STA-CNN: Convolutional Spatial-Temporal Attention Learning for Action Recognition

Hao Yang, Chunfeng Yuan, *Member IEEE,* Li Zhang, Yunda Sun, Weiming Hu, *Senior Member IEEE,* and Stephen J. Maybank, *Fellow IEEE*

Abstract—Convolutional Neural Networks have achieved excellent successes for object recognition in still images. However, the improvement of Convolutional Neural Networks over the traditional methods for recognizing actions in videos is not so significant, because the raw videos usually have much more redundant or irrelevant information than still images. In this paper, we propose a Spatial-Temporal Attentive Convolutional Neural Network (STA-CNN) which selects the discriminative temporal segments and focuses on the informative spatial regions automatically. The STA-CNN model incorporates a Temporal Attention Mechanism and a Spatial Attention Mechanism into a unified convolutional network to recognize actions in videos. The novel Temporal Attention Mechanism automatically mines the discriminative temporal segments from long and noisy videos. The Spatial Attention Mechanism firstly exploits the instantaneous motion information in optical flow features to locate the motion salient regions and it is then trained by an auxiliary classification loss with a Global Average Pooling layer to focus on the discriminative non-motion regions in the video frame. The STA-CNN model achieves the state-of-the-art performance on two of the most challenging datasets, UCF-101 (95.8%) and HMDB-51 (71.5%).

Index Terms—Temporal Attention, Spatial Attention, Convolutional Neural Network, Action Recognition

1 Introduction

Action recognition in videos has been extensively investigated in computer vision, owning to its great potential in many applications such as intelligent surveillance [1], human-computer interaction [2], [3], robotics [4], *etc*. Inspired by a series of successes in Convolutional Neural Networks (CNN) for object recognition in still images [5], [6], [7], many CNN based methods have been proposed for action recognition. However, recognizing actions in raw videos is still a challenging task because the raw videos have much more redundant or irrelevant information in the spatial and temporal domains, as compared with still images.

To focus on interesting regions in videos, visual attention has been applied in action recognition models [8], [9], [10], [11], [12], [13], [14], [15]. Most previous visual attention methods in action recognition [8], [9], [10], [16] are constructed with the Recurrent Neural Network (*i.e.* LSTM [17]). They generate an attention map at each timestep according to the input information at the current timestep and the history information obtained at previous timesteps. But these methods are costly in computation and have not achieved comparable results with the CNN based action recognition methods [18], [19], [20], [21]. Poses [11], [12], [13] and bounding boxes [14], [15] of actors are exploited to focus on human body parts. However, they rely on manually defined body parts with the following two limitations: (1) the precise annotations of poses and bounding boxes are labor-intensive or device-dependent; (2) not every part of a subject is discriminative for action recognition. For example, the discriminative part of example (a) in Fig. 1 is only around the eyes, while in example (b), the discriminative part is only around the mouth. Moreover, the previous attention models apply visual attention only in the spatial domain, while the temporal attention is lack of studies.

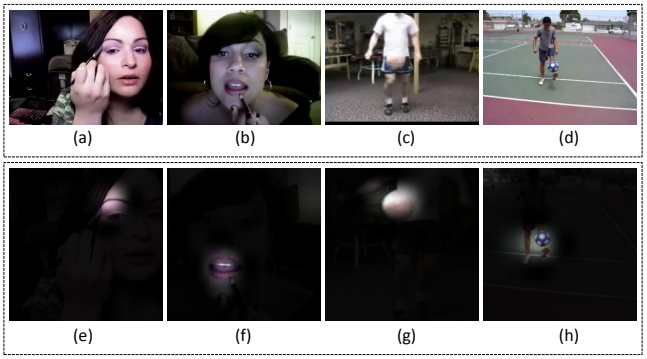


Fig. 1. Four video examples in the UCF-101 video dataset. (a) involves the action "ApplyEyeMakeup", (b) involves the action "ApplyLipstick", (c) and (d) both show the action "SoccerJuggling". The bottom row shows the Spatial Attention Map of each video example learned by the proposed Spatial Attention Network.

