



3D RANs: 3D Residual Attention Networks for action recognition

Jiahui Cai¹ · Jianguo Hu²

Published online: 25 July 2019
© Springer-Verlag GmbH Germany, part of Springer Nature 2019

Abstract

In this work, we propose 3D Residual Attention Networks (3D RANs) for action recognition, which can learn spatiotemporal representation from videos. The proposed network consists of attention mechanism and 3D ResNets architecture, and it can capture spatiotemporal information in an end-to-end manner. Specifically, we separately add the attention mechanism along channel and spatial domain to each block of 3D ResNets. For each sliced tensor of an intermediate feature map, we sequentially infer channel and spatial attention maps by channel and spatial attention mechanism submodules in each residual unit block, and the attention maps are multiplied to the input feature map to reweight the key features. We validate our network through extensive experiments in UCF-101, HMDB-51 and Kinetics datasets. Our experiments show that the proposed 3D RANs are superior to the state-of-the-art approaches for action recognition, demonstrating the effectiveness of our networks.

Keywords Action recognition · 3D ResNets · Video classification · Attention mechanism

1 Introduction

Human action recognition has been a very hot and challenging research task in recent decades, due to its potentially huge application value in the real world, such as surveillance systems, video indexing and human–computer interaction [1,2]. Motivated by the notable success of 2D convolutional neural networks (2D CNNs) in image domain, many advanced deep models are introduced into video domain for action recognition recently [3–8], which have obtained better recognition accuracy compared to previous methods [9–12]. Human action recognition in video domain needs to consider not only static appearance in each frame but also temporal relation across multiple frames. Therefore, an efficient deep architecture should be able to capture spatiotemporal information and obtain high performance on action recognition task.

Since the introduction of deep networks to this field, there are two different categories for video classification: (1) two-

stream CNNs [13], (2) 3D CNNs [14] and (3) 2D CNNs with LSTM [15–17]. Two-stream CNNs use spatial and temporal streams to capture appearance (RGB frames) and motion (stacked optical flow) information. Although it is an effective method, there is still a signification limitation. It can only represent motion information based on optical flows. We need to train two networks and calculate optical flows, which is expensive to compute. In order to overcome these drawbacks, using 3D CNNs in an end-to-end deep networks to capture spatiotemporal information from stack RGB frames is an effective method. 3D CNNs perform 3D convolution and 3D pooling. However, the performance of 3D CNNs is lower than two-stream CNNs. 2D CNNs with LSTM capture long-term temporal information from videos, but it is difficult to learn good representation for finer temporal relation in short-term frame.

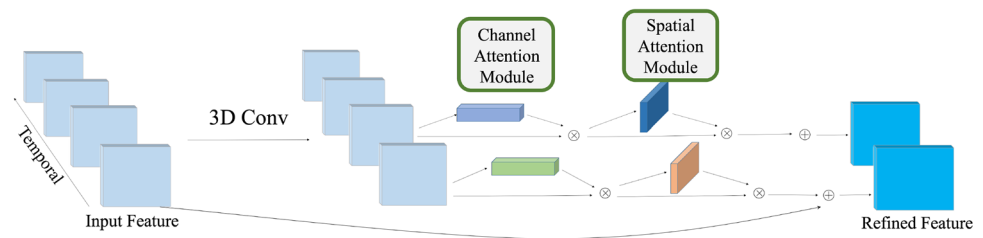
In this paper, we address these problems by proposing a new deep network architecture, named 3D Residual Attention Networks (3D RANs). Our 3D RANs are composed of 3D ResNets [18] and attention mechanism [19–21]. We use the 3D Residual Networks (3D ResNets) as our base networks owing to their good performance in training very deep neural network, which relieves the gradient vanishing problem by shortcut connection. The basic architecture of 3D RAN module is illustrated in Fig. 1. We sequentially add the channel and spatial attention module in the 3D ResNets building block to focus on meaningful feature along two

✉ Jiahui Cai
caijh33@mail2.sysu.edu.cn

¹ School of Electronics and Information Technology, Sun Yat-sen University, 132 East Waihuan Road, Guangzhou College Town, Guangzhou 510006, Guangdong, China

² School of Data and Computer Science, Sun Yat-sen University, 132 East Waihuan Road, Guangzhou College Town, Guangzhou 510006, Guangdong, China

Fig. 1 Diagram of 3D RAN module. For an input 3D signal, we first use a 3D convolution operation to fuse the spatio-temporal information to obtain the intermediate feature maps which are feed into the channel and spatial attention module separately to generate the refined feature maps



dimensions: channel and spatial axis, so that each building block of 3D RAN can learn what and where to focus in the channel and spatial domain. As a result, we use our network to focus on important features and suppress unimportant ones.

A 3D Residual Attention Network is generated by simply stacking multiple 3D RAN modules. We can also use one or more 3D RAN modules instead of its counterparts in the original network. Moreover, the depth can be directly extended to hundreds of layers. We validated the effectiveness of our attention modules through numerous ablation experiments. At the same time, compared with the base 3D ResNet and others state-of-the-art methods, our network can greatly improve the performance of action recognition on multiple benchmark datasets (UCF101 [22], HMDB51 [23] and Kinetics [24]).

2 Related work

In this section, we provide a simple overview about 3D ResNets and attention mechanism.

3D ResNet based In the field of video action recognition, we not only capture features from spatial dimensions but also capture motion information encoded between multiple consecutive frames. Since the success of the residual network (ResNet) [25] in image classification, there were several attempts to build effective residual architecture for video classification and action recognition. Feichtenhofer et al. [26] introduced spatiotemporal ResNets which combined two-stream and residual network to improve action recognition performance. They show the architecture of ResNets is effective for action recognition with 2D CNNs. Moreover, recent studies extended the ResNet architecture to 3D CNNs to learn spatiotemporal information for action recognition. Hara et al. [18] extended 2D-based ResNet to the 3D ones, to capture spatiotemporal features. 3D ResNets perform convolutional and pooling operating with the kernel size of $3 \times 3 \times 3$. 3D ResNets also introduce shortcut connections that bypass convolutional layer directly to the next layer. The connections pass through the gradient flows of network from later layers to early layers and ease training of very deep network. The 3D Residual Networks have been widely used in many subsequent studies on action recognition, action detection, video captioning and hand gesture detection.

