub-modules are highlighted as bold.

Method	FLOPs	Δ FLOPs	Param.	Top-1
TSN	33G	-	23.68M	83.1
TSM	33G	-	23.68M	92.1
Ours	STE	33.1G	+0.1G (+0.3%)	23.9M
	CE	33.16G	+0.16G (+0.5%)	26.08M
	ME	34.69G	+1.69G (+5.1%)	25.9M
	ACTION-Net	34.75G	+1.75G (+5.3%)	28.08M

Table 4: Ablation study of having ACTION included or not in each residual block stage regarding ResNet-50 backbone on EgoGesture dataset by using 8 frames input. More ACTION engaged yields better performance.

Stage	Top-1	Top-5
res ₂	92.3	98.2
res _{2,3}	92.9	98.5
res _{2,3,4}	93.1	98.5
res _{2,3,4,5}	94.2	98.7

Table 5: ACTION-Net generalizes well across backbones and datasets (TSM is used as a baseline). Accuracy and model complexity on three backbones using Something-Something V2, Jester and EgoGesture with 8 frames as input. The most significant improvements on accuracy and the least extra FLOPs are highlighted as bold.

Backbone	Method	FLOPs	Param.	Something V2	Jester	EgoGesture
ResNet-50	TSM	33G	23.68M	58.7	94.4	92.1
	ACTION-Net	34.75G (+5.3%)	28.08M	62.5 (+3.8%)	97.1 (+2.7%)	94.2 (+2.1%)
BNInception	TSM	16.39G	10.36M	53.5	94.6	92.3
	ACTION-Net	16.94G (+3.4%)	11.59M	56.7 (+3.2%)	96.6 (+2.0%)	93.2 (+0.9%)
MobileNet V2	TSM	2.55G	2.33M	54.9	95.0	92.4
	ACTION-Net	2.57G (+0.8%)	2.36M	58.5 (+3.6%)	96.7 (+1.7%)	93.5 (+1.1%)

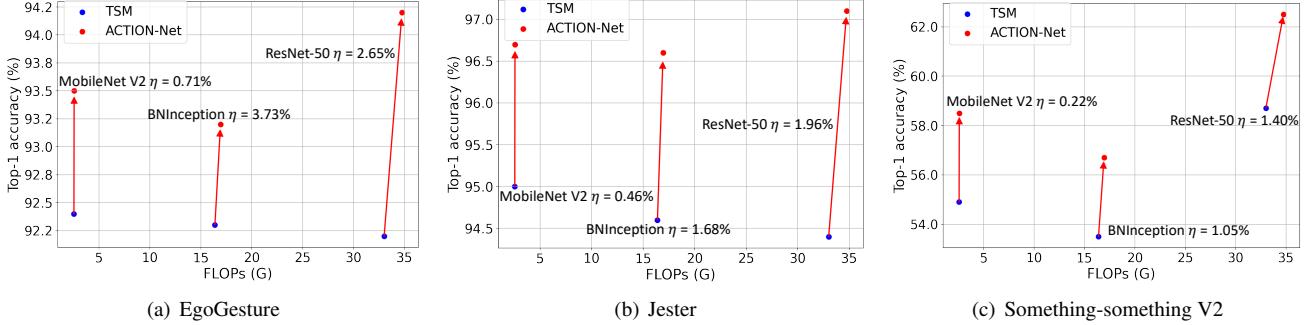


Figure 4: Top-1 accuracy and FLOPs for ACTION-Net and TSM on three backbones i.e., ResNet-50, BNInception and MobileNet V2 using three datasets (from left to right: EgoGesture, Jester and Something-something V2). η is calculated using equation (10) for each backbone on three datasets. EgoGesture: ResNet-50 $\eta = 2.65\%$, BNInception $\eta = 3.73\%$, MobileNet V2 $\eta = \textbf{0.71\%}$. Jester: ResNet-50 $\eta = 1.96\%$, BNInception $\eta = 1.68\%$, MobileNet V2 $\eta = \textbf{0.46\%}$. Something V2: ResNet-50 $\eta = 1.40\%$, BNInception $\eta = 1.05\%$, MobileNet V2 $\eta = \textbf{0.22\%}$.

on the EgoGesture dataset for inspecting these two aspects.

Efficacy of Three Excitations. To validate the contribution of each excitation sub-module, we compare the performance for each individual module and the combination of all sub-modules (ACTION-Net) in Table 3. We also provide visualization results in *Supplementary Materials*. Results show that each excitation module is able to improve the performance for 2D CNN baselines provided by TSN and TSM with limited added computational cost. Concretely, STE and CE both add negligible extra computation compared to TSM by averaging channels globally and averaging spatial information globally yet they both provide useful information to the network. ME adds more computation and parameters to the network than the previous two yet it is acceptable. It captures temporal differences on the spatial domain among adjacent frames over the time and achieves better performance compared to STE and CE. When integrating all these three sub-modules to constitute the ACTION, it can be seen that the ACTION-Net achieves the highest accuracy and increases 2.1% Top-1 accuracy together with increasing 5.3% FLOPs. To better capture the relation between boosted performance and add-on computation, we define the efficiency formulated as

$$\eta = \frac{\Delta \text{FLOPs}}{\Delta \text{Top-1}} \quad (10)$$

where both ΔFLOPs and $\Delta \text{Top-1}$ are in percent, η is the efficiency that represents how many extra FLOPs *in percent* are introduced with respect to increasing 1% Top-1 accuracy (*smaller indicates more efficient apparently*). Efficiency η for STE, CE, ME and ACTION-Net is 0.18%, 0.29%, 2.83% and 2.52% respectively. It can be noticed that STE is the most efficient when taking η into account.

