A Spatial-temporal Attention Module for 3D Convolution Network in Action Recognition

Shengwei Zhou, Liang Bai, Haoran Wang, Zhihong Deng, Xiaoming Zhu and Cheng Gong

ABSTRACT

Action recognition is a significant but challenging task in the field of computer vision. 3D convolutional neural network is one of the mainstream methods for action recognition because it can process three-dimensional information effectively. However, at present, the performance of 3D convolutional neural networks is not particularly prominent. The main reason is that the information of the video is mainly contained in the key areas of key frames in the video, yet the 3D convolutional neural network usually cannot extract the most critical information in the video effectively. Therefore, we propose a temporal attention and a spatial attention respectively, and combine them into a module called STAM to let models focus more on the key information. We introduced the STAM module into 3D ResNet, and conducted experiments on the UCF101 and HMDB51 datasets. The results demonstrate that our proposed attention module can improve the performance of 3D convolutional neural networks effectively.

KEYWORDS

Action recognition, 3D convolution, Attention, Neural Network.

INTRODUCTION

Action recognition has received extensive attention due to the application in many scenes such as human-computer interaction and abnormal video recognition. However, unlike image processing, action recognition remains a challenging problem because it has both temporal and spatial information. Thus it demands to process spatial-temporal information at the same time. Research has shown that 3D convolution can be used for action recognition because it can extract temporal and spatial information simultaneously[7-12]. However, due to the excessive amount of parameters, the development of 3D convolutional neural network is slow in the past few years. But recently, the evolvement of computing performance and the emergence of large action recognition datasets like Kinetics has greatly promoted the research of 3D convolutional neural networks.

Though considerable work has been done in the field of 3D convolutional neural networks for action recognition, it remains a challenging issue. The main point is that for a video, the information is mainly contained in the key-area of its...
key-frames. But in reality, 3D convolutional neural networks often lack the ability to extract such critical information. Historical research has shown that the attention mechanism is a method to extract key information while ignoring unimportant information. In the image field, many network models have introduced the attention mechanism and achieved good results[13-15]. However, the attention mechanism has hardly been introduced to the 3D convolutional neural network yet. Therefore, we propose to introduce the attention mechanism into the 3D convolutional neural network to better extract the key information of the video.

The main contribution of this article includes: (i) We propose a temporal attention that enables the network to focus more on important frames of the video; (ii) We propose a spatial attention that allows the network to focus more on important areas of the video ; (iii) We introduced the combination of these two attention mechanism (STAM) into 3D ResNet, and conducted experiments on two classic action recognition dataset named UCF101[16] and HMDB51[17]. The result indicates the STAM module can improve the performance of 3D convolutional neural networks obviously, showing the validity of the attention module. What’s more, our proposed STAM attention is a lightweight and embeddable module that can be equipped to any other 3D neural network.

RELATED WORK

METHOD FOR ACTION RECOGNITION

Since the convolutional neural network made a huge breakthrough in the ImageNet competition in 2012, a great number of neural network method have been proposed to action recognition. One of the mainstream methods is the two-stream network. K. Simonyan et al.[1] proposed the two-stream network framework for the first time, in which RGB streams are used to extract spatial information, while optical flows streams are used to extract temporal information, and finally the network integrate information from two streams to obtain results. L. Wang et al.[2] proposed a sparse sampling method and introduced it into two-stream network to construct a network model called temporal segment networks (TSN). The model first divides the video into three equal segments, and then randomly selects one short snippet from each of the three segments for prediction, and finally merging the results of the three snippets as the final result of the Entire video. Since then, a great deal of methods based on two-stream networks have emerged[3-5], mainly focusing on how to better extract the spatial-temporal information in the video. However, the two-stream method has a big drawback, which is that it needs to preprocess the video to extract the optical flow.

The other mainstream method of action recognition is 3D convolutional neural networks, S. Ji et al.[6] proposed using 3D convolutional neural networks for action recognition. D. Tran et al.[2] first proposed a standard 3D convolutional neural network architecture called C3D. And since then, a series of network frameworks based on 3D convolution have emerged[8-11]. In addition, the development of computers and the emergence of large datasets have also promoted the development of 3D convolutional neural networks. K. Hara et al.[12] has proved that current action recognition dataset has adequate data for training deep neural network. However, the current 3D convolutional neural network is usually not as effective as the two-stream network, indicating that the
3D convolution network cannot effectively mining the key spatial-temporal information in the video.