In raw videos, different temporal segments contribute to action recognition differently. Some segments are discriminative for the classification, while others mislead the action classifier. For example, the top row in Fig. 2 shows

- *Hao Yang and Yunda Sun are with the R&D Center of Artificial Intelligent, NUCTECH Company Limited, Beijing, China.*
  Li Zhang is with the Department of Engineering Physics, Tsinghua University, Beijing, China.
  E-mail: {yanghao1, zhangli, sunyunda}@nuctech.com
- *Chunfeng Yuan and Weiming Hu are with the National Laboratory of Pattern Recognition in the Institute of Automation, Chinese Academy of Sciences, Beijing, China.*
  E-mail: {cfyuan, wmhu}@nlpr.ia.ac.cn
- *Stephen J. Maybank is with the Department of Computer Science and Information Systems, Birkbeck College. London, United Kingdom.*
  E-mail: sjmaybank@dcs.bbk.ac.uk

Manuscript submitted August, 2019.

Fig. 2. Two video examples from action recognition datasets. The STA-CNN model employs the Temporal Attention Mechanism to select the discriminative temporal segments from long and noisy videos. The bars below the video segments show the learned discriminative confidences for the segments. These are used as averaging weights in the test stage.

the action "Biking", but the rider and the bike do not appear in the first few segments of the video. These segments are irrelevant for recognizing the action. The bottom row in Fig. 2 shows the action "LongJump". In this video, most segments show an athlete running in a sports ground. These segments have appeared in many actions of the same dataset, such as "HighJump", "JavelinThrow", "PoleVault", "SkipJump" and so on. So these segments are not discriminative for recognizing the action "LongJump". These irrelevant and undiscriminating segments mislead the action classifier when it averages the predictions from all segments. To eliminate the irrelevant and undiscriminating segments in videos, previous methods require input videos to be pre-processed. However, the pre-processing is usually performed manually, hence it is labor-intensive and cost-expensive in real-world applications. In this paper, we propose an unsupervised Temporal Attention Mechanism. It is able to automatically mine discriminative segments which strongly support the final decision.

In the spatial domain of videos, researchers [18], [20] have found that the most important information for action recognition is contained in the following two types of regions: (1) motion salient regions; for example (c) and (d) in Fig. 1 both contain the action "SoccerJumping". The motion salient regions are the leg and the soccer ball, which are sufficient for recognizing the action. (2) discriminative non-motion regions. For example (a) in Fig. 1 contains the action "ApplyEyeMakeup" and (b) contains the action "Applylipstick". They have the same scene and motion. The discriminative regions are the non-motion eyes in the example (a) and the non-motion mouth in the example (b). To focus on both types of regions described above, we propose a Spatial Attention Network (SAN). The SAN is pre-trained on optical flow predicting datasets to extract flow features and the locations of motion salient regions. In addition, we use a Global Average Pooling (GAP) layer to replace fully connected pooling layers in order to keep the remarkable location ability of convolutional units until the final layer [22], [23]. Meanwhile, the SAN is trained with a classification loss to highlight the discriminative non-motion regions. Finally, the Spatial Attention Layer generates the Spatial Attention Map to guide the Action Classification Network in learning effective spatial-temporal features from the motion salient regions and the discriminative non-motion regions in the video frame.

The main contributions of this work are summarized as follows:

- We propose a novel Temporal Attention Mechanism which automatically mines discriminative temporal segments from raw videos. Only the selected segments are used to update weights of the network. It eliminates the interference of irrelevant and undiscriminating segments in the raw videos.
- We propose a weakly-supervised Spatial Attention Mechanism which selectively focuses on motion salient and discriminative non-motion spatial regions. It generates a Spatial Attention Map to guide the Action Classification Network in learning effective spatial-temporal features for action recognition.
- The proposed Spatial-Temporal Attentive Convolutional Neural Network (STA-CNN) incorporates the Temporal Attention Mechanism and the Spatial Attention Mechanism into a unified convolutional network. It achieves the state-of-the-art performance on the UCF-101 and HMDB-51 datasets.

