Neural Processing Letters (2024) 56:124
https://doi.org/10.1007/s11063-024-11591-3

Multipath Attention and Adaptive Gating Network for Video Action Recognition

Haiping Zhang[1] · Zepeng Hu[1] · Dongjin Yu[1] · Liming Guan[1] · Xu Liu[2] · Conghao Ma[2]

Accepted: 1 March 2024

Abstract

3D CNN networks can model existing large action recognition datasets well in temporal modeling and have made extremely great progress in the field of RGB-based video action recognition. However, the previous 3D CNN models also face many troubles. For video feature extraction convolutional kernels are often designed and fixed in each layer of the network, which may not be suitable for the diversity of data in action recognition tasks. In this paper, a new model called *Multipath Attention and Adaptive Gating Network* (MAAGN) is proposed. The core idea of MAAGN is to use the *spatial difference module* (SDM) and the *multi-angle temporal attention module* (MTAM) in parallel at each layer of the multipath network to obtain spatial and temporal features, respectively, and then dynamically fuses the spatial-temporal features by the *adaptive gating module* (AGM). SDM explores the action video spatial domain using difference operators based on the attention mechanism, while MTAM tends to explore the action video temporal domain in terms of both global timing and local timing. AGM is built on an adaptive gate unit, the value of which is determined by the input of each layer, and it is unique in each layer, dynamically fusing the spatial and temporal features in the paths of each layer in the multipath network. We construct the temporal network MAAGN, which has a competitive or better performance than state-of-the-art methods in video action recognition, and we provide exhaustive experiments on several large datasets to demonstrate the effectiveness of our approach.

Keywords Action recognition · Attention mechanism · 3D convolution · Temporal modeling

1 Introduction

Video action recognition is becoming more and more crucial in many fields as everyone now has a cell phone, and various video-related businesses are popping up. For example, action

Haiping Zhang and Zepeng Hu have contributed equally to this work

✉ Zepeng Hu
hzp78468@gmail.com

Published online: 27 March 2024

recognition can be applied to the review of massive videos uploaded on youtube, for video surveillance of dangerous movements and dangerous behaviours, and even in areas such as robotics motion technology.

Although deep learning has made tremendous progress in the accuracy of action recognition [1–6], the accuracy of some complex action data sets, such as Something-Something v1 [7], is still not satisfactory. Video clips contain two key pieces of information: spatial information and temporal information, which are still not well modelled by existing methods. Spatial information represents static information in a single frame scene, such as action entities, forms, objects, and other information. Temporal information represents the integration of spatial information on multiple frames to capture the nature of the action.

In current deep learning approaches, action recognition is usually implemented through two mechanisms. One common practice is to use two-stream network [4, 8, 9]. One stream is located on RGB frames for extracting spatial information, and the other uses optical streams as input to capture temporal information. With the addition of the optical flow module in the two-stream network, this approach can greatly improve action recognition accuracy. However, the optical flow is very expensive to calculate. Another common method is to learn Spatio-temporal features from RGB images by 3D convolution [10]. 3DCNN can effectively capture spatio-temporal information, and it is widely used in the field of video understanding. However, It cannot obtain specific before and after action differences based on optical flow information as in two-stream networks. Therefore, starting from [1], action recognition 3D CNN networks have evolved from single-path towards multi-path. Similar to dual-stream networks, 3DCNN multipath networks usually use different sampling rates to acquire inputs, which are fed into different paths to acquire spatio-temporal information separately, and finally in aggregating multi-layer path features.

Previous work [1, 2, 11] has explored the importance of temporal and spatial features on action categories and has made excellent contributions. However, multipath networks still face many challenges. Some multipath networks simply extract temporal information or spatial information at each level, which can lead to a decrease in the accuracy of actions at different rates. And some multipath networks extract spatial features and temporal features serially in each layer, which can lead to the loss of some original feature information in the later extracted temporal features. Therefore, we propose MAAGN, which extracts both temporal and spatial features in parallel in each layer of a multilevel network and adaptively and dynamically fuses spatio-temporal features. First, we propose a *spatial difference module* (SDM) module that uses difference operations to reduce useless feature interference without using optical flow images and without the extra computational effort required by two-stream networks. Models such as TDN also propose frame difference operations, where TDN uses interval binning features in k copies, e.g., k5, and subtracts them sequentially to obtain frame difference features. In multipath networks, as stated in slowfast [1], due to the different sampling rates, the network at all levels naturally distinguishes between spatial and temporal semantic information in feature acquisition, so a small granularity feature difference operation may not yield better results. Unlike previous work, SDM directly employs a coarse-grained feature slicing operation, followed by efficiently stacking convolutional blocks using a multilayer neural perceptron, resulting in a feature difference operation more adapted to multilevel networks. Second, we present a *multi-angle temporal attention module* (MTAM) to focus on the video global focus features to determine the action's classification. Finally, we also propose a spatio-temporal *adaptive gating module* (AGM), which can adaptively fuse the temporal and spatial characteristics of each path in a multilevel network, and it is unique in each level of the network. To demonstrate the effectiveness of MAAGN, we conducted experiments on three datasets, Kinetics-400 and Something-Something V1&V2.

The evaluation results show that our MAAGN can achieve a competitive or better performance than state-of-the-art methods in video action recognition. We also performed a detailed ablation study to demonstrate the importance of spatial difference operation and temporal attention and investigated the effect of the specific design of MAAGN. In summary, our main contribution lies in the following four aspects:

- We design a spatial difference module based on the idea of RGB difference for reducing the interference of repetitive information in videos and prove that the difference operation is very effective in reducing the complexity of action recognition modeling.
- We design a new multi-angle temporal attention module for video action modeling and demonstrate experimentally that 3DCNN networks can effectively use attention in action recognition.
- We designed an adaptive gating module that can adaptively fuse the spatio-temporal characteristics of the paths at each layer in a multipath network.
- We have conducted extensive experiments to demonstrate its effectiveness on several action recognition datasets Kinetics-400 [8] and somethingV1&V2 [7]. We also provide ablation experiments to fully and effectively demonstrate the degree of enhancement of the model by each module.

2 Related Work

2.1 Action Video Recognition Method

Attempts at video action recognition, in general, can be divided into two categories, 2D CNN-based [4, 5, 12, 13] or 3D CNN-based [1, 8, 14, 15] methods. Simonyan and Zisserman [4] designed a two-stream network for RGB images as spatial streams and optical flow maps as temporal streams. The two-stream model is applied to each frame input to extract optical stream features and RGB image features, then aggregat each frame optical stream with image features for fusion. However, since the two-stream model requires densely sampled video and appears to be overwhelmed when dealing with long videos, many 2DCNN models based solely on RGB images have been proposed. For example, *TSN* [5] presents to divide the video into multiple snippets and randomly draw frames from each snippet for each training to achieve sparse sampling to reduce the computational cost and then aggregate the results for classification. Subsequent gradual improvements have proposed models related to shifting modules similar to *TRN* [16] and *TSM* [12], replacing the mean pool operation with interpretable relational modules to capture information in the time dimension better. Among them, many networks employ RGB and feature difference operations. For example, *TSN* [5] effectively demonstrated that RGB disparity is an important form of alternative optical flow, and *STM* [17] used a feature disparity approach to model motion information based on [5]. Neural networks are hierarchical, with different layers containing different levels of semantics. However, action recognition networks are fixed over all layers and lack the flexibility and ability to model the multi-level semantics contained in different layers. To address this problem, *GSN* [13] and *AAGCN* [18] present a gating module to control interactions in spatial-temporal decomposition. *TDN* [19] proposes to use the time difference operator to design time difference models for better temporal modeling of short- and long-term motion models.