ATTENTION MECHANISMS

Attention play an significant role in human visual system, it allows human to focus on important information while ignoring irrelevant information. Recently, many people have tried to introduce the attention mechanism into the neural network to improve the performance of models. Jaderberg et al.[13] proposed an attention module which can extract key information in the image by performing spatial transformation to the image. Hu et al.[14] proposed a channel attention module, which can strengthen attention to key channels by giving different weights to different channels. Wang et al.[15] proposed an encoder-decoder attention module and combined it with ResNet. After end-to-end learning, the network not only performs well but also has good robustness against noise. However, although the attention mechanism has achieved great success in the image field, the attention mechanism has rarely been introduced into the 3D convolutional neural network, especially in the deep neural network. Therefore, we propose an attention module that includes both spatial attention and temporal attention, and embeds it into the 3D convolutional neural networks to improve their performance.

PROPOSED METHOD

Our goal is to capture the key information in a video in both temporal and spatial dimensions. In this section, we first proposed a temporal attention for extracting key temporal information and a spatial attention for extracting key spatial information. Then we combine these two attention together to generate a module called spatial-temporal attention module(STAM), for it can extract both key temporal information and key spatial information. The entire attention frame is shown in Figure 1. The input features pass through the temporal attention module and the spatial attention module respectively, and obtained the output features. In addition, we selected 3D-ResNet[12] as our baseline and build our model by introducing STAM into it.

![Figure 1. The frame of STAM attention module. (This module consists of a temporal attention and a spatial attention, and ensures that the dimensions of the input and output are consistent.)](image)

TEMPORAL ATTENTION

The temporal information of the video is contained in the channels of the feature map, and each channel can be regarded as a feature detector. So we build
a temporal attention mechanism by looking for the inter-channel relationship of the feature map. This attention mechanism is mainly focuses on ‘when’ is meaningful in a video. It can assign different weights to different channels in the feature maps, assigning higher weights to channels containing key information, and assigning lower weights to other channels.

Specifically, for a feature map $F \in \mathbb{R}^{C \times H \times W}$, we first compress it in the spatial dimension, and by performing average-pooling on each channel, we obtain a temporal feature descriptor $f^t \in \mathbb{R}^{C \times 1 \times 1}$, the l-th element of $f^t$ is computed as:

$$f^t_l = \frac{1}{H \times W} \sum_{m=1}^{H} \sum_{n=1}^{W} F_l(m,n)$$  \hspace{1cm} (1)$$

where $F_l(m,n)$ represents value corresponding to the position $(m, n)$ on the l-th channel of the feature map. After that we constructed a three-layer perceptron, mapping $f^t$ to a channel attention map $M_t \in \mathbb{R}^{C \times 1 \times 1}$. The temporal attention map is calculated as follows:

$$M_t(F) = \sigma(MLP(f^t)) = \sigma(W_2(W_1(f^t)))$$  \hspace{1cm} (2)$$

Where $\sigma$ refers to the sigmod function, $F$ denotes the input feature map, $W_1$ and $W_2$ represent the parameters of perceptron. By training on the dataset, the perceptron parameters are constantly updated, and the model learns an excellent channel attention map $M_t(F)$. Finally, by integrating the original feature map and the channel attention map using element-wise summation, different channels got different weights, thence the key temporal information in the video can be effectively extracted, and the final output feature map dimension is consistent with the input dimension.

**SPATIAL ATTENTION**

For each single frame of a video, the importance of different regions is also different, so we also propose a spatial attention method. Different from the temporal attention focusing on ‘when’ to strengthen, the spatial attention focusing on ‘where’ to strengthen. The spatial attention mechanism can assign different weights to different spatial regions, assigning higher weights to key regions, and assigning lower weights to other regions.

Given a feature map $F \in \mathbb{R}^{C \times H \times W}$, we first compress it in the temporal dimension by applying average-pooling to points at each location, getting a spatial feature descriptor $f^s \in \mathbb{R}^{1 \times H \times W}$. The $(m,n)$ element of $f^s$ is computed as:

$$f^s_{(m,n)} = \frac{1}{C} \sum_{l=1}^{C} F_{(m,n)}(l)$$  \hspace{1cm} (3)$$

where $F_{(m,n)}(l)$ represents the value corresponding to the l-th channel on the $(m, n)$ position. After that we convolved this feature map with a $7 \times 7$ convolution kernel and used the result as our spatial attention map $M_s \in \mathbb{R}^{1 \times H \times W}$. The calculation process is as follows:

$$M_s(F) = \sigma(f^{7\times7}(f^s))$$  \hspace{1cm} (4)$$

Where $\sigma$ refers to the sigmod function, $f^{7\times7}$ references a convolution kernel of size $7 \times 7$, $F$ denotes the input feature map. After training, the model learns the appropriate convolution parameters so that the spatial attention map can know
where to emphasize or suppress. Finally, the spatial attention map is also merged into the original feature map using element-wise summation. The spatial attention give higher weight to the key areas, and the output dimensions are still consistent with the input dimensions.