2 Related Works

Action recognition in videos is a very challenging task and has long been an active research topic in computer vision. Inspired by the success of CNN models in object recognition [5], [6], [7], [24], [25], many convolutional models have been proposed to recognize actions [18], [19], [20], [21], [26], [27] in recent years. For example, the Slow Fusion model [26] fuses spatial and temporal information at multiple semantic levels. The Two-stream models [18], [19] train two convolutional networks separately, *i.e.*, the SpatialNet is trained on the RGB frame to extract appearance features and the TemporalNet is trained on flow frames to model motion features. The confidence scores of the two networks are fused in order to improve classification performance. The Fusion Two-stream model [21] demonstrates that fusing appearance and motion features after the last convolutional layer achieves a better performance. The Temporal Segment Network (TSN) [20] splits an input video into three segments in the temporal domain and trains the Spatial ConvNet and the Temporal ConvNet by averaging the predictions from the three segments for the long-range temporal structure modeling. The 3D convolutional models [27], [28], [29] extend the 2D convolution to the spatial-temporal domain. They can abstract the spatial-temporal features at multiple semantic levels naturally and effectively. The C3D model [27] is pre-trained on a large-scale video dataset to learn general features which are used to train a linear SVM for action classification. I3D [29] proposes a very deep Inflated 3D-CNN model by extending the Inception model [7] to 3D in order to extract spatial-temporal features of actions. The previous 3D convolutional deep models [27], [28] are typically learned within a short snippet of videos, so they fail to model actions over their full temporal extent. The LTC-CNN model [30] operates on longer temporal extents of videos in order to improve the accuracy of action recognition.

Human perception includes an important mechanism for focusing attention selectively on the interested regions in

a scene. This selective attention mechanism has long been an important topic in the vision community. An attention model is proposed in [31] for sequence to sequence training in machine translation, where two types of visual attention have been studied firstly, namely hard attention and soft attention. In hard attention, spatial regions are selected by making binary choices. For example, Mnih et al. [32] and Ba et al. [33] apply hard attention to object recognition to extract the most salient features in images. Soft attention mechanisms use the weighted averages instead of hard binary selection. Soft attention is extended to the image captioning task in [34] since image captioning can be essentially considered as image to language translation. To apply an attention mechanism in action recognition, we exploit the hard attention in the temporal domain to select the discriminative segments, and exploit the soft attention in the spatial domain to selectively focus on the motion salient regions.

Several previous action recognition approaches have employed attention mechanisms [8], [9], [10], [11], [12], [13], [14], [15]. Sharma, et al. [8] first propose a soft attention based Recurrent Neural Network (*i.e.*, LSTM) for action recognition. At each time step, an attention map is learned to weight convolutional features. The Attention VideoLSTM [10] replaces the full connections in the LSTM with convolutional connections. It is able to generate a 2D attention map directly for spatial features pooling. The work [16] proposes an interpretable and easy plug-in spatial-temporal attention mechanism. It learns a salience mask in order to focus on the most salient features in the spatial domain and it employs a convolutional LSTM based attention mechanism to identify the most relevant frames in the temporal domain. The Hierarchical Attention Network [9] proposes a hierarchical attention structure to model the temporal transitions between frames as well as video segments. It effectively incorporates the short-term motion information and long-term temporal structures. These LSTM based attention methods require very large computational resources but achieve inferior performances compared with the CNN based action recognition methods [18], [19], [20], [21]. A 3D-CNN based attention model [35] is proposed to provide tighter crops around relevant video regions using a saliency based attention transformation. Some recent works generate more fine-grained representations by extracting features around human pose keypoints [11], [12], [13] or from person bounding boxes [14], [15]. These forms of attention are helpful for classification performance, but they require annotating bounding boxes and poses of subjects. Optical flow is a good representation of instantaneous motion in video frames [36], [37]. So it is exploited to guide a convolutional network to pay attention on the motion salient regions of the video frame [36] or to enhance the spatial features using the corresponding motion features [37]. However, these methods require computing the optical flow fields for every two consecutive frames in videos.

