entropy

MDPI

Article

Video Action Recognition Using Motion and Multi-View Excitation with Temporal Aggregation

Yuri Yudhaswana Joefrie [1,2,*] and Masaki Aono [1]

[1] Department of Computer Science and Engineering, Toyohashi University of Technology, 1-1 Tenpaku-cho, Toyohashi 441-8580, Japan
[2] Department of Information Technology, Universitas Tadulako (UNTAD), Palu 94118, Indonesia
* Correspondence: yuri.yudhaswana.joefrie.gp@tut.jp

Abstract: Spatiotemporal and motion feature representations are the key to video action recognition. Typical previous approaches are to utilize 3D CNNs to cope with both spatial and temporal features, but they suffer from huge computations. Other approaches are to utilize (1+2)D CNNs to learn spatial and temporal features in an efficient way, but they neglect the importance of motion representations. To overcome problems with previous approaches, we propose a novel block which makes it possible to alleviate the aforementioned problems, since our block can capture spatial and temporal features more faithfully and efficiently learn motion features. This proposed block includes Motion Excitation (ME), Multi-view Excitation (MvE), and Densely Connected Temporal Aggregation (DCTA). The purpose of ME is to encode feature-level frame differences; MvE is designed to enrich spatiotemporal features with multiple view representations adaptively; and DCTA is to model long-range temporal dependencies. We inject the proposed building block, which we refer to as the META block (or simply "META"), into 2D ResNet-50. Through extensive experiments, we demonstrate that our proposed method architecture outperforms previous CNN-based methods in terms of "Val Top-1 %" measure with Something-Something v1 and Jester datasets, while the META yielded competitive results with the Moment-in-Time Mini dataset.

Keywords: action recognition; multi-view; excitation; multi-layer neural network; temporal convolution; videos

Citation: Joefrie, Y.Y.; Aono, M. Video Action Recognition Using Motion and Multi-View Excitation with Temporal Aggregation. *Entropy* **2022**, *24*, 1663. https://doi.org/10.3390/e24111663

Academic Editors: Andrea Prati, Luis Javier García Villalba and Vincent A. Cicirello

Received: 25 October 2022
Accepted: 11 November 2022
Published: 15 November 2022

1. Introduction

Video action recognition is still challenging for researchers and practitioners as it involves both spatiotemporal and motion understanding. In the meantime, as an impact of the growth of technology, more people are involved in social media. They often record, upload and share videos to media platforms. As a result, an abundant number of videos are available to the public. This leads to more researchers engaging in the topic of video understanding. Action recognition, the first step of video understanding, becomes critical in practical applications such as suspicious behavior detection in camera surveillance and video recommendation systems.

An action in videos can be recognized based on scenes with less temporal information, while other actions need more temporal aspects to recognize. Such examples of actions with fewer temporal cues are *'rafting'* and *'haircut'*. We can judge the aforementioned actions only by seeing the scene. Contrarily, more temporal information is needed to judge an action involved in a video, e.g., *'zooming in with two fingers'* and *'picking something up'*. With this condition in mind, one must consider having both spatiotemporal and motion information flow in the network.

Current existing convolutional neural networks (CNNs) for action recognition can be categorized by the type of convolution kernel, i.e., three-dimensional (3D) and two-dimensional (2D) CNN. Several researchers utilized 3D CNNs to learn both spatial and temporal information simultaneously [1–3]. While this approach works very well for video

action recognition tasks, the usage of a CNN-based 3D kernel certainly introduces more parameters compared with the 2D kernel type; hence, the computation cost increases. This will limit the implementation in real-time application. Additionally, 3D CNNs are typically laborious to optimize and prone to overfitting [3].

To overcome the real-time and optimization problem, recently, more researchers utilized 2D CNNs and equipped the network with temporal convolution for characterizing the temporal information [4–6]. TSN [7] employs a sparse temporal sampling strategy from the whole video to predict action in it, and this approach influenced several 2D CNN methods afterward. The approach by Lin et al. [6] utilized a shift-part strategy together with a sampling strategy to recognize an action very effectively. The shifting strategy is to shift part of the channels along the temporal axis in order to endow the network with movement learning. However, both approaches lack motion representations and spatiotemporal cues from different views. Meanwhile, since optical flow represents a motion as an input-level frame, many researchers take this modality into consideration. Two main drawbacks of optical flow are that it needs huge space storage and tremendous time for computation, which is not suitable for real-time application.

In different approaches from previous methods, several works factorized 3D cubes of a kernel into a (1+2)D kernel configuration [8–10] to lessen the heavy computation. Another interesting factorization strategy was introduced by the Deep HRI team [11] in the action recognition competition. They proposed their novel architecture, coined as multi-view CNN (MV-CNN), to act as a 3D convolution and showed a profound increase in the accuracy. Figure 1 illustrates their proposed idea and shows the comparison with two other convolution designs. Assuming k is the kernel with the size of 3: (a) the kernel convolves on temporal (T) and spatial ($H \times W$) axes simultaneously; (b) temporal and spatial feature learnings are constructed serially; and (c) multi-view feature learnings occur independently, and the resulting feature maps are aggregated with weighted average α_i to produce multi-viewed feature maps. Nevertheless, these factorization strategies still neglect the importance of motion features, which are beneficial for action recognition tasks.

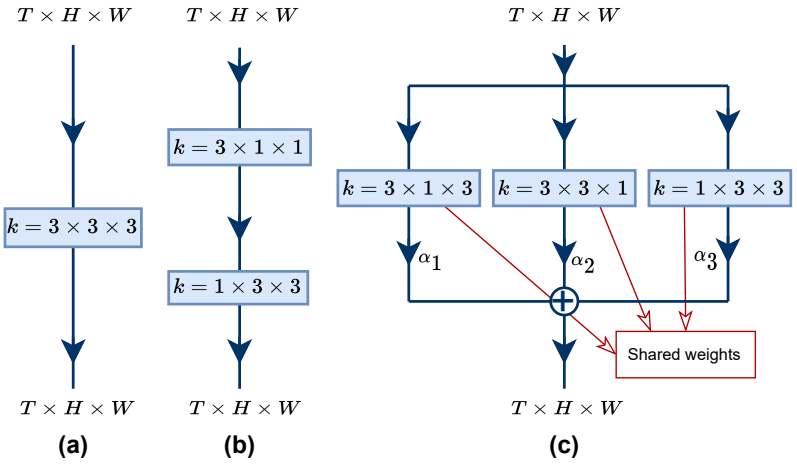

Figure 1. Comparison of three designs of spatiotemporal feature learning: (**a**) vanilla 3D convolution; (**b**) (1+2)D convolution; and (**c**) a multi-view design from [11].

Based on the aforementioned observation, we propose a novel building block, **M**otion and Multi-View **E**xcitation and **T**emporal **A**ggregation (META). Specifically, META comprises three submodules: (1) Motion Excitation (ME), (2) Multi-view Excitation (MvE), and (3) Densely Connected Temporal Aggregation (DCTA). These three submodules are integrated into the 2D ResNet-50 base model, and it provides the network the ability to learn spatiotemporal and motion features altogether. The ME submodule addresses issues with optical flow problems by calculating feature-level content displacement on the fly during training or inferencing; thus, no space storage is needed. It also introduces an

insignificant amount of FLOPs and time to calculate compared with optical flow. The MvE submodule, on the other hand, produces enhanced spatiotemporal representation of output features. This submodule adds a multi-view perspective to the original feature maps, and we design the MvE submodule to be complementary to ME, in that output from the MvE is directly added to the output from ME. For temporal feature learning, one-dimensional (1D) convolution is suitable for such a task. In our work, we insert densely connected 1D convolution layers inside a group of subconvolutions and arrange them in a hierarchical layout to model the temporal representation and long-range temporal dependencies. In a nutshell, the main contribution of our work is as follows:

1. We design three submodules, including ME, MvE and DCTA, to learn enriched spatiotemporal and motion representations in a very efficient way and in an end-to-end manner.
2. We propose META and insert it in 2D ResNet-50 with a few additional model parameters and low computation cost.
3. We conduct extensive experiments on three popular benchmarking datasets, including Something-Something v1, Jester and Moments-in-Time Mini, and the results show the superiority of our approach.

2. Related Work

In this section, we discuss in more detail several related works for action recognition, including MV-CNN and TEA, which highly motivated this work. We also add a Transformer-based approach in this section because it has contributed to recent advances in computer vision, particularly for the task of action recognition.

