Multimedia Systems
https://doi.org/10.1007/s00530-021-00831-4

SPECIAL ISSUE PAPER

Improved SSD using deep multi-scale attention spatial–temporal features for action recognition

Shuren Zhou[1] · Jia Qiu[1] · Arun Solanki[2]

Received: 19 January 2020 / Accepted: 2 July 2021

Abstract
The biggest difference between video-based action recognition and image-based action recognition is that the former has an extra feature of time dimension. Most methods of action recognition based on deep learning adopt: (1) using 3D convolution to modeling the temporal features; (2) introducing an auxiliary temporal feature, such as optical flow. However, the 3D convolution network usually consumes huge computational resources. The extraction of optical flow requires an extra tedious process with an extra space for storage, and is usually modeled for short-range temporal features. To construct the temporal features better, in this paper we propose a multi-scale attention spatial–temporal features network based on SSD, by means of piecewise on long range of the whole video sequence to sparse sampling of video, using the self-attention mechanism to capture the relation between one frame and the sequence of frames sampled on the entire range of video, making the network notice the representative frames on the sequence. Moreover, the attention mechanism is used to assign different weights to the inter-frame relations representing different time scales, so as to reasoning the contextual relations of actions in the time dimension. Our proposed method achieves competitive performance on two commonly used datasets: UCF101 and HMDB51.

Keywords Action recognition · Multi-scale spatial–temporal feature · Attention mechanism

1 Introduction

Video understanding [1] is an important area of computer vision, and one of the most important is the human action recognition. The action in a video is composed of a series of processes from beginning to end and in between, which is quite different from a still image [2] frozen at a single moment. Many actions in human's view to be classified by observing them change from beginning to end, and the same is true for computers. This brings a problem for action recognition in video: how to effectively construct a temporal feature representation to enable the computer to perform video action recognition task better.

Video data can be very intuitive explained as a 3D spatial–temporal signal [3, 4], some solutions [5–7] trying to find different forms of spatial–temporal fusion features. Although these methods' result is remarkable for some action recognition task, for more complex and time-costing longer action its' performance will drop a lot due to the differences between the same action category and the similarities and fuzziness between actions. There are a number of factors that can lead to big inter-class differences in actions such as appearance features, differences and variations in the people/objects that make up the motion, lighting and imaging conditions, self-occlusion, and cluttered backgrounds. To solve these problems, some methods [8, 9] extract the trajectory [10] of the points of interest from the video sequence to represent the discriminative spatial region. But overall, the challenge of recognizing more complex human action is not well addressed.

In the growing environment of deep learning [11, 12], 3D CNN [13] directly extends the existed 2D CNN network structure [14] to the 3D spatial–temporal domain, and then learns the convolution kernel parameters in the spatial–temporal domain. Karpathy et al. [15] learns the hierarchical structure composed of multi-layer convolution kernel, and to learn the long-range motion features through early fusion, late fusion and slow fusion. It may be due to the lack of

✉ Shuren Zhou
zsr@csust.edu.cn

[1] School of Computer and Communication Engineering, Changsha University of Science and Technology, Changsha 410114, China

[2] School of Information and Communication Technology, Gautam Buddha University, Noida, India

Published online: 14 July 2021

sufficient training video data [16] (compared with a large-scale image dataset [17]), the complexity and difficulty of 3D CNN are increased. The performance of 3D CNN is not particularly good in learning completely semantic information [18], and the recognition result of two-stream CNN network is often better than that of 3D CNN. In fact, compared with 2D images which only need to obtain spatial features [19], 3D CNN needs a huge amount of training data for network training to learn the features of an extra dimension through 3D convolution kernel, which brings an additional burden for calculation.

Two-stream CNN structure [20] uses an additional CNN stream to learn the temporal feature, which takes the optical flow calculated from the continuous video frame sequence as input. Using optical flow to capture the motion features, the two-stream CNN structure is not very effective in modeling the motion feature in long range, but it can more intuitively understand the semantic information at the temporal level. Optical flow information calculates the pixel vector movement between adjacent frames, which can intuitively display the motion information between adjacent frames, but its calculation is cumbersome and requires additional storage space, which is not conducive for end-to-end training.

Video human action recognition is not only dependent on one aspect of appearance or motion features, but should be recognized through both, we hope that the network can judge the occurrence of actions through both appearance and temporal relationship like humans. In addition, the existing methods often ignore the problem of feature imbalance in the time dimension. The weights of semantic information in different time points are different, so the frame feature at different moments should be treated differently. The convolutional neural network is superior in extracting image features, but there are still have challenges in inferring temporal relations.

