Space-Time Separate Modeling for Efficient Video Classification

Pei Cao¹, Shuo Wang²*, Jinneng Wu³, Yanbin Hao²

¹ Wuhan Research Institute of Posts and Telecommunications, Wuhan, Hubei, 430205, China
² School of Information Science and Technology, University of Science and Technology of China, Hefei, Anhui, 230026, China
³ Department of Electrical Engineering and Electronics, Wuhan Institute of Technology, Wuhan, Hubei, 430205, China

*Corresponding author’s e-mail: shuowang.hfut@gmail.com

Abstract. Efficient video classification requires the deep neural network models to be much lightweight. Current deep convolutional networks generally adopt 3D convolutions or similar spatio-temporal computational counterparts to process the 3D signal of videos. However, due to the heavy computational load of those 3D units, they suffer from the problems of hard training and inefficient inferring. To address these problems, this paper proposes a novel yet efficient space-time separate (STS) modelling module. STS splits the feature channels into multiple groups and separately model various types of content information from the channel groups using different convolutional operations. Since the channel splitting mechanism can significantly reduce the model complexity, STS is much more lightweight than the existing video models. Particularly, the joint use of spatial/temporal/spatio-temporal convolutions achieves paralleled information modelling in a single block. We conduct experiments on two benchmarked video datasets to evaluate the performance of STS and demonstrate its effectiveness and efficiency on the video classification task.

1. Introduction

With the well-rounded of image related deep learning method, a growing number of researchers pay their attention to the tasks in more potential video field, e.g., video classification [1, 2, 3, 14, 15]. Compared to images, videos have a qualitative increase in the amount of information. For example, we cannot simply make distinction between “open a door” and “close a door” from some still-image sequences. Therefore, apart from the spatial learning, temporal modeling becomes increasingly important for videos which require temporal relationship. Furthermore, with the emergence of large-scale video datasets, efficiency is another concern for video models. Because training a deep video neural network is much more time-consuming than training an image net for the extra time-axis of videos. In this case, building lightweight models is equally important for video processing. Based on the above analysis, this paper investigates the feasibility and advantages of building efficient deep video neural networks.

Currently, there are two major types of video modeling mechanisms. The first pipeline is that adopting a CNN model to extract spatial features from video frames and then passing those spatial features into a recurrent neural network (e.g., LSTM [2]) to organize the temporal relations along the
timeline. This kind of works follow the structure of CNN+LSTM. Here, the two blocks (i.e., CNN and LSTM) can be either separately or jointly trained. However, as the spatial and temporal patterns are modeled with different modules in a separate learning manner, they under-perform due to the lack of joint modeling of space-time features. Another group of works focus on constructing spatio-temporal block for learning 3D features from videos, e.g., C3D [1], I3D [3], R(2+1)D [4], P3D [5], S3D [6], TSM [7], GSM [8], GST [9], and so on. Among these works, C3D and I3D directly replace 2D convolutions in a 2D CNN with 3D convolution versions, R(2+1)D, P3D and S3D split the 3D convolution into a cascade of 2D spatial convolution and 1D temporal convolution, while TSM and GSM use a temporal shift module to model temporal pattern between consecutive frames. In terms of model complexity, C3D and I3D are much heavier than others for the use of 3D convolutions. As the temporal shift module in TSM and GSM are parameter-free, they can have the same parameters with the standard 2D CNNs (e.g., ResNets [10]). For more efficient modeling, GST proposes to split the feature channels into spatial (2D) and spatio-temporal (3D) groups and then work in parallel. As a result, GST contains less parameters than the standard 2D CNN.

Inspired by channel splitting strategy used in GST, in this paper we propose a new channel splitting scheme and build a much lightweight video model for efficient video classification. Different to GST that decomposes the channel into two groups to separately learn spatial (2D) and spatio-temporal (3D) features, our proposed scheme learn three kinds of feature patterns, including spatial (2D), temporal (1D) and spatio-temporal (3D) features in parallel. We name the proposed method as space-time separate (STS) modeling module. Specifically, we split the feature channels into 3 non-overlapped groups, based
on which three kinds of space-time features are separately modeled. The advantages of our proposed STS model mainly lie on two aspects: (1) STS can jointly learn spatial/temporal/spatio-temporal feature in a block; and (2) the channel splitting mechanism significantly reduces the model parameters, resulting in much lightweight model. To evaluate the efficiency and effectiveness of our STS model, we conduct extensive experiments on two benchmark video classification datasets.