3 Our Spatial-Temporal Attention Model

In this section, we present the details of the proposed Spatial-Temporal Attentive Convolutional Neural Network (STA-CNN). The STA-CNN model incorporates a Temporal Attention Mechanism and a Spatial Attention Mechanism. Firstly, the unsupervised Temporal Attention Mechanism is introduced to mine the discriminative segments in the temporal domain of videos. Then, we introduce the weakly-supervised Spatial Attention Mechanism which introduces a Spatial Attention Network to focus on both motion salient regions and discriminative non-motion regions in the spatial domain of videos. Finally, the overall architecture of the STA-CNN model is presented in Fig. 3.

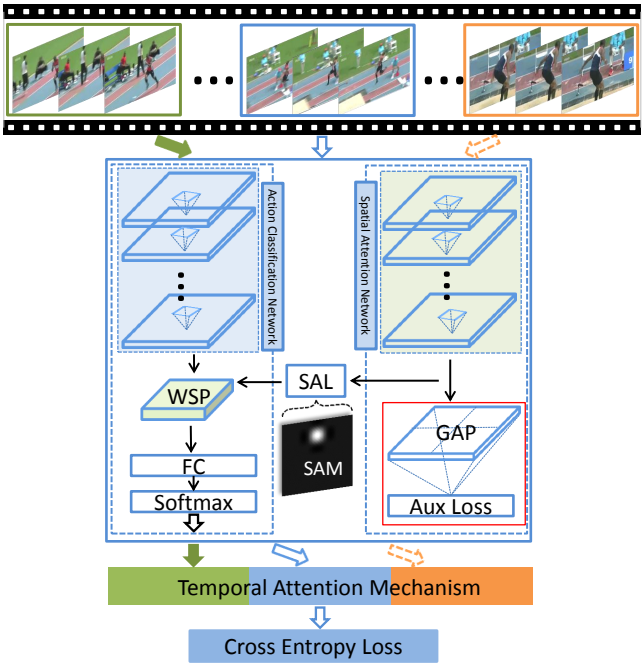


Fig. 3. The architecture of the STA-CNN model. The Spatial Attention Network generates a Spatial Attention Map to guide the Action Classification Network extracting effective features from informative spatial regions. The most discriminative segments are selected by the Temporal Attention Mechanism. Only the selected segments are used to update the weights in the backward pass.

3.1 Unsupervised Temporal Attention Mechanism

In the real-world application of action recognition, the raw videos usually contain many irrelevant or undiscriminating segments, as shown in Fig. 2. These segments usually mislead the action classifier. To eliminated the interference of the irrelevant and undiscriminating temporal segments, we propose an unsupervised Temporal Attention Mechanism. The Temporal Attention Mechanism mines the discriminative segments based on the prediction confidence of each video segment. It does not require any extra annotation than class label.

More specifically, we firstly divide each video V into N segments with equal time intervals in the temporal domain, denoted by $V = \{s_n\}_{n=1}^{N}$. Then an input snippet is randomly selected from each segment. Without loss of generality, the snippets selected from the video are also denoted by $s_n, n = 1, 2, ..., N$. Each snippet contains T frames $s_n = [i_1, i_2, ..., i_T], n = 1, 2, ..., N$. All of the snippets are fed into the classification network. The predictions of these segments are computed through the forward pass, denoted

Input: Dataset $(V, y) \in D$, SGD parameter η, λ, $iter$.
Output: Network weights W, and W is initalized from ImageNet pre-trained model.

$t = 0$;
while $t < iter$ **do**
  $t = t + 1$;
  Randomly sample a batch $B \in D$;
  Forward pass:
  For $V \in B$, $V = \{s_n\}_{n=1}^{N}$, compute predictions $\{p(s_n)\}_{n=1}^{N}$;
  Sort $p(s_n)$ by the discriminative confidence in Eqn. (1) and update indicator β_n: $\beta_n = 1[[\alpha_n \in top\_n\{\alpha_1, ..., \alpha_N\}]]$;
  $L = \sum_B \sum_n \beta_n l(s_n; \theta)$;
  Backward pass:
  Apply SGD to update the weights W using only the most discriminative segments.
  $W = W - \eta \bigtriangledown L - \lambda \bigtriangleup W$
end
return θ;

Algorithm 1: The training process of the unsupervised Temporal Attention Mechanism.

