Fuzzy Systems and Data Mining VI
A.J. Tallón-Ballesteros (Ed.)

doi:10.3233/FAIA200754

Hierarchy Spatial-Temporal Transformer for Action Recognition in Short Videos

Guoyong Cai, Yumeng Cai[1]
GuangXi Key Laboratory of Trusted Software, Guilin University of Electronic Technology, Guilin, China

Abstract. Short videos action recognition based on deep learning has made a series of important progress; most of the proposed methods are based on 3D Convolution neural networks (3D CNN) and Two Stream architecture. However, 3D CNN has a large number of parameters and Two Stream networks cannot learn features well enough. This work aims to build a network to learn better features and reduce the scale of parameters. A Hierarchy Spatial-Temporal Transformer model is proposed, which is based on Two Stream architecture and hierarchy inference. The model is divided into three modules: Hierarchy Residual Reformer, Spatial Attention Module, and Temporal-Spatial Attention Module. In the model, each frame's image is firstly transformed into a spatial visual feature map. Secondly, spatial feature learning is performed by spatial attention to generating attention spatial feature maps. Finally, the generated attention spatial feature map is incorporated with temporal feature vectors to generate a final representation for classification experiments. Experiment results in the hmdb51 and ucf101 data set showed that the proposed model achieved better accuracy than the state-of-art baseline models

Keywords. Spatial feature learning; hierarchy inference; spatial-temporal attention

1. Introduction

Video learning, which goal is to learn the content feature in each frame, consists of various tasks, such as object tracking, temporal action localization, action recognition, and et al. Action recognition in video learning has been researched for years but still a challenging task. Compared to single image recognition, temporal information between video frames must be considered for action recognition [1]; the computing efficiency must be considered due to the real-time property of video streams.

Recently, many researchers have developed novel methods for action recognition by using 3D Convolution Neural Network [2] or by Two Stream Convolution Neural Network(2S CNN) [3], which incorporate spatial and temporal CNN to extract spatial and temporal feature from each video frame.

[1] Corresponding Author: Yumeng Cai, GuangXi Key Laboratory of Trusted Software, Guilin University of Electronic Technology, Guilin, China. Email: cymqqqq@gmail.com

3D CNN can learn the features well in the video, but it has large scale parameters and is low efficient; 2S CNN can not learn the spatial and temporal feature in video accurately. Some works have explored methods of using spatial-temporal relation information, such as the C3D network [4], R3D network [5], but they still can not learn video features well.

To address the problem of reducing the parameter scale of 3D CNN, the paper used the mask network to avoid redundancy computation when learning spatial features in continuous frames.

To learn better features, the paper proposed to use the hierarchical inference principle designed in paper [6]. The hierarchy inference can calculate the posterior probability $p(y \mid x, z, w; \theta)$ under given initial prior knowledge $p_\theta(w)$ and conditional probability $p_\theta(z \mid w), p_\theta(y \mid x, z)$ in continuous states, where x,y,z,w is denoted be the feature map and θ represents the parameter of the network in this work. This will help us to design more useful learning models that can learn better features. The spatial feature of the image in the first frame can be taken as initial prior to infer the action context of the spatial features in the next frame.

This paper proposed a novel model, Hierarchy Spatial-Temporal Transformer, to learn the spatial-temporal related information based on Two Stream CNN and incorporates hierarchy inference. The main contributions of this paper are listed as follows.

(1). A Hierarchy Residual Reformer is proposed to calculate the spatial feature similarity between every two frames.

(2). A hierarchy residual network that incorporates the hierarchy inference method is proposed to calculate action feature similarity.

(3). Spatial attention and a hybrid spatial-temporal transformer are designed to learn spatial- temporal related information.

2. Related Works

Action recognition methods can be divided into three types: 3D CNN, 2S CNN, and the combination of 3D CNN and 2S CNN.

Baccouche and Ji et al. [7] first proposed 3D convolution for action recognition; and then Ji et al.[8] further proposed two techniques to exploit prior information for the 3D CNN. One is that the original image, image gradient, and optical flow of adjacent frames were all used as input of their network, and the other is that they also used 3D convolution to learn the motion information between each frame. Based on 3D convolution, Tran et al. [9] proposed spatial and temporal feature 3D convolution (C3D) for action recognition, and their experiments showed that the convolution network in the temporal direction works best when the size of the convolution kernel is 1x1x3. The reason is that C3D can handle multiple frames at a time. However, due to its scale of weight and parameters, C3D is difficult to train. To reduce the parameter scale and weight of C3D, Qiu et al. [10] decompose the 3D CNN into a 2D spatial

convolution and 1D temporal convolution. Then the two convolution networks are connected in series or parallel, and finally, a residual network is combined to construct a pseudo 3D (P3D) network. Based on the C3D network, Xu et al.[11] proposed Faster R-CNN to extract features from the input video clips, then extract candidate temporal information, and finally used Region of Interesting network(RoI)[12] to aggregate the features. Different from the C3D network, Faster R-CNN is mainly used for Object Detection. Shou et al.[13] proposed multi-stages CNN (MSCNN) for action recognition. Different from the above 3D networks, three different 3D convolution networks is proposed in MSCNN to perform different tasks.

