SSAN: Separable Self-Attention Network for Video Representation Learning

Xudong Guo\textsuperscript{*1} \hspace{1cm} Xun Guo\textsuperscript{2} \hspace{1cm} Yan Lu\textsuperscript{2}
\textsuperscript{1}Tsinghua University \hspace{1cm} \textsuperscript{2}Microsoft Research Asia
\texttt{gx20@mails.tsinghua.edu.cn} \hspace{1cm} \texttt{\{xunguo, yanlu\}@microsoft.com}

Abstract

Self-attention has been successfully applied to video representation learning due to the effectiveness of modeling long range dependencies. Existing approaches build the dependencies merely by computing the pairwise correlations along spatial and temporal dimensions simultaneously. However, spatial correlations and temporal correlations represent different contextual information of scenes and temporal reasoning. Intuitively, learning spatial contextual information first will benefit temporal modeling. In this paper, we propose a separable self-attention (SSA) module, which models spatial and temporal correlations sequentially, so that spatial contexts can be efficiently used in temporal modeling. By adding SSA module into 2D CNN, we build a SSA network (SSAN) for video representation learning. On the task of video action recognition, our approach outperforms state-of-the-art methods on Something-Something and Kinetics-400 datasets. Our models often outperform counterparts with shallower network and fewer modalities. We further verify the semantic learning ability of our method in visual-language task of video retrieval, which showcases the homogeneity of video representations and text embeddings. On MSR-VTT and Youcook2 datasets, video representations learnt by SSA significantly improve the state-of-the-art performance.

1. Introduction

Video representation learning is crucial for tasks such as detection, segmentation and action recognition. Although 2D and 3D CNN based approaches have been extensively explored to capture the spatial-temporal correlations for these tasks, learning strong and generic video representations is still challenging. One possible reason is that videos contain not only rich semantic elements within individual frames, but also the temporal reasoning across time which links those elements to reveal the semantic-level information for actions and events. Effective modeling of long range dependencies among pixels is essential to capture such contextual information, which current CNN operations can hardly achieve. RNN based methods [23] have been used for this purpose. However, they suffer from the high computational cost. More importantly, RNN cannot establish the direct pairwise relationship between positions regardless of their distance.

Self-attention mechanism has been recognized as an effective way to build long range dependencies. In natural language processing, self-attention based transformer [35] has been successfully used to capture contextual information from sequential data, \textit{e.g.}, sentences. Recent efforts have also introduced self-attention to computer vision domain for visual tasks such as segmentation and classification [40, 47, 37, 14]. The work from Wang \textit{et al.} [37] proposed a generic self-attention form, \textit{i.e.}, non-local mean, for video action recognition, which builds pairwise correlations for pixel locations from space and time simultaneously. However, the correlations from space and time represent different contextual information. The former often relates to scenes and objects, and the latter often relates to temporal reasoning for actions (short-term activities) and

\textsuperscript{*}The work was done when the author was with MSRA as an intern.
2. Related Work

2.1. Video Learning Networks

With the success of CNN [34, 38, 43, 4] in image tasks, great efforts have been put on video tasks. CNN networks are also extended from image to video. There are mainly two branches for video learning architectures: 2D based methods [16, 36, 24] and 3D based methods [2, 32, 39, 25]. I3D [2] network proposed to inflate ImageNet pre-trained models from 2D to 3D by copying the weights. S3D [39] proposed to decompose 3D convolutions to spatial and temporal convolutions. TSN [36] proposed an efficient 2D based video learning structure. And TSM [20] proposed to capture temporal dependency by simply shifting the features between frames. SlowFast [7] used two branches with different temporal resolutions to capture temporal correlations. 3D based methods suffer from the overhead of parameters and complexity, while 2D based methods need careful temporal feature aggregation. To take the flexibility of 2D based methods for video frames, we employ 2D CNN as our baseline in this work.

2.2. Action Recognition

Action recognition, also known as video classification [8, 34, 33], has been extensively explored in recent years. The accuracy of action recognition is highly related to video representation learning. Early works try to use 2D based methods to video. Later, 3D convolution networks are explored and achieve excellent performance. However, huge complexity makes these methods expensive to be used. Moreover, 3D based methods always take several consecutive frames as input, so that videos with complex actions can not be well handled. Recently, 2D convolution networks with temporal aggregation achieve significant improvements, which have flexible structures and inputs, as well as much lower computational and memory costs than 3D based methods. Representations learnt for action recognition are always used as initialization for other tasks.