as $p(s_n), n = 1, ..., N$. Each prediction is a C dimensional vector $p(s_n) \in R^C$, where C is the number of action classes. The discriminative confidence of each segment is defined as the reciprocal of information entropy which measures the average indeterminacy of a distribution. The discriminative confidence α_n of the prediction $p(s_n)$ is defined by:

$$\alpha_n = \frac{1}{H(s_n)}, \tag{1}$$

$$H(s_n) = -\sum_{j=1}^{C} p_j(s_n) \log p_j(s_n), \tag{2}$$

When the prediction $p(s_n)$ is reliable, it is usually sparse with a low entropy of the distribution, *i.e.*, only a few entries of $p(s_n)$ have large values, while the other entries are small or approach 0. Conversely, when $p(s_n)$ is not reliable, its entries (class probabilities) tend to spread evenly over all the action categories, so the entropy $H(s_n)$ will be large.

The proposed Temporal Attention Mechanism uses hard attention. It allocates a binary weight to each video segment, *i.e.*, the most discriminative (top_n) segments are allocated 1, and the others are allocated 0. In the backward pass during training, only the most discriminative segments are used to update weights of the network. The training process of the Temporal Attention Mechanism is shown in the box **Algorithm 1**.

Two temporal segment strategies for the Temporal Attention Mechanism are proposed. As described in above, the first temporal segment strategy divides a video into N segments with equal time intervals. Then, one input snippet is selected randomly from each segment. In the backward pass, only the most discriminative (top_n) video snippets are used to update the weights of the network. This temporal segment strategy is denoted by $TS(N, top\_n)$. The second temporal segment strategy divides a video into N segments with equal times intervals and selects M input snippets from each segment. Then, the most discriminative (top_m) input snippets are selected from each segment. In the backward pass, only the selected $N \times top\_m$ snippets are used to update weights of the network. This segment strategy can cover long-term temporal information for action recognition. It is denoted by $TS(N \times M, N \times top\_m)$.

3.2 Weakly-supervised Spatial Attention Mechanism

The Spatial Attention Mechanism includes two individual subnetworks: one is the Action Classification Network and the other is the Spatial Attention Network. The Action Classification Network is the backbone of the STA-CNN model. It is used to extract spatial-temporal features for classifying actions. The Spatial Attention Network is pre-trained on the flow predicting datasets and exploits the GAP layer followed with a classification loss to generate a Spatial Attention Map which focuses on the motion salient and discriminative non-motion spatial regions. The Spatial Attention Map is used to guide the Action Classification Network in learning effective features in the informative spatial regions for action recognition.

3.2.1 Spatial Attention on Motion Salient Regions

The optical flow information is a good indication of the instantaneous motion. So the Spatial Attention Network is pre-trained on the flow predicting datasets [38] to enable the network to extract the flow features and the locations of the motion salient regions in the video frame. The Spatial Attention Network consists of ten convolutional layers and six max-pooling layers, denoted as $f_{san}(X; W_{san})$, where W_{san} are the weights of the Spatial Attention Network and X denotes the input of the network. To expand the spatial size of the predicted flow and obtain the dense flow prediction, the Expanded Part is used at the end of Spatial Attention Network. The Expanded Part includes four convolutional layers and four unpooling layers (extending the feature maps, as opposed to pooling) and it is denoted as $f_{exp}(X; W_{exp})$, where W_{exp} are the weights of the Expanded Part. The architecture of the Spatial Attention Network and the Expanded Part is shown in Fig. 4.

Formally, the input snippets, selected from a video, are denoted as $s_n, n = 1, ..., N$. The video snippet s_n is fed into the network to predict flow:

$$\hat{f}_{flow}(s_n) = f_{exp}(f_{san}(s_n; W_{san}); W_{exp}), \tag{3}$$

where $\hat{f}_{flow}(s_n)$ denotes the predicted flow from the input snippet s_n. The real flow is estimated from the input snippet s_n, denoted as $f_{flow}(s_n)$. The Spatial Attention Network and the Expanded Part are jointly trained by the Mean Square Error with a relaxation factor:

$$L_{flow} = \sum_{V} \sum_{n=1}^{N} ||max(|\hat{f}_{flow}(s_n) - f_{flow}(s_n)| - \theta, 0)||^2 \tag{4}$$

where $\theta = e^{-2} f_{flow}$ is a threshold to relax the difference between the predicted flow and the real flow. The real flow $f_{flow}(s_n)$ usually has noise, so we ignore the cases in which $|\hat{f}_{flow}(s_n) - f_{flow}(s_n)|$ is smaller than θ. Therefore, by Eqn. 4, we only minimize the sum of relatively large squared distances. Introducing the relaxation factor to the