Based on the above considerations, in this work, we propose a multi-scale attention spatial–temporal features network based on improved SSD method (MAST). We process video data in a manner similar to TSN [21], sample the video at equal intervals, and extract features from the sampled frames. Unlike TSN, we just use RGB images as input, this greatly saves computing resources for video action recognition. We observe the SSD method [22] in the field of object detection, which extracts features of different scales to detect object, making good use of the appearance information in the image. In MAST, we perform self-attention and inter-frame relationship calculations on frame features on multiple feature scales to obtain appearance features and motion features at the same time. With the purpose of distinguishing the importance of different frames, we further added a sampling attention module, and calculated an additional sampling loss based on the output of this module. This allows the network to focus on more discriminative frames and further improve the recognition accuracy.

In general, our works are summarized as follows:

1. We improved the SSD model framework, added attention modules and temporal feature modeling capabilities based on multi-scale features, so that the model can perform video action recognition without using optical flow, which improves recognition efficiency.
2. Our network prefers to focus on the discriminative feature of a video in the time dimension. By reasoning the temporal relation between the inter-frames and pay more attention to the discriminative frames, our network can effectively improve action classification accuracy. The entire network structure showed in Fig. 1.

2 Related work

Research on video action recognition has made great progress in recent years. Before the development of deep learning, most methods used for video action recognition are

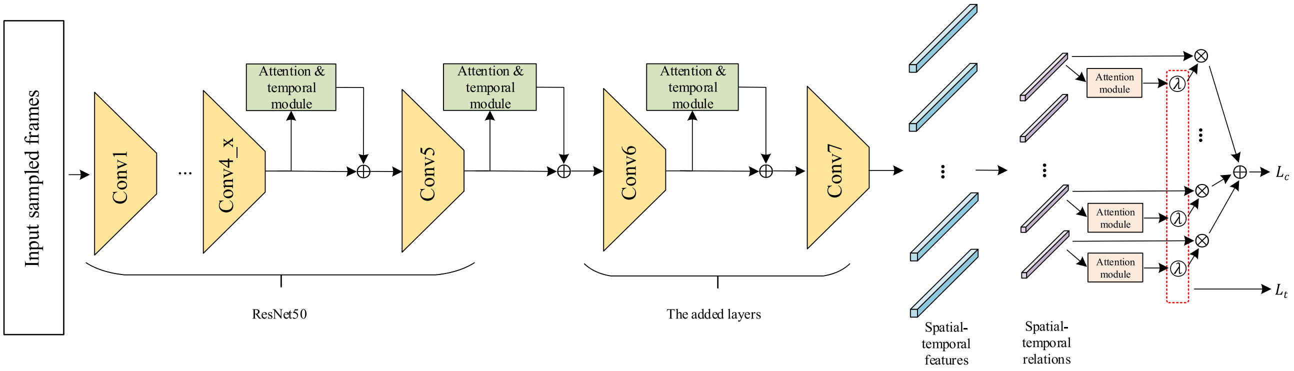

Fig. 1 The overall framework of MAST. The network using ResNet50 as its backbone, and using three additional convolutional layers and attention and temporal modules to replace the layer after conv5_x. The more details about attention and temporal module can be found in Fig. 2

traditional feature extraction methods in image recognition, such as HOG [23], HOF [24], etc., usually after the handcrafted feature extraction following a classifier for the classification of features. There also exist some methods have quit excellent performance such as the IDT [8], regarding the densely trajectory as a good video representation, but it's difficult to gain competitiveness in terms of efficiency. With the popularity of deep learning, the image features obtained by convolution neural network shows excellent ability, so more and more researchers hope that using convolution neural network to extract features of a video to cope with the action recognition task. Different from the image classification, the network building in the video action recognition task needs to have the ability to capture the temporal features [13]. using 3D convolution kernel construct 3D CNN network makes originally applied to RGB image convolution operation adding a dimension. Tran et al. [25] constructs a general and simple 3D CNN model dealing with large-scale data, and achieved remarkable performance. The methods based on 3D CNN also cause a problem: using 3D convolution kernel for capture the temporal features lead to an explosion in computation. Therefore, there is also spawned some variant of convolution kernels, such as $R(2+1)D$ [26], P3D [27], which the convolution can be divided into processing space dimension and time dimension separately, effectively reducing the amount of calculation. Li et al. [28] has chosen to making 3D convolution process on video frame sequence from three different perspectives, sharing the convolution kernels in the view of the three parameters, and thus reduced the number of parameters.