2.3. Self-Attention

Self-attention mechanism has been successfully used in machine translation domain [35]. Recently, there have been extensive efforts to study its applications on computer vision tasks such as classification and segmentation [40, 47, 37, 46, 13, 9, 10, 14]. There are also efforts to use attention instead of convolutions for feature extraction [1]. For video classification, non-local network [37] proposed to use non-local mean, which is based on self-attention, to capture global dependency among pixel positions. Each pixel position attends to all other positions from both spatial and temporal dimensions. The full connections among pixels do take broader attention fields. However, this also introduces irrelevant information. Our work is closely related to non-local block, and try to investigate the relation of spatial attention and temporal attention.

2.4. Visual-Language Learning

Recently, joint vision and language training [28] is more and more popular for vision tasks. By adding language models, such as BERT [5], the semantic information can be efficiently learned from videos for multi-modal tasks, such as retrieval and captioning [31, 45]. VideoBERT [30] proposed to use a visual-linguistic model to learn high-level features without any explicit supervision. Sun et al. [29] then proposed to use contrastive bidirectional transformer
Figure 2. Design of separable self-attention attention module. Spatial attention (SA) part is highlighted in yellow. Temporal attention (TA) part is highlighted in blue.

(CBT) to perform self-supervised learning for video representations. UniViLM [27] also proposed a joint video and language pre-training scheme by adding generation tasks in the pre-training process, which is more efficient than CBT in some downstream tasks. However, these methods only focus on the training of the transformer encoder and decoder, while take the video networks as feature extractors. Therefore, it is still important to improve video learning networks to efficiently learn real semantic information.

3. Separable Self-Attention Network

In this section, we describe our proposed separable self-attention network in details.

3.1. Self-Attention in Vision

Let $X \in \mathbb{R}^{T \times H \times W \times C}$ be the input features with $T$ frames. $H$, $W$ and $C$ denote spatial size, temporal size and channel number, respectively. A typical 3D self-attention/3D NL block [37] maps $X$ into query, key and value embeddings using three $1 \times 1 \times 1$ convolutions, which are denoted as $X_q$, $X_k$ and $X_v$. The three embeddings are then reshaped to the sizes of $THW \times C$, $C \times THW$ and $THW \times C$, respectively. After that, the similarity matrix $M \in \mathbb{R}^{THW \times THW}$, which builds the full pairwise relationships for locations from spatial and temporal dimensions, is calculated using matrix multiplication as

$$M = X_q \times X_k. \quad (1)$$

$M$ is then normalized with $softmax$ function and distributed to each location to generate the attention map $Y$ as

$$Y = softmax(M) \times X_v, \quad (2)$$

where $Y \in \mathbb{R}^{THW \times C}$. Each element $m_{ij}$ in matrix $M$ measures the similarity between position $i$ and position $j$ in spatial and temporal dimensions. Attention $Y$ is then transformed by $1 \times 1 \times 1$ convolution $W_z$ and added back to the original query feature $X$, like a residual connection, to generate the output feature $Z$:

$$Z = W_z(Y) + X. \quad (3)$$

3.2. Separable Self-Attention Module

The 3D based self-attention can successfully model long range dependencies from space and time simultaneously. However, such dependencies are first-order correlations which mainly capture the similarity between single pixels, but not semantic-level correlations. For example, if $i$ and $j$ are positions from different frames, lacking the prior spatial correlations between $i$ and other positions in the same frame with $i$, the calculated correlation between $i$ and $j$ may not describe the true temporal relationships of the scenes and objects they belong to. Furthermore, Equation 1 shows that the existing self-attention design, such as non-local block [37], considers more on position-wise correlations, but less on channel-wise correlations, which contain important classification information. This may lead to information loss for scenes and objects.
Based on above analysis, we carefully design a separable self-attention module, which follows two principles. Firstly, the spatial and temporal attentions are performed sequentially, so that temporal correlations can fully consider the spatial contexts. Secondly, spatial attention maps exploit as much context information as possible.