**PROPOSED ARCHITECTURE**

We connect the temporal attention and the spatial attention together as our attention module named as spatial-temporal attention module (STAM). We put the temporal attention mechanism in front of the spatial attention because we found that the effect of such combination is better than the reverse combination. The whole attention process is calculated as follows:

\[
F' = (F \odot M_t) \odot M_s
\]

Where \(F\) denotes the input feature map, \(M_t\) stands for the temporal attention map, \(M_s\) stands for the spatial attention map and \(\odot\) denotes element-wise multiplication.

**TABLE I. THE ARCHITECTURE OF OUR BASIC 3D RESNET-152.**

| Layer Name | Architecture | Output size |
|------------|--------------|-------------|
| conv1      | 7 x 7 x 7, stride 1,2,2 | 64 x 8 x 28 x 28 |
| conv2      | 3 x 3 x 3 max pool, stride 2 | 64 x 8 x 28 x 28 |
| conv3      | 3 x 3 x 3 | 128 x 4 x 14 x 14 |
| conv4      | 3 x 3 x 3 | 256 x 2 x 7 x 7 |
| conv5      | 3 x 3 x 3 | 256 x 1 x 4 x 4 |
| avg        | average pool | 256 x 1 x 1 x 1 |
| fc         | fully connected layer | 101-d / 51-d |

The STAM module can be flexibly applied to various network architectures because it is lightweight and ensures that the dimensions of the output are consistent with the input. Resnet\[17\] is one of the most successful network models in computer vision tasks due to its simplicity and effectiveness. Therefore, To demonstrate the effectiveness of this attention module, we chose 3D ResNet\[12\] as our baseline. The 3D ResNet is basically the same as 2D ResNet, except that its convolution and pooling kernel is three-dimensional. Specifically, we selected 3D ResNet-152 as our baseline, the architecture of the basic model is shown in Table 1. Thereafter, we embed STAM module into the benchmark to form our network structure. Specifically, we put the STAM module behind the last batch normalization of the main branch of the 3D ResNet. The merge details is showed in the Figure 2.
EXPERIMENT

DATASET

We experimented on two challenging action recognition datasets, UCF101[16] and HMDB51[17]. The UCF101 dataset includes 13320 video clips with a total of 101 human action categories, and the average length of each video is 7 seconds. The HMDB51 dataset contains 6776 video clip with a total of 51 human action categories, and each of the video spans about 3 seconds. For both dataset, the non-action frames in the video have been removed. We use standard training/test splits (70% training and 30% testing) and provided evaluation protocol.

RESULTS

We first compared the effect of each sub-attention modules and the merged STAM modules. So we added temporal attention, spatial attention and STAM to the basic model separately. For all of these models, we are training from scratch on UCF101 and HMDB51 and accuracies are averaged over three splits. All models have inputs of 3 channels × 16 frames × 112 pixels × 112 pixels. The experimental results are shown in TABLE II. It can be found that the accuracy of all models in this experiment is low, similar to the results of K. Hara et. al[12], proving that UCF101 and HMDB51 do not have enough data to train 3D convolutional neural networks and all models have been over-fitted. Nevertheless, it still can be seen that, compared to the baseline, our proposed spatial attention and temporal attention can effectively improve the performance, and the merged STAM is better than any single attention mechanism. What’s more, our proposed attention module is lightweight, and its parameter size is only slightly larger than basic model. So it can be simply equipped in other 3D convolutional neural networks.

| Model                | Parameters | UCF101 | HMDB51 |
|----------------------|------------|--------|--------|
| 3D ResNet152(baseline) | 118.22M    | 44.7   | 19.9   |
| 3D ResNet152 + temporal | 124.80M    | 45.7   | 21.1   |
| 3D ResNet152 + spacial | 118.26M    | 44.9   | 20.2   |
| 3D ResNet152+STAM    | 124.84M    | 45.8   | 21.4   |

To further verify the effectiveness of the STAM module, we also performed experiments on UCF101 and HMDB51 with pre-trained models on the Kinetics dataset. Kinetics[8] is a larger dataset than UCF101 and HMDB51, previous experiments have shown that it has enough data to train 3D convolutional neural networks, and the pre-trained models in Kinetics can be effectively migrated to
other video datasets. So we also selected the Kinetics pretrained 3D ResNet model for the fine-tuning experiments on UCF101 and HMDB51. Specifically, we selected 3D ResNet 152 model pre-trained by K. Hara et al[12] on Kinetics as our baseline, and added our STAM attention model for comparison. The initialization parameters of our model are the same as the baseline, except for the STAM module. We fine-tuned these two models on UCF101 and HMDB51, and the experimental results are shown in Table III. The results shows that STAM can still bring significant improvement in Kinetics pretrained models, confirming the validity of our proposed module on the 3D convolution network for action recognition.