Another popular capturing temporal features method is the two-stream network put forward by Simonyan [20]. Two-stream network consists of two convolution neural network, its space steam process on the RGB images to extract appearance feature, while the temporal stream processes the optical flow representing motion features to capture the temporal feature of video. The two streams extract the features of video separately and classify it, then fuse the classification score of two streams to obtain the final classification results, it also illustrates the effectiveness of optical flow in video action recognition. TSN [21] sampling the whole video get several video segments, each segment input to two-stream network to extract the features of the two streams, then fuse the classification result of each segment. Peng et al. [29] observed a situation in two-stream network that the recognition result of one stream success and the other stream failure, in this case the final recognition result is always inaccurate, so the two-stream collaboration methods add collaborative learning and attention module makes the features of the two streams interaction to achieve a more accurate classification result. Carreira et al. [30] making the convolution kernels of two-stream network structure inflated to 3D convolution kernels, achieved the optimal results in large datasets kinetics. Sun et al. [31] also proves that the optical flow cue is helpful to the video action recognition. Due to the optical flow needs to be extracted in advance and needs to be stored in an extra space, some methods considering regard the extraction of optical flow as a part of the whole network. Different from the traditional optical flow extraction method, [32] consider using CNN to predict optical flow. Zhu et al. [33] directly joint the optical flow to the front of the two-stream construct, achieving end-to-end training. Recently, to find the better representative video time information clues, Piergiovanni et al. [34] recently proposed a new temporal feature which can replace the optical flow, also has obtained the good effect.

For the mentioned methods above, however, most of them lack of the ability of modeling long-term feature sequence, they often face to the datasets which can be classified by the appearance feature or the short-term temporal feature, it almost would not treat the feature in different time dimension differently. The video action recognition network should have the ability that paying attention to the discriminative frame which including more information of temporal. In video action recognition task, compared to the methods specialize in dealing with long-term sequence data such as recurrent neural network [35] and long–short-term memory network [36], the attention mechanism [37] is more effective and convenient to extract the relative important features of sparse sample, capture the relation between features. The purpose of [38] is finding discriminative frames in a frames sequence of video frames. Liu et al. [39] select the discriminative frames based on the improved frame selection mode, its use is a blend of supervised pyramids of light flow movement features to look for interested.

To solve the VQA task, a relational reasoning network [40] is proposed, it also define the state relation of object. [41] reason the relationship between frames of multiply temporal scales based on the relation network, which has a strong temporal reasoning ability, but its performance in inferring appearance features is very ordinary. SSD method [22] in object detection field extract features of multi-scales to detect objects in the image, for considering the diversity of time cues in features of different scales, we choose to improve the method based on SSD. We consider that different motion cues can be captured in features of different scales, so we use the self-attention module and the inter-frame relationship inference module to perform temporal modeling.

3 Method

Now most method of modeling temporal clues are taken as well as the strategy of modeling the appearance feature: take the temporal clue as the input of a network or using

3D convolution kernels to extract the temporal features. These methods require additional computing resources while not paying attention to the temporal relationship of features at different scales and the differences in features at different time points. In this section we introduce the multi-scale attention spatial–temporal features network, in Sect. 3.1 we will introduce how to sample video frames to get a long-term video representation. In Sect. 3.2, we will detail the method of performing self-attention and temporal modeling on multiply feature scales. Finally, the method of temporal reasoning on the final spatial–temporal features and the method of assigning different weights to features at different moments are introduced in Sect. 3.3 (Table 1).

3.1 Video-level feature extraction

For a video include an action with a duration of a certain time, if a small continuous frame is simply selected from it and sent to the convolutional neural network for feature extraction, then the information obtained is only the information of the nearby domain at a certain instant in an action. But if choose a long range of continuous stack frames for feature extraction, due to the changes of action will not happen according to the frame is bigger, the result of feature extraction will contain a lot of redundant information, which will lead to additional calculate cost. We hope to get a feature representation that contains the action information from it start to finish, meanwhile, it will not mingle too much redundant information. So for a video V, we will be equally divided into segments, $V = \left[V_0, V_1 \ \dots, \ V_{N-1}\right]$,we using the N video snippets to get N images to construct an ordered frames sequence and using $\left(F_0, F_1, \ \dots, \ F_{N-1}\right)$, F_n is a video frame extracted from the corresponding video segment V_n. Then sending the frames sequence into the backbone convolutional neural network, we can obtain the feature sequence $\left(T_0, T_1, \ \dots, \ T_{N-1}\right)$ representing the whole video. Here we consider the more stable feature extraction process, we choose to use a backbone network similar to SSD as our feature extraction network, but different from the VGG16 [42] used in the SSD, we choose ResNet50 and add additional layers to replace the original avg pooling layer and fc layer to consistent with the model in SSD.

Table 1 Notations of variables

T_n	A frame in a sequence of frames
Y_i	Features calculated using self-attention
T_n'	Temporal feature calculated from two adjacent frames
R_m	The relationship between the inferred spatial–temporal features
λ_i	The weight of a temporal relationship in the time dimension