In the second approach, the researchers transformed the 2D-CNN practice into a 3DCN form or a variant of 2D convolution and 1D temporal convolution to better capture temporal information. C3D [20] is a VGG-based 3DCNN model that can learn features in the time dimension from sequence frames compared to 2D networks. To better represent the motion patterns, I3D [8] designed a Two-stream Inflated 3D ConvNets, combining the pre-computed optical flow with RGB images.Still, the computational cost grew exponentially as a result. 3D CNN networks have very many parameters and, therefore also, require more data to train the network. Due to this problem of 3D CNN networks, many researchers are exploring whether all the convolution cores need to be 3D and whether it is possible to use 3D cores for one part and 2D cores for the other part. Several variants try to reduce the computational cost of 3D convolution by decomposing 3D convolution into a 2D convolution and a 1D temporal convolution[21, 22]. In addition, some approaches starting with SlowFast, have tried to use the idea of two-stream networks to design multi-branch architectures to capture motion or contextual information. Along this line, *Non-local Net* [2] and *TPN* [11] use carefully designed temporal modules or multiple RGB inputs sampled at different FPS to get more information. Some recent work has increased research on temporal modeling, Chen et al. [23] and Yang et al. [24] use ATTENTION to fuse multiple pre-defined convolutional kernels to obtain global temporal information. While the recently proposed *TAM* [25] uses an adaptive temporal modeling approach, the temporal kernel parameters in *TAM* are decomposed into position-sensitive adaptive weights and position-independent adaptive convolution kernels to efficiently capture the dynamic video characteristics. *TAda* [26] modifies the 2D convolution operation in the action video model and proposes that relaxing the temporal invariance of convolution can effectively enhance the temporal modeling capability of convolution.

2.2 Attention Mechanism

It is well known that attention plays an important role in human perception [27–29]. The attention mechanism tells you where to focus and enhances feature expression in key regions. Starting from *SENet* [30], the dependency of channel features is established in 2D CNN networks shown by its *squeeze-and-excitation* (SE) block. *RAN* [31] uses an encoder-decoder style attention module to improve the top1 accuracy of large image classification datasets. Then, *CBAM*, to enhance the connection between the channel domain and the spatial domain in the 2D network, performs a maximum pooling operation and an average pooling operation on the intermediate features separately and then combines the two parts. In this case, the image recognition task is enhanced by explicitly modeling the channel-spatial interdependence. However, all the above networks are only applied in 2DCNN without considering critical information such as temporal properties for video action recognition. We use the 3DCNN multiple pyramid network *TPN* [11] combined with *CBAM* [32] and introduce a new temporal attention mechanism MTAM with a feature difference module SDM, which can make the 3D CNN network and attention mechanism compensate for each other to obtain better action recognition classification results.

3 Multipath Attention and Adaptive Gating Network

In this section, we will start to present the technical details of our proposed network. As illustrated in Fig. 1, the video is equally segmented into T Segments. Following [1], the input $F \in \mathbb{R}^{N\times\frac{T}{\tau}\times C\times H\times W}$ is obtained at different sampling rates τ. We use ResNet as the backbone network, and the i-th layer network extracts the basic features from the input features F_i first through the backbone. In SDM, a difference operation is performed at the

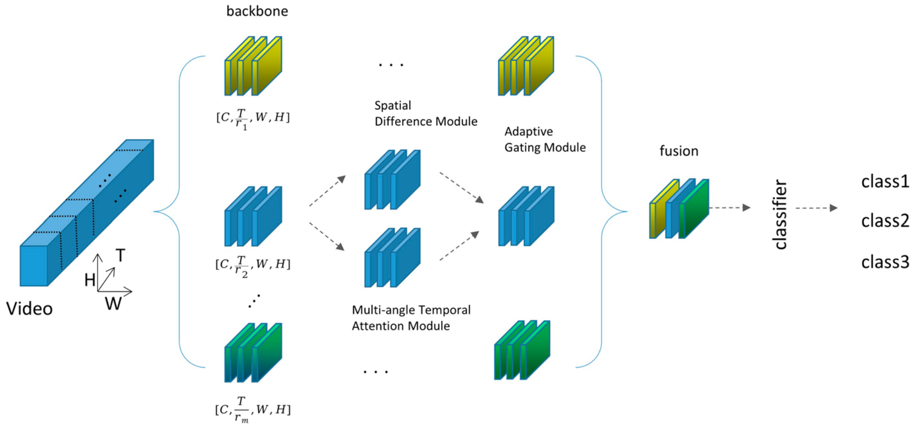

Fig. 1 The framework of MAAGN. The videos are sampled at different sampling rates to obtain multiple sequences of image collections as multiple inputs to the multipath network. MAAGN uses ResNet as the backbone to extract the underlying picture features in each network level so that each level obtains different spatio-temporal semantic features. The spatial and temporal features are subsequently extracted in parallel using a spatial difference module with a multi-scale temporal attention module. Then the spatio-temporal features are dynamically fused by the adaptive gating module. Finally, the features are quadratically aggregated, and their categories are output by the classifier

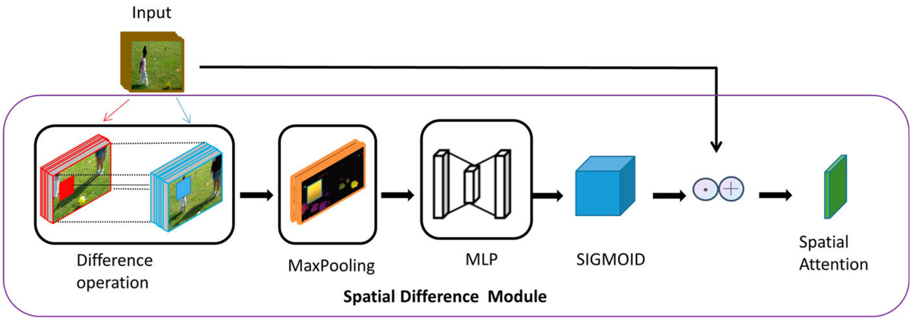

Fig. 2 The overview of SDM. SDM combines a difference operation with neural networks to generate feature maps

spatial level to remove the interference of unimportant information. At the same time, the intermediate features are subjected to MTAM to extract the global action temporal features. After that, AGM dynamically adjusts the gate unit so that each layer of the network gets its own unique gate unit for better fusion of spatio-temporal features. Finally, each branch feature is fused by the same weights to predict the classification result of its video. We will then introduce the SDM, MTAM and AGM sub-modules and specific technical details.