To capture long-term temporal information from videos, one general method is to use LSTM to completely model a video. Li et al. [27] proposed a bidirectional LSTM for action recognition by combining the segmented frames in the temporal domain and the local key information in the spatial domain. Song et al. [28] developed an LSTM network with attention modules to allocate different levels of attention on spatial and temporal dimension.

Attention mechanism It is well known that attention plays an important role in the human visual system [29–31]. By quickly scanning the whole scene, human vision obtains the target area that needs to be focused on, and then invests more attention resources in this area to obtain more detailed information about the target. There are two main aspects of attention mechanism: 1. Decide which part of the input needs to focus on. 2. Allocate available processing resources toward the most informative components of the input signal [21,32,33].

Recently, many studies attempt to incorporate attention mechanisms to improve the performance of convolutional neural networks (CNNs) in a range of visual tasks, such as image classification, image location and video understanding [34,35]. Wang *et al.* [36] introduced Residual Attention Network which use a trunk-and-mask module to achieve attention mechanism. By reweighting the feature map, the network not only has excellent performance, but also is robust to input noise. More relevant to our work, Hu et al. [37] propose Squeeze-and-Excitation module to recalibrate channel-wise feature response. They use global average pooling feature to explicitly model independencies between channels and compute channel-wise attention. Based on this, Woo et al. [38] introduced CBAM module sequentially, which infers attention maps along channel and spatial dimensions. Then the attention maps are multiplied to the input feature maps for adaptive feature refinement. They decompose the learning process to learn channel attention and spatial attention in turn. Compared with calculating 3D feature maps directly, the separate attention process has achieved excellent performance with less computation cost and parameters and can be inserted into any preexisting classic CNN architectures.

Toward action recognition, Sharma et al. [39] proposed a recurrent mechanism from RGB data, which integrates convolutional features from different parts of a space–time

volume. Kim et al. [40] proposed Space-Time Cubic Puzzles for self-supervised video representation learning from unlabeled videos dataset. Wang et al. [41] proposed a non-local block to model long-range relations among pixels based on the self-attention mechanism. The non-local operation computes the response at a position as a weighted sum of the features at all positions. All positions can be spatial, temporal and spatiotemporal domains.

3 3D Residual Attention Networks

Our 3D Residual Attention Networks are constructed by stacking multiple 3D attention modules. Each attention module is generated by adding channel and spatial attention mechanisms to the 3D ResNets counterpart module. In this section, we start with a detailed description for our 3D RAN module. Then, we introduce our simple and efficient network architecture.

3.1 3D RAN modules

Given a volume $F \in \mathbb{R}^{T \times H \times W \times C}$ as input, where C refers to the number of channels, T is the temporal duration and H and W denote the height and width in the spatial domain, we first perform 3D convolution (a convolution or a set of convolutions) operation on the input signal to extract spatial-temporal features and generate an intermediate feature map $F' \in \mathbb{R}^{T' \times H' \times W' \times C'}$. Kernels of a 3D convolutional layer can be represented as a 4D tensor $\mathcal{K} \in \mathbb{R}^{n_k \times t_k \times h_k \times w_k}$ (we omit the channel dimension for simplicity), where n_k is the number of kernels, t_k is the temporal depth of kernel and h_k and w_k are the kernel size in the spatial domain. The process of 3D convolution can be formulated as:

$$\begin{aligned}
 F' &= \mathcal{K} * F, \text{ where} \\
 F'_{x,y,z} &= \left[f_{x,y,z}^1, f_{x,y,z}^2, \dots, f_{x,y,z}^{n_k} \right]^T, \\
 f_{x,y,z}^n &= \sum_{t=0}^{t_k} \sum_{h=0}^{h_k} \sum_{w=0}^{w_k} \mathcal{K}_{t,h,w}^n \cdot F_{(x+t)(y+h)(z+w)}. \tag{1}
 \end{aligned}$$

Here $*$ denotes convolution, $\mathcal{K}_{t,h,w}^n$ denotes the value at (t, h, w) of p th filter, $F_{(x+t)(y+h)(z+w)}$ represents the values that start from the position (x, y, z) in F and have the same

size as the kernel \mathcal{K}^n . $f_{x,y,z}^n$ denotes the value at (x, y, z) on the n th output feature map.

For each sliced tensor $q_t \in \mathbb{R}^{H' \times W' \times C'}$ in F' , q_t represents the sliced tensor of intermediate feature map F' from time t to time $t+1$ and $t \in (0, T - 1)$. We sequentially add a channel attention module and a spatial attention module to infer a channel attention map M_c and a spatial attention map M_s , illustrated in Fig. 1. Finally, the attention maps are sequentially multiplied to the sliced tensor to reweight the output of each 3D RAN module. The attention process of a sliced tensor q_t can be expressed as [38]:

$$q'_t = M_c(q_t) \otimes q_t, \tag{2}$$

$$q''_t = M_s(q'_t) \otimes q'_t, \tag{3}$$

where \otimes refers to element-wise multiplication. q'_t is the channel attention output and q''_t is the final refined output. For simplicity, we only discuss the specific computation process of attention maps for a sliced tensor $q_t \in \mathbb{R}^{H' \times W' \times C'}$ in Sects. 3.1.1 and 3.1.2. Other sliced tensors repeat this process.