With large-scale video datasets for action recognition available publicly, more 3D CNNs are introduced for action recognition tasks. Researchers successfully implemented them and outperformed state-of-the-art methods [3,9,12–15]. Three-dimensional CNNs are thought to be capable of learning both spatial and temporal aspects of a video, which are a very important aspect of action recognition. Among these researchers, Carreira et al. [13] used an ImageNet [16,17] pre-trained cube-shaped –$N \times N \times N$ kernel to learn volumetric features. Another work by Hara et al. [15] used a 3D version of vanilla ResNet and proved the superiority of this architecture. Several attempts by [18–20] investigate the combination of a CNN and LSTM [21] or a densely connected LSTM. They claim that the LSTM layer can model the temporal relation of a series of features coming from either 2D CNNs or 3D CNNs. An alternative to LSTM for modeling temporal relationships is to use post hoc fusion [5] inside 2D CNNs. Later, a Temporal Shift Module (TSM) by Lin et al. [6] implemented a shifting operator on channel axes to learn temporal features without extra parameters and insignificant extra FLOPs. Afterwards, works proposed by [8–10] decomposed 3D convolution into 1D and 2D convolution to distill temporal and spatial features, respectively. This strategy was used to address the heavy computation and quadratic growth of model size when using 3D convolution. A mix of 2D and 3D CNNs in a unified architecture is also used to capture spatial and temporal features at the same time [22]. ECO [23] utilized this mixing strategy with top-heavy architecture. However, mixing the 2D CNN with the 3D CNN in one architecture will inevitably increase model parameters and require more time to optimize the parameters.

Meanwhile, partially inspired by the visual system of biological studies on the retinal ganglion cells in primates, Feichtenhofer et al. [24] advocated a two-stream architecture approach. Considering a video may contain more static as well as more dynamic objects, both streams have a different temporal rate, which makes their architecture unique compared with the existing two-stream architectures. Furthermore, learning from the three anatomical planes of the human body, i.e., sagittal, coronal and transverse, the DEEP HRI team [11] tried to simulate 3D convolution in their work. A video clip can be represented as a $T \times H \times W$ volumetric data, with T, H and W denoting the number of frames, height and width, respectively. They reshape it to 2D data (i.e., $T \times H$, $T \times W$ and $H \times W$) separately

and operate a shared 2D kernel to convolve over the reshaped 2D data. Figure 2 depicts images when they are seen in different views.

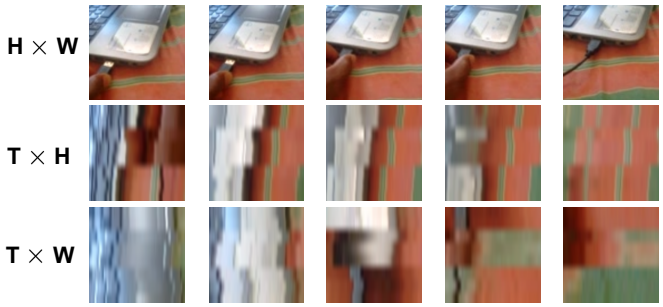

Figure 2. Image visualization with different views for action "Plugging something into something" class from the Something-Something v1 dataset.

Apart from previous work, SENet [25] also attracts researchers to employ its squeeze-and-excitation method. This method uses a global context pooling mechanism to enhance the spatially informative channels and was verified to be effective in image understanding tasks. A recent work by Hao et al. [26] studied the insertion of channel context into the spatio-temporal attention learning block for element-wise feature refinement.

In addition to spatiotemporal representation, motion information is also the key difference between action classification and image classification tasks; thus, exploiting motion features is mandatory. A comprehensive study by Sevilla-Lara et al. [27] analyzed the importance of optical flow for action recognition. They analyzed optical flow itself, conducted several experiments using optical flow modality, and proved the contribution to the accuracy. Another several attempts utilized optical flow as an additional input to RGB [7,28]. A two-stream architecture is deployed to learn both optical flow and RGB data, and an average result is generated to predict an action. Feichtenhofer et al. [29] experimented with several fusion schemes so as to enhance spatiotemporal features. While this approach demonstrates excellent results compared with RGB data alone, this approach cannot be implemented in a real-world application, as extracting the pixel-wise optical frame with the TV-L1 method [30] requires heavy computation as well as much storage space. In this light, the RGB difference was introduced [7,31], which is more lightweight. A work by Jiang et al. [32] with their STM module firstly outlined the motion calculation for end-to-end learning in a 2D ConvNet and proved to be effective in capturing instantaneous motion representation as short-range temporal evolution. Furthermore, Li et al. [33] introduced a new block termed as TEA to explore the benefits of the attention mechanism added to the motion calculation previously mentioned. Later, this attentive motion features module was adopted by [34–36]. In addition, the authors of TEA suggested overcoming the limitation of long-range temporal representation by introducing multiple temporal aggregations in a hierarchical design. In this work, we propose a motion calculation and a hierarchical structure of local temporal convolutions, similar to the previous work. We explain more details of our work and highlight the difference from the previous work in the subsequent section.

As Vision Transformers brought recent breakthroughs in computer vision, specifically for action recognition tasks, many researchers have adopted them as their model [37–40] or combined them with 2D CNN [41]. For example, Arnab et al. [37] proposed several factorization variants to model spatial and temporal representation effectively inside a transformer encoder. TimeSformer [38] investigated several self-attention combinations on frame-level patches and suggested that separated attention for spatial and temporal representation applied within each block yielded the best video classification accuracy. Another work by Tian et al. [41] introduced a 2D ResNet with Transformer injected at the top layer before the linear layer to accurately aggregate extracted local cues from preceding blocks into a video representation. Although these current approaches seem promising, a

Transformer-based network is not suited for real-world applications because it is highly computationally intensive [36].

3. Our Proposed Method

This section discusses the technical details of our work. Firstly, we present a method to extract motion representations to simulate the optical flow modality. Afterward, our novel Multi-view Excitation is introduced. Lastly, a simple stacking local temporal convolution with dense connection is also discussed here as a part of our improvement strategy. We also include a short discussion regarding comparing our work and TEA. Some notations written in this section are: N, T, C, H and W, indicating batch size, number of frames, number of channels, height and width, respectively.

3.1. Motion Excitation (ME)

Introduced firstly by STM [32], and later enhanced by TEA [33], the motion excitation submodule performs frame difference calculation in a unified framework for end-to-end learning. In principle, motion representation indicates content displacement of two adjacent feature maps, therefore called *feature-level*-based motion, rather than *pixel-level*-based motion, as in the concept of optical flow. Figure 3 illustrates the steps to measure approximate feature-level temporal differences.

The first step is to reduce the number of channels for efficiency with the ratio $r = 16$ by applying a 1×1 convolution layer K_{red} to the initial input X, formulated in Equation (1). Then, we slice feature maps at the temporal axis, followed by element-wise subtraction for every adjacent output feature and obtain M at time step t. Before subtraction, a 3×3 transformation convolution layer K_{transf} is applied to the output features X' at the time step $(t+1)$. Next, we concatenate motion representations M at all time steps according to the temporal axis with 0 padded to the last segment. Concretely, given $X \in \mathbb{R}^{NT \times C \times H \times W}$ are input features to the ME submodule, the above processes are expressed as follows:

$$X' = K_{red} * X, \quad X' \in \mathbb{R}^{NT \times \frac{C}{r} \times H \times W} \tag{1}$$

$$M_t = K_{transf} * X'_{t+1} - X'_t, \ \ 1 \geq t \geq T-1 \tag{2}$$

$$X' = concat(M_t, 0), \ \ 1 \geq t \geq T-1, \tag{3}$$

where $M_t \in \mathbb{R}^{N \times \frac{C}{r} \times H \times W}$ and the last $X' \in \mathbb{R}^{NT \times \frac{C}{r} \times H \times W}$.