3.2 Self-attention and temporal modeling on multi-scales

When human observe the occurrence of an action, they often observe two aspects: what is happening and at which temporal point the action occurs. We hope that the video action recognition network also has the ability to observe these two aspects. Based on the SSD method, we add a self-attention module and a temporal feature module between the layers of the backbone network. Since the receptive field sizes of the feature maps obtained from different layers are different, we consider that the temporal features of the feature maps on different scales are different. We want to capture the motion information of human in the video at different scales, so we establish temporal features on multi-scales. This is different from the optical flow that only captures the motion cues of the original image and then performs the convolution operation. The specific self-attention module and temporal feature module are shown in Fig. 2. Inspired by the non-local neural network, we use the non-local block as our feature self-attention module. In the self-attention module, T is the feature map sequences obtained by a certain convolutional layer, the self-attention module receives T as the input to perform self-attention operation: $SA\left(X_i, X_j\right)$, where X_i is the pixel value of a certain point of the input feature, X_j is the pixel value of all the possible positions of features. There are many fusion forms in the non-local block, we set the Embedded Gaussians model in our self-attention module. We define the output of the self-attention module is:

$$Y_i = \frac{1}{C(T)} \sum_{\forall j} \mathrm{SA}\left(X_i, X_j\right) g\left(X_j\right), \tag{1}$$

$C(T)$ means the normalized factor. Using the Embedded Gaussians model to describe self-attention fusion is:

$$\mathrm{SA}\left(X_i, X_j\right) = e^{\theta(X_i)^T \phi(X_j)}, \tag{2}$$

we can express it followed the form in [44] as

$$T_{\mathrm{att}} = \mathrm{softmax}\left(T^T W_\theta^T W_\phi F\right) g(T). \tag{3}$$

In the temporal feature module, we only use 2D convolution kernels to capture the temporal representation between frames. For the feature sequence T with size $N \times C \times H \times W$, we first reduce the number of channels to 1/16 of the original, then the feature T_{n+1} of the latter frame is subjected to a 2D convolution operation and then subtracted from the feature F_n of the previous frame. The module captures the motion cues between the two frames by calculating the

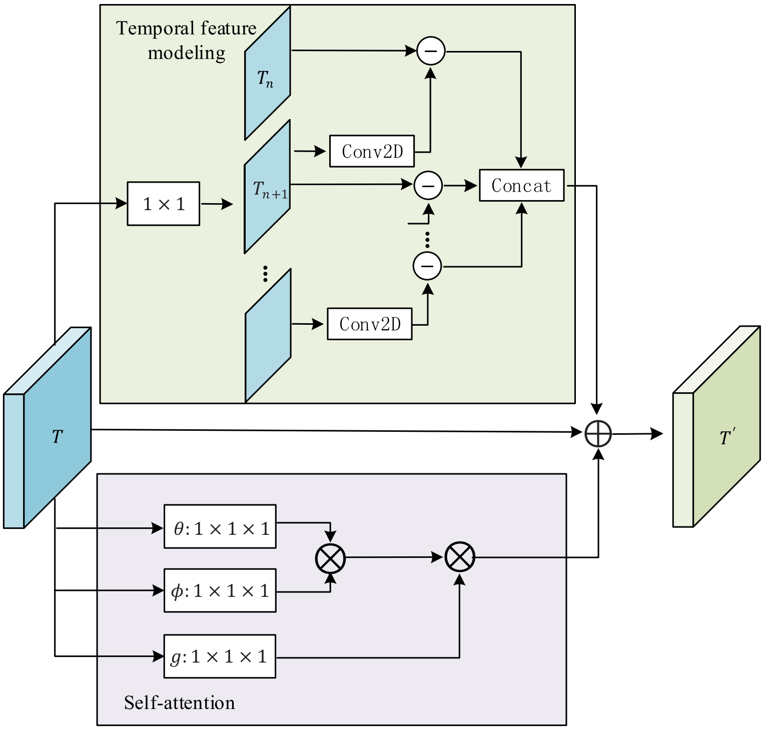

Fig. 2 The attention and temporal module consists of two parts: self-attention module focuses on the salient areas in the feature sequences (including both spatial and temporal dimension) and the temporal feature modeling module captures the motion cues between frames

pixel offset between the two frames, the formula is shown as follow:

$$T_n^{'} = \mathrm{Conv}\left(T_{n+1}\right) - T_n. \tag{4}$$

By calculating the motion information between N frames, the network obtains $N - 1$ temporal feature sequences. To be consistent with the original input size, we add a feature with all zeros at the end of the feature sequence to represent the motion information of the last time point. Finally, we restore the number of channels to ensure consistency with the input. After fusing the features obtained from the self-attention module with the features obtained from the temporal feature module, we then make a skip-layer connection with the original input features to obtain the spatial–temporal feature representation with attention and sent to the subsequent network:

$$T = T + T^{'} + T_{att}. \tag{5}$$

3.3 Temporal relation with attention weights

For a video frame sequence, more attention should be paid to the logical relationship between before and after of the action. Although some actions which contains obvious visual features (such as basketball-shooting and archery) can be recognized by its important visual features (such as the basketball for basketball-shooting, the bow and arrow for archery), but in real life a lot more action need to understand the temporal relations of its occurrence, such as judging whether to insert the USB to the computer or pull it from computer, we cannot determine it by only USB image feature. We reason the inter-frame relation which contains the long-term video spatial–temporal information. We consider to reason the inter-frame of multi subset temporal scales. First, the temporal relation between two frames which contain long-range spatial–temporal information can defined as