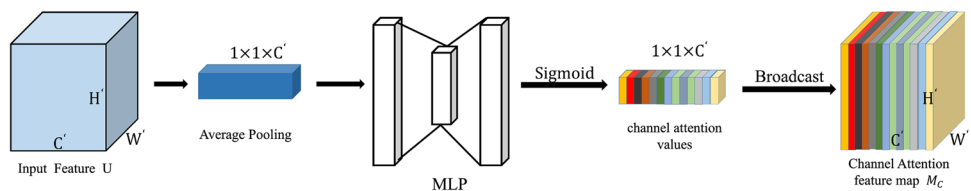
3.1.1 Channel attention module

We infer a channel attention map by utilizing the relationships within the feature channels. Channel attention focuses on what are the meaningful channels related to output target. Our goal is to improve the learning ability of the network by reweighting each channel signal in the intermediate feature maps. Figure 2 depicts the specific computation process of channel attention map for a sliced tensor $U \in \mathbb{R}^{H' \times W' \times C'}$ in the intermediate feature map (we use U instead of q_t for simplicity).

In order to capture the channel attention map efficiently in each sliced tensor, we first squeeze the spatial dimension $H' \times W'$ of the tensor to generate a channel descriptor F , which represents average-pooled feature [37]. This is achieved by using the global average pooling operation. The c -element of F is computed as:

$$F_c = \frac{1}{H' \times W'} \sum_{i=1}^{H'} \sum_{j=1}^{W'} U_c(i, j). \tag{4}$$

Fig. 2 Diagram of channel attention submodule in each sliced tensor of the intermediate feature map. The channel attention submodule uses average pooling outputs along spatial axis with MLP network to generate channel attention



The channel descriptor is then forwarded to a multi-layer perceptron (MLP) with one hidden layer to fully capture channel-wise dependencies. To limit model complexity and reduce the number of parameters, the hidden activation layer size is set to $\mathbb{R}^{1 \times 1 \times C'/r}$, where r is reduction ratio and usually sets to 16 for the best performance [38]. In short, the overall channel attention is summarized as:

$$M_C(F) = B_s(\sigma(MLP(F))) = B_s(\sigma(W_1(\delta(W_0F))), \tag{5}$$

where σ and δ separately refer to the sigmoid and ReLU function, $W_0 \in \mathbb{R}^{C'/r \times C'}$ and $W_1 \in \mathbb{R}^{C' \times C'/r}$. Note that W_0 and W_1 are the weights of MLP. B_s denotes broadcast channel attention values along the spatial dimension. Then, we use channel-wise multiplication between the feature map U and the $M_c(F)$ to get the channel attention feature map.

3.1.2 Spatial attention module

We infer a spatial attention map by utilizing spatial relationships of features. Different from the channel attention, the spatial attention focuses on where we need to pay more attention in an intermediate map. Figure 3 depicts the specific computation process of spatial attention map for a channel refined feature.

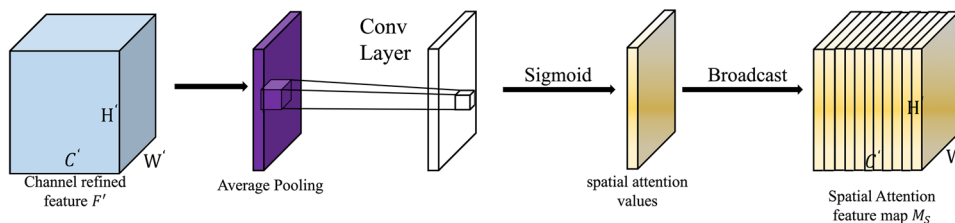
In order to compute spatial attention feature map efficiently, we first squeeze the channel information of feature map to generate a 2D spatial descriptor $H \in \mathbb{R}^{H' \times W' \times 1}$. This is achieved by using the global average pooling operation. Using pooling operating along the channel axis is shown to be effective in highlighting informative regions [43]. Elements at coordinates (i, j) of H are computed as:

$$H_{i,j} = \frac{1}{C'} \sum_{k=1}^{C'} F'_{i,j}(k). \tag{6}$$

We then use a convolution layer to infer a spatial attention map $M_s(F) \in \mathbb{R}^{H' \times W' \times C'}$, which encodes where to emphasize and where to suppress. The detail process is summarized as follows:

$$\begin{aligned} M_S(F) &= B_c\left(\sigma\left(f^{7 \times 7}(\text{Avg Pool}(F'))\right)\right) \\ &= B_c\left(\sigma\left(f^{7 \times 7}(H)\right)\right), \end{aligned} \tag{7}$$

Fig. 3 Diagram of spatial attention submodule in a sliced tensor of the intermediate feature map. The spatial submodule uses average pooling outputs along channel axis and forwards them to convolutional layer to generate spatial attention



where σ refers to the sigmoid function and $f^{7 \times 7}$ denotes a convolutional operation with the kernel size of 7×7 . B_c denotes broadcast spatial attention values along the channel dimension. Then, we use element-wise multiplication between the channel refined feature F' and the $M_s(F)$ to reweight each pixel value and get the spatial refined feature map.

Note Two attention modules, channel and spatial, can be placed in various manners: parallel or sequentially manner. We opt for simplest but the most effective, sequential channel—spatial. The effect of different module placement manners is demonstrated in Sect. 4.2.

3.2 Network architecture

After introducing the 3D RAN modules, we show the original 3D ResNet-34 [46] and our 3D RAN-34 architecture specifications in Table 1. For simplicity, we omit the batch normalization [44] layer and ReLU layer in the network architectures. Each network uses clips with the size of 3 channels \times 16 frames \times 112 pixels \times 112 pixels as input to keep balance between model capacity and processing efficiency. A spatial down-sampling is performed at *Conv1_X* with a stride of $1 \times 2 \times 2$. Then a max pooling layer before *Conv2_X* with a stride of $2 \times 2 \times 2$ is also applied for down-sampling, and three spatiotemporal down-samplings are performed at *Conv3_X*, *Conv4_X* and *Conv5_X* with a stride of $2 \times 2 \times 2$. When the number of feature maps increased, we use projection shortcut to match dimension. The difference between our networks and original 3D ResNets is that we add some fully connected and convolutional layers after the last 3D convolution layer of each module.