The main structure of our proposed SSA is illustrated in Figure 2 highlighted in grey. The input feature \( X \) is first mapped into spatial query, key and value embeddings using 2D \( 1 \times 1 \) convolutions, denoted as \( X^t_q, X^t_k \) and \( X^t_v \), where \( t \in [0, T] \) is time index. Then, the embeddings are used to generate \( T \) spatial attention maps independently, which are then concatenated together as a 4D intermediate attention maps \( \hat{X} \) with the dimension of \( T \times H \times W \times C \). After that, \( \hat{X} \) is transformed to temporal embeddings \( \hat{X}_q \) and \( \hat{X}_v \) using \( 3 \times 1 \times 1 \) convolution. Note that in our design, the spatial attention and temporal attention share the same mapping with different shapes, \( i.e., X^t_q \) and \( X^t_v \). The three temporal embeddings are then used to generate the temporal attention. We describe the details of spatial attention and temporal attention as follows.

**Spatial attention:** For spatial attention, we consider both position-wise attention and channel-wise attention. For this purpose, we design a two-branch structure, \( i.e., \) position branch and channel branch, to calculate them independently. The details are highlighted in yellow in Figure 2. The two branches share the same embeddings \( X^t_q \) and \( X^t_k \). Each of the embeddings is reshaped to the size of \( HW \times C \) and \( C \times HW \). The position branch generates spatial similarity matrix \( M_S \in \mathbb{R}^{N \times N} \), where \( N = H \cdot W \). This is also the design of most existing 2D self-attention methods [47]. The channel branch generates channel similarity matrix \( M_C \in \mathbb{R}^{C \times C} \) to explore the dependencies along channel dimension. The two branches are calculated as

\[
M_S = X^t_q(S) \times X^t_k(S) \tag{4}
\]

\[
M_C = X^t_q(C) \times X^t_k(C) \tag{5}
\]

where \( S \) and \( C \) denote position branch and channel branch, respectively. The spatial attention maps for time \( t \) are then calculated as

\[
\hat{X}_t = (M_S \times X^t_v(S)) + (M_C \times X^t_v(C)). \tag{6}
\]

The intermediate attention maps \( \hat{X} = Cat[\hat{X}_0, \hat{X}_1, ..., \hat{X}_T] \) can be generated for the next stage, \( i.e., \) temporal attention.

**Temporal attention:** Temporal attention is performed using the spatial attention maps \( \hat{X} \) as input. Unlike the 3D self-attention that uses three \( 1 \times 1 \times 1 \) convolutions to generate the embeddings, we use one \( 3 \times 1 \times 1 \) convolution instead. This design allows temporal fusion on spatial attention maps and builds the short range correlations along temporal dimension, so that the short-term activities can also be attended to. The feature maps of \( \hat{X} \) are then reshaped to generate \( \hat{X}_q \) and \( \hat{X}_k \) for calculating similarity matrix \( M_T \in \mathbb{R}^{T \times T} \) as

\[
M_T = \hat{X}_q \times \hat{X}_k \tag{7}
\]

Thus, the final output attention map \( Y \) is calculated as

\[
Y = M_T \times \hat{X}_v. \tag{8}
\]

### 3.3. Network Architecture

The recent works on video learning show the excellent performance of 2D CNN based frameworks such as TSN [36], TRN [44] and TSM [20]. Comparing with the 3D CNN based methods, 2D based methods can better decouple the spatial and temporal modeling, thereby achieve superior performance on video clips with high dynamics. We choose TSN framework as our baseline, which uniformly splits a video clip into \( T \) snippets and selects only one frame per snippet. We insert SSA module into different layers to establish separable self-attention network (SSAN). Figure 3 shows an example of SSAN on ResNet-50 backbone. In this architecture, SSA module is inserted into the residual blocks to capture spatial and temporal contextual information in different stages. Particularly, we insert SSA module in Res2 and Res3 blocks right after the first \( 1 \times 1 \) convolution. Please note that there is temporal fusion only at the end of TSN. Thus, the temporal information exchange in middle layers comes only from SSA module, which better illustrates the temporal modeling ability of SSA. SSA is a flexible building module and can also be easily added in 3D based architecture.