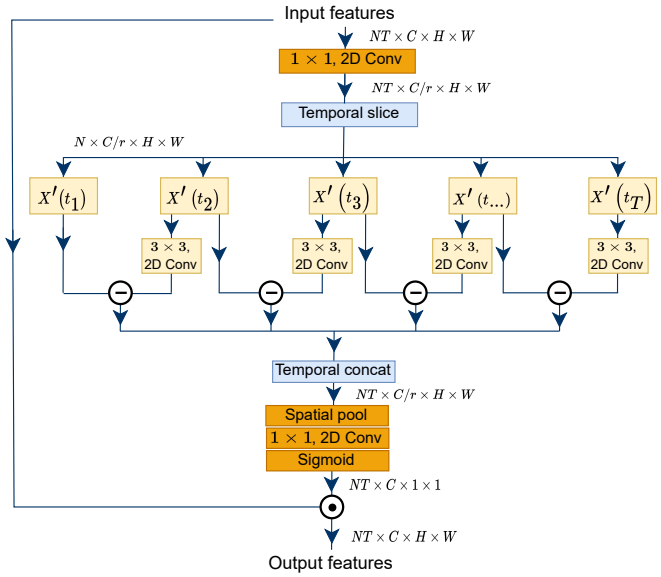

Figure 3. Two adjacent frames are subtracted to obtain motion representation. We firstly apply channel-wise 3×3 convolution on frames $[t+1]$ before subtraction.

At this point, we have a new X' as approximate feature-level motion representations. Since we want to emphasize the informative features and suppress less useful ones alongside with [25], we squeeze the global information from each channel of the motion representations by utilizing the global spatial pooling layer. Then, another 1×1 2D convolution layer K_{exp} performs channel expansion to restore the number of channels, and we obtain a new X, as in Equation (4). Lastly, attentive feature maps are obtained by feeding the new X to a sigmoid function δ, while final outputs X_{ME} are produced from a multiplication between the initial inputs X and attentive feature maps, as defined by Equation (5).

$$X = K_{exp} * pool(X'), X \in \mathbb{R}^{NT \times C \times 1 \times 1} \tag{4}$$

$$X_{ME} = \delta(F) * X, \quad X_{ME} \in \mathbb{R}^{NT \times C \times H \times W} \tag{5}$$

When subtracting the feature maps, we only calculate them one time: A collection of feature maps containing $[2 \sim T]$ timestamps minus $[1 \sim T-1]$.

3.2. Densely Connected Temporal Aggregation (DCTA)

Previously, learning temporal relationships in the task of action recognition was achieved by repeatedly stacking local temporal layers in deep networks. Unfortunately, it raises some problems. It is considered to be harmful to the features because the optimization message transmitted from distant frames has been weakened. To alleviate such a problem, we propose the Densely Connected Temporal Aggregation submodule. We follow the Res2Net design [42] to split feature maps in channel dimension into four subgroups of convolutions separately. Each subgroup consists of temporal and spatial convolutions configured serially, while one subgroup has temporal convolution only. In addition, output features from each subgroup flow to the next convolutional block and the neighboring subgroup through a residual connection, except for one subgroup without a residual-like connection (see DCTA submodule in Figure 4 for details). Thus, the last subgroup *aggregately* receives refined spatiotemporal features from former subgroups.

Regarding the temporal convolution, we arrange the layers in a stacked and *densely connected* fashion. Notably, its parameters are shared across subgroups. In this work, the number of temporal convolution layers for stacking is three, and these stacked layers are placed in three subgroups having a residual connection. More specifically, the first layer receives the encoded features from the summation of ME and MvE; the second layer receives input features from the first layer; and for the third layer, its input is formed from the summation of all the preceding layers' output features. Formally,

$$\left.\begin{aligned} X'_0 &= K_{temp} * X \\ X'_1 &= K_{temp} * X'_0 \\ X'_T &= K_{temp} * (X'_0 + X'_1) \end{aligned}\right\} \tag{6}$$

where K_{temp}, X, $X'_i \in \mathbb{R}^{NHW \times C \times T}$, $X'_T \in \mathbb{R}^{NHW \times C \times T}$ denote 1D convolution with a kernel size of 3, initial input features, output features from the i–th layer and the final result of the last temporal layer. We omit the necessary permutation and reshape X and X'_T for simplicity. After that, a 3×3 spatial convolution follows, as stated in the previous paragraph. For all subgroups in the DCTA submodules, the process can mathematically be expressed as:

$$\left.\begin{aligned} X'_0 &= K_{temp} * X \\ X'_1 &= K_{spa} * X'_T \\ X'_2 &= K_{spa} * (X'_1 + X'_T) \\ X'_3 &= K_{spa} * (X'_2 + X'_T) \end{aligned}\right\} \tag{7}$$

where $X'_i \in \mathbb{R}^{NT \times \frac{C}{4} \times H \times W}$, K_{spa} and K_{temp} are output features of the i–th subgroup, spatial convolution and part-shift temporal convolution from [6], respectively. Lastly, we concatenate across channel dimensions to obtain the final output features X':

$$X' = concat([X'_i]), \quad i = [0, 1, 2, 3] \tag{8}$$

where $X' \in \mathbb{R}^{NT \times C \times H \times W}$. Notice that in Figure 4, indices of subgroups from left to right are from 0 to 3, correspondingly.

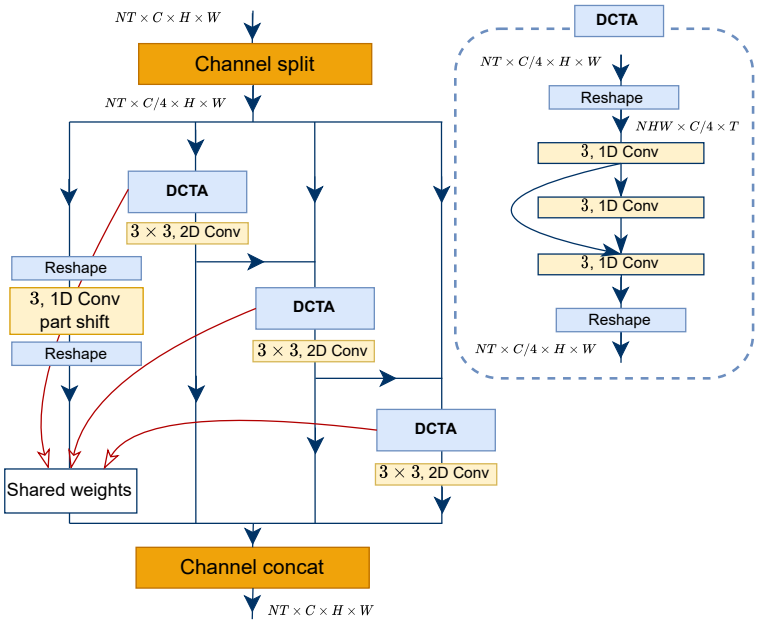

Figure 4. A detailed diagram showing the architecture of the DCTA submodule inside the Res2Net module.

3.3. Multi-View Excitation (MvE)

As illustrated in Figure 5, the MvE submodule has three branches to extract beneficial information from different views, similar to that in [11]. Given an input feature $X \in \mathbb{R}^{NT \times C \times H \times W}$ for branch TH, we utilize a 1×1 2D convolution layer K_{red} to reduce the channel number for efficiency with a ratio of r = 16, identical to Equation (1). Then, the tensor dimension is reshaped to comply with the desired dimension, i.e., $NT \times \frac{C}{r} \times H \times W \rightarrow NW \times \frac{C}{r} \times T \times H$. After that, a shared *channel-wise* convolution layer K is utilized to produce transformed feature maps X'_{TH}. Formally,

$$X'_{TH} = K * X', \quad X'_{TH} \in \mathbb{R}^{NW \times \frac{C}{r} \times T \times H} \tag{9}$$

The last step is to reshape back the tensor dimension, i.e., $NW \times \frac{C}{r} \times T \times H \rightarrow NT \times \frac{C}{r} \times H \times W$. The rest of the branches are processed accordingly to produce X'_{TW} and X'_{HW}. If we have obtained all the outputs from the other two branches, then the new X' is a convex combination of the X'_i:

$$X' = \sum_i \alpha_i * X'_i, \quad i \in [TH, TW, HW] \tag{10}$$

where α is a weighted average with constraints of $\sum_i \alpha_i = 1$ and each of the $\alpha_i \geq 0$. We argue that each branch will contribute differently to the performance of the model. The rest of operations are identical to Equations (4) and (5) to obtain attentive multi-view feature maps X_{MvE}.