Impact of the Number of ACTION Blocks. The architec-

ture of ResNet-50 can be divided into 6 stages i.e., conv₁, res₂, res₃, res₄, res₅ and FC. The ACTION module can be inserted into any residual stage from res₂ to res₅. We investigate the impact of the number of residual stages that contain the ACTION module as shown in Table 4. Results show that more stages including the ACTION module results in better performance, which indicates the efficacy for our proposed approach.

4.6. Analysis of Efficiency and Flexibility

Compared to recent 2D CNN approaches e.g., STM [14] and TEA [21], our ACTION module enjoys a plug-and-play manner like TSM, which can be embedded to any 2D CNN. To validate the efficacy for our proposed approach, we compare ACTION-Net with TSM on three backbones i.e., ResNet-50, BNInception and MobileNet V2. We report FLOPs and Top-1 accuracy for ACTION-Net and TSM respectively. We also report η calculated using equation 10 when replacing TSM with ACTION, which indicates the penalization of computation when improving the accuracy. Table 5 demonstrates recognition performance and computation for ACTION-Net employing three backbones. TSM is used as a baseline since TSM benefits good performance and zero extra introduced computational cost compared to TSN. It can be noticed that ACTION-Net outperforms TSM consistently regarding the accuracy for all three backbones yet with limited add-on computational cost. ResNet-50 is boosted most significantly regarding the performance and MobileNet V2 holds the least added FLOPs. Figure 4 demonstrates the efficiency η for ACTION-Net based on TSM on different backbones using three datasets. It can be noticed that MobileNet V2 achieves the lowest η (the most efficient) while ResNet-50 achieves the highest η (the least efficient) for all three datasets, which indicates that MobileNet V2 benefits mostly from the ACTION module

regarding the efficiency.

5. Conclusion

We target at designing a module to be inserted to 2D CNN models for video action recognition and introduce a novel ACTION module that utilizes multipath excitation for spatio-temporal features, channel-wise features and motion features. The proposed module could be leveraged by any 2D CNN to build a new architecture ACTION-Net for video action recognition. We demonstrate efficacy and efficiency for ACTION-Net on three large-scale datasets. We show that ACTION-Net achieves consistently improvements compared to 2D CNN counterparts with limited extra computations introduced.