### 4. Experiments

To demonstrate the effectiveness of our approach, we conduct comprehensive experiments on the standard vi-
<table>
<thead>
<tr>
<th>Method</th>
<th>Backbone</th>
<th>Modality</th>
<th>Frames</th>
<th>Val Top-1</th>
<th>Val Top-5</th>
<th>Test Top-1</th>
</tr>
</thead>
<tbody>
<tr>
<td>I3D [2]</td>
<td>Inception</td>
<td>RGB</td>
<td>64</td>
<td>45.8</td>
<td>76.5</td>
<td>27.2</td>
</tr>
<tr>
<td>NL I3D + GCN [2]</td>
<td>ResNet-50</td>
<td>RGB</td>
<td>32+32</td>
<td>46.1</td>
<td>76.8</td>
<td>45.0</td>
</tr>
<tr>
<td>S3D [39]</td>
<td>Inception</td>
<td>RGB</td>
<td>64</td>
<td>47.3</td>
<td>78.1</td>
<td>-</td>
</tr>
<tr>
<td>S3D-G [39]</td>
<td>Inception</td>
<td>RGB</td>
<td>64</td>
<td>48.2</td>
<td>78.7</td>
<td>42.0</td>
</tr>
<tr>
<td>TSM [36]</td>
<td>BNInception</td>
<td>RGB</td>
<td>8</td>
<td>19.5</td>
<td>-</td>
<td></td>
</tr>
<tr>
<td>TRN [44]</td>
<td>BNInception</td>
<td>RGB</td>
<td>8</td>
<td>34.4</td>
<td>-</td>
<td>33.6</td>
</tr>
<tr>
<td>bLVNet-TAM [6]</td>
<td>bLResNet-50</td>
<td>RGB</td>
<td>16×2</td>
<td>48.4</td>
<td>78.8</td>
<td>-</td>
</tr>
<tr>
<td>bLVNet-TAM [6]</td>
<td>bLResNet-101</td>
<td>RGB</td>
<td>16×2</td>
<td>49.6</td>
<td>79.8</td>
<td>48.9</td>
</tr>
<tr>
<td>TSM [20]</td>
<td>ResNet-50</td>
<td>RGB</td>
<td>8</td>
<td>45.6</td>
<td>74.2</td>
<td></td>
</tr>
<tr>
<td>TSM [20]</td>
<td>ResNet-50</td>
<td>RGB</td>
<td>16</td>
<td>47.2</td>
<td>77.1</td>
<td>46.0</td>
</tr>
<tr>
<td>TSM_{E_n} [20]</td>
<td>ResNet-50</td>
<td>RGB</td>
<td>24</td>
<td>49.7</td>
<td>78.5</td>
<td>-</td>
</tr>
<tr>
<td>TSM 2-stream [20]</td>
<td>ResNet-50</td>
<td>RGB+flow</td>
<td>16+16</td>
<td>52.6</td>
<td>81.9</td>
<td>50.7</td>
</tr>
<tr>
<td>SSA(Ours)</td>
<td>ResNet-50</td>
<td>RGB</td>
<td>8</td>
<td>49.5</td>
<td>79.5</td>
<td>-</td>
</tr>
<tr>
<td>SSA(Ours)</td>
<td>ResNet-50</td>
<td>RGB</td>
<td>16</td>
<td>51.7</td>
<td>81.3</td>
<td>-</td>
</tr>
<tr>
<td>SSA_{E_n}(Ours)</td>
<td>ResNet-50</td>
<td>RGB</td>
<td>16+8</td>
<td>55.1</td>
<td>84.9</td>
<td>54.0</td>
</tr>
</tbody>
</table>

Table 1. Comparisons with state-of-the-art methods on Something-Something-V1 validation and test sets.

sion task of video action recognition where the large-scale Something-Something dataset and Kinetics benchmark are used. Furthermore, to verify the efficiency of the video representations learnt by SSAN, we also conduct a vision-language task of text-based video retrieval which searches the corresponding video clips by text query. This task showcases the homogeneity between language and video representations, thereby the semantic-level information the video representations contain.

4.1. Dataset

**Something-Something**: Something-Something dataset [12] is a large scale benchmark dataset for video action recognition, containing 174 video categories of human-object interactions, e.g., “moving something across a surface without it falling”. Temporal reasoning is critical to infer the actions in this dataset. There are two versions, which contain 108k and 220k videos, respectively. Videos in Something-Something-V1 are split into 86K, 11K and 11K as training, validation and test sets. Something-Something-V2 is an updated version and contains 129K, 25K and 27K videos for training, validation and test. We conduct the experiments on both validation and test sets. It is noteworthy that labels for test sets are not publicly available. We submitted the inference results to the benchmark and report the scores published in the leaderboards.