Karpathy et al.[14] first proposed a 2S CNN for action recognition. One branch of the two-stream network inputs the spatial RGB image, and the other branch inputs the temporal-dimensional optical flow information of each frame. The final convolution features of the two branches are stitched together as the input into fully-connected layers to perform the action classification task. Based on a two-stream network, Zolfaghari et al. [15] proposed an efficient convolution network (ECO) for online video action recognition. Because the adjacent frames of the video have information redundancy, ECO samples several frames from the video. 2D convolution is used to extract spatial features of each frame, then the spatial features are stitched along the temporal dimension, and finally, the temporal relationship between each frame is captured by 3D convolution. The performance of ECO is better than 2S CNN. Based on 2S CNN, Tran et al. [16] proposed a residual two-stream network (ResNet (2+1)D) for action recognition. In their model, 3D convolution is decomposed integral into a 2D spatial convolution, and a 1D temporal convolution and spatial and temporal convolution are used to extract spatial features and temporal dimensional optical flow features respectively.

Shou et al.[17] proposed overlapping convolution networks (CDC) for action recognition, CDC incorporates 2S CNN and 3D CNN to obtain classification prediction score according to each frame. Then spatial convolution is used for down-sampling in the spatial dimension, and temporal-transposed convolution is used for up-sampling in the temporal dimension. Finally, spatial and temporal features are fused to classify actions.

Although C3D, P3D, and MSCNN networks have some improvements in performance, the computational efficiency is very low and the parameters scale is still large. The ECO, ResNet (2+1)D has some improvements in computational efficiency, but the ability to learn feature is still not enhanced.

3. Hierarchy Spatial-Temporal Transformer

In this section, the proposed model, the Hierarchy Spatial-Temporal Transformer, is presented and the overall architecture of the model is shown in figure 1. The hierarchy spatial-temporal transformer consists of three components: hierarchy residual reformer, spatial attention module, and spatial-temporal attention module. The Hierarchy residual reformer aims to transform short video into spatial visual feature maps and learn

temporal features. The spatial attention module incorporates an attention mechanism to perform spatial feature learning for producing spatial attention vector. The spatial-temporal attention module aims to fuse the spatial feature vector and temporal feature to produce final features. The details the components are given in the following section 3.1-3.3.

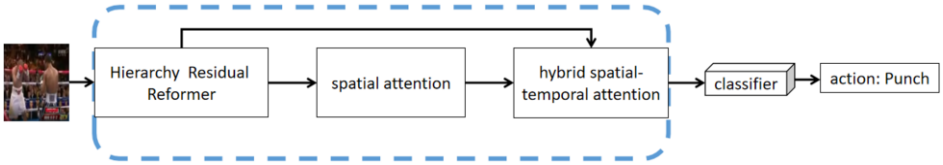

Figure 1 Overall Architecture of The Model

3.1Hierarchy Residual Reformer

In figure 2, the hierarchy residual reformer consists of three parts: temporal convolution, spatial-visual slide window, and hierarchy residual network. Temporal convolution is used to extract the optical flow feature of each frame. The spatial visual slide window contains a merge operation that is used to aggregate features of two frames and spatial visual normalization (SVN) module that is used to calculate spatial feature similarity of images in two frames. The hierarchy inference method is applied by the hierarchy residual network to calculate the action similarity of the spatial visual feature map. In figure 2, each video that the time is within thirty seconds is clipped into 16 frames, denoted by x1 to x16.

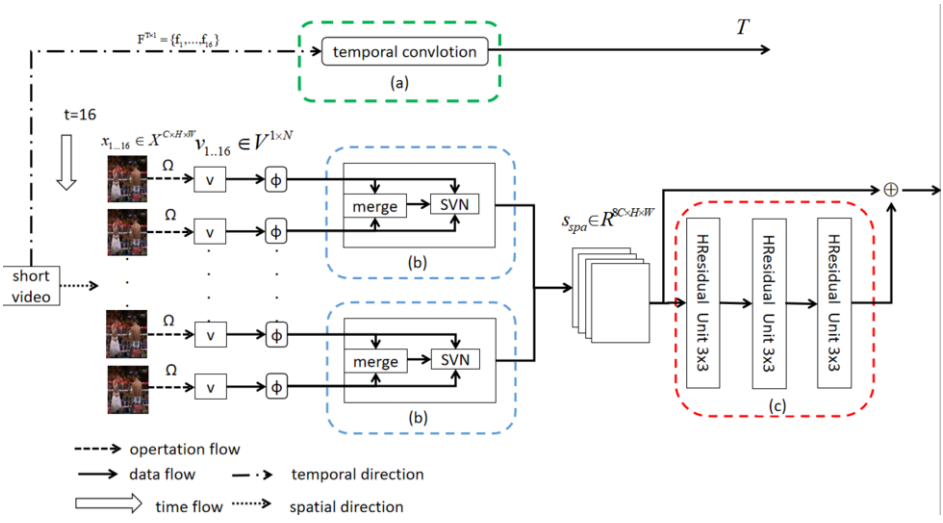

Figure 2 Hierarchy Residual Reformer

In this work, the short video is transformed by a map function along the spatial direction that is denoted by a 2D image of each frame $X^{C\times H\times W} = \{x_1,...,x_{16}\}$, where