References

- [1] Mahdi Abavisani, Hamid Reza Vaezi Joze, and Vishal M Patel. Improving the performance of unimodal dynamic hand-gesture recognition with multimodal training. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 1165–1174, 2019. [6](#)
- [2] Fabian Caba Heilbron, Victor Escorcia, Bernard Ghanem, and Juan Carlos Niebles. Activitynet: A large-scale video benchmark for human activity understanding. In *Proceedings of the ieee conference on computer vision and pattern recognition*, pages 961–970, 2015. [1](#)
- [3] Joao Carreira and Andrew Zisserman. Quo vadis, action recognition? a new model and the kinetics dataset. In *proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 6299–6308, 2017. [1, 3](#)
- [4] Yunpeng Chen, Yannis Kalantidis, Jianshu Li, Shuicheng Yan, and Jiashi Feng. A[^] 2-nets: Double attention networks. In *Advances in neural information processing systems*, pages 352–361, 2018. [1](#)
- [5] Christoph Feichtenhofer, Haoqi Fan, Jitendra Malik, and Kaiming He. Slowfast networks for video recognition. In *Proceedings of the IEEE international conference on computer vision*, pages 6202–6211, 2019. [3, 6](#)
- [6] Shanghua Gao, Ming-Ming Cheng, Kai Zhao, Xin-Yu Zhang, Ming-Hsuan Yang, and Philip HS Torr. Res2net: A new multi-scale backbone architecture. *IEEE transactions on pattern analysis and machine intelligence*, 2019. [3](#)
- [7] Raghad Goyal, Samira Ebrahimi Kahou, Vincent Michalski, Joanna Materzynska, Susanne Westphal, Heuna Kim, Valentin Haenel, Ingo Fruend, Peter Yianilos, Moritz Mueller-Freitag, et al. The “something something” video database for learning and evaluating visual common sense. In *ICCV*, volume 1, page 5, 2017. [3, 5](#)
- [8] Kensho Hara, Hirokatsu Kataoka, and Yutaka Satoh. Can spatiotemporal 3d cnns retrace the history of 2d cnns and imagenet? In *Proceedings of the IEEE conference on Computer Vision and Pattern Recognition*, pages 6546–6555, 2018. [1, 3](#)
- [9] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 770–778, 2016. [2, 3, 5](#)
- [10] 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. *arXiv preprint arXiv:1704.04861*, 2017. [14](#)
- [11] Jie Hu, Li Shen, and Gang Sun. Squeeze-and-excitation networks. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 7132–7141, 2018. [3, 4](#)
- [12] Eddy Ilg, Nikolaus Mayer, Tonmoy Saikia, Margret Keuper, Alexey Dosovitskiy, and Thomas Brox. Flownet 2.0: Evolution of optical flow estimation with deep networks. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 2462–2470, 2017. [2](#)
- [13] Sergey Ioffe and Christian Szegedy. Batch normalization: Accelerating deep network training by reducing internal covariate shift. *arXiv preprint arXiv:1502.03167*, 2015. [3, 13](#)
- [14] Boyuan Jiang, MengMeng Wang, Weihao Gan, Wei Wu, and Junjie Yan. STM: Spatiotemporal and motion encoding for action recognition. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 2000–2009, 2019. [2, 3, 5, 6, 7, 8](#)
- [15] Andrej Karpathy, George Toderici, Sanketh Shetty, Thomas Leung, Rahul Sukthankar, and Li Fei-Fei. Large-scale video classification with convolutional neural networks. In *Proceedings of the IEEE conference on Computer Vision and Pattern Recognition*, pages 1725–1732, 2014. [2](#)
- [16] Will Kay, Joao Carreira, Karen Simonyan, Brian Zhang, Chloe Hillier, Sudheendra Vijayanarasimhan, Fabio Viola, Tim Green, Trevor Back, Paul Natsev, et al. The kinetics human action video dataset. *arXiv preprint arXiv:1705.06950*, 2017. [1](#)
- [17] Okan Köpüklü, Ahmet Gunduz, Neslihan Kose, and Gerhard Rigoll. Real-time hand gesture detection and classification using convolutional neural networks. In *2019 14th IEEE International Conference on Automatic Face & Gesture Recognition (FG 2019)*, pages 1–8. IEEE, 2019. [6, 7](#)
- [18] Okan Köpüklü, Thomas Ledwon, Yao Rong, Neslihan Kose, and Gerhard Rigoll. Drivermhg: A multi-modal dataset for dynamic recognition of driver micro hand gestures and a real-time recognition framework. *arXiv preprint arXiv:2003.00951*, 2020. [6](#)
- [19] Hildegard Kuehne, Hueihan Jhuang, Estibaliz Garrote, Tomaso Poggio, and Thomas Serre. Hmdb: a large video database for human motion recognition. In *2011 International Conference on Computer Vision*, pages 2556–2563. IEEE, 2011. [1](#)
- [20] Myunggi Lee, Seungeui Lee, Sungjoon Son, Gyutae Park, and Nojun Kwak. Motion feature network: Fixed motion filter for action recognition. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 387–403, 2018. [2, 3, 5, 7](#)
- [21] Yan Li, Bin Ji, Xintian Shi, Jianguo Zhang, Bin Kang, and Limin Wang. TEA: Temporal excitation and aggregation for action recognition. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 909–918, 2020. [2, 3, 5, 7, 8](#)

