Supplementary Materials for ACTION-Net: Multipath Excitation for Action Recognition

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 [3], we insert TSM/ACTION into two blocks in stage_4 and the first block in stage_5, 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 nothing that four possible location for both ResNet-50 and MobileNet V2 but two possible locations for BNIncetion 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 regarding the results in Table 1.

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, stride 1, 2, 2			$T \times 112 \times 112$
pool_1	$1 \times 3 \times 3$, max, stride 1, 2, 2			$T \times 56 \times 56$
res_2	$\begin{bmatrix} 1\times1\times1, 64 \\ 1\times3\times3, 64 \\ 1\times1\times1, 256 \end{bmatrix} \times 3$	TSM $\begin{bmatrix} 1\times1\times1, 64 \\ 1\times3\times3, 64 \\ 1\times1\times1, 256 \end{bmatrix} \times 3$	ACTION $\begin{bmatrix} 1\times1\times1, 64 \\ 1\times3\times3, 64 \\ 1\times1\times1, 256 \end{bmatrix} \times 3$	$T \times 56 \times 56$
res_3	$\begin{bmatrix} 1\times1\times1, 128 \\ 1\times3\times3, 128 \\ 1\times1\times1, 512 \end{bmatrix} \times 4$	TSM $\begin{bmatrix} 1\times1\times1, 128 \\ 1\times3\times3, 128 \\ 1\times1\times1, 512 \end{bmatrix} \times 4$	ACTION $\begin{bmatrix} 1\times1\times1, 128 \\ 1\times3\times3, 128 \\ 1\times1\times1, 512 \end{bmatrix} \times 4$	$T \times 28 \times 28$
res_4	$\begin{bmatrix} 1\times1\times1, 256 \\ 1\times3\times3, 256 \\ 1\times1\times1, 512 \end{bmatrix} \times 6$	TSM $\begin{bmatrix} 1\times1\times1, 256 \\ 1\times3\times3, 256 \\ 1\times1\times1, 1024 \end{bmatrix} \times 6$	ACTION $\begin{bmatrix} 1\times1\times1, 256 \\ 1\times3\times3, 256 \\ 1\times1\times1, 1024 \end{bmatrix} \times 6$	$T \times 14 \times 14$
res_5	$\begin{bmatrix} 1\times1\times1, 512 \\ 1\times3\times3, 512 \\ 1\times1\times1, 2048 \end{bmatrix} \times 3$	TSM $\begin{bmatrix} 1\times1\times1, 512 \\ 1\times3\times3, 512 \\ 1\times1\times1, 2048 \end{bmatrix} \times 3$	ACTION $\begin{bmatrix} 1\times1\times1, 512 \\ 1\times3\times3, 512 \\ 1\times1\times1, 2048 \end{bmatrix} \times 3$	$T \times 7 \times 7$
global average pool, FC				$T \times CLS$
temporal average				CLS

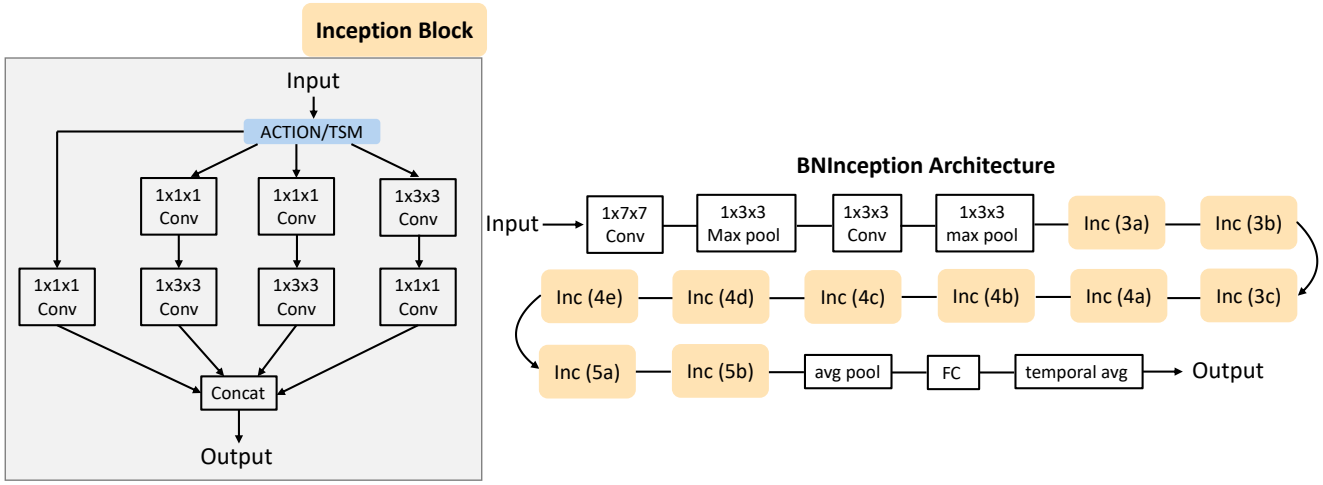
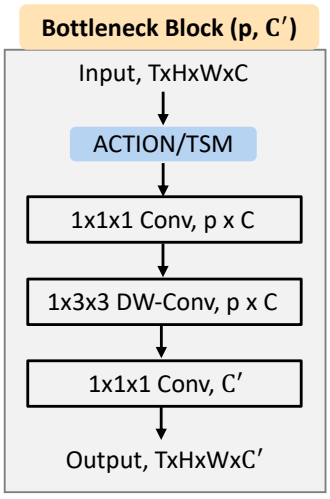