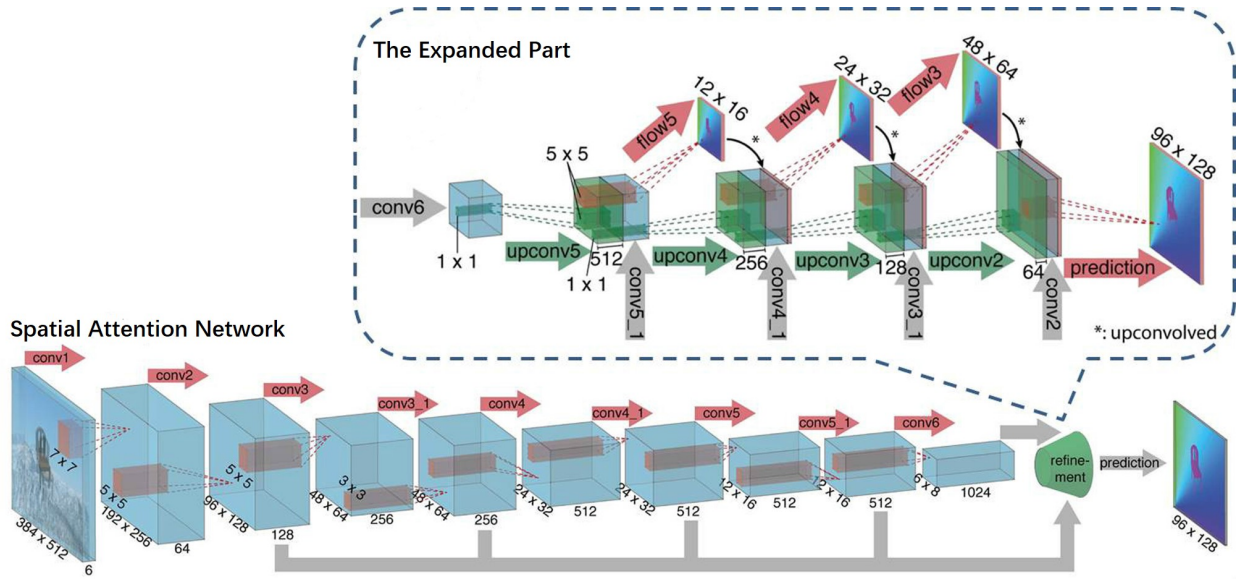


Fig. 4. The architecture of the Spatial Attention Network and the Expanded Part which are extended from the FlowNetSimple model [38].

loss function for training not only avoids noise interference and also accelerates the training of the network.

3.2.2 Spatial Attention on Discriminative non-motion Regions

Pre-training on the flow predicting datasets enables the Spatial Attention Network to locate the motion salient regions in the video frame, but it ignores the information in non-motion spatial regions and some of these information is discriminative for action recognition. To focus on the discriminative non-motion regions, we introduce classification information to the Spatial Attention Network. A GAP layer, instead of usually used multiple fully connected pooling layers, is added at the end of the Spatial Attention Network. It is followed with a *softmax* and cross-entropy loss layer. On the one hand, the GAP layer retains the remarkable localization ability of the convolutional units until the final layer [22], [23]. On the other hand, adding the classification loss to the Spatial Attention Network enables it to focus only on the remarkably discriminative regions from all the spatial locations.