- [22] Ji Lin, Chuang Gan, and Song Han. TSM: Temporal shift module for efficient video understanding. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 7083–7093, 2019. 2, 3, 5, 6, 7, 12
- [23] Xingyu Liu, Joon-Young Lee, and Hailin Jin. Learning video representations from correspondence proposals. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 4273–4281, 2019. 7
- [24] Zhaoyang Liu, Donghao Luo, Yabiao Wang, Limin Wang, Ying Tai, Chengjie Wang, Jilin Li, Feiyue Huang, and Tong Lu. Teinet: Towards an efficient architecture for video recognition. In *AAAI*, pages 11669–11676, 2020. 2, 3, 5, 7
- [25] Zhaoyang Liu, Limin Wang, Wayne Wu, Chen Qian, and Tong Lu. Tam: Temporal adaptive module for video recognition. *arXiv preprint arXiv:2005.06803*, 2020. 2
- [26] Chenxu Luo and Alan L Yuille. Grouped spatial-temporal aggregation for efficient action recognition. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 5512–5521, 2019. 7
- [27] Joanna Materzynska, Guillaume Berger, Ingo Bax, and Roland Memisevic. The jester dataset: A large-scale video dataset of human gestures. In *Proceedings of the IEEE International Conference on Computer Vision Workshops*, pages 0–0, 2019. 3, 5
- [28] Joanna Materzynska, Tete Xiao, Roei Herzig, Huijuan Xu, Xiaolong Wang, and Trevor Darrell. Something-else: Compositional action recognition with spatial-temporal interaction networks. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 1049–1059, 2020. 6
- [29] Matheus Monfort, Alex Andonian, Bolei Zhou, Kandan Ramakrishnan, Sarah Adel Bargal, Tom Yan, Lisa Brown, Quanfu Fan, Dan Gutfreund, Carl Vondrick, et al. Moments in time dataset: one million videos for event understanding. *IEEE transactions on pattern analysis and machine intelligence*, 42(2):502–508, 2019. 1
- [30] Zhaofan Qiu, Ting Yao, and Tao Mei. Learning spatio-temporal representation with pseudo-3d residual networks. In *proceedings of the IEEE International Conference on Computer Vision*, pages 5533–5541, 2017. 3, 6
- [31] Mark Sandler, Andrew Howard, Menglong Zhu, Andrey Zhmoginov, and Liang-Chieh Chen. Mobilenetv2: Inverted residuals and linear bottlenecks. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 4510–4520, 2018. 3, 14
- [32] Lei Shi, Yifan Zhang, Jian Cheng, and Hanqing Lu. Skeleton-based action recognition with directed graph neural networks. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 7912–7921, 2019. 6
- [33] Lei Shi, Yifan Zhang, Jian Cheng, and Hanqing Lu. Two-stream adaptive graph convolutional networks for skeleton-based action recognition. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 12026–12035, 2019. 1, 6
- [34] Lei Shi, Yifan Zhang, Jian Cheng, and Hanqing Lu. Skeleton-based action recognition with multi-stream adaptive graph convolutional networks. *IEEE Transactions on Image Processing*, 29:9532–9545, 2020. 6
- [35] Karen Simonyan and Andrew Zisserman. Two-stream convolutional networks for action recognition in videos. In *Advances in neural information processing systems*, pages 568–576, 2014. 1, 2
- [36] Khurram Soomro, Amir Roshan Zamir, and Mubarak Shah. Ucf101: A dataset of 101 human actions classes from videos in the wild. *arXiv preprint arXiv:1212.0402*, 2012. 1
- [37] Jonathan Stroud, David Ross, Chen Sun, Jia Deng, and Rahul Sukthankar. D3d: Distilled 3d networks for video action recognition. In *The IEEE Winter Conference on Applications of Computer Vision*, pages 625–634, 2020. 1
- [38] Swathikiran Sudhakaran, Sergio Escalera, and Oswald Lanz. Gate-shift networks for video action recognition. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 1102–1111, 2020. 3
- [39] Du Tran, Lubomir Bourdev, Rob Fergus, Lorenzo Torresani, and Manohar Paluri. Learning spatiotemporal features with 3D convolutional networks. In *Proceedings of the IEEE international conference on computer vision*, pages 4489–4497, 2015. 1, 3
- [40] Du Tran, Heng Wang, Lorenzo Torresani, Jamie Ray, Yann LeCun, and Manohar Paluri. A closer look at spatiotemporal convolutions for action recognition. In *Proceedings of the IEEE conference on Computer Vision and Pattern Recognition*, pages 6450–6459, 2018. 3
- [41] 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 *European conference on computer vision*, pages 20–36. Springer, 2016. 2, 3, 5, 6
- [42] Xiaolong Wang, Ross Girshick, Abhinav Gupta, and Kaiming He. Non-local neural networks. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 7794–7803, 2018. 6
- [43] Zhengwei Wang, Qi She, Tejo Chalasani, and Aljosa Smolic. Catnet: Class incremental 3d convnets for lifelong egocentric gesture recognition. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops*, pages 230–231, 2020. 6
- [44] Ceyuan Yang, Yinghao Xu, Jianping Shi, Bo Dai, and Bolei Zhou. Temporal pyramid network for action recognition. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 591–600, 2020. 5
- [45] Jingran Zhang, Fumin Shen, Xing Xu, and Heng Tao Shen. Temporal reasoning graph for activity recognition. *IEEE Transactions on Image Processing*, 29:5491–5506, 2020. 6
- [46] Yifan Zhang, Congqi Cao, Jian Cheng, and Hanqing Lu. Egogesture: a new dataset and benchmark for egocentric hand gesture recognition. *IEEE Transactions on Multimedia*, 20(5):1038–1050, 2018. 3, 5, 7
- [47] Bolei Zhou, Alex Andonian, Aude Oliva, and Antonio Torralba. Temporal relational reasoning in videos. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 803–818, 2018. 2

- [48] Bolei Zhou, Aditya Khosla, Agata Lapedriza, Aude Oliva, and Antonio Torralba. Learning deep features for discriminative localization. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 2921–2929, 2016. [2](#)
- [49] Mohammadreza Zolfaghari, Kamaljeet Singh, and Thomas Brox. Eco: Efficient convolutional network for online video understanding. In *Proceedings of the European conference on computer vision (ECCV)*, pages 695–712, 2018. [5](#)

A. Backbone Architecture in Experiments

In the main content of our paper, we evaluate efficiency and performance for our ACTION-Net on three different backbones i.e., ResNet-50, BNInception and MobileNet V2. We provide details that insert ACTION/TSM into each backbone in this section. To keep consistence, the number and the inserted position of TSM and ACTION are same.

ResNet-50. We insert TSM/ACTION into each residual block i.e., from res_2 to res_5 at the start. It is summarized in Table 2. There are 3, 4, 6, 3 TSM/ACTION modules that are inserted into res_2 , res_3 , res_4 and res_5 respectively. Therefore, ResNet-50 is equipped with 16 TSM/ACTION modules totally.