3.3 Implementation

Training and evaluation We use stochastic gradient descent (SGD) with momentum of 0.9 to train our network models on Kinetics training set from scratch. Initial learning rate is 0.1 and is divided by 10 after the validation loss saturates. For all datasets, the dropout ratio and weight decay rate are set to 0.5 and $10e^{-3}$, respectively. The optimization is done at 150 epochs.

Table 1 3D RAN architecture details for the Kinetics dataset: convolution residual blocks are shown in external brackets, next to the number of times each block is repeated in the stack

Layer name	Output size	Architecture	
		3D ResNet-34	3D RAN-34
conv1	16 × 56 × 56	7 × 7 × 7, 64, <i>stride</i> 1 × 2 × 2	
conv2_x	8 × 28 × 28	3 × 3 × 3, <i>maxpool</i> , <i>stride</i> 2	
		$\left[\begin{matrix} \text{conv}, 3 \times 3 \times 3, 64 \\ \text{conv}, 3 \times 3 \times 3, 64 \\ \text{conv}, 3 \times 3 \times 3, 64 \end{matrix} \right] \times 3$	$\left[\begin{matrix} \text{conv}, 3 \times 3 \times 3, 64 \\ \text{conv}, 3 \times 3 \times 3, 64 \\ \left[\begin{matrix} \text{fc}, 16, 64 \\ \text{conv}, 7 \times 7 \end{matrix} \right] \times 8 \end{matrix} \right] \times 3$
conv3_x	4 × 14 × 14	$\left[\begin{matrix} \text{conv}, 3 \times 3 \times 3, 128 \\ \text{conv}, 3 \times 3 \times 3, 128 \end{matrix} \right] \times 4$	$\left[\begin{matrix} \text{conv}, 3 \times 3 \times 3, 128 \\ \text{conv}, 3 \times 3 \times 3, 128 \\ \left[\begin{matrix} \text{fc}, 16, 128 \\ \text{conv}, 7 \times 7 \end{matrix} \right] \times 4 \end{matrix} \right] \times 4$
conv4_x	2 × 4 × 7	$\left[\begin{matrix} \text{conv}, 3 \times 3 \times 3, 256 \\ \text{conv}, 3 \times 3 \times 3, 256 \end{matrix} \right] \times 6$	$\left[\begin{matrix} \text{conv}, 3 \times 3 \times 3, 256 \\ \text{conv}, 3 \times 3 \times 3, 256 \\ \left[\begin{matrix} \text{fc}, 16, 256 \\ \text{conv}, 7 \times 7 \end{matrix} \right] \times 2 \end{matrix} \right] \times 6$
conv5_x	1 × 4 × 4	$\left[\begin{matrix} \text{conv}, 3 \times 3 \times 3, 512 \\ \text{conv}, 3 \times 3 \times 3, 512 \end{matrix} \right] \times 3$	$\left[\begin{matrix} \text{conv}, 3 \times 3 \times 3, 512 \\ \text{conv}, 3 \times 3 \times 3, 512 \\ \left[\begin{matrix} \text{fc}, 16, 512 \\ \text{conv}, 7 \times 7 \end{matrix} \right] \times 1 \end{matrix} \right] \times 3$
	1 × 1 × 1	Global average pool, 400-d fully connected, softmax	

Attention blocks (channel + spatial) are shown in inner brackets, next to the number of each inner block is parallel place in the stack. The values followed by fc and conv indicate the output dimension of two fully connected layers and convolution kernel size

During training, we will perform data augmentation for all training datasets to enhance the perform of network architectures. Our data augmentation includes temporal sampling, random clipping, brightness and contrast adjustment [45]. We first select the temporal location of a sample frame, and we randomly select the remaining 15 frames around the selected frame. If the videos are not enough, we can loop the videos many times until reaching 16 frames. Next, we use random cropping strategy which selects a spatial position from 4 corners and 1 center. In addition to the positions, we use multi-scale cropping methods with scales selected from to train our networks. The procedure is similar to [45]. Finally, we spatially resize each frame to 112 × 112 pixels. All operations are consistent across all frames in each training clip.

During evaluation, we generate test clips (16-frame clips) by sliding window manner on kinetics validation set. Each clip uses spatially cropped around center position with scale 1. We use trained network to evaluate each clip in validation set and get the class scores. The maximum recognition score denotes the corresponding class label.

4 Experiments

4.1 Dataset

We evaluate our models on three well-known benchmark datasets: UCF-101 [22], HMDB-51 [23] and Kinetics [24].

UCF101 is a realistic action videos database, collected from YouTube, with 13320 short videos from 101 different categories. The action categories can be divided into five types: (1) Human–Object interaction, (2) Body-Motion Only, (3) Human–Human Interaction, (4) Playing Musical Instruments, and (5) Sports. The average length of each video is 7 seconds. This dataset is random spilt into three subdatasets, 70% of which are used to train and 30% for testing.

HMDB-51 was released by Brown University, most of which comes from movies and some from public databases and online video libraries such as YouTube. The dataset contains 6766 videos, divided into 55 different categories, each of which contains at least 101 samples. Similar to UCF-101, the videos were temporally trimmed. This dataset provides 3 subdatasets, 70% of which are used to train 30% for testing.

Kinetics contains approximately 300,000 video clips from 400 different categories. Each clip is about 10 seconds long and is tagged with an action category. All clips are subject to multiple rounds of manual annotation, so the quality of annotation is extremely high. These actions include a wide range of human–object interactions and human–human interactions.