Specifically, the convolutional features in the last convolutional layer of the Spatial Attention Network are denoted as $f_{san}(s_n; W_{san}) \in R^{M \times K \times K}$, where K and M denote the spatial size and the number of channels respectively. The weights of the Spatial Attention Network W_{san} are initialized from the model which is pre-trained on the flow predicting datasets. Then the features $f_{san}(s_n; W_{san})$ are reorganized as a matrix $X_n \in R^{M \times K^2}$. The Matrix X_n consists of stacked feature vectors from K^2 spatial locations, *i.e.*, $X_n = [x_{n1}, x_{n2}, ..., x_{nK^2}]$, where $x_{ni} \in R^M$ denotes the feature at the spatial location i. For a class c, the class scores S_c are denoted by

$$S_c = \sum_{m=1}^{M} w_c^m \sum_{i=1}^{K^2} x_{ni}^m = \sum_{i=1}^{K^2} \sum_{m=1}^{M} w_c^m x_{ni}^m, \tag{5}$$

where $c = 1, ..., C$, and C is the number of action classes. w_c^m is the weight corresponding to the class c for unit m. Then, we define $\hat{a}_{ni} = \sum_{m=1}^{M} w_c^m x_{ni}^m, i = 1, 2, ..., K^2$. It indicates the discrimination of the spatial location i in supporting the identification of class c. The scores of all the classes are denoted as $\hat{y}_{san} = [S_1, S_2, ..., S_C]$. Based on these class scores $\hat{y}_{san}$, the auxiliary classification loss is computed as:

$$L_{aux}(\hat{y}_{san}, y_{action}) = \sum_{c=1}^{C} y_{action}^c log S_c, \tag{6}$$

where y_{action} is the ground truth of the input video, *i.e.*, if the input video belongs to the class c, then y_{action}^c is set as 1, and the other dimensions $y_{action}^t, t \neq c$ are set as 0. The convolutional weights of the Spatial Attention Network are updated by the auxiliary classification loss with the GAP layer in Eqn. 6. It stores the localization information of discriminative spatial regions in the convolutional features.

3.2.3 Spatial Attention Layer

After the pre-training on the flow predicting datasets and the training by auxiliary classification loss with a GAP layer, the locations of motion salient regions and discriminative non-motion regions (called informative spatial regions in following) are extracted by the Spatial Attention Network and are stored in the convolutional feature maps. The Spatial Attention Layer is designed to generate the Spatial Attention Map by using the location information of informative spatial regions. The Spatial Attention Layer is added at the end of the Spatial Attention Network. It is shown as SAL in Fig. 3. In the Spatial Attention Layer, firstly, the convolutional features of the Spatial Attention Network $f_{san}(s_n; W_{san}) \in R^{M \times K \times K}$ are organized as a matrix $X_n = [x_{n1}, x_{n2}, ..., x_{nK^2}]$. Then, the Spatial Attention Layer learns a unified spatial importance from the convolutional features at each spatial location as:

$$a_{ni} = \sum_{m=1}^{M} w_i^m x_{ni}^m, i = 1, ..., K^2 \tag{7}$$

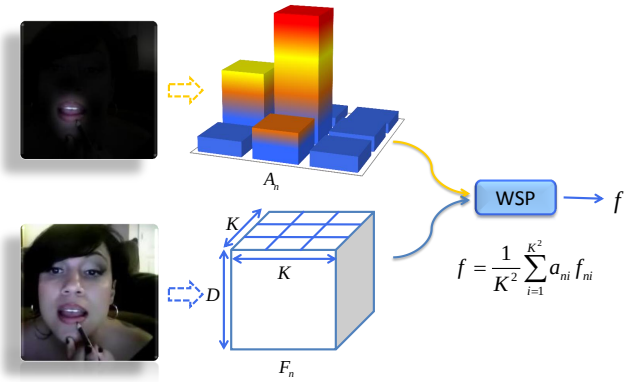


Fig. 5. The Weighted Spatial Pooling layer weights the convolutional feature cube F_n by the Spatial Attention Map A_n.

where a_{ni} denotes the importance at spatial location i for identifying the action in the video. The $w_i \in R^M$ are the weights of the Spatial Attention Layer corresponding to the spatial location i. Then, a *softmax* function is used to normalize the spatial importance as:

$$\bar{a}_{ni} = \frac{e^{a_{ni}}}{\sum_{k=1}^{K^2} e^{a_{nk}}}, \tag{8}$$

where, $\bar{a}_{ni} \in [0,1], i = 1, ..., K^2$. So the Spatial Attention Map is defined as $A_n = [\bar{a}_{n1}, \bar{a}_{n2}, ..., \bar{a}_{nK^2}]$. The *softmax* function can enlarge the contrast of the spatial importance.