BNInception. Figure 1 illustrates the details of BNInception used in our study. Similar to ResNet-50, we insert TSM/ACTION into each inception block at the starting point. In summary, there are 10 TSM/ACTION modules added into BNInception.

MobileNet V2. Figure 2 and Table 3 summarize details of MobileNet V2 architecture and positions that insert TSM/ACTION. We insert TSM/ACTION into each bottleneck at the start. In order to keep consistent with adding TSM to MobileNet V2 in the original work [22], we insert TSM/ACTION into two blocks in stage₄ and the first block in stage₅, which results in 10 TSM/ACTION modules have been added into MobileNet V2.

B. The Location of ACTION

As mentioned in the previous section, we insert ACTION at the beginning of each block for three backbones studied in this work. Here we provide ablation studies for different locations that insert ACTION to three backbones. It is worth noting that four possible location for both ResNet-50 and MobileNet V2 but two possible locations for BN-Inception i.e., start and before the concatenate operation as seen in Fig. 1. It can be noticed that inserting ACTION at the beginning (Loc 1) is more effective compared to other locations for all three backbones.

Table 1: Top-1 accuracy of different embedded locations on EgoGesture dataset using 8 segments. Loc 1 is the default used in the main paper.

Model	Loc 1 (default)	Loc 2	Loc 3	Loc 4
ResNet-50	94.2	94.0	93.8	94.0
BNInception	93.2	92.6	NA	NA
MobileNet V2	93.5	93.1	93.1	93.3

C. Visualization Results

Visualization for three actions ‘*Rotate fists counterclockwise*’, ‘*Applaud*’ and ‘*Draw circle with hand in horizontal surface*’ using two baselines (i.e., TSN and TSM), three excitation sub-modules and our proposed ACTION-Net is shown in Fig. 3, Fig. 4 and Fig. 5. The first row is the presented video sequence and rest of rows are CAM obtained by each approach. It can be noticed that both TSN and TSM can only recognize objects but are unable to produce CAM for the movement smoothly. Compared to TSN and TSM, it can be noticed that our proposed ACTION and each three sub-module are all able to extract meaningful temporal information from a presented video sequence by addressing smooth CAM for the action movement. Although STE and CE more focus on spatial modeling since the temporal modeling ability is limited in these two modules i.e., one 3D convolutional layer with size $3 \times 3 \times 3$ and one 1D convolutional layer with size 3, they are able to produce smooth CAM to some extent, which are much more convincing than TSM and TSN. From the visualization for ME, it can be noticed that ME is able to produce the most smooth CAM for the action movement between adjacent frames. However, spatial information for objects in video is somewhat limited e.g., it is hard to figure out one hand or two hands in the presented video for Fig. 3 and Fig. 4. Our proposed ACTION-Net, which integrates three excitation above, is able to not only recognize objects (first two frames) but also address action movements (middle frames), which takes advantages from each excitation sub-module.

Table 2: ResNet-50 backbone with TSN, TSM and ACTION used in this work.

Stage	TSN	TSM	ACTION-Net	Output size
Input		—		$T \times 224 \times 224$
conv ₁		$1 \times 7 \times 7, 64, \text{stride } 1, 2, 2$		$T \times 112 \times 112$
pool ₁		$1 \times 3 \times 3, \text{max, stride } 1, 2, 2$		$T \times 56 \times 56$
res ₂	$\begin{bmatrix} 1 \times 1 \times 1, 64 \\ 1 \times 3 \times 3, 64 \\ 1 \times 1 \times 1, 256 \end{bmatrix} \times 3$	$\begin{bmatrix} 1 \times 1 \times 1, 64 \\ 1 \times 3 \times 3, 64 \\ 1 \times 1 \times 1, 256 \end{bmatrix} \times 3$	$\begin{bmatrix} \text{ACTION} \\ 1 \times 1 \times 1, 64 \\ 1 \times 3 \times 3, 64 \\ 1 \times 1 \times 1, 256 \end{bmatrix} \times 3$	$T \times 56 \times 56$
res ₃	$\begin{bmatrix} 1 \times 1 \times 1, 128 \\ 1 \times 3 \times 3, 128 \\ 1 \times 1 \times 1, 512 \end{bmatrix} \times 4$	$\begin{bmatrix} \text{TSM} \\ 1 \times 1 \times 1, 128 \\ 1 \times 3 \times 3, 128 \\ 1 \times 1 \times 1, 512 \end{bmatrix} \times 4$	$\begin{bmatrix} \text{ACTION} \\ 1 \times 1 \times 1, 128 \\ 1 \times 3 \times 3, 128 \\ 1 \times 1 \times 1, 512 \end{bmatrix} \times 4$	$T \times 28 \times 28$
res ₄	$\begin{bmatrix} 1 \times 1 \times 1, 256 \\ 1 \times 3 \times 3, 256 \\ 1 \times 1 \times 1, 512 \end{bmatrix} \times 6$	$\begin{bmatrix} \text{TSM} \\ 1 \times 1 \times 1, 256 \\ 1 \times 3 \times 3, 256 \\ 1 \times 1 \times 1, 1024 \end{bmatrix} \times 6$	$\begin{bmatrix} \text{ACTION} \\ 1 \times 1 \times 1, 256 \\ 1 \times 3 \times 3, 256 \\ 1 \times 1 \times 1, 1024 \end{bmatrix} \times 6$	$T \times 14 \times 14$
res ₅	$\begin{bmatrix} 1 \times 1 \times 1, 512 \\ 1 \times 3 \times 3, 512 \\ 1 \times 1 \times 1, 2048 \end{bmatrix} \times 3$	$\begin{bmatrix} \text{TSM} \\ 1 \times 1 \times 1, 512 \\ 1 \times 3 \times 3, 512 \\ 1 \times 1 \times 1, 2048 \end{bmatrix} \times 3$	$\begin{bmatrix} \text{ACTION} \\ 1 \times 1 \times 1, 512 \\ 1 \times 3 \times 3, 512 \\ 1 \times 1 \times 1, 2048 \end{bmatrix} \times 3$	$T \times 7 \times 7$
global average pool, FC				$T \times \text{CLS}$
temporal average				CLS