3.1 Spatial Difference Module

We believe that blind stacking of convolutional layers after a certain level does not significantly impact action classification accuracy. Humans also analyze a specific activity by capturing the before-and-after action differences to precisely determine the action type. Therefore, our SDM module focuses on before-and-after feature differences to produce

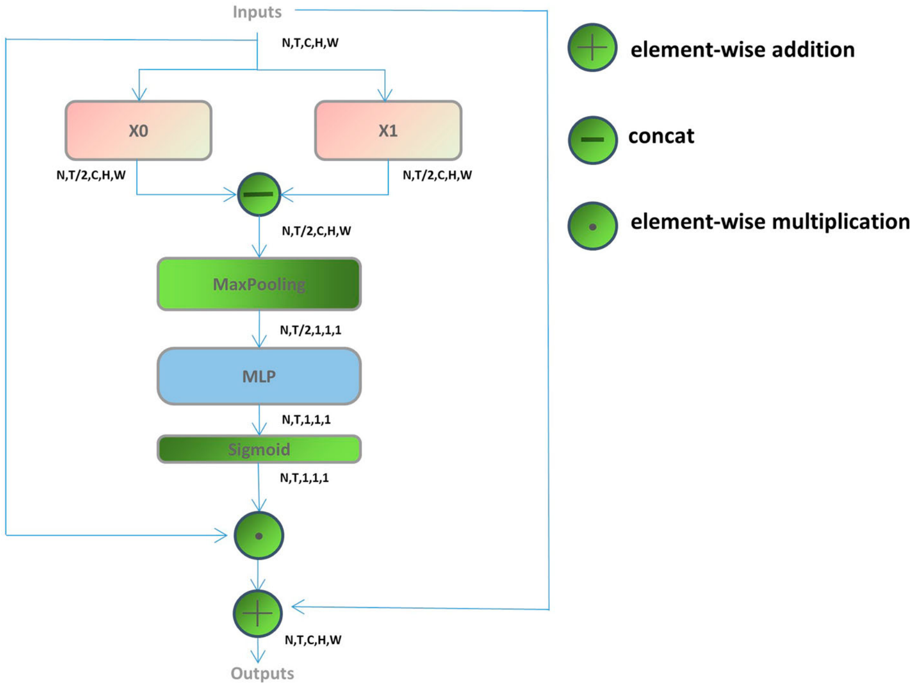

Fig. 3 The illustration of SDM. Our SDM operates based on intermediate feature differences and increases module robustness through a multi-layer perceptron. Eventually, it is fused with the original input by the residual operation to reduce noise interference

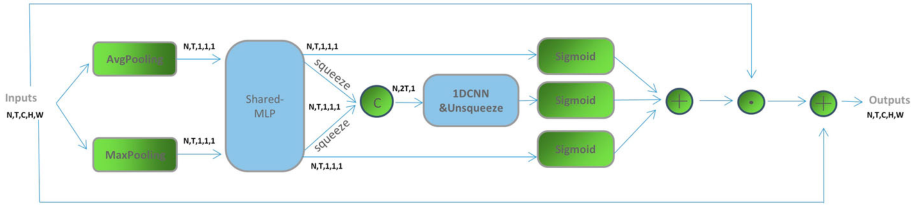

Fig. 4 The illustration of MTAM. Our MTAM proposes a multi-angle extraction Mask mechanism to enhance temporal information extraction using multi-scale information

efficient feature extraction results. As shown in Fig. 2, SDM extracts the before-and-after difference features from the video information just like humans judge an action.

Specifically, SDM performs the difference operation on the intermediate features, effectively extracts the front and back feature variability. As illustrated in Fig. 3, given a central feature $X \in \mathbb{R}^{N \times T \times C \times H \times W}$, input to SDM, we separate them into X_0, X_1, where X_0 is the front half of the feature matrix and X_1 is the back half. We use X_1 minus X_0 to extract the spatial difference, and the result is called F_m, which can be formulated respectively:

$$X_0 = X[:, 0 : T/2, :, :, :] \tag{1}$$

$$X_1 = X[:, T/2 : T, :, :, :] \tag{2}$$

$$F_m = X_1 - X_0 \tag{3}$$

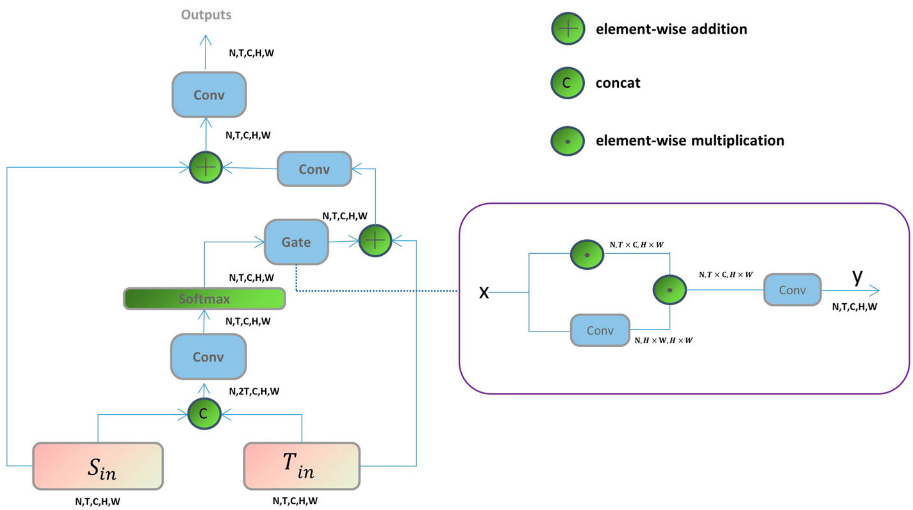

Fig. 5 Illustrating how the adaptive gating module work. In the multipath network training, a unique Gate unit is generated for each part of the path, thus adaptively fusing the two feature maps of the input

We access the spatial information of F_m by maximum pooling, and the result is $F_{max} \in \mathbb{R}^{N\times\frac{T}{2}\times1\times1\times1}$. The F_{max} are then forwarded to a *multi-layer perceptron*(MLP) to produce our feature map $F_{mlp} \in \mathbb{R}^{N\times T\times1\times1\times1}$. Note that the multilayer perceptron consists of two 3D convolutional layers and one ReLU layer, where the two 3D convolutional layers are named W_0 and W_1, and the ReLU follows after W_0 .To reduce the parameter overhead and improve the extraction effect, W_0 is set with a reduction factor r, while W_1 is set with an amplification factor $2r$, where r is *16*. It can be interpreted as:

$$\begin{aligned} F_{mlp} &= MLP(Max(F_m)) \\ &= W_1(ReLU(W_0(Max(F_m)))) \end{aligned} \tag{4}$$

Feeding F_{mlp} to a Sigmoid activation to obtain the spatial mask F^* as:

$$F^* = \sigma(F_{mlp}) \tag{5}$$

where $F^* \in \mathbb{R}^{N\times T\times1\times1\times1}$. In short, the final spatial feature can be represented as:

$$\begin{aligned} Y &= X + X \odot (\sigma(MLP(Max(\mathbb{D}(X_0, X1))))) \\ &= X + X \odot \mathbb{H}(X) \end{aligned} \tag{6}$$

3.2 Multi-angle Temporal Attention Module

MTAM differs from SDM in that SDM focuses on removing features that have not changed in the input video, while MTAM focuses on parts that have changed in the input video. Previous work [30, 32–34] used average-pooling with max-pooling in the intermediate features to gather important cues. Beyond the earlier works, we propose the MTAM module, which can efficiently aggregate the average pooling features and the maximum pooling features to capture sensitive information about the actions in the time dimension, shown in Fig. 4. Compared with the previous multi-scale temporal module, MTAM adds Shared-MLP to the module, focusing on better combining local temporal features with global temporal features. The use of MLP to scale down and then scale up the features can effectively superimpose multiple convolutional blocks to obtain stronger local temporal information, which makes up for the deficiency of SeNet's insensitivity to temporal features. In addition, MTAM uses the squeeze operation, which compresses the feature dimension and speeds up the computation based on reducing the computation of the model. Thanks to the above operation, it is possible to use a simple 1D convolution for feature aggregation.