$$R_2 = f_\phi\left(\sum_{m<n} g_\theta\left(T_m, T_n\right)\right), \tag{6}$$

for each T_m represent the frame feature fusing multi-scale spatial–temporal information we extract from the video.f_ϕ and g_θ is the reasoning method for two frames, here we adopt the MLP (Multilayer Perceptron) as our reasoning method. Spontaneously, we can extend the temporal relation between two frames to three frames even multi frames:

$$R_3 = f_\phi\left(\sum_{m<n<l} g_\theta\left(T_m, T_n, T_l\right)\right), \tag{7}$$

For any scale $m < N$, N frame features contain several relations of m frames. For ease of calculation, we only randomly selected k relations which the temporal scale is m from N features. Considering the number of frame is different in different temporal-scale temporal relation, and the sample interval we sampled is different, we hope to pay attention to the discriminative temporal point in an action, so we give every temporal relation a corresponding weight of attention λ_i, then we can obtain the attention temporal relation by multiplying λ_i with corresponding different temporal scales, add the attention temporal relation with different temporal scale and then we can using a SoftMax function to get the action classification result:

$$P = \mathrm{softmax}\left(\sum_{i=2}^{N-1} \lambda_i R_i\right), \tag{8}$$

we use two fully connected layer and a ReLU layer to get the module generating the attention weights λ_i, and the using a sigmoid function to restrict it to [0,1]:

$$\lambda_i = \mathrm{sigmoid}\left(\mathrm{Att\_Moudle}\left(R_i\right)\right). \tag{9}$$

These weights correspond to temporal relations at different scales, focusing on more discriminative inter-frame temporal relations. In our work the final loss L_f consists of two parts: classification loss L_c and temporal scale attention loss L_t, the calculation is as follows:

$$L_f = L_c + \alpha L_t. \tag{10}$$

α is the coefficient used to weigh the proportion between the two losses. Among them, we use the finally obtained prediction P to calculate the classification loss L_c between the predicted value and the ground-truth Y, and we use the cross-entropy loss function as the loss measurement method. The formula is shown as follows:

$$L_c = \mathrm{CE}(P, Y), \tag{11}$$

where CE is the cross-entropy loss function. For the temporal scale attention loss L_t, we obtain it by calculating the $L1$ loss of the vector $\overline{\lambda}$ formed by the attention weight λ_i of different temporal scales, which is shown as:

$$L_\mathrm{t} = L1\left(\overline{\lambda}\right). \tag{12}$$

Through L_t, we can effectively find more noteworthy inter-frame relationships, that is, key frames that are more representative of the entire video.

4 Experiment

4.1 Datasets

We use widely used datasets UCF101 [45] and HMDB51 [46] to validate our MAST method. UCF101 is a widely used action recognition dataset which consists of 13,320 videos and contains 101 different categories of human action recognition, each category of action contains videos of 25 different people. The content of this dataset is very diverse, and can be divided into five categories: human–object interaction, human action, human–human interaction, musical instrument performance, and sports. HMDB51 contains 51 types of actions, a total of 6849 videos, with a resolution of 320*240, mainly collected on YouTube, google and other websites, and can be divided into categories such as facial actions, body actions, human–object interactions and human actions.

4.2 Training strategy and implementation details

We train MAST on 2 Nvidia Titan X GPUs (12 GB memory), implemented with pytorch 1.0. GPU perform massively parallel operations, which can accelerate network computations. The size of all input images is 224*224. We use stochastic gradient descent algorithm to learn the network parameters, where the batch size is set to 64 and the momentum is set to 0.9. We initialize the learning rate as 0.001 and decreases to its 1/10 after 40 and 80 epochs. The total number of epochs is 100. In our experiment, the number of uniformly sampled frames N is set to 8. The coefficient α used to balance the two different losses is set to 0.0001. The backbone Resnet50 model is pre-trained on ImageNet, we have replaced the layers after conv_4x, see Table 2 for specific network parameters.

4.3 Evaluation

We obtained the recognition results of our model on the validation set of the above two datasets, and compared them with the competitive methods. The results were shown in Table 2. Some early methods using manual features have obvious disadvantages. Compared with the double-stream method, our method has a better improvement. It can be seen that there is no need to rely on additional temporal information, and the temporal features can also be effectively captured by calculating the inter-frame motion of RGB frames. The two-stream method has obvious disadvantages in terms of speed and computational. The TSN method uses

Table 2 The backbone structure parameter using ResNet50, s means stride of convolution operation