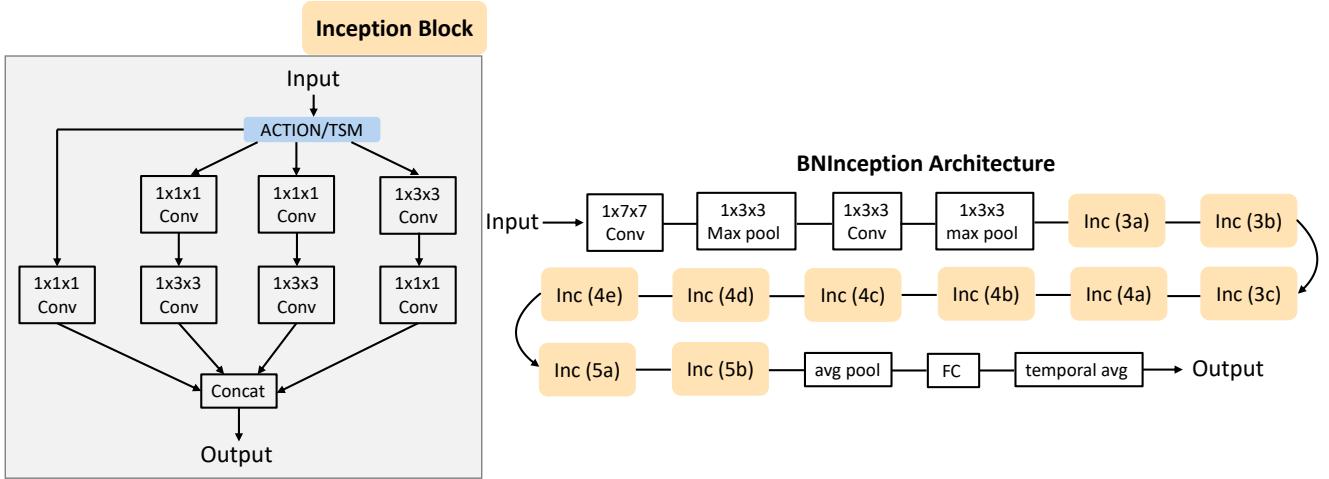
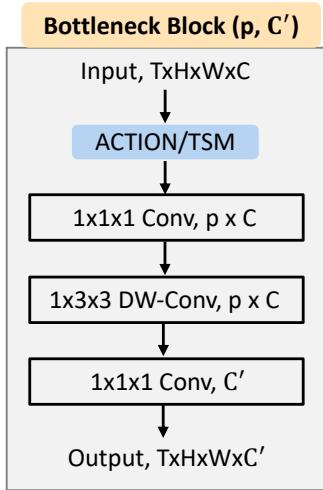


Figure 1: BNInception with ACTION and TSM used in this study. We insert ACTION/TSM into the start in each Inception block [13].



Stage	MobileNet V2	Output size
Input	—	$T \times 224 \times 224$
conv ₁	$1 \times 7 \times 7, 32, \text{stride}1, 2, 2$	$T \times 112 \times 112$
Stage ₂	Bottleneck(1, 16)	$T \times 56 \times 56$
	Bottleneck(1, 16) $\times 2$	
Stage ₃	Bottleneck(6, 25) $\times 3$	$T \times 28 \times 28$
	Bottleneck(6, 64) $\times 4$	
Stage ₄	Bottleneck(6, 96) $\times 3$	$T \times 14 \times 14$
	Bottleneck(6, 160) $\times 3$	
Stage ₅	Bottleneck(6, 320)	$T \times 7 \times 7$
	$1 \times 1 \times 1, 1280, \text{stride}1, 1, 1$	
	global average pool, FC, temporal average	CLS

Figure 2 & Table 3: Figure on the left: Bottleneck block (p , C') with ACTION/TSM in MobileNet V2. We insert ACTION/TSM into the bottleneck block at the start. DW-Conv refers to depth-wise convolution [10]. Table on the right: MobileNet-V2 backbone. Bottleneck blocks with ACTION/TSM illustrated in figure on the left are applied to the backbone. [31].