C, H, W represents a channel, height, and width of the image, respectively. And the short video is learned by a temporal convolution along the temporal direction that is in the form of motion across the frames is denoted by continuous multi-frame optical flow $\mathrm{F}^{1\times\mathrm{N}} = \{\mathrm{f}_1,...,\mathrm{f}_{16}\}$, where N represents the numbers of frames. Optical flow is the motion of objects between consecutive frames of the sequence; it is obtained by the function inside the OpenCV.

A green part of figure 2(a), temporal convolution $\mathrm{temporal\_CNN}()$ that is denoted by a 3D CNN layer with kernel size 1x1x3 is used to extract optical flow motions along the temporal direction to produce the temporal feature T, the operation is denoted by equation(1).

$$T = \mathrm{temporal\_CNN}(f) \tag{1}$$

The images $x_{1...16}$ are firstly processed by channel shuffle[18] operation that is a tensor operation inside the Pytorch framework to help information flow across feature channels. Then a map function $\Omega : x \to v, x \in X^{C\times H\times W}, v \in V^{1\times N}$, where $N = C \times H \times W$, is used to transform the processed images $x_{1...16}$ into feature vectors $v_{1...16}$, where Ω represents a fully-connected neural network. The feature vector is flattened into a 3D tensor by the reshape function that is inside the Pytorch framework. The tensor is then learned by a 3D CNN layer Φ with kernel size 1x3x3.

In figure 2(b) that is marked by a blue square, the output of the 3D CNN layer is then sent to a spatial visual slide window, which consists of a merge operation that is denoted by equation(2) and a spatial visual normalization(SVN) module. The goal of the merge operation is to aggregate features of two frames to produce the spatial feature ξ.

$$\xi = \Phi(v_i) \oplus \Phi(v_{\mathrm{i+1}}) \tag{2}$$

where $\oplus$ represents matrix addition. Φ represents 3D CNN with kernel size 1x3x3.

The goal of the SVN module is to calculate the similarity of spatial features while keeping the data distribution consistent. The procedure of SVN includes the following steps. First, the feature vector of two frames is used to calculate mean value *u*, denoted by equation (3).

$$u = \frac{1}{M}\sum_{i=1}^{M}\left\|\Phi(v_i) - \Phi(v_{\mathrm{i+1}})\right\| \tag{3}$$

where M represents the number of frames.

In the second step, the mean value and the feature vector of the i^{th} frame are used as the input to calculate variance, the operation is shown by equation (4). Equation (5) is used to perform a normalization operation that keeps the data distribution consistent and reduces the complexity of the model and obtain feature map s, where ε represents noise from Gaussian distribution. ξ,μ,σ represents the output of equation(2), equation(3), and equation(4), respectively. The output of equation (5) is stacked together to produce the final spatial feature map $s_{spa} \in R^{8C\times H\times W}$, where the channel numbers C of each window is stacked together to 8C, and the operation is denoted by equation(6), where CONCAT represents connect operation.

$$\sigma = \frac{1}{M}\sum_{i=1}^{M}\left\|\Phi(v_i) - u\right\|^2 \tag{4}$$

$$s = \frac{\xi - u}{\sqrt{\sigma + \varepsilon}} \tag{5}$$

$$s_{spa} = \mathrm{CONCAT}[s_1,...,s_8] \tag{6}$$

In figure 2(c) that is marked by a red square, the hierarchy residual network is performed to learn the feature of the action context of each spatial-visual feature map. The detail of the HResidual unit $\mathrm{spatial\_CNN}()$ is given in figure 3. The HResidual unit consists of a 3D CNN layer, a down-sample module φ, a δ module, and a LeakyReLU activation function. The goal of this module is to produce a spatial feature $\mathrm{s} \in R^{C\times H\times W}$, the overview operation of this module is denoted by equation(7):

$$s = \mathrm{spatial\_CNN}(s_{spa}^{i};\theta) + s_{spa}^{i-1} \tag{7}$$

where s_{spa}^{i} represents the spatial-visual feature of the i^{th} channel.