A 3D maximum pooling layer processes the given intermediate feature input $X \in \mathbb{R}^{N\times T\times C\times H\times W}$ to obtain the global pooling information $F_{max} \in \mathbb{R}^{N\times T\times 1\times 1\times 1}$. Feeding F_{max} to the shared-multi-layer perceptron(SMLP) is similar to SDM, but the amplification and scaling factors are also r. After the shared network is applied to the F_{max}, we generate $F_{temp1} \in \mathbb{R}^{N\times T\times 1\times 1\times 1}$ using the sigmoid activation function. The above operations can be summarized as:

$$F_{temp1} = \sigma(SMLP(Max(X))) \tag{7}$$

Again, X is pooled by a 3D average pooling layer, producing our average pooling feature $F_{avg} \in \mathbb{R}^{N\times T\times 1\times 1\times 1}$. F_{avg} is then convolved by the same SMLP, the SMLP parameter sharing, to generate the second mask $F_{temp2} \in \mathbb{R}^{N\times T\times 1\times 1\times 1}$, which can be interpreted as:

$$Avg(X) = \frac{1}{H\times W}\sum_{i=1}^{H}\sum_{j=1}^{W} X[:,:,:,i,j] \tag{8}$$

$$F_{temp2} = \sigma(SMLP(Avg(X))) \tag{9}$$

The F_{max} and F_{avg} obtained from the previous operation are squeezed separately to generate $F^{'}_{max}$ and $F^{'}_{avg}$, respectively, where $F^{'}_{max}$ and $F^{'}_{avg}$ dimensions are $\mathbb{R}^{N\times T\times 1}$. Those are then concatenated to produce a rough aggregation feature $F_{ios} \in \mathbb{R}^{N\times 2T\times 1}$. The association of average and maximum features is further strengthened by using 1D convolution, where the convolution kernel size is 3 × 1. Finally, applying the unsqueeze operation to restore its original dimension and passing through a sigmoid activation function, we obtain our third mask and the most important one, $F_{temp3} \in \mathbb{R}^{N\times T\times 1\times 1\times 1}$. The calculation of the third mask can be represented as:

$$F^{'}_{avg} = Sqz(SMLP(Avg(X))) \tag{10}$$

$$F^{'}_{max} = Sqz(SMLP(Max(X))) \tag{11}$$

$$F_{temp3} = \sigma(Unsqz(Conv([F^{'}_{avg}, F^{'}_{max}]))) \tag{12}$$

Finally, after summing the three masks and then multiplying them with the original input, the multi-scale mask temporal attention is generated. In short, combining Eq. 7, Eq. 9 and Eq. 12, the temporal attention is computed as:

$$Y = X + X \odot (\mathbb{F}(X) + \mathbb{G}(X) + \mathbb{Z}(X)) \tag{13}$$

3.3 Adaptive Gating Module

Gating modules have been used before in previous works for action recognition [13, 18]. They have split a feature map or transformed a feature map to form a new feature map and then gated it with the original feature map. In AGM, there are two inputs, which are the spatial residual attentional feature map and the multiscale temporal attentional feature map. In previous attention networks, it is usual to serialize these two modules, the spatial attention module, and the multi-angle temporal attention module. Due to this operation, it is impossible to avoid that the feature maps after the extraction of spatial features lose much of the representation of temporality. It is necessary to perform spatial feature processing operation and temporal processing operation on the original feature map separately. Therefore, we chose to parallelize temporal feature extraction and spatial feature extraction. Parallelization allows obtaining optimization at the programming language coding level and performing operations on the feature tensor of multiple modules simultaneously, resulting in a shorter convergence time of the model parameters while obtaining a higher utilization of computing power. Moreover, when serialization is performed, the model is modified at the module level, and the scaling cost is higher than that of parallel processing. In particular, when we modify a mature network model, if we want to add a new module, we need to modify the neighbouring modules, which may impact the extraction effect of the neighbouring modules. The original feature tensor shape extracted from the video is often fixed, and the parallelization process can obtain better results with less cost in terms of scalability. In a multipath network, frames are fed to each layer at multiple pre-defined rates and processed through convolutional blocks. As the number of network layers deepens, each layer can receive various visual information at the feature level. The slowest sampling rate will capture the most spatial semantic variation depending on the sampling rate. In contrast, the fastest sampling rate will best capture the subtle motion of the limbs. Thus, simple summation operations can impact the modeling ability of multipath networks, which is also reflected in our experiments. Based on these observations, we incorporate a gating module that adaptively and dynamically changes to adjust the importance of individual features in different layers. A lot of work [23–25, 35] related to adaptive convolution has been proposed recently. One of them, TAM [25], differs in the model architecture by making changes to multiple convolutional modules of ResNet to obtain global and local temporal information. We use adaptive modules to aggregate features at the fusion layer to obtain better spatiotemporal information. The Dynamic convolution proposed by Wu et al. [23] uses a convolutional kernel generated by a linear module based on the information of the current moment to obtain features independently for each moment. Unlike [23], AGM uses a gate module to generate convolution parameters that better fit the temporal and spatial semantics using nonlinear modules. Thus, the temporal and spatial feature weights are dynamically adjusted to combine the two parallel branches at the global level effectively.

The structure of AGM is shown in Fig. 5, where Gate consists of a neural unit initialized to all zeros, called alpha, and two convolutional layers, with the neural unit alpha solved in a back propagation gradient update. The overall solving process of the Gate module can be expressed as:

$$g = Conv((X \odot \alpha) \odot Conv(x)) \tag{14}$$

At the beginning of the network training, given the spatial features $S_{in} \in \mathbb{R}^{N \times T \times C \times H \times W}$ and temporal features $T_{in} \in \mathbb{R}^{N \times T \times C \times H \times W}$, the output of this module is simply summing S_{in} and T_{in} and convolving them since the Gate module parameter α is 0. Input the features from the upper module to the gate module, and adjust the weight of spatial features adaptively according to the formula Eq. 14. After a period of training, the AGM can find that the spatio-temporal feature map is helpful for the model to determine the action's class, and each layer of the multipath network dynamically acquires the unique gate module parameters for that layer. Specifically, the second dimension of the temporal and spatial features are linked by the concat operation, and they are convolved, where Conv1 is a 3D convolution with a convolution kernel size of 3 × 3. And the spatio-temporal features are processed by normalization to enhance the two feature map links. Subsequently, this spatio-temporal feature is manipulated using a

Gate unit, where the parameters of the Gate unit are constructed based on the input features. The formula is summarized as:

$$Y_0 = Gate(Softmax(Conv(Concat(S_{in}, T_{in})))) \tag{15}$$

We add the gated features with the temporal features. Finally, the initial spatial feature map and the gated temporal feature map are summed and convolved to obtain the gated temporal feature map, where the size of Conv2 and Conv3 convolution kernels are both 3 × 3, and this process is computed as:

$$Y = Conv(S_{in} + Conv(Y_0 + T_{in})) \tag{16}$$

4 Experiments

In this section, we show the experiment results of MAAGN on some relevant action recognition datasets [7, 8]. First, we introduce the relevant datasets and the details of the specific experiments. Next, MAAGN is compared with the state-of-the-art. Finally, we perform ablation experiments for each part of the model and analyze them to show the role of each module.