Layers	Output size (spatial dimension)	Convolution blocks
Conv1	112*112	7*7, 64, s = 2
Conv2_x	56*56	1*1, 64; 3*3, 64 × 3; 1*1, 256
Conv3_x	28*28	1*1, 128; 3*3, 128 × 4; 1*1, 512
Conv4_x	14*14	1*1, 256; *3, 256 × 6; 1*1, 1024
Conv5	7*7	1*1, 512; 3*3, 256 × 3; 1*1, 2048
Conv6	3*3	3*3, 256
Conv7	1*1	3*3, 256

Table 3 Results on UCF101 and HMDB51 datasets

Methods	UCF101	HMDB51
MVSV [47]	0.835	0.559
Mv + FV [48]	0.785	0.467
EMV [49]	0.802	–
C3D [25]	0.823	0.568
STC(ResNet101) [50]	0.901	0.626
P3D [27]	0.886	–
Two-stream [20]	0.880	0.594
TSN(ResNet50) [21]	0.862	0.547
TSN(RGB)(BN-Inception) [21]	0.911	–
STM [51]	0.962	0.722
Ours	0.932	0.66

All the results are the accuracy of classification

Table 4 The accuracy on different backbone structures (test on UCF101)

Backbone	Accuracy
VGG16	0.905
BN-Inception	0.919
ResNet50	0.932

Table 5 The performances of adding module in different layers (test on UCF101)

Position	Accuracy
After conv4_x	0.884
After conv4_x and conv5	0.902
After conv5 and conv6	0.912
After conv4_x, conv5 and conv6	0.932

Table 6 Comparison between different module strategies (test on UCF101)

Method	Accuracy
Self-attention only	0.874
Temporal modeling only	0.892
MAST (using two sub-modules)	0.932

a variety of data input forms, comparing to its RGB input, the accuracy of our method is improved by 2.1% on UCF101 it mainly due to TSN only performs a simple weighted average on the extracted features at different times, lacking in-depth temporal modeling. Compared with the method using 3D convolution kernel, our method is 10% higher than C3D on UCF101 and 9% on HMDB51 respectively, it can be seen that spatial–temporal features can also be extracted without using a 3D convolution kernel that consumes a lot of calculation. The method of modeling temporal features used in STM is similar to ours. Our performance is slightly worse than that of STM. We consider the replacement of the entire residual module in STM improves its accuracy, but our method pays more attention to salient spatial–temporal areas in different scales and effectively verify the structure of the SSD. In addition, paying attention to the feature relationship between channels is what we need to further study in the future. We did not compare the operating efficiency with the two-stream method or the method using 3D convolution kernel, because the speed of these methods is incomparable to using 2D convolution.

4.4 Ablation experiments

4.4.1 Structures

We use ablation experiments to prove the effectiveness of each part of our method. We first considered the impact of using different network structures to extract features on our method in video behavior recognition. ResNet itself is a network with a residual structure [43], and the multi-scale spatiotemporal attention module we introduced in the additional layer can also be regarded as a residual structure. We compared the experimental results of using VGG16 and BN-Inception as the backbone. The results were shown in Table 3. It can be seen that using ResNet50 as the backbone has the best performance. We consider that because Resnet can solve the problems of network degradation and gradient explosion, making the neural network deeper and have better classification effect (Table 4).

4.4.2 Influence of the added modules

We analyzed the role of our modules added in different convolutional layers. The receptive field size of the feature map obtained by different layers is different, so we tried several different module addition schemes: only add a module after conv4_x, add module after conv5 and conv6, add after conv4_x and conv5, add after conv4 _x, conv5 and conv6. The different results are shown in Table 5. Since the high-level feature map contains more semantic information, we can see that adding our module after the higher-level convolutional layer is better than adding it in the lower level. In terms of quantity, adding a module after all additional layers can better integrate spatial–temporal information.

In addition, we explored the impact of two different sub-modules on our network. Compared with the temporal feature modeling module, the self-attention module explores the features that are more worthy of highlighting in the spatial and temporal dimensions, while the latter only extracts temporal information. We found through experiments that only a single module is not as effective as a combination using the both, it can be seen that the information in the spatial domain and the information in the time domain are mutually helpful (Table 6).

5 Conclusion

This paper proposes a method to introduce the multi-scale attention spatial–temporal features into the advanced framework SSD, effectively enhancing the representation capability of the features used for video action recognition, and improving the recognition performance. The results prove that the effectiveness of our proposed method. It is worth mentioning that we only use RGB frames as the input, and we can still capture effective temporal representations in features of multi different spatial scales. In future work, we will study a more robust network structure to allow deeper backbone to be embedded in it. In addition, we will explore the impact of high-resolution images on video action recognition and explore different forms of attention mechanisms.

Acknowledgements This work was supported by the Scientific Research Fund of Hunan Provincial Education Department of China ProjectNo.17A007; and the Degree and Post-graduate Education Reform Project of Hunan Province of China (Project No. 2020JGZD043).