A 3D CNN layer Φ with kernel size 1x3x3 is firstly performed to learn the spatial visual feature. The down-sample module is then used to extract the maximum value of the feature map, and function δ is used to calculate similarity that is a conditional probability under given spatial- visual feature map of the i^{th} channel, the operation is denoted by equation (8).

$$\mathrm{spatial\_CNN}(s_{spa}^{i};\theta) = \phi(W_{s_{spa}^{i-1}}{}^{T} * \Phi(s_{spa}^{i})) + \mathrm{LeakyReLU}(W_{s_{spa}^{i-1}}{}^{T} * \delta(s_{spa}^{i},\Phi(s_{spa}^{i}))) \tag{8}$$

where φ,δ represents the down-sample module and action feature similarity computation module, respectively. LeakyReLU represents activation function. $W^{T}{}_{s_{spa}^{i}}$

represents the weight matrix of the action feature. θ represents the hyper-parameters of the neural network.

Finally, a mask is used to measure the weights of the action of two frames. If the similarity of action of the two frames is low, then the Hadamard product is performed, otherwise, the only operation that is performed of the part is convolution. The advantage of this operation is that only features with significant weight are calculated, which can greatly reduce the scale of the parameter. The operation is denoted by equation (9), where $\otimes$ represents Hadmard product. $*$ represents a matrix product.

$$\delta(s_{spa}^{i}, \Phi(s_{spa}^{i})) = (1 - mask(s_{spa}^{i})) * \Phi(s_{spa}^{i}) + mask(s_{spa}^{i}) * \Phi(s_{spa}^{i}) \otimes W_{s_{spa}^{i}} \quad (9)$$

After all operations of this part are accomplished, the final feature is converged into one backbone to produce the final spatial-visual feature s, where $\oplus$ represents matrix addition.

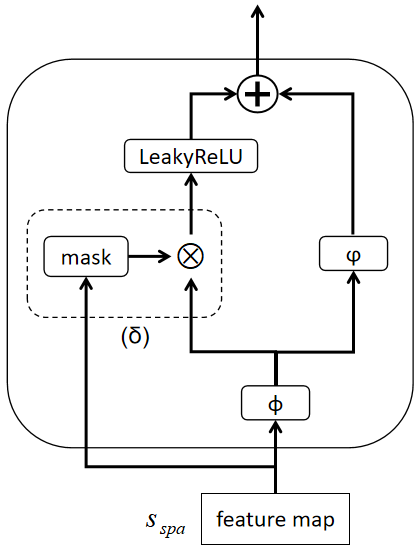

Figure 3 HResidual Unit

where mask represents a 3D convolution layer with kernel size 1x3x3 and output 1 channel.

3.2 Spatial feature attention

In figure 4, spatial feature attention consists of two Gaussian error linear units(GELU), a softmax function, and a single-layer feed-forward neural network f. The two spatial feature maps s^1, s^2 are used as the input to spatial attention module that is incorporated attention mechanism to further learn better spatial feature.

Two Gaussian error linear units(GELU) is used for spatial-visual feature learning, because which takes advantage of Dropout operation[19] and non-linear activation function to prevent model over-fitted and enhance the learning ability of the model. Then Hadmard product is performed to fuse two features, the operation is shown by equation(10).

$$\gamma = \mathrm{GeLU}(s^1) \otimes \mathrm{GeLU}(s^2) \tag{10}$$

where $\otimes$ represents Hadmard product.

For attention, the fused feature is sent to the softmax function that is denoted by equation(11) is used to obtain the attention score β.

$$\beta = \mathrm{softmax}(\gamma) \tag{11}$$

The fused feature γ and the output of softmax function are used as the input to the single-layer feed-forward neural network f that is denoted by equation(12) to generate an attention feature vector λ.

$$\lambda = \mathrm{f}(\beta, \gamma) \tag{12}$$

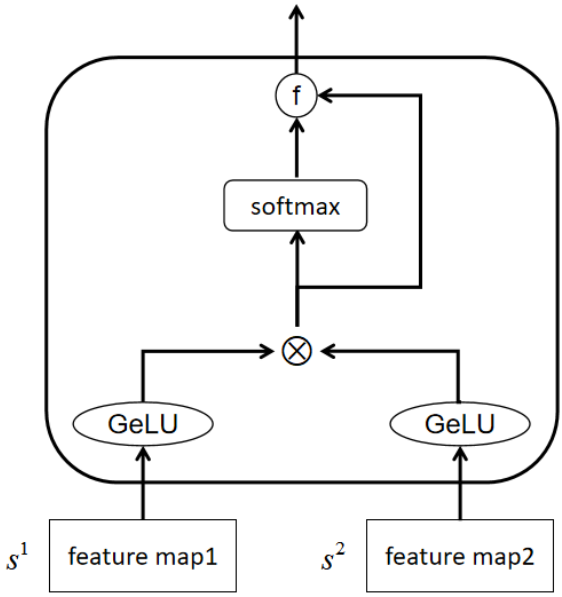

Figure 4 spatial feature attention module

3.3 Hybrid spatial-temporal attention

In figure 5, to incorporate spatial and temporal features to produce the final feature, the hybrid spatial-temporal attention module that is inspired by the Google transformer [20] is proposed. The hybrid spatial-temporal attention consists of a split function, a

pair of the max-pooling network, a fully-connected network, and a block-in attention module which is given by figure 7 in detail.