4.2 Ablation studies

Arrangement of the attention submodules In this experiment, we verify the effectiveness of the basic network with different ways of arranging attention submodules. The design

of proposed network mentioned above can be split into two steps: We first infer and add the channel to attention submodule and then the same to spatial attention submodule. Except this manner, we also could first place a spatial attention submodule and then a channel attention submodule or two submodules can be added in a parallel. We compare different ways of adding the channel and the spatial attention submodules: single channel, single spatial, sequential channel–spatial, sequential spatial–channel and parallel use two attention submodules.

We use 3D ResNet-34 as the basic network architecture. Hara et al. [46] showed Kinetics dataset is big enough to train 3D ResNet-34 without over-fitting; thus, all networks are trained at kinetics dataset from scratch using its training and validation datasets. Table 2 shows the comparison results of using different attention submodules. From these results, we can find that the accuracy of using the single channel attention submodule is better than using the single spatial attention submodule and both higher than the original network. We can also observe that adding channel attention maps and spatial attention maps simultaneously could further increase performance. Obviously, the order of arranging channel and spatial submodules may affect the performance of overall network. Adding feature maps in sequence can achieve better performance than doing in parallel. In addition, the channel first order could get the best performance.

By comparing the experimental results in Table 2, we choose to arrange the channel and spatial submodules sequentially as our final module design, as shown in Fig. 1. Our final module (3D RAN) outperforms benchmark network (3D ResNet-34) by a certain margin with a 1.6% improvement on top-1 accuracy and a 1.3% improvement on top-5 accuracy, as shown in Table 2.

Comparison with the Baseline 3D CNN on Kinetics We compare the 3D RANs against 3D ResNets with different network depths. All networks are trained on the Kinetics datasets from scratch. As shown in Table 3, the 3D RANs consistently improved the performance of action recognition separately under different depths, demonstrating that introducing attention mechanism to 3D ResNets works well on Kinetics.

Particularly, the 3D ResNet-34 has achieved validation accuracy over top-1 of 61.7% and top-5 of 83.2% and even outperforms the deeper ResNet-50 network (61.3% over top-1 accuracy and 83.1 top-5 accuracy) with very fewer parameters. We can also see that accuracy over top-1 and top-5 increases with the raise in network depth. This result supports that Kinetics dataset is sufficiently large for training 3D CNNs, just like ImageNet dataset for 2D CNNs.

Note We can find that attention modules can improve network performance at minimal addition parameters.

Comparison with the Baseline 3D CNN on UCF-101 and HMDB-51. We further compare our proposed 3D RANs

Table 2 Action recognition accuracy rates (%) for different ways of arranging attention submodules on the kinetics validation set

Method	Params	Top-1	Top-5	Average
3D ResNet-34 [46]	63.72M	60.1	81.9	71.0
3D ResNet-34 + channel	63.96M	61.2 _(1.1)	82.6 _(0.7)	71.9 _(0.9)
3D ResNet-34 + spatial	63.73M	61.0 _(0.9)	82.4 _(0.5)	71.7 _(0.7)
3D ResNet-34 + channel + spatial	63.97M	61.7 _(1.6)	83.2 _(1.3)	72.5 _(1.5)
3D ResNet-34 + spatial + channel	63.97M	61.5 _(1.4)	83.0 _(1.1)	72.3 _(1.3)
3D ResNet-34 + channel & spatial in parallel	63.97M	61.4 _(1.3)	82.9 _(1.0)	72.2 _(1.2)

It can be observed that 3D ResNet with sequential channel and spatial attention submodule is the best combining strategy. The numbers in brackets denote the performance improvement over the basic network

Table 3 Accuracy rates (%) on Kinetics validation set with data augmentation

	Original				3D RAN			
	Params	Top-1	Top-5	Average	Params	Top-1	Top-5	Average
3D ResNet-18 [46]	33.41M	54.2	78.1	66.1	33.53M	55.3 _(1.1)	79.0 _(0.9)	67.2 _(1.1)
3D ResNet-34 [46]	63.72M	60.1	81.9	71.0	63.97M	61.7 _(1.6)	83.2 _(1.3)	72.5 _(1.5)
3D ResNet-50 [46]	47.02M	61.3	83.1	72.2	50.89M	62.9 _(1.6)	84.1 _(1.0)	73.5 _(1.3)
3D ResNet-101 [46]	86.06M	62.8	83.9	73.3	94.39M	63.3 _(0.5)	84.5 _(0.6)	73.9 _(0.6)
3D ResNet-152 [46]	118.22M	63.0	84.4	73.7	130.49M	63.5 _(0.5)	84.8 _(0.4)	74.2 _(0.5)
3D ResNet-200 [46]	127.42M	63.1	84.4	73.7	141.78M	63.7 _(0.6)	84.8 _(0.4)	74.3 _(0.6)

Average is averaged accuracy over top-1 and top-5. The original column indicates the results experimented in original papers. The 3D RAN column indicates the experiment results after we added the attention module in original networks. The numbers in brackets denote the performance improvement over the original networks

Table 4 Top-1 accuracy rates (%) on UCF-101 and HMDB-51 datasets with data augmentation

Method	Pre-trained	Original		3D RAN	
		UCF-101	HMDB-51	UCF-101	HMDB-51
3D ResNet-18	None (scratch)	42.4	17.1	47.6	21.3
3D ResNet-18	Kinetics	84.4	56.4	86.7	58.3
3D ResNet-34	Kinetics	87.7	59.1	88.6	60.5
3D ResNet-50	Kinetics	89.3	61.0	89.9	61.8
3D ResNet-101	Kinetics	88.9	61.7	89.4	62.4
3D ResNet-152	Kinetics	89.6	62.4	90.5	63.4
3D ResNet-200	Kinetics	89.6	63.5	91.0	64.7