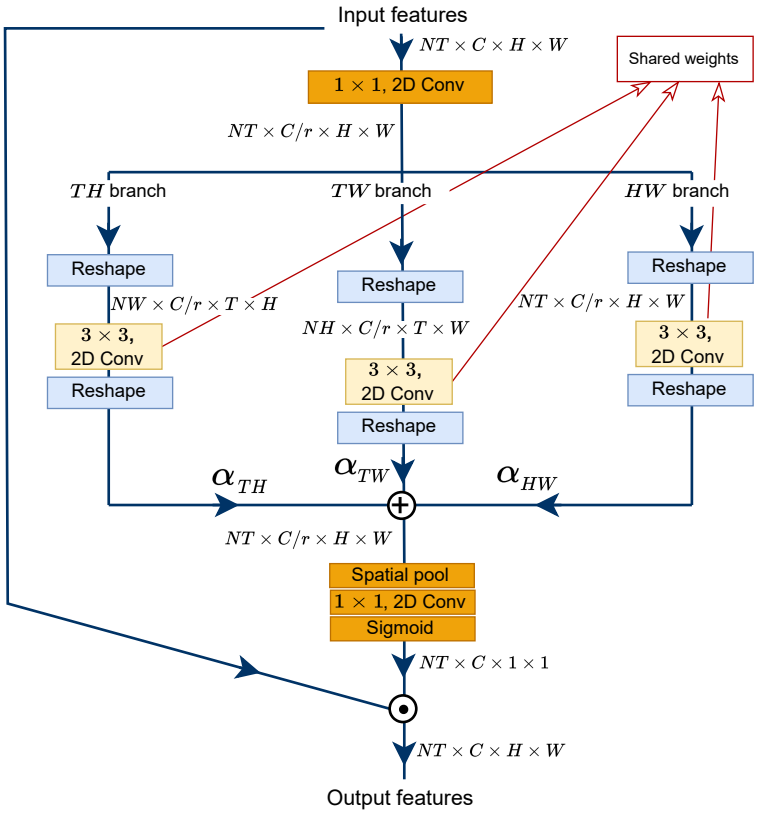

Figure 5. Detailed architecture of MvE submodule.

The initial work of the multi-view design was proposed by Li et al. with their team DEEP HRI [11]. Different from their work, our work introduces the excitation algorithm to the MvE submodule so that it has a kind of attention mechanism.

3.4. Meta Block

For comparative purposes, we adopt 2D ResNet-50 as a backbone like other state-of-the-art methods [33–35]. As shown in Figure 6, each "conv2" in all residual blocks (conv2_x until conv5_x) is replaced by META. In total, we insert 16 blocks of META to endow the network with the ability to learn both spatiotemporal and motion representations efficiently. When feeding feature maps to the DCTA submodule, we sum all of the output features generated from ME, MvE and the former convolution block (denoted by X_{ME}, X_{MvE} and X, respectively) to obtain X'.

$$X' = X_{ME} + X_{MvE} + X, \quad X' \in \mathbb{R}^{NT \times C \times H \times W} \tag{11}$$

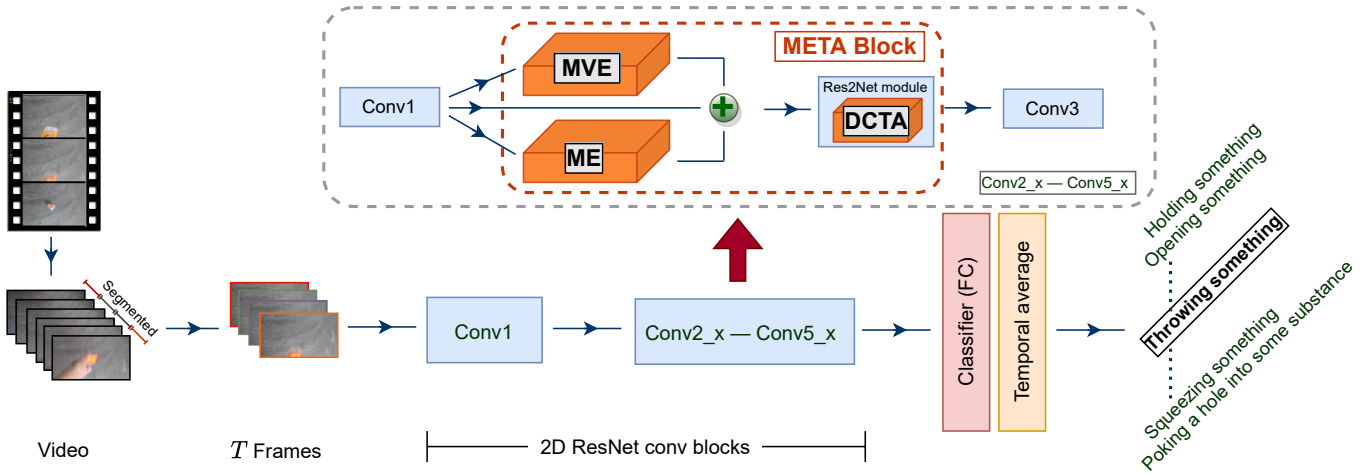

Figure 6. An overview of our proposed model implemented in 2D ResNet50 [43] architecture. We replace the original "conv2" with ME, MvE and DCTA inside every residual block to construct the META block. Inside the Res2Net [42] module, we insert a DCTA submodule. Details on data flow are given in Section 4.2.1.

3.5. Discussion with TEA

We want to highlight the differences between our work and TEA in this subsection. TEA adopted feature-level motion representation and enhanced it by excitation strategy with negligible extramodel parameters. Unlike TEA, which only considers X in parallel with the output of ME, we also added output features from the MvE submodule, as in Equation (11). Moreover, the network enjoys richer spatiotemporal and motion representation features since we re-calibrate the features by *both* ME and MvE submodules.

Regarding temporal aggregation inside the Res2Net module, TEA adopted it to enable their network to model the long-range spatiotemporal relationship by adding a local temporal convolution to each subgroup of convolution. However, in our work, we also added local temporal convolutions in each subgroup of convolution and arranged them in a *stacked up* and *densely connected* manner.

4. Experiment and Evaluation

In the following section, we explain our experiments in detail. Firstly, we describe the datasets we used and explain how we implement training and testing strategies, including hyperparameter settings. We also perform certain ablation experiments to investigate the contribution of each component of META. Later, we present the results and analysis along with the discussion.

4.1. Datasets

Our proposed method is evaluated on three large-scale action recognition benchmark datasets, i.e., Something-Something v1, Jester and Moments-in-Time Mini.

An action classification on the Something-Something v1 [44], a motion-centric type of dataset with 174 classes, requires temporal understanding to classify an action. This dataset is designed to emphasize the interaction between human and object, for example, "*Throwing something*" and "*Throwing something in the air and catching it*". It contains 108,499 videos, with 86,017 in the training set and 11,522 in the validation set. Jester [45], which is also considered a temporal-related dataset, consists of 118,562 training videos, 14,787 validation videos and fewer categories than the Something-Something v1 dataset, i.e., 27. Example actions are "*Swiping up*" and "*Zooming out with two fingers*". The Moments-in-Time Mini dataset [46] is a large-scale human-annotated collection of one hundred thousand short videos corresponding to dynamic events unfolding within three seconds; "*boxing*" and "*repairing*" are the two examples of categories. This dataset provides 100,000 videos for training and 10,000 for validation. It involves 200 action categories and offers a balanced

number of videos in each category. While the previous datasets are more temporal-related, the Moments-in-Time Mini dataset can be considered both a temporal- and scene-related dataset. Frames have already been extracted from all videos in the Something-Something v1 and Jester datasets when they are made publicly available. However, in the Moments-in-Time Mini dataset, we must extract RGB frames from the videos at 30 frames per second at a resolution of 256 by 256. Figure 7 shows some images with their classes for the aforementioned datasets.

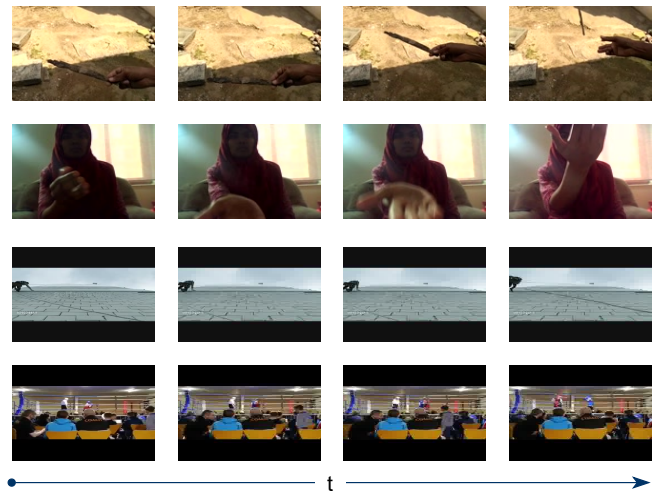

Figure 7. Examples of frames from (top-down) Something-Something v1 (“*Throwing something*”), Jester (“*Swiping up*”) and Moments-in-Time Mini (“*repairing*”, “*boxing*”) datasets.