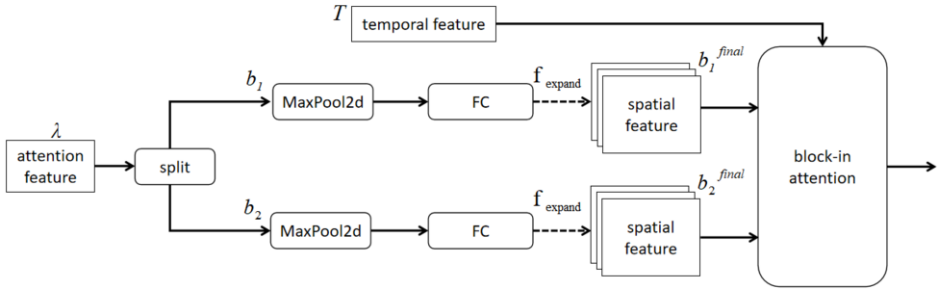

Figure 5 spatial-temporal attention module

The output of the spatial attention module is used as the input, and then the attention feature λ is divided into two branches $b_1, b_2 \in R^{2C\times H/2\times W/2}$, the operation is denoted by equation (13). In figure 6, the principle of split function is that numbers of channel C are divided into 2C through a tensor operation that can enhance the ability of information interaction between channels for image; the operation is denoted by equation (14).

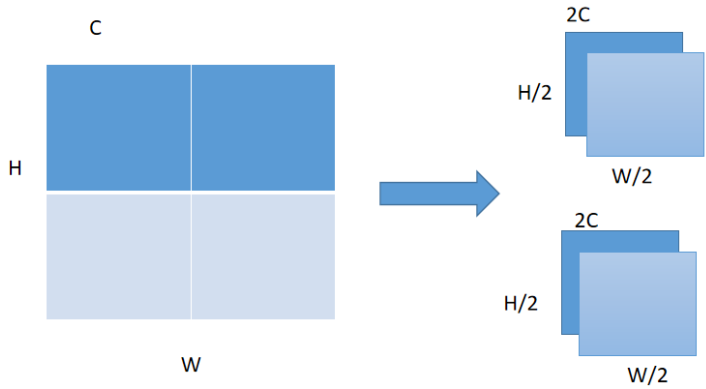

Figure 6 channels split

$$b_1 = b_2 = \text{split}_{2c}(\lambda) \tag{13}$$

$$\text{split}_{2c}[C,H,W] = [[C,H/2,W/2],[C,H/2,W/2]] \tag{14}$$

Then, the feature map of each branch is processed by max-pooling, fully-connected layer fc respectively, the operation is denoted by equation(15). Finally, the expand function f_{expand} that is denoted by equation(16) is used to expand the numbers of the channel to feature map $b_1^{final}, b_2^{final} \in R^{C\times H\times W}$ to enhance the

information redundancy between channels, the operation of this part is denoted by equation(16).

$$\begin{aligned} b_1^{final} &= \mathrm{f}_{scale}(\mathrm{fc}(\mathrm{F}_{\max}(b_1))) \\ b_2^{final} &= \mathrm{f}_{scale}(\mathrm{fc}(\mathrm{F}_{\max}(b_2))) \end{aligned} \tag{15}$$

$$\mathrm{f}_{\mathrm{expand}}(2C, H/2, W/2) = [C, H, W] \tag{16}$$

In figure 7, the feature map b_1^{final}, b_2^{final} and the temporal feature T are used as input to the block-in attention module. The block-in module consists of matrix product operation, softmax function, and element-wise product operation.

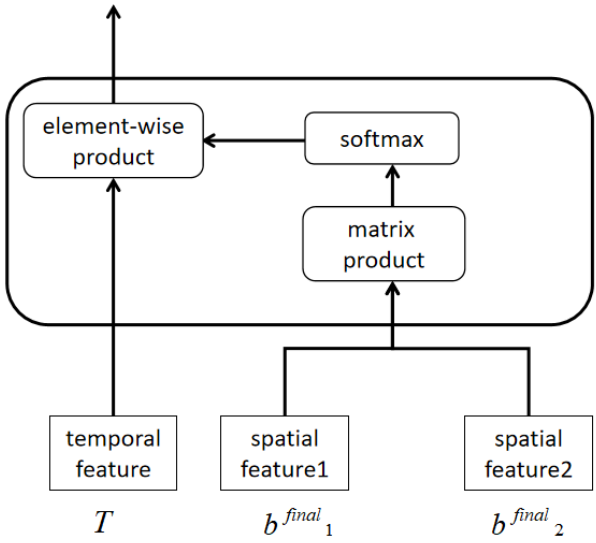

Figure 7 block-in attention module

Matrix product is firstly performed to fuse two feature maps to enrich the information of spatial features. Then the softmax operation is performed to obtain the hybrid attention score. Finally, the attention score and temporal feature T are used as the input to perform the element-wise product to obtain the final attention feature vector, then the feature vector is sent to the classifier to predict the final result, the operation of this part is denoted by equation (17).