**Kinetics-400**: Kinetics-400 [2] is a popular benchmark for action recognition collected from Youtube, which contains 400 action categories. There are totally 300K video samples, which are divided to 240K, 20K and 40K as training, validation and test sets, respectively. Videos in Kinetics-400 are relatively longer and more complex. Each video is trimmed to around 10-second clip. Kinetics is less sensitive to temporal relationships compared with Something-Something.

4.2. Experimental Setups

**Video action recognition**: We adopt the sparse sampling and data augmentation in TSN [44] to train our model. Specifically, we first divide a video clip into T uniform segments and then select one random frame from each segment as the input. The input frames are resized as 256×256 and randomly cropped to the size of 224×224. We take single clip and the 224×224 central crop for evaluation unless otherwise specified.

Our model is initialized with the weights pre-trained on ImageNet. For Something-Something dataset, we train our models for around 80 epochs. And for Kinetics-400 dataset, the models are trained for around 200 epochs. For all the models, the initial learning rate is 0.01 and decays by 0.1 when the validation loss reaches the plateau. We adopt a linear warmup strategy [11] for the first epoch. Batch size is set as 64. We utilize the Nesterov momentum optimizer with a weight decay of 0.0005 and a momentum of 0.9. Dropout rate of 0.5 is also used to reduce over-fitting.

**Text-based video retrieval**: This task aims at retrieving the most relevant video clip given an input text query. We adopt the method in [22], which uses a gated recurrent unit (GRU) based text-video joint embedding network to measure the similarity between video representations and text embeddings. We adopt the training and inference strategies in [22] including text pre-processing, except that the video representations are extracted using our SSA with ResNet-50 instead of the original 3D ResNeXt-
101 pre-trained on Kinetics dataset. We also pre-train our SSA with ResNet-50 on Kinetics dataset using video classification task. Then, we use SSA as a feature extractor to get video representations for retrieval. The GRU network can be pre-trained on HowTo100M [22] dataset, which is a video-language pre-training dataset containing one million narrated instructional web videos, then fine-tuned on MSR-VTT and Youcook2 datasets. The evaluation metric is “Recall at K” (R@K), which is the percentage of video candidates retrieved in the top K video clips. We report this recall metrics on two popular video-language datasets, i.e., MSR-VTT and Youcook2.

4.3. Experiments on Action Recognition

Comparisons with non-local block: Non-local block [37] has been proved to significantly improve performance in video classification. We compare our approach with NL block to demonstrate the effectiveness of our separable design. In Table 2, we show the results of 3D NL (spatial-temporal), 2D NL (spatial) and 1D NL (temporal). We can see obvious performance improvement of SSA over 3D NL. This demonstrates that separating spatial and temporal attention is a right way to better model temporal reasoning.

We also show the results of 2D+1D NL, which is a straightforward way to separate self-attention module along spatial and temporal dimensions. In specific, 1D NL is performed right after 2D NL. We insert NL block into the same locations with SSA for fair comparison, and observe superior performance of SSA. This interesting result shows that separable self-attention does need careful design.

Comparisons with state-of-the-art: We compare our approach with previous state-of-the-art methods on Something-Something and Kinetics datasets. For Something-Something-V1 and V2, we report our results on both validation and test sets.

The results of SSA and other state-of-the-art methods on Something-Something-V1 are summarized in Table 1. The first section shows the results of I3D and S3D, as well as their enhanced variants NL I3D + GCN and S3D-G, which enable full temporal fusion and attention mechanism. With much fewer input frames, our 16-frame model outperforms these methods by a large margin (5.9%, 4.3%, 5.6% and 3.2%, respectively). The second section shows the results of TSN and TRN. TSN has no temporal fusion operation, so that the performance is much lower. Our approach is built upon TSN framework. From the table, we can see a significant improvement of SSA (30.0%) when using 8-frame input. Though TRN has a temporal fusion at the end of trunk network, the performance is still relatively low. This shows the fact that modeling temporal context across different layers is important. That is also why we choose to insert SSA module into the layers of the backbone, but not add it at the end as a head. Our RGB-only ensemble model (SSAEn), which is the ensemble of 8-frame input and 16-frame input,
Table 5. Ablation study of individual performance for spatial attention (SA), temporal attention (TA) and shared value embedding (SVE) on Something-Something-V2.