4.2. Implementation Details

4.2.1. Training

We conduct all experiments on one Nvidia Quadro P6000 GPU card with PyTorch as the deep learning framework. We follow a sparse sampling strategy by TSN [7]. We extract T frames randomly from a number of evenly divided segments (in all our experiments, $T = 8$). Selected frames go through the network, and simple temporal pooling strategy is utilized to averagely predict an action for an entire video. Random scaling and cropping are applied as data augmentation. The size of the shorter side of the frame is cropped to 256 and resized to 224×224 to serve as the final frame size; hence, the final input shape is $NT \times 3 \times 224 \times 224$. Before the training started, we loaded our base model with weights trained on the popular ImageNet dataset [16,17]. As we adopt the Res2Net module for residing the DCTA submodule, we select the publicly available *res2net50 26w 4s* (available at: https://shanghuagao.oss-cn-beijing.aliyuncs.com/res2net/res2net50_26w_4s-06e79181.pth, accessed on 2 February 2022) pre-trained weights.

Regarding the hyperparameters for the Something-Something v1 and Moments-in-Time Mini datasets, the batch size, initial learning rate and dropout rate are set to 8, 0.0025 and 0.5, respectively. Moreover, the learning rates are decreased by factors of 10 at 30, 40 and 45 epochs and stops at 50 epochs. For the Jester dataset, the model is optimized for 30 epochs and the dropout is set to 0.5. Then, a learning rate is started at 0.0025 and reduced by factors of 10 at 10, 20 and 25 epochs. In addition, we unfreeze all instances of batch normalization layers during training. For the network optimizer, we select SGD with a momentum of 0.9 and a weight decay of 5×10^{-4}. When setting the learning rate and weight decay for the classification layer on the three datasets above, we follow [33], i.e., $5\times$ higher than other layers.

As a final thought, as suggested by [47], the learning rate must be matched with the batch size, i.e., the corresponding learning rate must be $2\times$ higher when the batch size is scaled up by two. For example, if the learning rate changes from 2.5×10^{-3} to 5×10^{-3}, then batch size should increase from 8 to 16.

4.2.2. Testing

We follow settings from [33] to adopt two methods as testing protocols: (1) efficient protocol, with frames × crops × clips is 8 × 1 × 1 and cropped 224 × 224 at central region as final frame size; and (2) accuracy protocol, with frames × crops × clips is 8 × 3 × 10, full resolution images (256 × 256 for final input size for frames) and averaged softmax scores for all clips for final prediction. When comparing with other recent works, we apply the accuracy protocol, as in Tables 1 and 2. For the Moments-in-Time Mini dataset, as in Table 3, we apply the efficient protocol.

4.3. Results on Benchmarking Datasets

We report our experimental results and compare them with state-of-the-art methods. We list TSM to act as the baseline for Tables 1 and 2. Since META is designed to function on CNN-based networks, we primarily compare our work with others whose networks are the same type as META to make relevant comparisons. Nevertheless, we still include recent Transformer-based networks in our comparison to demonstrate that META can achieve competitive accuracy while still being lightweight. We also include some successful predictions of META compared with other works on the three datasets we used.

4.3.1. Something-Something V1

Something-Something v1 can be categorized as a temporal-related dataset; thus, ME and DCTA play important roles here. We divide Table 1 into four compartments; the upper part contains 3D CNNs, followed by 2D-based CNNs, Transformer networks and lastly, our model.

Table 1. The comparison result of META against other state-of-the-art methods on the Something-Something v1 dataset. RN in the column backbone indicates ResNet. We list UniFormer methods with 16 input frames for relevant comparison. The highest accuracies for CNN-based networks are highlighted in bold.

Methods	Backbone	Pre-Train	Inputs	FLOPs	Param.	Top-1 (%)	Top-5 (%)
Three-Dimensional CNNs: ECO RGB from [6]	BNInception + 3D RN-18	Kinetics	8 × 1 × 1	32.0 G	47.5 M	39.6	–
ECO RGB from [6]			16 × 1 × 1	64.0 G	47.5 M	41.4	–
I3D NL RGB [14]	3D RN-50	ImgNet + Kinetics	32 × 1 × 2	168.0 G × 2	35.3 M	44.4	76.0
I3D NL+GCN RGB [14]				303.0 G × 2	62.2 M	46.1	76.8
Two-Dimensional CNNs: TSM RGB [6]	2D RN-50	ImgNet	8 × 1 × 1	33.0 G	24.3 M	45.6	74.2
TSM RGB			16 × 1 × 1	65.0 G	24.3 M	47.2	77.1
TSN from [6]			8 × 1 × 1	33.0 G	24.3 M	19.7	46.4
STM [32]			8 × 3 × 10	33.3 G × 30	24.0 M	49.2	79.3
STM			16 × 3 × 10	67.0 G × 30	24.0 M	50.7	80.4
TEA [33]			8 × 1 × 1	35.1 G [2]	26.1 M [2]	48.9	78.1
TEA			8 × 3 × 10	35.1 G × 30 [2]	26.1 M [2]	51.7	**80.5**
ACTION-NET [34]			8 × 1 × 1	34.7 G	28.1 M	47.2 [3]	75.2 [3]
MEST [35]			8 × 1 × 1	34.0 G	25.7 M	47.8	77.1
MEST			16 × 1 × 1	67.0 G	25.7 M	50.1	79.1
AIA TSM [26]			8 × 1 × 1	33.1 G	23.9 M	49.2	77.5
SMNet [36]			8 × 3 × 10	33.1 G × 30	23.9 M	49.8	79.6
Transformers:							
UniFormer-B [40]	Transformer	Kinetics	16 × 1 × 1	96.7 G	49.7 M	55.4	82.9
UniFormer-S [40]				41.8 G	21.3 M	53.8	81.9
EAN RGB+LMC [41]	Transformer + 2D RN-50	ImgNet	(8 × 5) × 1 × 1	37.0 G	36.0 M [1]	53.4	81.1
Ours:							
META	2D RN-50	ImgNet	8 × 1 × 1	35.6 G	26.6 M	50.1	78.5
META			8 × 3 × 1	35.6 G × 3	26.6 M	51.0	79.3
META			8 × 3 × 10	35.6 G × 30	26.6 M	**52.1**	80.2

[1] Not counting Latent Motion Code (LMC) module parameters. [2] Re-counted using official public code for digit precision. [3] Our implementation using official public code.

According to the table, our methods outperform the baseline for the 8 × 1 ×1 layout by sizable margins of 4.5% and 4.3% for top-1 and top-5 accuracy, respectively, while the FLOPs are only 1.08× higher. Our work is superior in that it significantly outperforms the baseline method's 16 frames with a 2.9% accuracy improvement and low FLOPs, even when only eight frames and the one clip–one crop methodology are used. For the methods listed in the first compartment, we outperformed their work significantly. We considerably outscored I3D NL+GCN by only using eight frames, about 17× fewer FLOPs and fewer than half the parameters the I3D network used. A more competitive result is shown in the third compartment, where we outperform all current state-of-the-art methods in terms of top-1 accuracy. The nearest score to ours is TEA, where we obtain a substantially higher margin (52.1% vs. 51.7%), except top-5 accuracy is 0.3% lower (80.2% vs. 80.5%) when employing 10 clips. For comparison with SMNet [36], a more recent work, we noticeably outperform their work by big margins of 2.3% and 0.6% for top-1 and top-5 accuracy, respectively. This definitely demonstrates our superior submodules of MvE and DCTA combined with ME, considering SMNet also equipped their network with motion encoding.

When comparing our work with recent Transformer-based state-of-the-art methods, however, META is inferior to those methods presented in the middle part. Without considering FLOPs and the number of parameters, META is 3.3% less accurate than UniFormer-B in terms of top-1 accuracy and 1.3% lower than EAN RGB+LMC. According to [48], Transformer strength comes from its architecture, which was built to aggregate global information earlier due to self-attention. In addition to striking differences in architecture concept, we found that UniFormer used Kinetics as its pre-trained model, while we only pre-trained our model from ImageNet. Moreover, our model takes eight frames as the input image, while the Transformer-based models require more frames than us to serve as an input image.