$$\begin{aligned} &\mathrm{transformer}(T, b_1^{final}, b_2^{final}) \\ &= \mathrm{softmax}\left(\frac{\mathrm{mat}(W^T{}_1 b_1^{final}, W_2{}^T b_2^{final})}{\sqrt{\max(1,b)}} \bullet T\right) \\ &= \mathrm{softmax}\left(\frac{W_1{}^T * b_1^{final} + W_2{}^T * b_2^{final}}{\sqrt{\max(1,b)}} \bullet T\right) \end{aligned} \tag{17}$$

where W_1^T, W_2^T represents the weight matrix of $b^{final}{}_1, b^{final}{}_2$, respectively. $*$ represents a matrix product. $\bullet$ represents an element-wise product.

4. Experiment

This section starts with the data set and training details, followed by network architectures,e.g. R3D, R2Plus1D[21], C3D, Hierarchy Spatial-Temporal transformer, and result.

Data Set HMDB51[22] is used in our model, which includes 51 action categories and 7000 short videos, and the UCF101 data set which includes 101 action categories and 30000 short videos. Both of them are open-source data set that contains various scenes in daily life, and a word is used to describe the characters in the scene are doing certain actions, such as running, brushing teeth, playing football, etc.

Training Details In this work, the model is trained in a completely supervised manner on the HMDB51 and UCF101 data set, meaning video sequences is the only needed information. As a video process, a variable t is denoted by the frame rate and set it to 16, to decode the short videos with a frame rate of 16fps, and resize all frames to 224x224x3. In this experiment, the kernel size of the first HResidual Unit is reshaped as 1x1 in every layer to reduce the parameter scale.ResNet-21 and VGG-13 are used as a backbone network. The model is trained end-to-end using a batch size of 64 for 100 iterations with an Adam optimizer. Four-way Tesla P100 GPU is used in this experiment for two days on average. The learning rate has different settings for four different networks: for C3D networks, the experiment found that when the learning rate is set to 0.1, the loss function will appear, NAN, so it is set to 0.0001. When the learning rates of R2Plus1D and R3D are set to 0.001, the network converges the fastest. The learning rate of the hierarchy spatial-temporal transformer proposed in this work is set to 0.01.

Network Architecture The overall network architecture is listed in Table1. The network from left to right in the table is sorted in order of the parameter scale. ResNet[23] with layer size 4,7,7,3 and VGG[24] with layer size 4, 3, 3, 3 is used as the backbone network. The three columns on the left are the baseline models, in which C3D and R2Plus1D network all use 3 x 3 kernel with stride 2. The model R3D uses 7 x 7 convolution with input channel 3 and stride 2, then passes through a maximum pooling layer with kernel size 3 x3 and stride 2. Our model hierarchy spatial-temporal transformer, three sections in the table is interpreted as a three-stage calculation model, which are: hierarchy residual reformer, spatial attention module, and hybrid spatial-temporal attention. Finally, all modes are processed by global average pooling and fully-connected layers, and then classified by Softmax.

Table 1. The four columns refer to ResNet-21. In detail is the general shape of a residual block, including filter sizes and feature dimensions. The number of stacked blocks on each stage is presented on the right of the grid. Batch norm suggests Batch Normalization. The symbol FC indicates the output dimension of the two fully-connected layers.

Output	C3D	R2Plus1D	R3D	HST transformer
224x224	3 x 3,64,stride 2	3 x 3,64,stride 2	7 x 7,3,stride 2	1x1,3,stride 1
112x112	Relu activation	relu activation	3 x 3 max pool,stride 2	LeakyReLU activation
112x112	3 x 3 , 64 3 x 3 , 64 x4 3 x 3 , 128	7 x 7 , 3 x4 3 x 3 , 64	3 x 3 , 64 Batch Norm x4 3 x 3 ,128	1 x 1 , 64 three stages x4 3 x 3 ,128
56 x 56	3 x 3 , 128 3 x 3 , 128 x6 3 x 3 , 256	7 x 7 , 64 x6 3 x 3 ,128	3 x 3 ,128 Batch Norm x6 3 x 3 ,256	1 x 1 , 64 three stages x6 3 x 3 ,128
28 x 28	3 x 3 , 256 3 x 3 , 256 x6 3 x 3 , 512	7 x 7 , 128 x6 3 x 3 , 256	3 x 3 ,256 Batch Norm x6 3 x 3 ,512	1 x 1 , 64 three stages x6 3 x 3 ,128
14 x 14	3 x 3 , 512 3 x 3 , 512 x3 3 x 3 , 512	7 x 7 , 256 x3 3 x 3 , 512	3 x 3 , 512 Batch Norm x3 3 x 3 , 512	1 x 1 , 64 three stages x3 3 x 3 ,128
7 x 7	3 x 3 global average pool, 1024 FC, softmax	3 x 3 global average pool, 1024 FC, softmax	3 x 3 global average pool, 1024 FC, softmax	3 x 3 global average pool, 1024 FC, softmax

Table 2. List of the baseline model and our model, as well as the backbone network, parameter scale, and accuracy.