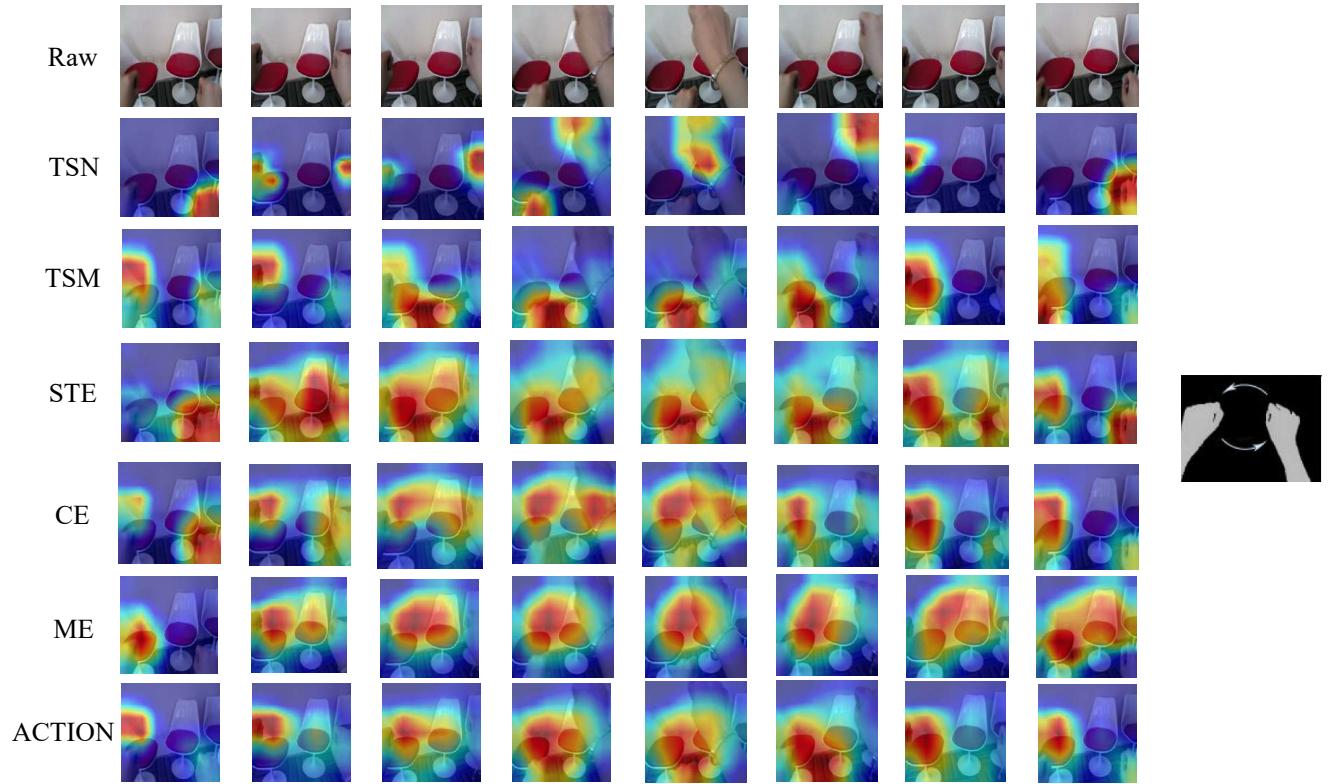


Figure 3: ‘Rotate fists counterclockwise’