4.1 Dataset and Training Detail

4.1.1 Video Datasets

We evaluated MAAGN on three video datasets, Kinetics-400 and Something-Something V1&V2, focusing on different action aspects. Kinetics-400 is an important benchmark in behavior recognition. It has about 300k YouTube videos covering 400 categories with video durations under 10 s. The Kinetics-400 dataset contains activities from daily life, and some classes are highly correlated with interactive objects or scene contexts. We train our MAAGN on training data (around 240k videos) and report the performance on validation data (about 20k videos). Something-Something is a large dataset created by crowdsourcing. It is a large labelled dataset of actions between humans and some objects in everyday life. The dataset focuses mainly on some complex movements, so action recognition should focus on motion features rather than objects or scene contexts. V1 contains about 86K training videos and 11K validation videos for over 174 categories, while v2 has more videos, including about 220K videos for test and validation sets. We report performance on the validation sets of Something V1 and V2.

4.1.2 Training

Our model integrates *TPN* [11] to add modules on top of this. Following the setting [1], the input frames are sampled at a specific interval τ from a set of *64* consecutive frames. As with most model setups [2, 38], each sample frame is randomly cropped so that it is [256,320] pixels wide and high, and the image is flipped horizontally to increase the amount of data and a dropout [39] of *0.5* to reduce overfitting. We use ResNet50 [40] pre-trained on the ImageNet [41] dataset as the backbone part of the initialized model. In terms of model parameters, the learning rate is 0.01, the momentum is *0.9*, the weight decay parameter is *0.0001*, and the batch size is *32*. We adopted SGD as optimizer to trained the model on *4* GPUs, and each GPU was trained with *8* batch-size at a time. For Kinetic-400, the learning rate will decay by a factor of *10* at both *100* and *125* epochs and stop training at *150* epochs. For Something-Something V1&V2, the learning rate will decay by a factor of *10* at *75* and *125* epochs, and training will stop at *150* epochs.

4.1.3 Inference

We implement two kinds of the testing scheme. The first way is three-crop [17, 42], which was used in slowfast, where three random crop sizes of 256 × 256 were sampled from the action video for inference. The other way is ten-crop [5], which is the method adopted by *TSN*. The ten-crop samples five 224 × 224 pixel crops and flips them over for a total of ten crops. We used three-crop to test the Kinetic-400 for the three datasets and employed the ten-crop to test the Something-Something V1&V2.

4.2 Comparison with the State-of-the-Art

In this subsection, we will directly compare MAAGN with the previous State-of-the-Art network [1, 12] on the Something-SometingV1 &V2 and Kinetics-400. It is well known that an increase in the sampling frame rate is accompanied by an increase in computational effort and a corresponding increase in the accuracy of the network. Therefore, we will use the same sampling rate and the same test method on various datasets for comparison with other networks.

The experimental results are summarized in Tables 1 and 2. We will use center crop and one clip on the Something-Something and three-crop on the Kinetics-400. Since the behaviours in Something-Something V1& V2, you can hardly infer their specific categories from a single frame, for example, pushing something from left to right. Therefore, we compare the results on the Something-Something V1& V2 with some models focusing on temporal modeling, such as TANet [25], TSM [12], etc., to demonstrate the effectiveness of our model for temporal modeling. As shown in Table 1, we tested the accuracy of all models in the table on the validation sets of SomethingV1&V2. Since [5] does not have a temporal modeling module, it is logical that it has the lowest accuracy ranking, which illustrates the importance of temporal modeling for the Something-Something. Among them, TSM uses the temporal shift module, and TPN uses the multiple pyramid network which can fully utilize the information of temporal features in the data. They can achieve much better accuracy than TSN, 45.6% and 49.0%, respectively. Our MAAGN can perform multiscale mask extraction of the overall temporal features due to the consideration of global temporal. Under the condition that the test input is 8-segments, MAAGN can obtain the top performance on both V1& V2, where the accuracy of Something-Something V1 reaches 50.4%, and the accuracy of 62.7% can be obtained in V2 data. When the test input reaches 16-segments, MAAGN increases the accuracy as the input frames increase, indicating that the performance shown in the 8-segments condition does not reach the upper limit of the model. Compared to TDN and TAda, which use more frames as input, MAAGN still shows competitive results.

On the Kinetics-400 dataset, we compare many methods that have achieved excellent results on this dataset. Analyzing the experimental results, the method using ResNet [40] as the backbone can outperform the way using Inception [43, 44] as the backbone. With ResNet-50 backbone and 32 frames input, our MAAGN outperformed R(2+1)D [14], Nonlocal-Net [2], slowfast [1], and TSM [12] by 1.1%. In addition, with the change of backbone to ResNet-101, MAAGN has an advantage over CorrNet [10] in terms of Top5 accuracy, although Top1

Table 1 The validation set test results for Something-SomethingV1& V2

Method	Backbone	Frames×clips×crops	V1-Top1	V1-Top5	V2-Top1	V2-Top5
TRN-multiscale [16]	BNInception	8×1×1	38.9	–	48.8	76.7
ECO [36]	BNINception	8×1×1	39.6	–	–	–
TSN [5]	Resnet-50	8×1×1	19.7	–	30	–
TSM [12]	Resnet-50	8×1×1	45.6	77.5	59.1	87.5
TPN [11]	Resnet-50	8×1×1	49	78.5	62	88.4
GST [13]	Resnet-50	16×1×1	48.6	76.9	62.6	88.9
TimeSformer-HR [37]	TimeSformer	16×1×3	–	–	62.5	–
TANet [25]	Resnet-50	16×1×1	47.6	77.7	62.5	87.6
TDN [19]	Resnet-50	(8+32)×1×1	52.3	80.6	64.0	88
TAda [26]	Resnet-50	8×2×3	–	–	63.8	87.7
MAAGN	Resnet-50	8×1×1	50.4	79.7	62.4	89.6
MAAGN	Resnet-50	16×1×1	51.9	80.4	63.1	90.8

Table 2 In comparison with other state-of-the-art methods on the Kinetics-400 validation set

Method	Backbone	Frames	Top1	Top5
S3D-G [15]	Inception	64	77.2	93
I3D [8]	Inception	16	71.6	90
R(2+1)D [14]	Resnet-50	16	73.9	90.9
Nonlocal-Net [2]	Resnet-50	32	76.5	92.6
SlowFast [1]	Resnet-50	32	77	92.6
TDN [19]	ResNet-50	16	77.5	93.2
TSM [12]	Resnet-50	16	74.3	93
TAM [25]	ResNet-50	16	76.9	92.9
TADA [26]	ResNet-50	16	76.9	92.7
Nonlocal-R101 [2]	Resnet-101	32	77.7	93.3
SlowFast [1]	Resnet-101	32	77.9	93.2
CorrNet [10]	ResNet-101	32	**79.2**	93.3
MAAGN	Resnet-50	32	**78.1**	**93.2**
MAAGN	Resnet-101	32	78.9	**93.6**

Bold denotes top performance

accuracy is slightly lower than that of CorrNet. SlowFast plus Non-Local Net can achieve better results than us in Kinetics-400 but requires much more computing power.

4.3 Ablation Study

We demonstrate ablation experiments on the Something-Something V1 dataset whit 8-segemtns as input and the following experimental results. We show the Top1 performance of the combination of individual modules and the composition of each module tested using the ten-crop method with an input image of 224× 224 pixels.