Model	Dataset	Backbone	Params(M)	Acc(%)
C3D	HMDB51	ResNet-21	78.20M	76.54%
R2Plus1D	HMDB51	ResNet-21	65.53M	80.64%
R3D	HMDB51	ResNet-21	57.68M	83.25%
Hirearachy Spatial-Temporal Transformer	HMDB51	ResNet-21	35.54M	85.53%

Table 3. The four columns refer to VGG-13. In detail is the general shape of a VGG block, including filter sizes and feature dimensions. The number of stacked blocks on each stage is presented on the right of the grid. Batch norm suggests Batch Normalization. The symbol FC indicates the output dimension of the two fully-connected layers.HST short for Hierarchy Spatial-Temporal Transformer

Output	C3D	R2Plus1D	R3D	HST transformer
224x224	3 x 3,64,stride 2	3 x 3,64,stride 2	7 x 7,3,stride 2	1x1,3,stride 1
112x112	Relu activation	Relu activation	3 x 3 max pool,stride 2	LeakyReLU activation
112x112	3 x 3 , 64 3 x 3 , 64 x4 3 x 3 , 128	7 x 7 , 3 x4 3 x 3 , 64	3 x 3 , 64 Batch Norm x4 3 x 3 ,128	1 x 1 , 64 three stages x4 3 x 3 ,128

56 x 56	3 x 3 , 128 3 x 3 , 128 x3 3 x 3 , 256	7 x 7 , 64 X3 3 x 3 ,128	3 x 3 ,128 Batch Norm x3 3 x 3 ,256	1 x 1 , 64 three stages x3 3 x 3 ,128
28 x 28	3 x 3 , 256 3 x 3 , 256 x3 3 x 3 , 512	7 x 7 , 128 X3 3 x 3 , 256	3 x 3 ,256 Batch Norm x3 3 x 3 ,512	1 x 1 , 64 three stages x3 3 x 3 ,128
14 x 14	3 x 3 , 512 3 x 3 , 512 x3 3 x 3 , 512	7 x 7 , 256 x3 3 x 3 , 512	3 x 3 , 512 Batch Norm x3 3 x 3 , 512	1 x 1 , 64 three stages x3 3 x 3 ,128
7 x 7	3 x 3 global average pool, 1024 FC, softmax	3 x 3 global average pool, 1024 FC, softmax	3 x 3 global average pool, 1024 FC, softmax	3 x 3 global average pool, 1024 FC, softmax

Table 4. List of the baseline model and our model, as well as the backbone network, parameter scale, and accuracy.

Model	Dataset	Backbone	Params(M)	Acc(%)
C3D	UCF101	VGG-13	42.5M	77.89%
R2Plus1D	UCF101	VGG-13	35.8M	79.3%
R3D	UCF101	VGG-13	26.78M	82.65%
Hierarchy Spatial-Temporal Transformer	UCF101	VGG-13	24.1M	86.3%

The following results of inference from the cases were selected from the HMDB51and UCF101 data set. We took four frames from the test short videos and four different videos from UCF101 test videos. The actions of the two figures are ApplyLipStick and BaseBallPitch and the upper left corner of the figure suggests the current action and the probability of recognition, as shown in Figure 8 and Figure 9.

Figure 8 result of test short video from HMDB51

Figure 9 result of test short video from UCF101

5. Conclusion

This work proposed a hierarchy spatial-temporal transformer mainly addresses the problem that how to enhance the ability to learn the spatial feature of the network in action recognition of short videos and reduce parameter scale. The work introduces a hierarchy inference method that is incorporated attention mechanisms to further learn the spatial feature. The model consists of three components: hierarchy residual reformer that transforms video vector into a spatial visual feature map, spatial attention module that is used to further learn spatial feature, and spatial-temporal attention that incorporates spatial and temporal features to produce the final hybrid feature vector. The result suggests that accuracy of the model better than the baseline model. However, due to the limitation of hardware resources, the model in this work can only handle short videos within 30 seconds, and cannot handle large-scale videos. In the future, we will continue to study better local building modules to study the field of action recognition in short videos.