Figure 1: BNInception with ACTION and TSM used in this study. We insert ACTION/TSM into the start in each Inception block [2].



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	$T \times 28 \times 28$
Stage_4	Bottleneck(6, 64) $\times$ 4	$T \times 14 \times 14$
	Bottleneck(6, 96) $\times$ 3	
Stage_5	Bottleneck(6, 160) $\times$ 3	$T \times 7 \times 7$
	Bottleneck(6, 320)	
	$1 \times 1 \times 1, 1280, \text{stride} 1, 1, 1$	
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 [1]. *Table on the right*: MobileNet-V2 backbone. Bottleneck blocks with ACTION/TSM illustrated in *figure on the left* are applied to the backbone. [4].

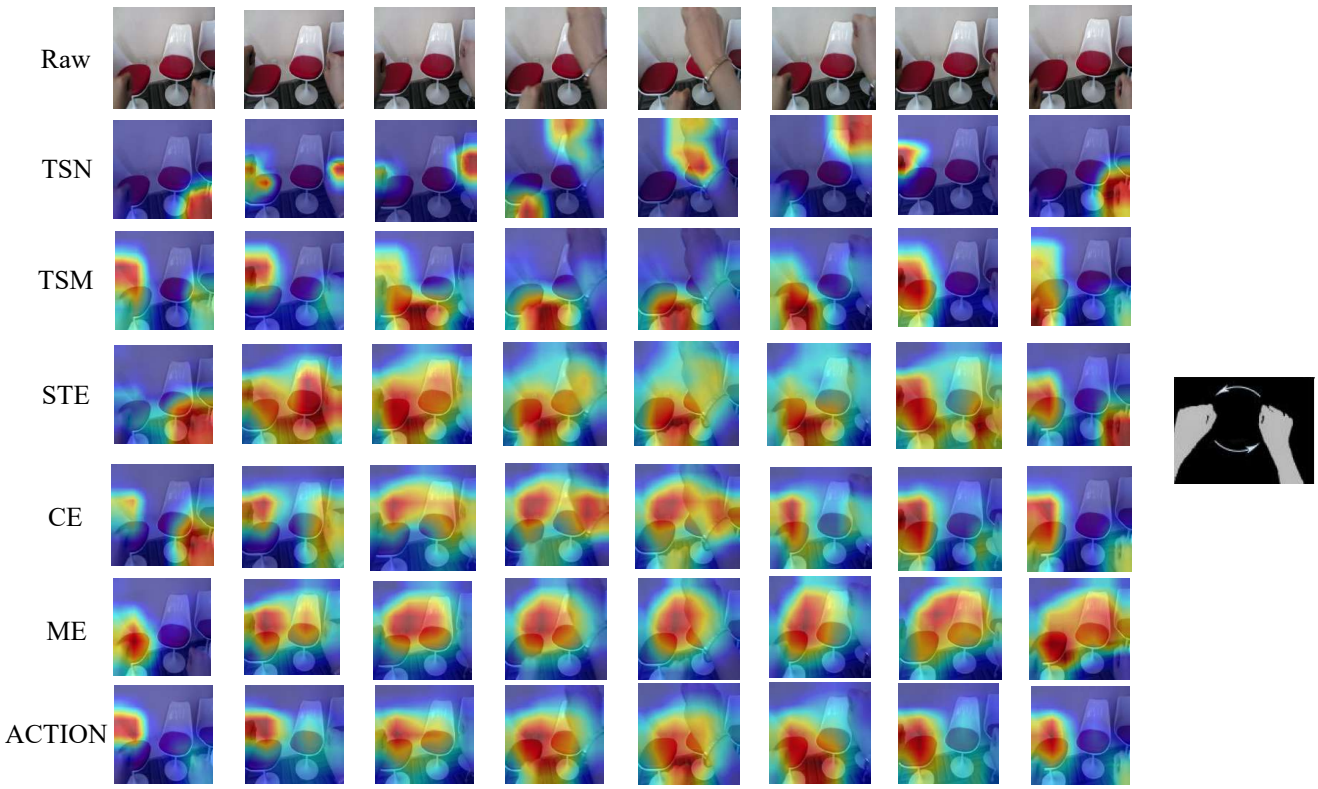


Figure 3: *'Rotate fists counterclockwise'*

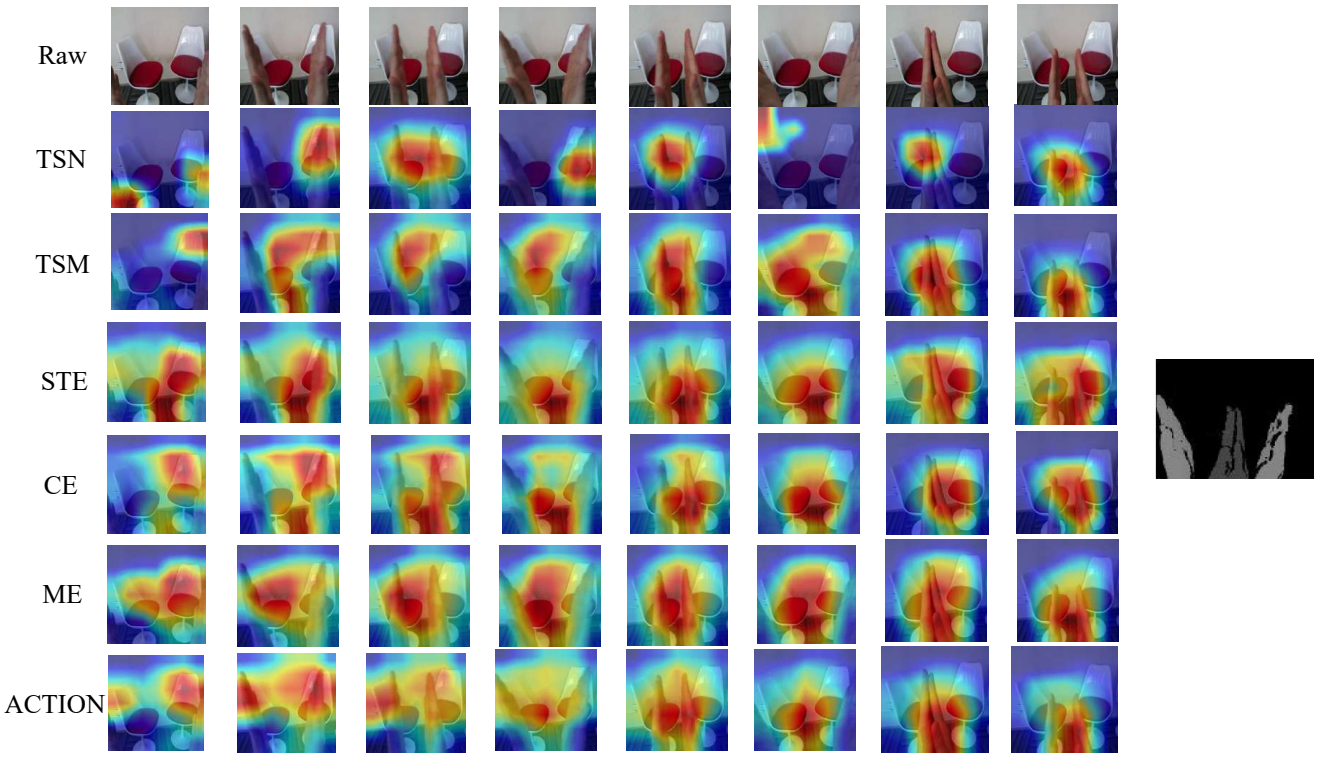


Figure 4: *'Applaud'*

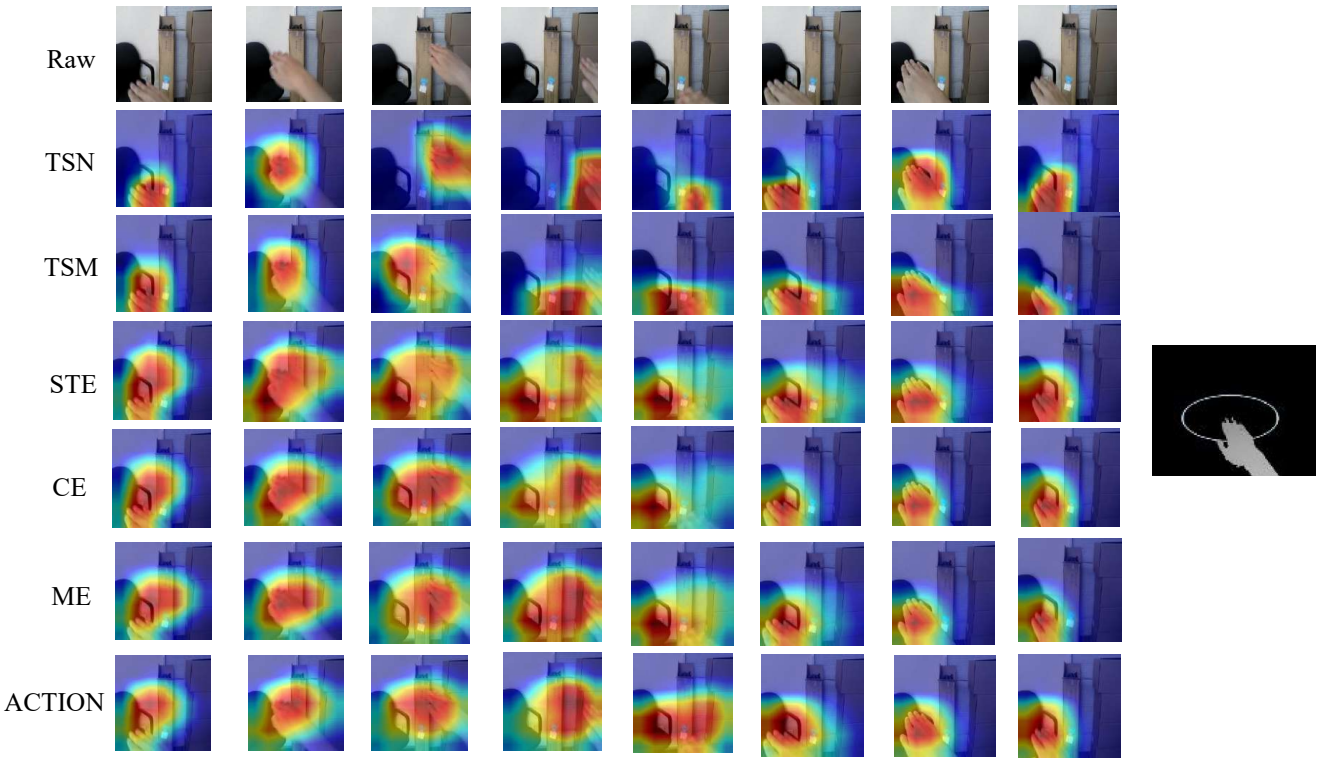


Figure 5: *'Draw circle with hand in horizontal surface'*

References

[1] 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.

[2] Sergey Ioffe and Christian Szegedy. Batch normalization: Accelerating deep network training by reducing internal covariate shift. *arXiv preprint arXiv:1502.03167*, 2015.

[3] 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.

[4] 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.