Table 3 Study on module combination: we compared the accuracy of the temporal module and spatial module in series with that of the temporal module and spatial module adding AGM in parallel

SDM	MTAM	Top 1
(a) Serial combination module		
		50.80
✓		51.30
	✓	51.40
✓ (first)	✓ (second)	51.70
✓ (second)	✓ (first)	51.60

SDM	MTAM	Summation	AGM	Top 1
(b) Parallel combination module accuracy study				
✓	✓	✓		51.10
✓	✓		✓	51.90

SDM	MTAM	Summation	AGM	Speed on sthv1	Speed on sthv2
(c) Parallel combination module speed study					
✓	✓	✓		10.17 task/s	10.75 task/s
✓	✓		✓	13.26 task/s	14.21 task/s

4.3.1 Study on Module Combinations

We started our ablation study by how to combine the two modules. We explored the performance of each module by adding SDM alone or MTAM alone on the same baseline and finally, the two modules were serially integrated to validate the overall results. Table 3(a) shows the top1 accuracy of each module on the Something-SomethingV1 dataset. We conclude the following: (1) The best performance is achieved by extracting spatial information first and then temporal information. (2) For video extracting temporal information alone works better than extracting spatial information alone. In Table 3(b), we explored the parallel use of SDM and MTAM modules by using the summation operation and adding AGM for the integration of the two modules, respectively. In Table 3(c), we calculate the average speed of the model trained on sthv1 and sthv2 using four Nvidia 3080 graphics cards. Comparing Table 3(a), (b) with (c) , we empirically obtained the following results: (1) The effect of spatio-temporal features obtained by simple parallel summation is much less than that obtained in the serial mode. (2) Using the adaptive gate unit can effectively increase the accuracy in the parallel mode. (3) In parallel mode, spatial and temporal features can be computed simultaneously, thus speeding up the task and reducing the time spent on model training while effectively utilizing the graphics card's computing power.

4.3.2 Study on SDM

We explore how the extracted spatial residual information is aggregated with the original features. As shown in Table 4, we used the following methods: (1) spatial attention points multiplied by original features, (2) spatial attention plus original features, (3) spatial attention points multiplied by original features plus original features, and (4)spatial attention points minus original features. From the experimental results of each SDM method, method (3) is the

Table 4 Study on SDM: we compared different implementations of SDM, including extracting different ways of combining spatial attention

Fusion	Top1
F ⊙ H	50.90
F + H	51.10
F + F ⊙ H	51.30
F − H	50.70

Table 5 Study on MTAM: we compared different implementations of MTAM

MaxPool	AvgPool	ISO	Top1
✓			51.10
	✓		50.90
✓	✓		51.20
✓	✓	✓	51.40

Table 6 Study on model parameters: we compared the parameters under different model compositions

Method	Params	Top1 on sthv1
TPN (our baseline)	33.62 M	50.8
TPN + SDM	34.51 M	51.3
TPN + MTAM	34.84 M	51.4
MAAGN (Serial)	35.76 M	51.7
MAAGN (Parallel)	36.19 M	51.9

best. Different differential operations with larger granularity in multipath networks, such as subtracting the first half of the feature map from the second half of the feature map, can obtain a good correspondence between the front and back feature points. Thanks to these operations, the redundant static information in the video can be effectively removed to prevent overfitting of the convolution parameters, and the dynamic visual information can be better captured by reweighting the features. As the results show, SDM can use approach (3) to further widen the weight gap between the important and unimportant features after performing the difference operation to weaken the trivial action features. Furthermore, we find that method (4) is the least effective, which proves from the side that the extracted attentional features are the most sensitive points in the feature map.

4.3.3 Study on MTAM

In the *MTAM*, it is necessary to perform a maximum pooling and an average pooling of the original features to extract important information. We verified the impact of the following four methods on the accuracy of MTAM: (1) maximum pooling only, (2) average pooling only, (3) average pooling and maximum pooling, (4) average pooling with maximum pooling and aggregation of both, generating multiple masks. As shown in Table 5, we can conclude that the fourth approach can maximize the accuracy of the module, from 51.2% to 51.4%, far exceeding the original effect. Therefore, we conclude that combining average pooling and maximum pooling can extract a greater degree of video feature information. Although it is possible to enhance the temporal field of perception by stacking convolutional blocks and network layers to capture multiple visual temporal information, this approach is too cumber-

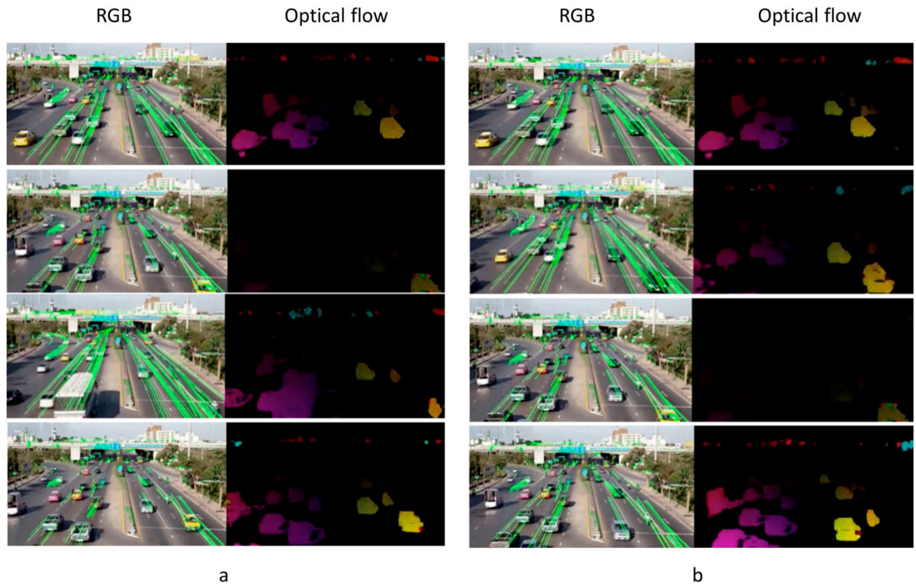

Fig. 6 The input of the motion recognition network contains RGB images and optical flow images

some, and the visual temporal information obtained needs to be richer. A better approach is to extract the information by multi-scale and large-scale algorithms. The combination of mean pool and maximum pool algorithms can extract video feature information deeper, which confirms the effectiveness of larger receptive fields for modeling global and local temporal sequences. The map extracted with multiple masks has more sensitive features.

4.3.4 Study on Model Parameters

As shown in Table 6, we compare the changes in the number of parameters after assembling each module with the baseline. All models use ResNet50 as the backbone, and the rest of the parameter settings are kept the same. As the results show, the classification effect of the baseline model is effectively enhanced with the enhancement of a not-too-large number of parameters.

4.4 Advantage Analysis

In this subsection, we visualize the effect of MAAGN and use it to demonstrate that MAAGN is indeed superior to other networks.

4.4.1 MAAGN Versus Traditional Two-Stream Network

The input requirements for two-stream networks [4] are very strict, requiring both optical flow and RGB maps, as shown in Fig. 6, which is unfavourable for practical industrial applications. As a comparison, MAAGN only needs to be the RGB image as input, the left part of a and b in Fig. 6, which makes more sense for practical applications.