Reference

[1] X. Wang, R. Girshick, A. Gupta, and K. He, “Non-local Neural Networks,” arXiv 1711.07971.
[2] K. Hara, H. Kataoka, and Y. Satoh, “Can Spatiotemporal 3D CNNs Retrace the History of 2D CNNs and ImageNet?,” arXiv 1711.09577.
[3] X. Li, W. Wang, X. Hu, and J. Yang, “Selective Kernel Networks,” arXiv 1903.06586.
[4] Z. Shou, D. Wang, and S.-F. Chang, “Temporal Action Localization in Untrimmed Videos via Multi-stage CNNs,” arXiv 1601.02129.
[5] L. Wang, W. Li, W. Li, and L. Van Gool, “Appearance-and-Relation Networks for Video Classification,” arXiv 1711.09125.
[6] Yoon, KiJung & Liao, Renjie & Xiong, Yuwen & Zhang, Lisa & Fetaya, Ethan & Urtasun, Raquel & Zemel, Richard & Pitkow, Xaq. (2019). Inference in Probabilistic Graphical Models by Graph Neural Networks. arXiv 1803.07710.
[7] Baccouche M., Mamalet F., Wolf C., Garcia C., Baskurt A. (2011) Sequential Deep Learning for Human Action Recognition. In: Salah A.A., Lepri B. (eds) Human Behavior Understanding. HBU 2011. Lecture Notes in Computer Science, vol 7065. Springer, Berlin, Heidelberg.
[8] D. Tran, L. Bourdev, R. Fergus, L. Torresani and M. Paluri, "Learning Spatiotemporal Features with 3D Convolutional Networks," 2015 IEEE International Conference on Computer Vision (ICCV), Santiago, 2015, pp. 4489-4497.
[9] S. Ji, W. Xu, M. Yang and K. Yu, "3D Convolutional Neural Networks for Human Action Recognition," in IEEE Transactions on Pattern Analysis and Machine Intelligence, 2013, Jan.,vol. 35, no. 1, pp. 221-231.
[10] Z. Qiu, T. Yao, and T. Mei, “Learning Spatio-Temporal Representation with Pseudo-3D Residual Networks,” arXiv 1711.10305.
[11] H. Xu, A. Das, and K. Saenko, “R-C3D: Region Convolutional 3D Network for Temporal Activity Detection,” arXiv 1703.07814.

[12] W. Du, Y. Wang and Y. Qiao, "RPAN: An End-to-End Recurrent Pose-Attention Network for Action Recognition in Videos," 2017 IEEE International Conference on Computer Vision (ICCV), Venice, 2017, pp. 3745-3754.
[13] J. Gao, Z. Yang, C. Sun, K. Chen, and R. Nevatia, "TURN TAP: Temporal Unit Regression Network for Temporal Action Proposals," arXiv 1703.06189.
[14] A. Karpathy, G. Toderici, S. Shetty, T. Leung, R. Sukthankar and L. Fei-Fei, "Large-Scale Video Classification with Convolutional Neural Networks," 2014 IEEE Conference on Computer Vision and Pattern Recognition, Columbus, OH, 2014, pp. 1725-1732.
[15] M. Zolfaghari, K. Singh, and T. Brox, "ECO: Efficient Convolutional Network for Online Video Understanding," arXiv 1804.09066.
[16] D. Tran, H. Wang, L. Torresani, J. Ray, Y. LeCun and M. Paluri, "A Closer Look at Spatiotemporal Convolutions for Action Recognition," 2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition, Salt Lake City, UT, 2018, pp. 6450-6459
[17] Z. Shou, J. Chan, A. Zareian, K. Miyazawa, and S.-F. Chang, "CDC: Convolutional-De-Convolutional Networks for Precise Temporal Action Localization in Untrimmed Videos," arXiv 1703.01515.
[18] J. Carreira and A. Zisserman, "Quo Vadis, Action Recognition? A New Model and the Kinetics Dataset," arXiv 1705.07750.
[19] X. Zhang, X. Zhou, M. Lin and J. Sun, "ShuffleNet: An Extremely Efficient Convolutional Neural Network for Mobile Devices," 2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition, Salt Lake City, UT, 2018, pp. 6848-6856.
[20] Vaswani, Ashish & Shazeer, Noam & Parmar, Niki & Uszkoreit, Jakob & Jones, Llion & Gomez, Aidan & Kaiser, Lukasz & Polosukhin, Illia. (2017). Attention Is All You Need. ArXiv, abs/1706.03762
[21] C. Szegedy et al., "Going deeper with convolutions," 2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Boston, MA, 2015, pp. 1-9.
[22] H. Kuehne, H. Jhuang, E. Garrote, T. Poggio and T. Serre, "HMDB: A large video database for human motion recognition," 2011 International Conference on Computer Vision, Barcelona, 2011, pp. 2556-2563
[23] K. He, X. Zhang, S. Ren and J. Sun, "Deep Residual Learning for Image Recognition," 2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Las Vegas, NV, 2016, pp. 770-778.
[24] Simonyan, Karen & Zisserman, Andrew. (2014). Very Deep Convolutional Networks for Large-Scale Image Recognition. arXiv 1409.1556.