All accuracies are averaged over three splits

with advanced methods on two common datasets, UCF-101 and HMDB-51 datasets. According to previous experiments, since the parameters of 3D CNNs are far more than 2D counterparts, training them in a relatively small data set will lead to over-fitting problems and have lower performance compared to 2D CNNs pre-trained in large-scale datasets, such as ImageNet. Specifically, we use 3D ResNet-18 and 3D RAN-18 as our test model, which are the shallowest model in all modules, and we trained these two models from scratch on UCF-101 and HMDB-51, respectively. Table 4 reports the comparison result in terms of accuracy over top-1. It can be seen that both ResNet-18 and RAN-18 pre-trained on kinetics obviously outperformed counterparts trained on UCF-101 and HMDB-51 from scratch. These results show that the network has suffered seriously over-fitting problems when they trained from scratch in UCF-101 and HMDB-51 datasets. So, we trained our model in Kinetics datasets and fine-tune on UCF-101 and HMDB-51 datasets, respectively. We can also notice from Table 4 that the recognition performance gradually increases when the layers increase. At the same time, 3D RANs consistently outperform all baselines networks significantly across different depth. Moreover, unlike the results of training on the Kinetics dataset in Table 3, the RAN-200 still improves recognition accuracy on these two datasets. We think this is because the fine-tuning only trained

the full connected layer, and the number of parameters for pre-trained is the same from RAN-50 to RAN-200. These results show that the pre-trained early layers of RAN-200 are more suitable for UCF-101 and HMDB-51 datasets.

4.3 Comparison with the state-of-the-art methods

Table 5 shows the accuracy of our 3D RAN-200, which achieved best performance on both datasets when compared with other state-of-the-art network architectures. Our 3D RANs capture spatial-temporal information using only RGB frames as input. For fairness, all networks in Table 5 use only RGB frames as input, which is reported by these works. The results are achieved by using inputs at length of 16 frames. Simultaneously, for 3D networks, we pre-trained on the Kinetics dataset, and for 2D networks, we pre-trained on the ImageNet dataset. Here, we can see that RAN-200 also achieves the best performance compared with C3D, P3D, two-stream CNN and TDD. In particular, we can see TSN and two-stream I3D, which use optical flow and RGB frames as input, achieved higher accuracy. We believe that the time domain information provided by optical flow directly is more than we extracted through 3D convolution, but it is time-consuming to train two networks and calculate optical flow. Based on these results, we can draw a conclusion that our pro-

Table 5 Action recognition accuracy rates (%) comparison with the state of the art on UCF-101 and HMDB-51 datasets with only RGB frames as input

Method	Dim	Pre-trained	UCF-101	HMDB-51
RAN-200	3D	Kinetics	91.0	64.7
ResNeXt-101 [47]	3D	Kinetics	90.7	63.8
C3D [1]	3D	Kinetics	82.3	–
P3D ResNet [48]	3D	Kinetics	88.6	–
MiCT-Net [4]	3D/2D	Kinetics+ImageNet	88.9	63.8
Two-stream I3D [3]	3D	Kinetics+ImageNet	98.0	80.7
Two-stream CNN [13]	2D	ImageNet	88.0	59.4
TDD [4]	2D	ImageNet	90.3	63.2
TSN [6]	2D	ImageNet	94.0	68.5

All accuracies are averaged over three splits. Dim denotes the dimension of convolution kernel

posed 3D RAN significantly promoted the research of video classification on multiple benchmark datasets.

4.4 Visualization

In order to understand the role of the attention mechanism more intuitively, we apply the Grad-CAM to basic network architectures (3D ResNet-34 and 3D RAN-34) to visualize some video sequences from the UCF101 validation set in Fig. 4. The frames are selected from the long video sequence. From the Grad-CAM mask which covers the object regions in the input, we can clearly see that they are important

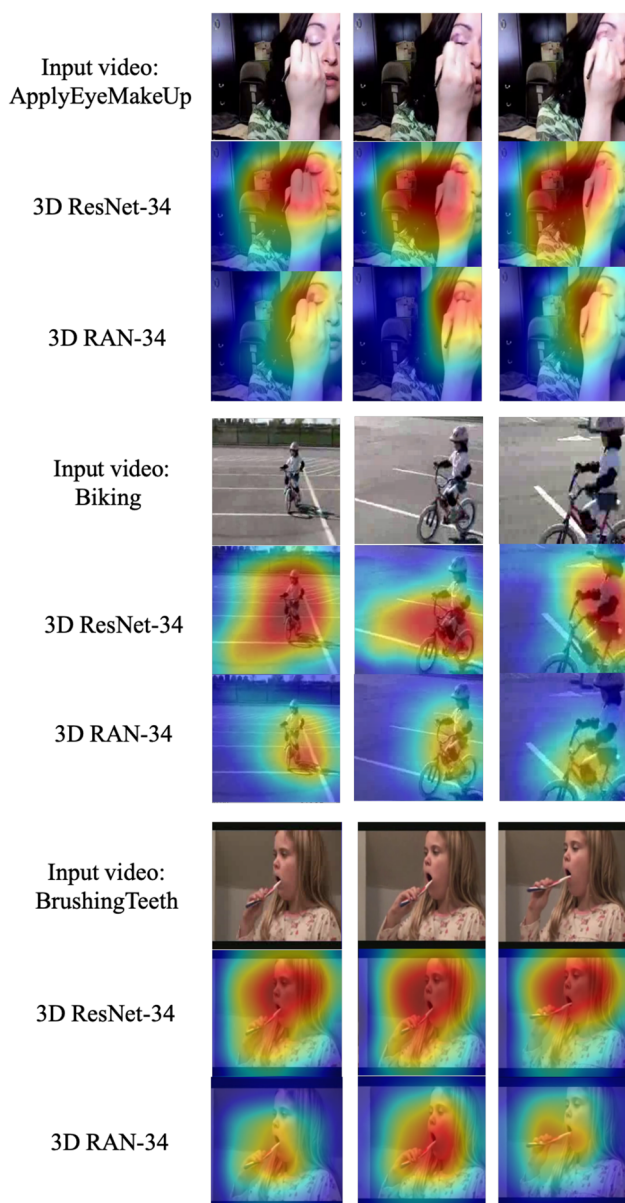


Fig. 4 Examples of visualization results of 3D ResNet-34 and 3D RAN-34 on the UCF101 validation set

regions for predictions. We can also notice that, compared to 3D ResNet-34, 3D RAN-34 generates more accurate mask regions for prediction.