Figure 8 shows a visual comparison of CNN-based techniques using a ball chart. We report the top-1 accuracy with respect to floating-point operations in gigabyte (GFLOPs). Accuracies are calculated using only center crop and single forward pass unless otherwise specified. The plot demonstrates how we consistently excel over comparable works while keeping FLOPs to a minimum level (only 1.08× as many as TSM). For our method and TEA, we find that total accuracy may be improved by a factor of ±1.05, at the expense of computational costs that increase to well over a thousand GFLOPs. The plot shows that overall, 2D CNNs may outperform 3D CNNs when the 2D-based network is provided with sufficient temporal feature learning.

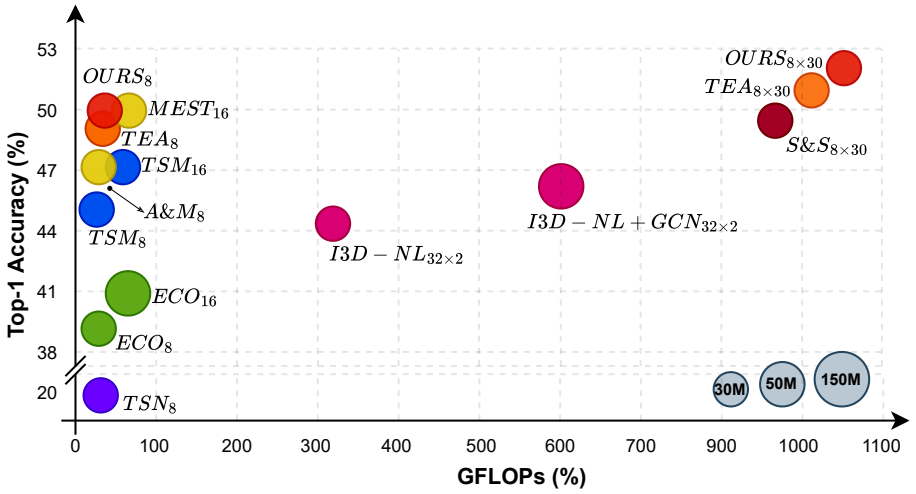

Figure 8. Ball chart reporting the top-1 accuracy *vs.* computational complexity (in GFLOPs). The size of each ball indicates model complexity. "A & M" corresponds to ACTION-NET [34] and MEST [35], while "S & S" denotes STM [32] and SMNet [36], respectively. We merge their icon since they share similar numbers of accuracy and GFLOPs.

4.3.2. Jester

Likewise, Jester is classified as a temporal-related dataset. Our experiment result is provided in Table 2. Clearly, our work demonstrates superiority on the Jester dataset in terms of top-1 accuracy compared with the baseline (97.1% vs. 97.0%) and other state-of-the-art methods, except for ACTION-NET, where we obtained the same accuracy. Our interesting finding according to this table is: Both META and ACTION-NET, which are equipped with a motion representation module, achieved only a slightly higher accuracy than TSM (△0.1% of accuracy) without a motion representation module in it. Though, admittedly, we need further experiments to verify this, we think that motion encoding may have less meaning for this dataset. Moreover, results from TEA and MEST confirm our thoughts, as both methods proposed this module, and the performance is inferior compared with ours ({96.5%, 96.6%} vs. 97.1%).

Table 2. Comparison with state-of-the-art methods on Jester validation set. These methods used 8 frames as model input. The highest accuracies for CNN-based networks are highlighted in bold.

Methods	FLOPs × Views	Top-1 (%)	Top-5 (%)
Two-Dimensioal ResNet-50:			
TSM [6]	33.0 G × 2	97.0	**99.9**
TSN from [6]	–	83.9	99.6
STM [32]	33.3 G × 30	96.6	**99.9**
TEA from [34]	–	96.5	99.8
ACTION-NET [34]	34.7 G × 30	**97.1**	99.8
MEST [35]	34.0 G × 2	96.6	**99.9**
Transformers:			
ViViT-L/16x2 320 [37] from [39]	–	81.7	93.8
TimeSFormer [38] from [39]	–	94.1	99.2
DirecFormer [39]	196.0 G × 3	98.2	99.6
META (Ours)	35.6 G × 30	**97.1**	99.8

Different from the previous dataset comparison, where our work has lesser predictive top-1 and top-5 accuracies than the methods utilizing Transformer, our work demonstrates very competitive results on the Jester dataset. META barely falls short of DirecFormer's top-1 accuracy by 1.1% but surprisingly achieves a slightly higher top-5 accuracy (△0.2%). With the other two Transformer-based methods, we constantly outperform their works in top-1 and top-5 accuracies with significant gaps, proving the superiority of our work.

4.3.3. Moments-In-Time Mini

Unlike the above datasets, this dataset possesses characteristics of temporal-related and scene-related datasets. The performance of our proposed work is still impressive, and Table 3 confirms this. We achieved the highest accuracy in terms of top-5 accuracy. While we obtain lower top-1 accuracy compared with IR-Kinetics400, we want to emphasize that IR-Kinetics400 utilized a Kinetics-400 [13] dataset as their pre-trained weights, whereas we only used ImageNet pre-trained weights. The closest accuracy to META is from I3D-DenseLSTM (Δ0.9% of top-1 acc.), where in their work, they utilized optical flow modality for encoding motion representations and LSTM to model long-range temporal representation, similar to META. Obviously, META is more efficient than I3D-DenseLSTM, as we estimate motion representations in a unified framework.

Table 3. The comparison result of META against other CNNs on the Moments-in-Time Mini validation set. The highest accuracies are highlighted in bold.

Methods	Backbone	Top-1 (%)	Top-5 (%)
TRN from [49]	BNInception + InceptionV3	26.1	48.5
P3D from [49]	P3D ResNet	14.7	33.4
P3D-Kinetics from [50]	P3D ResNet	26.3	–
IR-Kinetics from [50]	Inception-ResNetV2	**30.3**	–
I3D-DenseLSTM [19]	I3D + ResNext	26.5	52.4
META (Ours)	2D ResNet-50	27.4	**53.2**

4.3.4. Example of Successful Predictions

We illustrate some accurate predictions of META over other works in Figure 9. To obtain the probability score in (a), we re-train the model using the official code publicly available (https://github.com/Phoenix1327/tea-action-recognition) (accessed on 11 July 2022), whereas we only load the model with the official weights (https://github.com/V-Sense/ACTION-Net) (accessed on 11 July 2022) for (b). In all scenarios, we confidently achieve top-1 accuracy (indicated by a number in parenthesis) with substantial difference in probability score, whereas other works rank below ours. For instance, (a) informs that META exactly predict an action of "Lifting something with something on it" while TEA measures such action 8th out of the softmax outputs in descending order. This fact demonstrates our predominance over existing related works in three datasets.

Figure 9. Examples of prediction results showing three frames and their probability scores (**a**–**c**): Something-Something v1 ("*Lifting something with something on it*"), Jester ("*Pulling hand in*") and Moments-in-Time Mini ("*Tying*") datasets.

4.3.5. Learning Curve Analysis

During model training, we generate log statements of accuracies for each iteration and save them in a plain text file for further analysis. Figure 10 shows a training visualization in terms of top-1 and top-5 accuracies, with one crop–one clip for both training phase and inference. It is clear from the visualization that after 50 training epochs the model has not improved. Meanwhile, at epoch 30, the accuracy graph turned upward. This is due to our strategy of changing the learning rate at the epoch with our optimization method SGD.

Figure 10. Training curve of our model on Something-Something v1 dataset.

4.4. Ablation Study

We perform some evaluations of our META comprehensively on the Something-Something v1 validation dataset and report the result in this subsection. All experiments utilize ImageNet pre-trained weights and are conducted using the efficient protocol. TSM serves as our baseline.

1. Impact of each module
   We examine how each submodule affects the performance and present the findings in Table 4. It is clear that, in comparison with the baseline, each submodule continuously improves the performance of the 2D ResNet on video action recognition. The DCTA submodule makes the most contribution, improving top-1 accuracy by 2.4% while being computationally efficient with only a 1.7 G overhead gap and the least number of parameters, whereas the other two add 2.0 G of extra FLOPs.
2. Location of META
   We examine the number of META implemented inside four convolution blocks toward accuracy. From Table 5, it is evident that better precision can be attained with more profound METAs placed in convolution blocks. Interestingly, META only requires installing one convolution block to dramatically increase the performance, with top-1 and top-5 accuracies exceeding the baseline by 2.6% and 2.9%, respectively.