**TABLE III. TOP-1 ACCURACIES ON UCF101 AND HMDB51 DATASET (FINE-TUNED WITH PRETRAINED MODEL).**

| Model              | UCF101 | HMDB51 |
|--------------------|--------|--------|
| 3D ResNet152       | 89.2   | 61.4   |
| 3D ResNet152+ STAM | 89.5   | 61.9   |

**CONCLUSION**

We proposed an attention module called STAM that can be equipped into 3D convolutional neural network for action recognition. This module consists of two sub-modules: temporal attention and special attention. The temporal attention mechanism enables the network to focus more on the key frames of the video, while the spatial attention mechanism makes the network to focus more on the key areas. And thus the entire module allows the network to pay more attention on the key information of the video. To certify the validity of our STAM module, we selected 3D-ResNet as our baseline and introduced the STAM module into it. We conducted experiments on UCF101 and HMDB51. The results show that the STAM module can greatly improve the performance of the basic model, whether it is training from scratch or fine-tuning with the pre-training model, showing the effectiveness of our STAM module. We believe that the attention mechanism is much crucial for video action recognition. In future work, we will introduce the STAM module into more other 3D convolutional neural networks to further verify its universality and effectiveness.

**REFERENCES**

1. K. Simonyan and A. Zisserman. Two-stream convolutional networks for action recognition in videos. In NIPS, 2014.
2. L. Wang, Y. Xiong, Z. Wang, Y. Qiao, D. Lin, X. Tang, and L. V. Gool. Temporal Segment Networks: Towards Good Practices for Deep Action Recognition. In ECCV, 2016.
3. Z. Lan, Y. Zhu and A. G. Hauptmann. Deep local video feature for action recognition. In CVPR, 2017.
4. R. Girdhar, D. Ramanan, A. Gupta, J. Sivic and B. Russell. Action VLAD: Learning spatio-temporal aggregation for action recognition. In CVPR, 2017.
5. G. A. Sigurdsson, S. Divvala, A. Farhadi and A. Gupta. Asynchronous temporal fields for action recognition. In CVPR, 2017.
6. S. Ji, W. Xu, M. Yang, and K. Yu. 3d convolutional neural networks for human action recognition. IEEE TPAMI, 35(1):221–231, 2013.
7. D. Tran, L. Bourdev, R. Fergus, L. Torresani and M. Paluri. Learning spatio-temporal features with 3D convolutional networks. In ICCV, 2015.
8. J. Carreira and A. Zisserman. Quo vadis, action recognition? A new model and the Kinetics dataset. In CVPR, 2017.
9. Z. Qiu, T. Yao and T. Mei. Learning spatio-temporal representation with pseudo-3D residual networks. In ICCV 2017.
10. D. Tran, H. Wang, L. Torresani, J. Ray, Y. LeCun and M. Paluri. A closer look at spatio-temporal convolutions for action recognition. In CVPR, 2018.
11. X. Wang, R. Girshick, A. Gupta and K. He. Non-local neural networks. In CVPR, 2018.
12. K. Hara, H. Kataoka and Y. Satoh. Can Spatiotemporal 3D CNNs Retrace the History of 2D CNNs and ImageNet? In CVPR, 2018.
13. Jaderberg, Max, Karen Simonyan, and Andrew Zisserman. “Spatial transformer networks.” Advances in neural information processing systems. 2015.
14. Hu, J., Shen, L., Sun, G.: Squeeze-and-excitation networks. arXiv preprint, arXiv:1709.01507, 2017.
15. Wang, F., Jiang, M., Qian, C., Yang, S., Li, C., Zhang, H., Wang, X., Tang, X.: Residual attention network for image classification. arXiv preprint, arXiv:1704.06904 (2017).
16. Soomro, K., Zamir, A.R., Shah, M.: UCF101: A dataset of 101 human actions classes from videos in the wild. CorR abs/1212.0402 (2012).
17. Kuehne, H., Jhuang, H., Garrote, E., Poggio, T.A., Serre, T.: HMDB: A large video database for human motion recognition. In: ICCV. (2011) 2556–2563.
18. K. He, X. Zhang, S. Ren, and J. Sun. Deep residual learning for image recognition. In CVPR, 2016.