5 Conclusion

In this paper, we propose the 3D Residual Attention Networks (3D RANs) by introducing the attention mechanism into residual networks (ResNets). The benefits of our network are that it can significantly improve the capacity of capturing spatiotemporal information. Extensive experiments demonstrate that our 3D RAN outperforms traditional 3D ResNets on Kinetics dataset and other state-of-the-art methods on both UCF-101 and HMDB-51 datasets with RGB input.

One explanation of our network that could obtain great performance improvements for action recognition is that our network could learn what and where to emphasize or suppress. We reweight each channel and pixel of the intermediate feature map. This allows the system to focus more on finding useful information in the input data. In this way, we can enhance the representation of the network. In our future work, we will transfer our networks to other video-related tasks.

Acknowledgements This work was supported in part by the 2016 Guangzhou Innovation and Entrepreneurship Leader Team under Grant CXLJTD-201608 and the Development Research Institute of Guangzhou Smart City.

References

1. Tran, D., Bourdev, L., Fergus, R., Torresani, L., Paluri, M.: Learning spatiotemporal features with 3d convolutional networks. In: Proceedings of the IEEE International Conference on Computer Vision, pp. 4489–4497 (2015)
2. Li, Y., Wang, Z., Yang, X., Wang, M., Poiana, S.I., Chaudhry, E., Zhang, J.: Efficient convolutional hierarchical autoencoder for human motion prediction. *Vis. Comput.* **35**, 1143–1156 (2019)
3. Carreira, J., Zisserman, A.: Quo vadis, action recognition? a new model and the kinetics dataset. In: IEEE Conference on Computer Vision and Pattern Recognition (CVPR), pp. 4724–4733 (2017)
4. Wang, L., Qiao, Y., Tang, X.: Action recognition with trajectory-pooled deep-convolutional descriptors. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 4305–4314 (2015)
5. Wang, X., Farhadi, A., Gupta, A.: Actions~ transformations. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 2658–2667 (2016)
6. Wang, L., Xiong, Y., Wang, Z., Qiao, Y., Lin, D., Tang, X., Van Gool, L.: Temporal segment networks: towards good practices for deep action recognition. In: European Conference on Computer Vision, pp. 20–36 (2016)
7. Karpathy, A., Toderici, G., Shetty, S., Leung, T., Sukthankar, R., Fei-Fei, L.: Large-scale video classification with convolutional neural networks. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 1725–1732 (2014)
8. Zhang, B., Wang, L., Wang, Z., Qiao, Y., Wang, H.: Real-time action recognition with enhanced motion vector CNNs. In: Pro-

- ceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 2718–2726 (2015)
9. Wang, H., Schmid, C.: Action recognition with improved trajectories. In: Proceedings of the IEEE International Conference on Computer Vision, pp. 3551–3558 (2013)
 10. Scovanner, P., Ali, S., Shah, M.: A 3-dimensional sift descriptor and its application to action recognition. In: Proceedings of the 15th ACM International Conference on Multimedia, pp. 357–360 (2007)
 11. Wang, H., Kläser, A., Schmid, C., Liu, C. L.: Action recognition by dense trajectories. In: IEEE Conference on Computer Vision and Pattern Recognition (CVPR), pp. 3169–3176 (2011)
 12. Laptev, I., Marszalek, M., Schmid, C., Rozenfeld, B.: Learning realistic human actions from movies. In: IEEE Conference on Computer Vision and Pattern Recognition. CVPR, pp. 1–8 (2008)
 13. Simonyan, K., Zisserman, A.: Two-stream convolutional networks for action recognition in videos. *Adv. Neural Inf. Process. Syst.* **27**, 568–576 (2014)
 14. Ji, S., Xu, W., Yang, M., Yu, K.: 3D convolutional neural networks for human action recognition. *IEEE Trans. Pattern Anal. Mach. Intell.* **35**, 221–231 (2013)
 15. Wang, Y., Jiang, L., Yang, M. H., Li, L. J., Long, M., Fei-Fei, L.: Eidetic 3D LSTM: A Model for Video Prediction and Beyond (2013)
 16. Ma, Z., Sun, Z.: Time-varying LSTM networks for action recognition. *Multimed. Tools Appl.* **77**, 32275–32285 (2018)
 17. Liang, D., Liang, H., Yu, Z., Zhang, Y.: Deep convolutional BiLSTM fusion network for facial expression recognition. *Vis. Comput* (2019). <https://doi.org/10.1007/s00371-019-01636-3>
 18. Hara, K., Kataoka, H., Satoh, Y.: Learning spatio-temporal features with 3D residual networks for action recognition. In: Proceedings of the ICCV Workshop on Action, Gesture, and Emotion Recognition, pp. 4 (2017)
 19. Nair, V., Hinton, G. E.: Rectified linear units improve restricted boltzmann machines. In: Proceedings of the 27th International Conference on Machine Learning (ICML-10), pp. 807–814 (2010)
 20. Ba, J., Mnih, V., Kavukcuoglu, K.: Multiple object recognition with visual attention (2014)
 21. Mnih, V., Heess, N., Graves, A.: Recurrent models of visual attention. *Adv. Neural Inf. Process. Syst.* **27**, 2204–2212 (2014)
 22. Soomro, K., Zamir, A.R., Shah, M.: UCF101: A dataset of 101 human actions classes from videos in the wild (2012)
 23. Kuehne, H., Jhuang, H., Stiefelhagen, R., Serre, T.: Hmdb51: a large video database for human motion recognition. *High Perform. Comput. Sci. Eng.* **12**, 571–582 (2013)
 24. Kay, W., Carreira, J., Simonyan, K., Zhang, B., Hillier, C., Vijayanarasimhan, S., Suleyman, M.: The kinetics human action video dataset (2017)
 25. He, K., Zhang, X., Ren, S., Sun, J.: Deep residual learning for image recognition. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 770–778 (2016)
 26. Feichtenhofer, C., Pinz, A., Zisserman, A.: Convolutional two-stream network fusion for video action recognition. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 1933–1941 (2016)
 27. Li, W., Nie, W., Su, Y.: Human action recognition based on selected spatio-temporal features via bidirectional LSTM. In: IEEE Access, pp. 44211–44220 (2018)
 28. Song, S., Lan, C., Xing, J., Zeng, W., Liu, J.: Spatio-temporal attention based LSTM networks for 3D action recognition and detection. *IEEE Trans. Image Process.* **27**, 3459–3471 (2018)
 29. Itti, L., Koch, C., Niebur, E.: A model of saliency-based visual attention for rapid scene analysis. *IEEE Trans. Pattern Anal. Mach. Intell.* **11**, 1254–1259 (1998)
 30. Rensink, R.A.: The dynamic representation of scenes. *Vis. Cognit.* **7**, 17–42 (2000)
 31. Corbetta, M., Shulman, G.L.: Control of goal-directed and stimulus-driven attention in the brain. *Nat. Rev. Neurosci.* **3**, 201 (2002)
 32. Larochelle, H., Hinton, G.E.: Learning to combine foveal glimpses with a third-order Boltzmann machine. *Adv. Neural Inf. Process. Syst.* **23**, 1243–1251 (2010)
 33. Olshausen, B. A., Anderson, C. H., Van Essen, D. C.: A neurobiological model of visual attention and invariant pattern recognition based on dynamic routing of information. In: Proceedings of the IEEE International Conference on Computer Vision, pp. 4700–4719 (1993)
 34. Cao, C., Liu, X., Yang, Y., Yu, Y., Wang, J., Wang, Z., Ramanan, D.: Look and think twice: capturing top-down visual attention with feedback convolutional neural networks. In: Proceedings of the IEEE International Conference on Computer Vision, pp. 2956–2964 (2015)
 35. Jaderberg, M., Simonyan, K., Zisserman, A.: Recurrent spatial transformer networks. In: *Computer Science*, (2015)
 36. Wang, F., Jiang, M., Qian, C., Yang, S., Li, C., Zhang, H., Tang, X.: Residual attention network for image classification. In: *Computer Vision and Pattern Recognition*, pp. 6450–6458 (2017)
 37. Hu, J., Shen, L., Sun, G.: Squeeze-and-excitation networks (2017)
 38. Woo, S., Park, J., Lee, J. Y., Kweon, I. S.: CBAM: Convolutional Block Attention Module. In: *Proceedings of European Conference on Computer Vision* (2018)
 39. Sharma, S., Kiros, R., Salakhutdinov, R.: Action recognition using visual attention. In: *Computer Science* (2015)
 40. Kim, D. , Cho, D. , Kweon, I. S.: Self-supervised video representation learning with space-time cubic puzzles. *arXiv preprint arXiv:1811.09795* (2018)
 41. Wang, X., Girshick, R., Gupta, A., He, K.: Non-local neural networks. In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 7794–7803 (2018)
 42. Hinton, G. E.: Rectified linear units improve restricted Boltzmann machines Vinod Nair (2010)
 43. Zhou, B., Khosla, A., Lapedriza, A., Oliva, A., Torralba, A.: Learning deep features for discriminative localization. In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 2921–2929 (2016)
 44. Ioffe, S., Szegedy, C.: Batch normalization: accelerating deep network training by reducing internal covariate shift. pp. 448–456 (2015)
 45. Wang, L., Xiong, Y., Wang, Z., Qiao, Y.: Towards good practices for very deep two-stream ConvNets. In: *Computer Science* (2015)
 46. Hara, K., Kataoka, H., Satoh, Y.: Can spatiotemporal 3D CNNs retrace the history of 2D CNNs and ImageNet?. In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 18–22 (2017)
 47. Xie, S., Girshick, R., Dollár, P., Tu, Z., He, K.: Aggregated Residual transformations for deep neural networks. In: *IEEE Conference on Computer Vision and Pattern Recognition*, pp. 5987–5995 (2017)
 48. Qiu, Z., Yao, T., Mei, T.: Learning spatio-temporal representation with Pseudo-3D residual networks. In: *2017 IEEE International Conference on Computer Vision (ICCV)*, pp. 5534–5542 (2017)
 49. Zhou, Y., Sun, X., Zha, Z. J., Zeng, W.: MiCT: mixed 3D/2D convolutional tube for human action recognition. In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 449–458 (2018)



Jiahui Cai received his bachelor's degree from the University of Wuhan Institute of Technology, Wuhan, China, in 2016. He is currently pursuing the master's degree at the University of Sun Yat-sen, Guangzhou, China. His research interest includes computer vision, machine learning and deep learning.



Jianguo Hu received the B.S. and M.S. degrees in National University of Defense Technology, in 2000 and 2004, respectively, and Ph.D. degree in Sun Yat-sen University, Guangzhou, China, in 2010. He is currently a professor in the Sun Yat-sen University. And he is also the director of Development Research Institute of Guangzhou Smart City. He is a Scientific and Technological Innovation Leader of the Guangdong Special Branch Project. He is also an Innovative Team Leader in Guangzhou City. His research interests include mixed-signal ICs, RFID, Internet of things and artificial intelligence.