Table 4. The comparison result of an individual component against the baseline, including FLOPs and the number of parameters.

Methods	FLOPs	Param.	Top-1 (%)	Top-5 (%)
TSM [6]	33.0 G	23.7 M	45.6	74.2
ME	35.0 G	26.1 M	47.9	77.8
MvE	35.0 G	26.1 M	46.3	76.9
DCTA	34.7 G	25.7 M	48.0	77.0
META	35.6 G	26.6 M	50.1	78.5

Table 5. Examination of the quantity of location METAs inserted into 2D ResNet-50 residual convolutional blocks.

Location	Top-1 (%)	Top-5 (%)	△ Top-1 (%)	△ Top-5 (%)
TSM [6]	45.6	74.2	–	–
conv{2}_x	48.2	77.1	+2.6	+2.9
conv{2,3}_x	49.5	78.1	+3.9	+3.9
conv{2,3,4}_x	49.9	78.2	+4.3	+4.0
META	50.1	78.5	+4.5	+4.3

5. Conclusions

This paper presents a novel building block to overcome the existing problems for the video action recognition task by designing three submodules to construct a META block and integrating it into each residual block of 2D ResNet-50. The proposed block includes excitation of motion and multi-view features followed by densely connected temporal aggregation. While retaining modest computations, our META achieves competitive results on three large-scale datasets compared with its 2D/3D CNN counterparts. Compared with recent Transformer-based networks, our work still achieves competitive results on the Jester dataset, while being inferior on the Something-Something v1 dataset. In the future, we would like to investigate another fusion approach, i.e., channel concatenation in the DCTA submodule, so that all layers are connected, and the current input is the concatenation of the preceding layers. This fusion will guarantee that new information is added to the collective knowledge.

Author Contributions: Conceptualization, Y.Y.J. and M.A.; software, Y.Y.J.; data curation, Y.Y.J.; writing—Original draft, Y.Y.J.; writing—Review and editing, M.A.; visualization, Y.Y.J.; funding acquisition, M.A.; supervision, M.A. All authors have read and agreed to the published version of the manuscript.

Funding: This work is partially supported by Grant-in-Aid for Scientific Research (C), issue number 22K12040.

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: Somthing-Something v1: Not applicable; Jester is available at https://developer.qualcomm.com/software/ai-datasets/jester (accessed on 2 February 2022); Moments-in-Time Mini: Not applicable.

Conflicts of Interest: The authors declare no conflict of interest.

References

1. Stroud, J.C.; Ross, D.A.; Sun, C.; Deng, J.; Sukthankar, R. D3D: Distilled 3D Networks for Video Action Recognition. In Proceedings of the 2020 IEEE Winter Conference on Applications of Computer Vision, WACV 2020, Snowmass Village, CO, USA, 1–5 March 2018; pp. 614–623. [CrossRef]
2. Brezovský, M.; Sopiak, D.; Oravec, M. Action recognition by 3d convolutional network. In Proceedings of the Elmar-International Symposium Electronics in Marine, Zadar, Croatia, 16–19 September 2018; pp. 71–74. [CrossRef]
3. Hara, K.; Kataoka, H.; Satoh, Y. Learning spatio-Temporal features with 3D residual networks for action recognition. In Proceedings of the 2017 IEEE International Conference on Computer Vision Workshops, ICCVW 2017, Venice, Italy, 22–29 October 2017; pp. 3154–3160. [CrossRef]
4. Wang, L.; Li, W.; Van Gool, L. Appearance-and-Relation Networks for Video Classification. In Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition, Honolulu, HI, USA, 21–26 July 2017; pp. 1430–1439. [CrossRef]
5. Zhou, B.; Andonian, A.; Oliva, A.; Torralba, A. Temporal Relational Reasoning in Videos. *Lect. Notes Comput. Sci. Including Subser. Lect. Notes Artif. Intell. Lect. Notes Bioinform.* **2017**, *11205 LNCS*, 831–846. [CrossRef]
6. Lin, J.; Gan, C.; Han, S. TSM: Temporal Shift Module for Efficient Video Understanding. In Proceedings of the IEEE International Conference on Computer Vision, Salt Lake City, UT, USA, 18–23 June 2018; pp. 7082–7092. [CrossRef]
7. Wang, L.; Xiong, Y.; Wang, Z.; Qiao, Y.; Lin, D.; Tang, X.; Van Gool, L. Temporal Segment Networks: Towards Good Practices for Deep Action Recognition. In Proceedings of the Computer Vision—ECCV 2016, Amsterdam, The Netherlands,

11–14 October 2016; Leibe, B., Matas, J., Sebe, N., Welling, M., Eds.; Springer International Publishing: Cham, Switzerland, 2016; pp. 20–36.
8. Diba, A.; Fayyaz, M.; Sharma, V.; Karami, A.H.; Arzani, M.M.; Yousefzadeh, R.; Gool, L.V. Temporal 3D ConvNets: New Architecture and Transfer Learning for Video Classification. *arXiv* **2017**, arXiv:1711.08200.
9. Qiu, Z.; Yao, T.; Mei, T. Learning Spatio-Temporal Representation with Pseudo-3D Residual Networks. In Proceedings of the IEEE International Conference on Computer Vision, Venice, Italy, 22–29 October 2017; pp. 5534–5542. [CrossRef]
10. Tran, D.; Wang, H.; Torresani, L.; Ray, J.; Lecun, Y.; Paluri, M. A Closer Look at Spatiotemporal Convolutions for Action Recognition. In Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition, Honolulu, HI, USA, 21–26 July 2017; pp. 6450–6459. [CrossRef]
11. Li, C.; Hou, Z.; Chen, J.; Bu, Y.; Zhou, J.; Zhong, Q.; Xie, D.; Pu, S. *Team DEEP-HRI Moments in Time Challenge 2018 Technical Report*; Hikvision Research Institute: Hangzhou, China 2018.
12. Arunnehru, J.; Chamundeeswari, G.; Bharathi, S.P. Human Action Recognition using 3D Convolutional Neural Networks with 3D Motion Cuboids in Surveillance Videos. *Procedia Comput. Sci.* **2018**, *133*, 471–477. [CrossRef]
13. Carreira, J.; Zisserman, A. Quo Vadis, Action Recognition? A New Model and the Kinetics Dataset. In Proceedings of the 2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Honolulu, HI, USA, 21–26 July 2017; pp. 4724–4733. [CrossRef]
14. Wang, X.; Gupta, A. Videos as Space-Time Region Graphs. In Proceedings of the European Conference on Computer Vision (ECCV), Munich, Germany, 8–14 September 2018.
15. Hara, K.; Kataoka, H.; Satoh, Y. Can Spatiotemporal 3D CNNs Retrace the History of 2D CNNs and ImageNet? In Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition, Honolulu, HI, USA, 21–27 July 2017; pp. 6546–6555. [CrossRef]
16. Deng, J.; Dong, W.; Socher, R.; Li, L.J.; Kai, L.; Li, F.-F. ImageNet: A large-scale hierarchical image database. In Proceedings of the 2009 IEEE Conference on Computer Vision and Pattern Recognition, Miami, FL, USA, 20–25 June 2009; pp. 248–255. [CrossRef]
17. Russakovsky, O.; Deng, J.; Su, H.; Krause, J.; Satheesh, S.; Ma, S.; Huang, Z.; Karpathy, A.; Khosla, A.; Bernstein, M.; et al. ImageNet Large Scale Visual Recognition Challenge. *Int. J. Comput. Vis.* **2014**, *115*, 211–252. [CrossRef]
18. Wang, X.; Miao, Z.; Zhang, R.; Hao, S. I3D-LSTM: A New Model for Human Action Recognition. *IOP Conf. Ser. Mater. Sci. Eng.* **2019**, *569*, 32035. [CrossRef]
19. Joefrie, Y.Y.; Aono, M. Action Recognition by Composite Deep Learning Architecture I3D-DenseLSTM. In Proceedings of the 2019 International Conference of Advanced Informatics: Concepts, Theory and Applications (ICAICTA), Yogyakarta, Indonesia, 20–21 September 2019; pp. 1–6. [CrossRef]
20. Mutegeki, R.; Han, D.S. A CNN-LSTM Approach to Human Activity Recognition. In Proceedings of the 2020 International Conference on Artificial Intelligence in Information and Communication (ICAIIC), Fukuoka, Japan, 19–21 February 2020; pp. 362–366. [CrossRef]
21. Hochreiter, S.; Schmidhuber, J. Long Short-Term Memory. *Neural Comput.* **1997**, *9*, 1735–1780. [CrossRef] [PubMed]
22. Xie, S.; Sun, C.; Huang, J.; Tu, Z.; Murphy, K. Rethinking Spatiotemporal Feature Learning: Speed-Accuracy Trade-offs in Video Classification. In Proceedings of the European Conference on Computer Vision (ECCV), Munich, Germany, 8–14 September 2018.
23. Zolfaghari, M.; Singh, K.; Brox, T. ECO: Efficient Convolutional Network for Online Video Understanding. *Lect. Notes Comput. Sci. Incl. Subser. Lect. Notes Artif. Intell. Lect. Notes Bioinform.* **2018**, *11206*, 713–730. [CrossRef]
24. Feichtenhofer, C.; Fan, H.; Malik, J.; He, K. Slowfast Networks for Video Recognition. In Proceedings of the IEEE International Conference on Computer Vision, Seoul, Republic of Korea, 27 October–2 November 2019; pp. 6201–6210. [CrossRef]
25. Hu, J.; Shen, L.; Albanie, S.; Sun, G.; Wu, E. Squeeze-and-Excitation Networks. *IEEE Trans. Pattern Anal. Mach. Intell.* **2017**, *42*, 2011–2023. [CrossRef] [PubMed]
26. Hao, Y.; Wang, S.; Cao, P.; Gao, X.; Xu, T.; Wu, J.; He, X. Attention in Attention: Modeling Context Correlation for Efficient Video Classification. *IEEE Trans. Circuits Syst. Video Technol.* **2022**, *32*, 7120–7132. [CrossRef]
27. Sevilla-Lara, L.; Liao, Y.; Güney, F.; Jampani, V.; Geiger, A.; Black, M.J. On the Integration of Optical Flow and Action Recognition. In Proceedings of the Pattern Recognition, Long Beach, CA, USA, 16–20 June 2019; Brox, T., Bruhn, A., Fritz, M., Eds.; Springer International Publishing: Cham, Switzerland, 2019; pp. 281–297.
28. Abdelbaky, A.; Aly, S. Two-stream spatiotemporal feature fusion for human action recognition. *Vis. Comput.* **2021**, *37*, 1821–1835. [CrossRef]
29. Feichtenhofer, C.; Pinz, A.; Wildes, R. Spatiotemporal Multiplier Networks for Video Action Recognition. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Honolulu, HI, USA, 21–26 July 2017.
30. Zach, C.; Pock, T.; Bischof, H. A Duality Based Approach for Realtime TV-L 1 Optical Flow. In *Pattern Recognition*; Springer: Berlin/Heidelberg, Germany, 2007; pp. 214–223. [CrossRef]
31. Wang, L.; Xiong, Y.; Wang, Z.; Qiao, Y.; Lin, D.; Tang, X.; Gool, L.V. Temporal Segment Networks for Action Recognition in Videos. *IEEE Trans. Pattern Anal. Mach. Intell.* **2019**, *41*, 2740–2755. [CrossRef] [PubMed]
32. Jiang, B.; Wang, M.; Gan, W.; Wu, W.; Yan, J. STM: Spatiotemporal and motion encoding for action recognition. In Proceedings of the IEEE International Conference on Computer Vision, Seoul, Republic of Korea, 27 October–2 November 2019; pp. 2000–2009. [CrossRef]