2. Approach

The proposed STS module is built upon a Residual block. In the following, we first elaborate the details of STS and then give analysis to the model complexity.

2.1 STS Module

The schema of the proposed STS module is shown in Figure 1-(e). Initially, let \( X \in \mathbb{R}^{C \times T \times H \times W} \) be the output feature map of the convolutional layer with a \( 1 \times 1 \times 1 \) kernel size, where \( C, T, H, W \) denote channel-size, time-length, space-height and space-width, respectively. Three hyperparameters \( \{ \beta_1, \beta_2, \beta_3 \} \) are used to control the split ratios of channels, resulting in three feature groups \( \{ X_1, X_2, X_3 \} \) where \( X_1 \in \mathbb{R}^{C_1 \times T \times H \times W} \), \( X_2 \in \mathbb{R}^{C_2 \times T \times H \times W} \), \( X_3 \in \mathbb{R}^{C_3 \times T \times H \times W} \) and \( C_1 + C_2 + C_3 = C \). It is different to GST (Figure 1-(d)) that divides the channel into two groups with fixed channel number of \( \frac{1}{2}C \), our STS tunes the channel ratios to be an optimal setting.

Next, we use three convolution operators to separately process the three feature groups \( \{ X_1, X_2, X_3 \} \). Specifically, we have

\[
Y_1 = ReLU(Conv1d(X_1; w_{1d}))
\]

\[
Y_2 = ReLU(Conv2d(X_2; w_{2d}))
\]

\[
Y_3 = ReLU(Conv3d(X_3; w_{3d}))
\]

where \( w_{1d} \in \mathbb{R}^{3 \times 1 \times 1} \), \( w_{2d} \in \mathbb{R}^{1 \times 3 \times 3} \), \( w_{3d} \in \mathbb{R}^{3 \times 3 \times 3} \) are the kernels of convolutions. In particular, the function \( Conv1d \) with \( w_{1d} \in \mathbb{R}^{3 \times 1 \times 1} \) focuses on modeling the temporal relations, \( Conv2d \) with \( w_{2d} \in \mathbb{R}^{1 \times 3 \times 3} \) extracts the spatial features, and \( Conv3d \) with \( w_{3d} \in \mathbb{R}^{3 \times 3 \times 3} \) models the spatio-temporal patterns from their corresponding input feature groups.

Finally, the resulting features \( \{ Y_1, Y_2, Y_3 \} \) are further concatenated together along channel dimension as

\[
Y = Concat(Y_1, Y_2, Y_3)
\]

where \( Y \in \mathbb{R}^{C \times T \times H \times W} \) is the ultimate learned feature map of STS.

2.2 Computational Analysis

We analyze the model complexity of our STS by making comparison with existing modules. Table 1 lists the comparison of parameters of different modules. As these modules only differ in the middle filter layer, we just show the number of parameters of middle layer parts. It can be found that the number of parameters of STS is controlled by the three hyperparameters \( \beta_1, \beta_2, \beta_3 \). For example, when setting \( \beta_1, \beta_2, \beta_3 = \frac{1}{4}, \frac{1}{2}, \frac{1}{4} \), the number of parameters of STS module is \( \frac{23}{8}C^2 \), which is only 46% of C2D, 15% of C3D and 61% of GST. This number of parameters is much smaller than others.

| Module                        | Parameters                |
|-------------------------------|---------------------------|
| C2D (Figure 1-(a))            | \( 9 \times C^2 \)        |
| C3D (Figure 1-(b))            | \( 27 \times C^2 \)       |
| R(2+1)D (Figure 1-(c))       | \( 12 \times C^2 \)       |
| GST (Figure 1-(d))           | \( 9 \times \frac{3}{4} \times C^2 \) |
| STS (ours)                    | \( (3\beta_1^2 + 9\beta_2^2 + 27\beta_3^2) \times C^2 \) |
3. Results & Discussion

3.1 Datasets
We mainly evaluate our module on two video datasets that require spatial and temporal modelling, i.e., Something-Something V1 [11] and EGTEA Gaze+ [12].

**Something-Something V1 (SSV1)** [11] is a large-scale video dataset for action recognition. There is totally about 110k videos for 174 fine-grained classes with diverse objects, backgrounds, and viewpoints. The fine-grained level classes need extensive temporal reasoning to differentiate them.