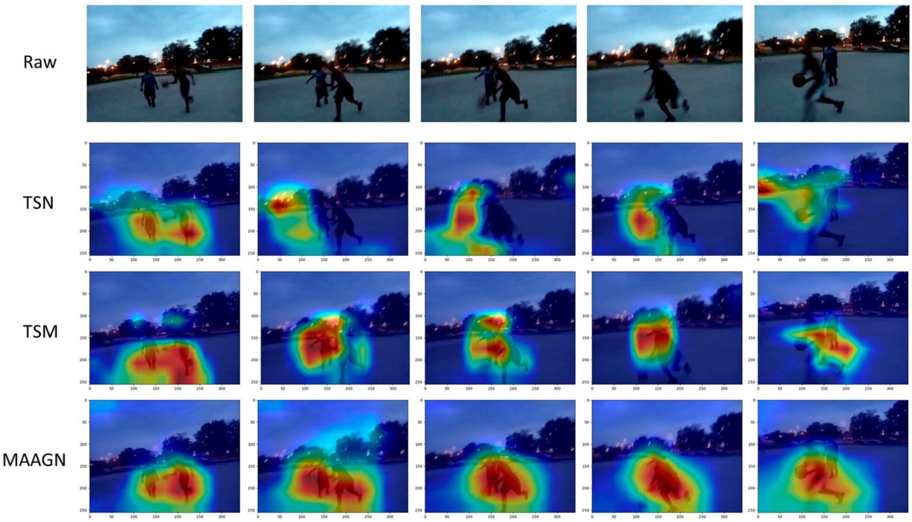

Fig. 7 Visualization of activation maps with GradCAM. Activation maps are generated by TSN, TSM, and our MAAGN for the action of playing basketball. In contrast to TSN and TSM, we can be noticed that MAAGN can focus on the action subject and ignore the irrelevant background

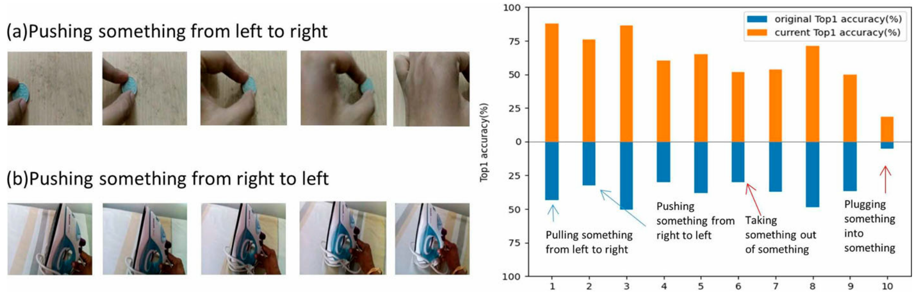

Fig. 8 MAAGN versus TSM on Something-Something v1 dataset with significant improvement in 10 classes. The backgrounds of the two action examples (**a**) and (**b**) are unchanged, the main change is the position of the hand and the object, and the classification of these two actions requires strong temporal modeling. When humans recognize these two actions, they hardly focus on the background, only on the subject of the action. Compared with TSM, we can see that the classification effect for these similar action categories is better than it

4.4.2 MAAGN Versus Other RGB-Based Networks

We used *GradCAM* [45] to visualize the class activation map of the three models, TSN [5], TSM [12], and our MAAGN, for the class playing basketball, and the results are shown in Fig. 7. The results of this visualization demonstrate that (1) *TSN* [5] focuses on the wrong motion regions in the video, (2) *TSM* [12] can localize the athletes but is still not accurate enough, and (3) our *MAAGN* can effectively localize more and more accurate regions thanks to MAAGN's removal of spatial noise and global temporal modeling.

4.4.3 MAAGN Versus TSM

MMAGN can effectively improve the accuracy of motion judgment for a group of movements that are similar but in opposite directions. For example, throwing a ball upward and dropping it downward, with the hand under the ball on top, is throwing up and vice versa for dropping both foci mainly on the position of the hand and the ball, without caring about the background or the face. As shown in Fig. 8, on the Something-Something V1 dataset, our best results on some categories requiring strong temporal modeling are significantly better than previous methods [46–48].

5 Conclusion

In this paper, we propose a new video action recognition network named MAAGN. The main contribution of MAAGN is to present two modules, and one is SDM which captures the information of video before and after frames in space using a difference operation. The other is MTAM which captures the information of global video in time sequence using a temporal attention mechanism. In addition, we applied the attention mechanism to multiple path networks and empirically demonstrated that the attention mechanism works very well on them. We also demonstrate that adaptive fusion can efficiently aggregate spatio-temporal features and reduce training time under parallel operation. In the future, we will continue to expand MAAGN in video understanding, possibly focusing on operations such as the temporal residual module or improving the efficiency of the model for application to real industrial scenarios.

References

1. Feichtenhofer C, Fan H, Malik J, He K (2019) Slowfast networks for video recognition. In: Proceedings of the IEEE/CVF international conference on computer vision, pp 6202–6211
2. Wang X, Girshick R, Gupta A, He K (2018) Non-local neural networks. In: Proceedings of the IEEE conference on computer vision and pattern recognition, pp 7794–7803
3. Karpathy A, Toderici G, Shetty S, Leung T, Sukthankar R, Fei-Fei L (2014) Large-scale video classification with convolutional neural networks. In: Proceedings of the IEEE conference on computer vision and pattern recognition, pp 1725–1732
4. Simonyan K, Zisserman A (2014) Two-stream convolutional networks for action recognition in videos. Adv Neural Inf Process Syst 27
5. Wang L, Xiong Y, Wang Z, Qiao Y, Lin D, Tang X, Gool LV (2016) Temporal segment networks: towards good practices for deep action recognition. European conference on computer vision. Springer, Cham, pp 20–36
6. Wang L, Qiao Y, Tang X (2015) Action recognition with trajectory-pooled deep-convolutional descriptors. In: Proceedings of the IEEE conference on computer vision and pattern recognition, pp 4305–4314