References

1. Fusier, F., Valentin, V., Bremond, F.: Video understanding for complex activity recognition[J]. Mach. Vis. Appl. **18**(3–4), 167–188 (2007)
2. Qin, J., Li, H., Xiang, X., Tan, Y., Pan, W., Ma, W., Xiong, N.N.: An encrypted image retrieval method based on Harris corner optimization and LSH in cloud computing. IEEE Access **7**(1), 24626–24633 (2019)
3. Gu, K., Jia, W., Wang, G., et al.: Efficient and secure attribute-based signature for monotone predicates. Acta Inform. **54**, 521–541 (2017)
4. Wang J, Gao Y, Yin X, Li F, Kim H (2018) An enhanced PEGASIS algorithm with mobile sink support for wireless sensor networks. Wirel. Commun. Mob. Comput. (2018). https://doi.org/10.1155/2018/9472075
5. Gorelick, L., Blank, M., Shechtman, E., Irani, M., Basri, R.: Actions as space–time shapes. TPAMI **29**(12), 2247–2253 (2007)
6. Jia, K., Yeung, D.-Y.: Human action recognition using local spatio-temporal discriminant embedding. In CVPR, p. 1 (2008)
7. Klaeser, A., Marszalek, M., Schmid, C.: A spatio-temporal descriptor based on 3d-gradients. In: BMVC, p. 1 (2008)
8. Wang, H., Schmid, C.: Action recognition with improved trajectories. ICCV **1**(5), 8 (2013)
9. Laptev, I.: On space-–time interest points. IJCV **64**(2–3), 5 (2005)
10. Xia, Z., Hu, Z., Luo, J.: UPTP vehicle trajectory prediction based on user preference under complexity environment. Wirel. Pers. Commun. **97**, 4651–4665 (2017). https://doi.org/10.1007/s11277-017-4743-9
11. He, S., Li, Z., Tang, Y., Liao, Z., Li, F., Lim, S-J.: Parameters compressing in deep learning. CMC **62**(1), 321–336 (2020)
12. Tang, Q., Xie, M.Z., Yang, K., Yuansheng, L. Dongdai, Z. Yun, S.: A decision function based smart charging and discharging strategy for electric vehicle in smart grid. Mob. Netw. Appl. **24**, 1722–1731 (2019)
13. Ji, X., Xu, W., Yang, M., Yu, K.: 3d convolutional neural networks for human action recognition. IEEE Trans. Pattern Anal. Mach. Intell. **35**(1), 221–231 (2013)
14. Krizhevsky, A., Sutskever, I., Hinton, G.E.: Imagenet classification with deep convolutional neural networks. In: NPIS, pp. 1097–1105 (2012)
15. Karpathy, A., Toderici, G., Shetty, S., Leung, T.; Sukthankar, R.: Largescale video classification with convolutional neural networks. In: IEEE Conference on Computer Vision and Pattern Recognition, pp. 1725–1732 (2014)
16. Long, M., Peng, F., Li, H.: Separable reversible data hiding and encryption for HEVC video. J. Real Time Image Proc. **14**, 171–182 (2018)
17. Deng, J., Dong, W., Socher, R., Li, L.-J., Li, K., FeiFei, L.: ImageNet: a large-scale hierarchical image database. In: CVPR, pp. 248–255 (2009)
18. Zhang, J., Jin, X., Sun, J., Wang, J., Arun, K.S.: Spatial and semantic convolutional features for robust visual object tracking. In: Multimedia Tools and Applications, pp. 15095–15115 (2020)
19. Gui, Y., Zeng, G.: Joint learning of visual and spatial features for edit propagation from a single image. In: The Visual Computer, pp. 36:469–482 (2019)
20. Simonyan, K.; Zisserman, A.: Two-stream convolutional networks for action recognition in videos. In: Advances in Neural Information Processing Systems, pp. 568–576 (2014)
21. Wang, L.M., Xiong, Y.J., Wang, Z., Qiao, Y., Lin, D.H.O., et al.: Temporal segment networks: towards good practices for deep action recognition. In: European Conference on Computer Vision, pp. 20–36 (2016)
22. Liu, W., Anguelov, D., Erhan, D., Christian, S., Scott R., Cheng-Yang F., Alexander C.: SSD: Single Shot MultiBox Detector. In: European Conference on Computer Vision, pp. 21–37 (2016)
23. Dalal, N.F., Triggs, B.S.: Histograms of oriented gradients for human detection. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition 2005. USA: IEEE, pp. 886–893 (2005)
24. Laptev, I., Marszalek, M., Schmid, C., Rozenfeld, B.: Learning realistic human actions from movies. In: IEEE Conference on Computer Vision and Pattern Recognition, Anchorage, AK, pp. 1–8 (2008)
25. Tran, D., Bourdev, L., Fergus, R., Torresani, L., Palur, M.: Learning spatiotemporal features with 3D convolutional networks. In: 2015 IEEE International Conference on Computer Vision (ICCV), Santiago, pp. 4489–4497 (2015)
26. Tran, D., Wang, H., Torresani, L., Ray, J., LeCun, Y., Paluri, M.: A closer look at spatiotemporal convolutions for action recognition. In: IEEE Conference on Computer Vision and Pattern Recognition, pp. 6450–6459 (2018)
27. Qiu, Z., Yao, T., Mei, T.: Learning spatio-temporal representation with pseudo-3D residual networks (2017). arXiv:1711.10305
28. Li, C., Zhong, Q., Xie, D, et al.: Collaborative spatio-temporal feature learning for video action recognition. In: IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR). IEEE (2019)
29. Peng, Y.X., Zhao, Y.Z., Zhang, J.C.: Two-stream collaborative learning with spatial-temporal attention for video classification. In: IEEE Transactions on Circuits and Systems for Video Technology, vol. 29, no. 3, pp. 773–786 (2018)
30. Carreira, J., Zisserman, A.: Quo vadis, action recognition?a new model and the kinetics dataset. CVPR **2**(4), 5 (2017)
31. Sun, S., Kuang, Z, Ouyang, W., Sheng, L., Zhang, W: Optical flow guided feature: a fast and robust motion representation for video action recognition (2017). arXiv:1711.11152