**EGTEA Gaze+ (EGaze+)** [12] is a first-vision dataset for action recognition. Specifically, EGTEA Gaze+ contains ~10k instances of fine-grained actions for 106 activity classes. We use the first official train/validation split (split1) for performance examination. We also conduct ablation study in this dataset.

3.2 Implementation detail
Our STS module is implemented in Pytorch, running on several GPUs (2080Ti or 3090) and adopts ResNet-50 pretrained on ImageNet [3] as backbone. We obtain the input frames same as method described in TSN [13] and later resize them with the shorter size as 256 for the datasets. During training, a 224×224 patch is cropped out of the center of the frame. We train our module for 50 epochs in total with an initial learning rate of 0.01 and decay it by a factor of 10 every 20 epochs. In the inference, we uniformly sample 8/16 frames from a video and fix the center crop as 224×224 for the final result report.

3.3 Result

Table 2. Performance comparison of different methods on Something-Something V1 and EGTEA Gaze+ datasets. The results of competing methods are obtained by re-running their released codes.

| Module       | Backbone   | Pretrain | Frames | Params (SSV1) | FLOPs (SSV1) | Top-1 (SSV1) | Top-1 (EGaze+) |
|--------------|------------|----------|--------|---------------|--------------|--------------|----------------|
| C2D (TSN [13]) | ResNet-50  | ImageNet | 8      | 23.9M         | 32.9G        | 17.7%        | 61.6%          |
| C3D [1]      |            |          | 8      | 42.5M         | 62.5G        | 46.2%        | 62.1%          |
| R(2+1)D [4] |            | Kinetics | 8      | 29.4M         | 37.8G        | 35.0%        | ---            |
| GST [9] |            |          | 8      | 21.0M         | 29.5G        | 44.4%        | 63.3%          |
| STS (ours)   |            | ImageNet | 8      | 16.3M         | 23.0G        | 44.5%        | 58.8%          |
| (β₁ = 1/4, β₂ = 1/2, β₃ = 1/4) | | 8×2 clips | 23.0G×2 | 46.0%        | 59.8%        |
|              |            |          | 16     | 43.0G         | 46.0%        | 59.0%        | 59.3%          |

We compare STS with state-of-the-art networks in this section. The result comparison follows the same protocol of using RGB frames as input and adopting ResNet-50 or its variants as backbones.

Table 2 shows the top-1 accuracies of these competing methods, as well as the numbers of parameters and FLOPs, on the two used datasets. Particularly, our proposed STS model only need 16.3M parameters and 23.0G (8 frames) FLOPs. This is about 2.7× efficient to the advanced 3D model C3D (62.5G FLOPs). In terms of top-1 accuracy, STS achieves the results of 44.5%/58.8% with 8 frames and 46.0%/59.0% with 16 frames on SSV1/EGaze+ datasets. We also report the results of using twice sampling strategy for much better performance obtaining. As shown in this table, STS can achieve the highest performance of 46.7% with 16×2 clips on SSV1 dataset. Being different to SSV1, we observe that there is not a very significant performance improvement on EGaze+ dataset. The possible reason can be that the first-vision activities occurred at fixed scenes (kitchen places) and the objects are more important for the video categorization. This can be verified by the good performance of 2D model C2D. Overall, considering the low computation cost, STS is better or competitive to the existing methods.
3.4 Ablation study

Table 3. Accuracy of STS with different settings of \(\{\beta_1, \beta_2, \beta_3\}\).
Results are obtained with 8 input frames (centre crop).

| dataset   | \(\beta_1, \beta_2, \beta_3\) | Top-1 Accuracy |
|-----------|-------------------------------|----------------|
| EGaze+    | \(\frac{1}{2}, \frac{1}{4}, \frac{1}{4}\) | 53.31%         |
|           | \(\frac{1}{4}, \frac{1}{2}, \frac{1}{4}\) | 58.75%         |
|           | \(\frac{1}{4}, \frac{1}{4}, \frac{1}{2}\) | 53.66%         |