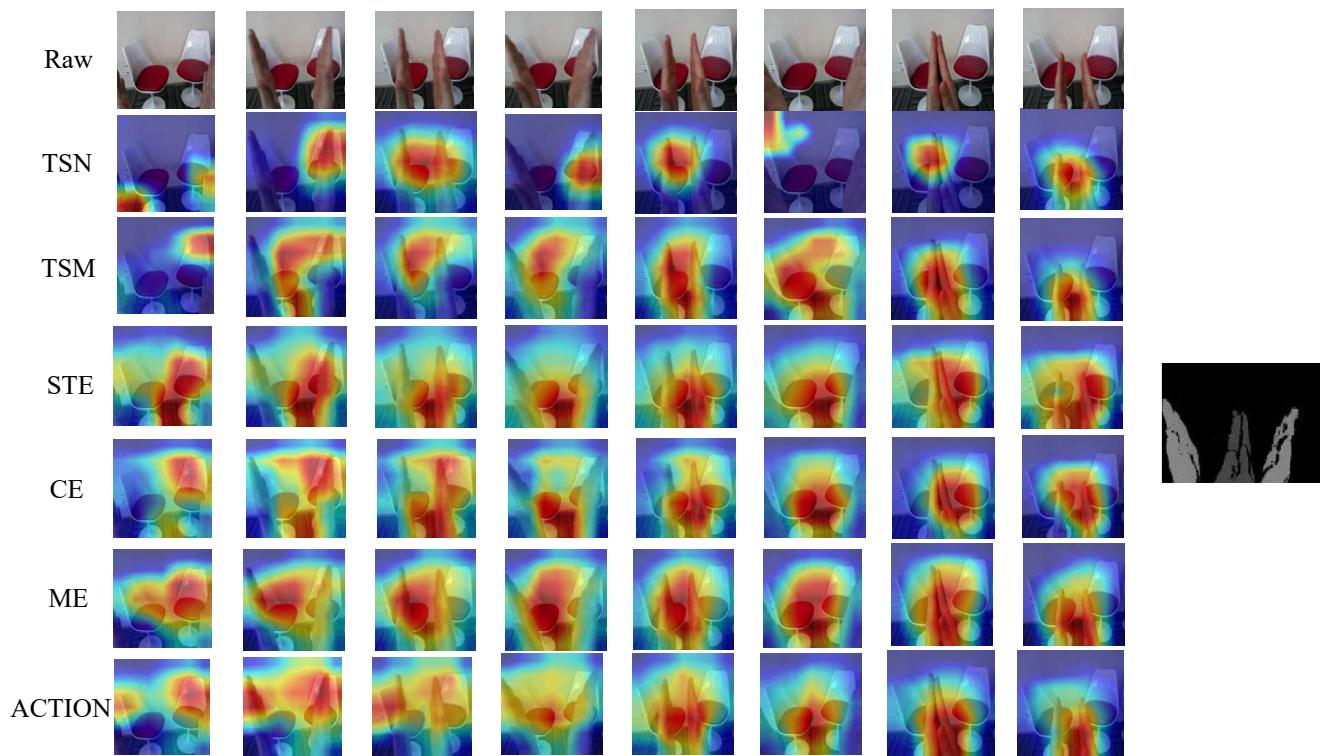


Figure 4: ‘Applaud’

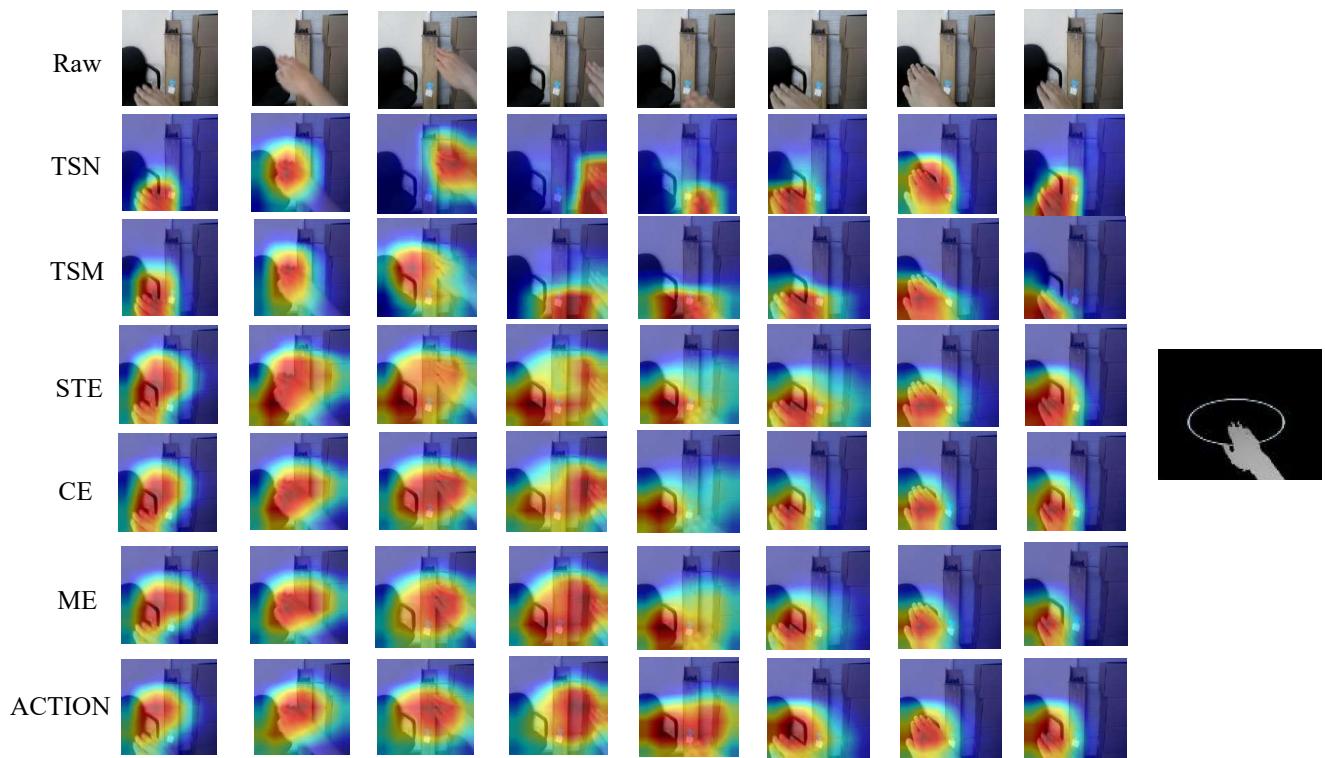


Figure 5: ‘Draw circle with hand in horizontal surface’