33. Li, Y.; Ji, B.; Shi, X.; Zhang, J.; Kang, B.; Wang, L. TEA: Temporal Excitation and Aggregation for Action Recognition. In Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition, Seattle, WA, USA, 14–19 June 2020; pp. 906–915. [CrossRef]
34. Wang, Z.; She, Q.; Smolic, A. ACTION-Net: Multipath Excitation for Action Recognition. In Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition, Nashville, TN, USA, 20–25 June 2021; pp. 13209–13218. [CrossRef]
35. Zhang, Y. MEST: An Action Recognition Network with Motion Encoder and Spatio-Temporal Module. *Sensors* **2022**, *22*, 6595. [CrossRef] [PubMed]
36. Yang, Q.; Lu, T.; Zhou, H. A Spatio-Temporal Motion Network for Action Recognition Based on Spatial Attention. *Entropy* **2022**, *24*, 368. [CrossRef] [PubMed]
37. Arnab, A.; Dehghani, M.; Heigold, G.; Sun, C.; Lucic, M.; Schmid, C. ViViT: A Video Vision Transformer. In Proceedings of the 2021 IEEE/CVF International Conference on Computer Vision (ICCV), Montreal, QC, Canada, 10–17 October 2021; pp. 6816–6826.
38. Bertasius, G.; Wang, H.; Torresani, L. Is Space-Time Attention All You Need for Video Understanding? In Proceedings of the 38th International Conference on Machine Learning, Virtual, 18–24 July 2021.
39. Truong, T.D.; Bui, Q.H.; Duong, C.N.; Seo, H.S.; Phung, S.L.; Li, X.; Luu, K. DirecFormer: A Directed Attention in Transformer Approach to Robust Action Recognition. In Proceedings of the Computer Vision and Pattern Recognition, Vancouver, BC, Canada, 21–24 June 2022.
40. Li, K.; Wang, Y.; Gao, P.; Song, G.; Liu, Y.; Li, H.; Qiao, Y. UniFormer: Unified Transformer for Efficient Spatiotemporal Representation Learning. *arXiv* **2022**, arXiv:2201.04676.
41. Tian, Y.; Yan, Y.; Min, X.; Lu, G.; Zhai, G.; Guo, G.; Gao, Z. EAN: Event Adaptive Network for Enhanced Action Recognition. *arXiv* **2021**, arXiv:2107.10771.
42. Gao, S.H.; Cheng, M.M.; Zhao, K.; Zhang, X.Y.; Yang, M.H.; Torr, P. Res2Net: A New Multi-scale Backbone Architecture. *IEEE Trans. Pattern Anal. Mach. Intell.* **2019**, *43*, 652–662. [CrossRef] [PubMed]
43. He, K.; Zhang, X.; Ren, S.; Sun, J. Deep Residual Learning for Image Recognition. In Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition, Boston, MA, USA, 7–12 June 2015; pp. 770–778. [CrossRef]
44. Goyal, R.; Kahou, S.E.; Michalski, V.; Materzynska, J.; Westphal, S.; Kim, H.; Haenel, V.; Fründ, I.; Yianilos, P.; Mueller-Freitag, M.; et al. The "something something" video database for learning and evaluating visual common sense. In Proceedings of the IEEE International Conference on Computer Vision, Venice, Italy, 22–29 October 2017; pp. 5843–5851.
45. Materzynska, J.; Berger, G.; Bax, I.; Memisevic, R. The Jester Dataset: A Large-Scale Video Dataset of Human Gestures. In Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV) Workshops, Seoul, Republic of Korea, 27 October–2 November 2019.
46. Monfort, M.; Andonian, A.; Zhou, B.; Ramakrishnan, K.; Bargal, S.A.; Yan, T.; Brown, L.; Fan, Q.; Gutfreund, D.; Vondrick, C.; et al. Moments in Time Dataset: One Million Videos for Event Understanding. *IEEE Trans. Pattern Anal. Mach. Intell.* **2020**, *42*, 502–508. [CrossRef] [PubMed]
47. Goyal, P.; Dollár, P.; Girshick, R.; Noordhuis, P.; Wesolowski, L.; Kyrola, A.; Tulloch, A.; Jia, Y.; Facebook, K.H. Accurate, Large Minibatch SGD: Training ImageNet in 1 Hour. *arXiv* **2017**, arXiv:1706.02677.
48. Raghu, M.; Unterthiner, T.; Kornblith, S.; Zhang, C.; Dosovitskiy, A. Do Vision Transformers See Like Convolutional Neural Networks? *Adv. Neural Inf. Process. Syst.* **2021**, *34*, 12116–12128.
49. Cai, D. Trimmed Event Recognition (Moments in Time): Submission to ActivityNet Challenge 2018. Available online: http://xxx.lanl.gov/abs/1801.03150 (accessed on 12 July 2022).
50. Guan, S.; Li, H. *SYSU iSEE Submission to Moments in Time Challenge 2018*; Technical Report; School of Data and Computer Science Sun Yat-Sen University: Guangzhou, China, 2018.