<table>
<thead>
<tr>
<th>Method</th>
<th>Backbone</th>
<th>Res2</th>
<th>Res3</th>
<th>Val Top1</th>
</tr>
</thead>
<tbody>
<tr>
<td>SSA</td>
<td>ResNet-50</td>
<td>✓</td>
<td></td>
<td>61.2</td>
</tr>
<tr>
<td>SSA</td>
<td>ResNet-50</td>
<td>✓</td>
<td>✓</td>
<td>61.8</td>
</tr>
<tr>
<td>SSA</td>
<td>ResNet-50</td>
<td>✓</td>
<td>✓ ✓</td>
<td>62.3</td>
</tr>
</tbody>
</table>

Table 6. Ablation study of different locations that SSA is inserted in. Note that we insert SSA into 5 residual blocks (2 from Res2 and 3 from Res3).

outperforms TSM 2-stream model with fewer input frames (24 vs. 32).

We also compare our approach on Something-Something-V2 with state-of-the-art 2D CNN based methods in Table 3. Compared with Something-Something-V1, the videos in Something-Something-V2 are more sensitive to the number of input frames. It can be seen that SSA model with 16-frame input achieves comparable performance with TSM 2-stream model with 32-frame input. And compared with bLVNet-TAM, which has two branches of different spatial resolutions, our method achieves much better results.

In both Table 1 and Table 3, SSA outperforms deeper network (ResNet-101) by a large margin (4.5% and 5.5% respectively), which show the strong video learning ability.

We also show per class results in Figure 4 to figure out the efficiency of our method in individual classes. We list the results of TSM, which uses temporal shifting to exchange information among frames, to better understand the difference between temporal fusion and temporal attention.

The performance comparisons on Kinetics-400 are summarized in Table 4, including two recent self-attention based methods, i.e., A²-Nets [3] and NL block [37]. Since Kinetics-400 dataset is less sensitive to the depth of network, we compare the results mainly on ResNet-50 backbone. From the table, we can see that our models outperform other methods when the numbers of input frames are the same. Our result with 16-frame input is comparable with that of NL 13D with 128-frame input, which demonstrates the strong temporal modeling ability of SSA. And our ensemble model (SSA Ensemble) achieves 1% higher accuracy than NL 13D with much fewer input frames (24 vs. 128).

**Ablation study:** Although SSA has been proven to be efficient for video representation learning, we still like to fully investigate and understand this idea. Therefore, we conduct ablation experiments to showcase how each design affects the overall performance.

As shown in Figure 2, our design consists of two sequential parts, i.e., spatial attention (SA) and temporal attention (TA). SA has a two-branch structure, which contains position branch and channel branch. The position branch is similar to 2D NL block. Therefore, SA is an enhanced version of 2D NL block by adding channel branch. In TA, we use $3 \times 1 \times 1$ convolution instead of $1 \times 1 \times 1$ convolution in NL block. We use 2D+1D NL as baseline and add our modifications one by one to demonstrate their contributions to the overall performance. Besides SA and TA, we also evaluate the performance of shared value embedding (SVE), which aims to reduce complexity by removing a $1 \times 1 \times 1$ convolution. The ablation study is shown in Table 5. The results show that both SA and TA improve performance. When using them together, there is significant performance boost. Table 6 shows the results of different locations that SSA is inserted in. We can see that inserting SSA to either one of Res2 and Res3 can take significant improvement. According to our experiments, inserting SSA to the later blocks such as Res4/Res5 has lower performance than Res2/Res3 (around 0.5%-0.7%). This may be due to the reduction of spatial resolution.

### 4.4. Experiments on Video Retrieval

Video retrieval is a standard video-language task to find video candidates by text query. The normal process is to jointly train a cross encoder of video representations and text embeddings to learn their similarity. Text embeddings contain explicit semantic information. As a consequence, the more semantic information that video representations contain, the better the accuracies are. That is also major reason that we verify our method on this task. We employ two popular large-scale video-language datasets, i.e., MSR-VTT and Youcook2. MSR-VTT contains videos in 20 domains such as music, sports and movies. For each video, 20 captioning sentences are annotated by human workers. There are totally 200K unique clip-caption pairs. We adopt the test strategy of JSFusion [41] and use 1000K clip-caption pairs as test data. Youcook2 is a cooking video data set, including 14K video clips from 89 recipes. Since the video clips are much longer (5.26 minutes on average) and contain some uncorrelated scenes other than cooking instructions, Youcook2 is a challenging dataset.