32. Fischer, P., Dosovitskiy, A., Ilg, E., Husser, P., Hazrba, C., Golkov, V., van der Smagt, P., Cremers, D., Brox, T.: FlowNet: learning optical flow with convolutional networks. In: International Conference on Computer Vision (ICCV) (2015)
33. Zhu, Y., Lan, Z.Z., Newsam, S., XHauptmann, S: Hidden two-stream convolutional networks for action recognition. In: Asian Conference on Computer Vision, pp. 363–378 (2018)
34. Piergiovanni, A., Ryoo, M.S: Representation flow for action recognition (2018). arXiv:1810.01455
35. Mnih, V.F., Heess, N.S.: Recurrent models of visual attention. In: Advances in Neural Information Processing Systems, NIPS (2014)
36. Qu, Z.W., Cao, B.Y., Wang, X.R., Li, F., Xu, P.R., et al.: Feedback lstm network based on attention for image description generator. Comput. Mater. Contin. **59**(2), 575–589 (2019)
37. Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A.N., Kaiser, L., Polosukhin, I.: Attention is all you need. In: Neural Information Processing Systems (NIPS) (2017)
38. Zhao, Z., Elgammal, A.M.: Information theoretic key frame selection for action recognition. In: British Machine Vision Conference (BMVC), pp. 1–10 (2008)
39. Liu, L., Shao, L., Rockett, P.: Boosted key-frame selection and correlated pyramidal motion-feature representation for human action recognition. Pattern Recogn. **46**(7), 1810–1818 (2013)
40. Santoro, A., Raposo, D., Barrett, D.G., Malinowski, M., Pascanu, R., Battaglia,P., Lillicrap, T.: A simple neural network module for relational reasoning (2017). arXiv:1706.01427
41. Zhou, B., Andonian, A., Oliva, A., Torralba, A.: Temporal relational reasoning in videos (2017). arXiv:1711.08496
42. Simonyan, K., Zisserman, A.: Very deep convolutional networks for large-scale image recognition[J] (2014). arXiv:1409.1556
43. He, K., Zhang, X., Ren, S., et al.: Deep residual learning for image recognition[J] (2015). arXiv:1512.03385
44. Wang, X., Girshick, R., Gupta, A., He, K.: Non-local neural networks. In: CVPR (2018)
45. Soomro, K., Zamir, A.R., Shah, M.: UCF101: a dataset of 101 human actions classes from videos in the wild[J] (2012). arXiv:1212.0402
46. Kuehne, H., Jhuang, H., Garrote, E., Poggio, T., Serre, T.: Hmdb: a large video database for human motion recognition. International Conference on Computer Vision, pp. 2556–2563 (2011)
47. Cai, Z.W., Wang, L.M., Peng, X.J.: Qiao, Y.: Multi-view super vector for action recognition. IEEE Conference on Computer Vision and Pattern Recognition, pp. 596–603 (2014)
48. Kantorov, V., Laptev, I.: Efficient feature extraction, encoding and classification for action recognition. IEEE Conference on Computer Vision and Pattern Recognition, pp. 2593–2600 (2014)
49. Zhang, B.W., Wang, L.M, Wang, Z., Qiao, Y., Wang, H.L.: Real-time action recognition with enhanced motion vector CNNs. IEEE Conference on Computer Vision and Pattern Recognition, pp. 2718–2726 (2016)
50. Diba, A. et al.: Spatio-temporal channel correlation networks for action classification. In: Computer Vision—ECCV 2018, vol. 11208, pp. 299–315 (2018)
51. Jiang, B., Wang, M., Gan, W., Wu, W.: STM: spatio-temporal and motion encoding for action recognition. In: ICCV (2019)

Publisher's Note Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.