7. Goyal R, Ebrahimi Kahou S, Michalski V, Materzynska J, Westphal S, Kim H, Memisevic, R (2017) The something something video database for learning and evaluating visual common sense. In: Proceedings of the IEEE international conference on computer vision, pp 5842–5850
8. Carreira J, Zisserman A (2017) Quo vadis, action recognition? A new model and the kinetics dataset. In: proceedings of the IEEE conference on computer vision and pattern recognition, pp 6299–6308
9. Feichtenhofer C, Pinz A, Wildes RP (2017) Spatiotemporal multiplier networks for video action recognition. In: Proceedings of the IEEE conference on computer vision and pattern recognition, pp 4768–4777
10. Wang H, Tran D, Torresani L, Feiszli M (2020) Video modeling with correlation networks. In: Proceedings of the IEEE/CVF conference on computer vision and pattern recognition, pp 352–361
11. Yang C, Xu Y, Shi J, Dai B, Zhou B (2020) Temporal pyramid network for action recognition. In: Proceedings of the IEEE/CVF conference on computer vision and pattern recognition, pp 591–600
12. Lin J, Gan C, Han S (2019) Tsm: temporal shift module for efficient video understanding. In: Proceedings of the IEEE/CVF international conference on computer vision, pp 7083–7093
13. Sudhakaran S, Escalera S, Lanz O (2020) Gate-shift networks for video action recognition. In: Proceedings of the IEEE/CVF conference on computer vision and pattern recognition, pp 1102–1111
14. Tran D, Wang H, Torresani L, Ray J, LeCun Y, Paluri M (2018) A closer look at spatiotemporal convolutions for action recognition. In: Proceedings of the IEEE conference on computer vision and pattern recognition, pp 6450–6459
15. Xie S, Sun C, Huang J, Tu Z, Murphy K (2018) Rethinking spatiotemporal feature learning: speed-accuracy trade-offs in video classification. In: Proceedings of the European conference on computer vision (ECCV), pp 305–321
16. Zhou B, Andonian A, Oliva A, Torralba A (2018) Temporal relational reasoning in videos. In: Proceedings of the European conference on computer vision (ECCV), pp 803–818
17. Jiang B, Wang M, Gan W, Wu W, Yan J (2019) Stm: spatiotemporal and motion encoding for action recognition. In: Proceedings of the IEEE/CVF international conference on computer vision, pp 2000–2009
18. Shi L, Zhang Y, Cheng J, Lu H (2020) Skeleton-based action recognition with multi-stream adaptive graph convolutional networks. IEEE Trans Image Process 29:9532–9545
19. Wang L, Tong Z, Ji B, Wu G (2021) Tdn: temporal difference networks for efficient action recognition. In Proceedings of the IEEE/CVF conference on computer vision and pattern recognition, pp 1895–1904
20. Tran D, Bourdev L, Fergus R, Torresani L, Paluri M (2015) Learning spatiotemporal features with 3d convolutional networks. In: Proceedings of the IEEE international conference on computer vision, pp 4489–4497
21. Li Y, Ji B, Shi X, Zhang J, Kang B, Wang L (2020) Tea: temporal excitation and aggregation for action recognition. In: Proceedings of the IEEE/CVF conference on computer vision and pattern recognition, pp 909–918
22. Liu Z, Luo D, Wang Y, Wang L, Tai Y, Wang C, Lu T (2020) Teinet: towards an efficient architecture for video recognition. In: Proceedings of the AAAI conference on artificial intelligence, vol 34, No 07, pp 11669–11676
23. Chen Y, Dai X, Liu M, Chen D, Yuan L, Liu Z (2020) Dynamic convolution: attention over convolution kernels. In: Proceedings of the IEEE/CVF conference on computer vision and pattern recognition, pp 11030–11039
24. Yang B, Bender G, Le QV, Ngiam J (2019) Condconv: conditionally parameterized convolutions for efficient inference. Adv Neural Inf Process Syst, 32
25. Liu Z, Wang L, Wu W, Qian C, Lu T (2021) Tam: temporal adaptive module for video recognition. In: Proceedings of the IEEE/CVF international conference on computer vision, pp 13708–13718
26. Huang Z, Zhang S, Pan L, Qing Z, Tang M, Liu Z, Ang Jr MH (2021) TAda! temporally-adaptive convolutions for video understanding. arXiv preprint arXiv:2110.06178
27. Corbetta M, Shulman GL (2002) Control of goal-directed and stimulus-driven attention in the brain. Nat Rev Neurosci 3(3):201–215
28. Itti L, Koch C, Niebur E (1998) A model of saliency-based visual attention for rapid scene analysis. IEEE Trans Pattern Anal Mach Intell 20(11):1254–1259
29. Vaswani A, Shazeer N, Parmar N, Uszkoreit J, Jones L, Gomez AN, Kaiser L, Polosukhin, I (2017) Attention is all you need. Adv Neural Inf Process Syst, 30
30. Hu J, Shen L, Sun G (2018) Squeeze-and-excitation networks. In: Proceedings of the IEEE conference on computer vision and pattern recognition, pp 7132–7141
31. Wang F, Jiang M, Qian C, Yang S, Li C, Zhang H, Wang X, Tang X (2017) Residual attention network for image classification. In: Proceedings of the IEEE conference on computer vision and pattern recognition, pp 3156–3164

32. Woo S, Park J, Lee JY, Kweon IS (2018) Cbam: convolutional block attention module. In: Proceedings of the European conference on computer vision (ECCV), pp 3–19
33. Zhou B, Khosla A, Lapedriza A, Oliva A, Torralba A (2016) Learning deep features for discriminative localization. In: Proceedings of the IEEE conference on computer vision and pattern recognition, pp 2921–2929
34. Zeiler MD, Fergus R (2014) Visualizing and understanding convolutional networks. European conference on computer vision. Springer, Cham, pp 818–833
35. Wu F, Fan A, Baevski A, Dauphin YN, Auli M (2019) Pay less attention with lightweight and dynamic convolutions. arXiv preprint arXiv:1901.10430
36. Zolfaghari M, Singh K, Brox T (2018) Eco: efficient convolutional network for online video understanding. In: Proceedings of the European conference on computer vision (ECCV), pp 695–712
37. Bertasius G, Wang H, Torresani L (2021) Is space-time attention all you need for video understanding?. In: ICML, vol 2, No 3
38. Simonyan K, Zisserman A (2014) Very deep convolutional networks for large-scale image recognition. arXiv preprint arXiv:1409.1556
39. Hinton GE, Srivastava N, Krizhevsky A, Sutskever I, Salakhutdinov RR (2012)Improving neural networks by preventing co-adaptation of feature detectors. arXiv preprint arXiv:1207.0580
40. He K, Zhang X, Ren S, Sun J (2016) Deep residual learning for image recognition. In: Proceedings of the IEEE conference on computer vision and pattern recognition, pp 770–778
41. Deng J, Dong W, Socher R, Li L J, Li K, Fei-Fei L (2009) Imagenet: a large-scale hierarchical image database. In: 2009 IEEE conference on computer vision and pattern recognition, pp 248–255. IEEE
42. Zhang J, Shen F, Xu X, Shen HT (2020) Temporal reasoning graph for activity recognition. IEEE Trans Image Process 29:5491–5506
43. Szegedy C, Liu W, Jia Y, Sermanet P, Reed S, Anguelov D, Erhan D, Vanhoucke V, Rabinovich, A (2015) Going deeper with convolutions. In: Proceedings of the IEEE conference on computer vision and pattern recognition, pp 1–9
44. Ioffe S, Szegedy C (2015) Batch normalization: accelerating deep network training by reducing internal covariate shift. In: International conference on machine learning, pp 448-456. PMLR
45. Selvaraju RR, Cogswell M, Das A, Vedantam R, Parikh D, Batra D (2017) Grad-cam: visual explanations from deep networks via gradient-based localization. In: Proceedings of the IEEE international conference on computer vision, pp 618–626
46. Luo C, Yuille AL (2019) Grouped spatial-temporal aggregation for efficient action recognition. In: Proceedings of the IEEE/CVF international conference on computer vision, pp 5512–5521
47. Wang Z, She Q, Smolic A (2021) Action-net: multipath excitation for action recognition. In: Proceedings of the IEEE/CVF conference on computer vision and pattern recognition, pp 13214–13223
48. Lee M, Lee S, Son S, Park G, Kwak N (2018) Motion feature network: fixed motion filter for action recognition. In: Proceedings of the European conference on computer vision (ECCV), pp 387–403

Authors and Affiliations

Haiping Zhang[1] · Zepeng Hu[1] · Dongjin Yu[1] · Liming Guan[1] · Xu Liu[2] · Conghao Ma[2]

Haiping Zhang
zhanghp@hdu.edu.cn

Dongjin Yu
yudj@hdu.edu.cn

Liming Guan
glm@hdu.edu.cn

Xu Liu
lxu505576@gmail.com

Conghao Ma
mch@hdu.edu.cn

[1] School of Computer Science and Technology, Hangzhou Dianzi University, Hangzhou 310018, Zhejiang, China

[2] School of Electronics and Information, Hangzhou Dianzi University, Hangzhou 310018, Zhejiang, China