Table 7 summarizes the performance comparisons on MSR-VTT dataset. To demonstrate the affection of video features to the retrieval performance, no pre-training is applied on these methods. The first section shows previous state-of-the-art methods. From the results, we can see that text-to-video retrieval is really a challenging task. The sec-
Figure 4. The top 15 categories which are improved by SSA (orange) compared to TSM (blue) on Something-Something-V2. There are some fine-grained categories that contain challenging actions such as “Putting something that can’t roll onto a slanted surface, so it stays where it is”. Our model also works better on some categories with very small differences such as “Putting something behind something” and “Putting something into something”.

<table>
<thead>
<tr>
<th>Method</th>
<th>R@1†</th>
<th>R@5†</th>
</tr>
</thead>
<tbody>
<tr>
<td>Random</td>
<td>0.1</td>
<td>0.5</td>
</tr>
<tr>
<td>C+LSTM+SA [31]</td>
<td>4.2</td>
<td>12.9</td>
</tr>
<tr>
<td>VSE-LSTM [18]</td>
<td>3.8</td>
<td>12.7</td>
</tr>
<tr>
<td>Kaufman et al. [17]</td>
<td>4.7</td>
<td>16.6</td>
</tr>
<tr>
<td>CT-SAN [42]</td>
<td>4.4</td>
<td>16.6</td>
</tr>
<tr>
<td>JSFusion [41]</td>
<td>10.2</td>
<td>31.2</td>
</tr>
<tr>
<td>GRU+ResNeXt-101 [22]</td>
<td>12.1</td>
<td>35.0</td>
</tr>
<tr>
<td>GRU+SSA</td>
<td>24.4</td>
<td>49.3</td>
</tr>
</tbody>
</table>

Table 7. Results of text-based video retrieval on MSR-VTT dataset. The only difference between our method and GRU+ResNeXt is the video representation.

<table>
<thead>
<tr>
<th>Method</th>
<th>R@1†</th>
<th>R@5†</th>
</tr>
</thead>
<tbody>
<tr>
<td>Random</td>
<td>0.03</td>
<td>0.15</td>
</tr>
<tr>
<td>HGLMM FV CCA [19]</td>
<td>4.6</td>
<td>14.3</td>
</tr>
<tr>
<td>GRU+ResNeXt-101 [22]</td>
<td>4.2</td>
<td>13.7</td>
</tr>
<tr>
<td>GRU+ResNeXt-101† [22]</td>
<td>8.2</td>
<td>24.5</td>
</tr>
<tr>
<td>GRU+SSA</td>
<td>5.5</td>
<td>15.9</td>
</tr>
<tr>
<td>GRU+SSA†</td>
<td>10.9</td>
<td>28.4</td>
</tr>
</tbody>
</table>

Table 8. Results of text-based video retrieval on Youcook2 dataset. † represents pre-trained on a subset of HowTo100M dataset (around 1.2M videos).

Table 8 summarizes the performance comparisons on Youcook2 dataset. The results show that pre-trained GRU using video representations learnt by SSA can be significantly improved (5.4%), which is larger than 3D ResNeXt (4.0%). This also shows that SSA is generic on different videos.

It is noteworthy that MIL-NCE in [21] also reports the retrieval results on the two datasets. However, MIL-NCE is pre-trained on HowTo100M (around 1.2M videos), while SSA is pre-trained on Kinetics-400 (around 240K videos). Our experiments show that SSA outperforms MIL-NCE on MSR-VTT by 14.5%, but is outperformed by MIL-NCE on Youcook2. Since they are not on the same basis, we don’t include the comparison in above tables.

5. Conclusion

We proposed a separable self-attention network (SSAN) for video representation learning, which learns spatial and temporal correlations in a separable way. In specific, we investigate the relationship between spatial attention and temporal attention and design a sequential structure to model temporal reasoning with the priori of spatial contextual information. By adding the SSA module into 2D CNN backbone, we built a SSA network based on TSN framework. We conducted extensive experiments on large-scale Something-Something dataset and Kinetics-400 dataset to verify our approach. Our SSAN outperforms state-of-the-art methods on both datasets. Furthermore, we also verify the video representations learnt by our method on video-language task of video retrieval. On MSR-VTT and Youcook2 datasets, our method significantly improves state-of-the-art performance.
References