In this part, we present the ablation study on EGaze+ dataset to study the impacts of different settings of \(\{\beta_1, \beta_2, \beta_3\}\). The three hyperparameters control the splitting ratios for different information modeling. Here, we empirically select three groups of settings, i.e., \(\{\frac{1}{2}, \frac{1}{4}, \frac{1}{4}\}\), \(\{\frac{1}{4}, \frac{1}{2}, \frac{1}{4}\}\) and \(\{\frac{1}{4}, \frac{1}{4}, \frac{1}{2}\}\). Table 3 shows the results of different settings. It can be found when setting \(\{\beta_1, \beta_2, \beta_3\} = \{\frac{1}{4}, \frac{1}{2}, \frac{1}{4}\}\), STS achieves the best result of 58.75%. This result is much better than the others (i.e., 53.31% and 53.66%). The phenomenon gives evidence that modeling spatial patterns is more essential than modeling temporal relations. Results in the work of GST also demonstrate this. But this does not mean that the temporal modeling is needless. Jointly learning space-time patterns from videos is a reliable solution.

4. Conclusion

In this paper, we have presented a simple yet efficient network for video classification. The proposed STS module splits the feature channels into three non-overlapped groups and uses three kinds of convolutions to conduct paralleled modeling. STS can achieve joint spatial/temporal/spatio-temporal modeling in a single module. The experimental results obtained on two benchmark video datasets demonstrate the effectiveness and efficiency on video classification.

Acknowledgments

This work is supported the National Key Research and Development Program of China (2020YFB1406703).

References

[1] Du Tran, Lubomir Bourdev, Rob Fergus, Lorenzo Torresani, and Manohar Paluri. (2015) Learning spatiotemporal features with 3d convolutional networks. In: Proceedings of the IEEE international conference on computer vision, pages 4489–4497.
[2] S. Hochreiter and J. Schmidhuber. (1997) Long Short-Term Memory. Neural Comput., vol. 9: pages 1735–1780.
[3] J. Carreira and A. Zisserman. (2017) Quo vadis, action recognition? a new model and the kinetics dataset. In: proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pages 6299–6308.
[4] D. Tran, H. Wang, L. Torresani, J. Ray, Y. LeCun, and M. Paluri. (2017) a closer look at spatiotemporal convolutions for action recognition. In: Proceedings of the IEEE conference on Computer Vision and Pattern Recognition, pages 6450–6459.
[5] T. Y. Zhao and Tao Mei. (2017) Learning Spatio-Temporal Representation with Pseudo-3D Residual Networks. In: proceedings of the IEEE International Conference on Computer Vision, pages 5533–5541.
[6] S. Xie, R. Girshick, and P. Doll. (2017) Aggregated Residual Transformations for Deep Neural Networks. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pages 7794–7803.
[7] J. Lin, C. Gan, and S. Han. (2018) Temporal shift module for efficient video understanding. arXiv preprint arXiv:1811.08383.

[8] S. Sudhakaran, S. Escalera, and O. Lanz. (2019) Gate-shift networks for video action recognition. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pages 1102–1111.

[9] C. Luo and A. Yuille. (2019) Grouped spatial-temporal aggregation for efficient action recognition, In: Proceedings of the IEEE/CVF International Conference on Computer Vision, pages 5512–5521.

[10] K. He, X. Zhang, S. Ren, and J. Sun. (2016) Deep residual learning for image recognition. In: Proceedings of the IEEE conference on computer vision and pattern recognition, pages 770–778.

[11] R. Goyal et al. (2017) The ‘Something Something’ Video Database for Learning and Evaluating Visual Common Sense. In: Proceedings of the IEEE International Conference on Computer Vision, volume 2: page 8.

[12] Y. Li, M. Liu, and J. Rehg. (2018) In the Eye of the Beholder: Gaze and Actions in First Person Video. In: The European Conference on Computer Vision.

[13] L. Wang, Y. Xiong, Z. Wang, Y. Qiao, D. Lin, X. Tang, and L. Van Gool. (2016) Temporal segment networks: Towards good practices for deep action recognition. In: European conference on computer vision, pages 20–36.

[14] Y. Hao, H. Zhang, C. Ngo, Q. Liu, X. Hu. (2020) Compact Bilinear Augmented Query Structured Attention for Sport Highlights Classification. In: Proceedings of the 28th ACM International Conference on Multimedia, pages 628-636.

[15] Y. Hao, Z. Liu, H. Zhang, B. Zhu, J. Chen, Y. Jiang, C. Ngo. (2020) Person-level Action Recognition in Complex Events via TSD-TSM Networks. In Proceedings of the 28th ACM International Conference on Multimedia, pages 4699-4702.