3.2.4 Classification Using Spatial Attention Map

The Spatial Attention Map $A_n \in R^{K^2}$ stores the location information of the motion salient regions and the discriminative non-motion regions. The map is used to guide the Action Classification Network to automatically extract features from the informative spatial regions of the video frame. In the Action Classification Network, the feature cube in the last convolutional layer is denoted as $f(s_n; W_{cla}) \in R^{D \times K \times K}$, where W_{cla} are the weights of the Action Classification Network, and the terms K and D denote the spatial size and the number of channels of convolutional feature maps respectively. Same with the Spatial Attention Network, the feature cube $f(s_n; W_{cla})$ is sliced as K^2 feature vectors in the spatial domain. Each feature is a D dimensional vector, denoted as a matrix $F_n = [f_{n1}, f_{n2}, ..., f_{nK^2}]$. Meanwhile, each feature vector $f_{ni}, i = 1, ..., K^2$ has a corresponding weight a_{ni} from the Spatial Attention Map A_n. The weight a_{ni} indicates the importance of the corresponding feature f_{ni} at the spatial location i for action recognition. The process of the Weighted Spatial Pooling, denoted as WSP module in Fig. 3, is formulated as:

$$f = \frac{1}{K^2} \sum_{i=1}^{K^2} a_{ni} f_{ni}, \tag{9}$$

where f is the pooled feature. It is fed to the following fully-connected layer and *softmax* layer for classification. Intuitively, the detail process of the Weighted Spatial Pooling is illustrated in Fig. 5. The Spatial Attention Network and the Action Classification Network are jointly trained by the classification loss:

$$L = L(\hat{y}_{cla}, y_{action}) + \lambda \cdot L_{aux}(\hat{y}_{san}, y_{action}). \tag{10}$$

The first item is the cross entropy loss of the Action Classification Network, $L(\hat{y}_{cla}, y_{action}) = \sum_{c=1}^{C} y_{action}^{c} log \hat{y}_{cla}^{c}$, where $\hat{y}_{cla}^{c}$ denotes the prediction scores of class c from the Action Classification Network. The second item is the auxiliary classification loss of the Spatial Attention Network, as shown in Eqn. 6. The parameter λ is a balance between the two losses.

TABLE 1
Evaluating the effectiveness of the Temporal Attention Mechanism and temporal sampling strategies, finetuned on the UCF-101 dataset split1.

Models	Spatial ConvNet		Temporal ConvNet	
	DCW	AVE	DCW	AVE
BN-Inception [20]	84.1%	84.5%	86.5%	87.2%
TAM-$TS(3,1)$	85.0%	84.7%	87.4%	87.3%
TAM-$TS(6,1)$	85.3%	84.9%	87.5%	87.3%
TAM-$TS(9,1)$	85.5%	84.9%	87.7%	87.2%
TAM-$TS(9,3)$	86.1%	85.4%	88.1%	87.5%
TAM-$TS(3\times3,3\times1)$	**86.5%**	**85.8%**	**88.4%**	**87.9%**

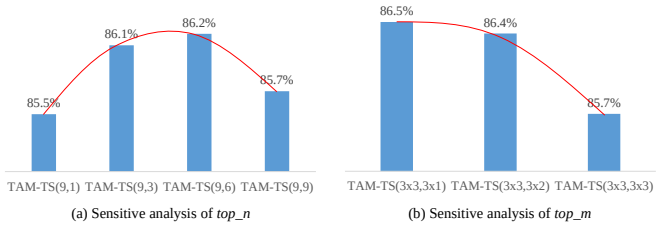
3.3 Overall Architecture

In this work, a novel STA-CNN model is proposed. The architecture of the STA-CNN is constructed by the Spatial Attention Network and the Action Classification Network, followed by the Temporal Attention Mechanism for selecting temporal segments from videos, as shown in Fig. 3. The Action Classification Network in the STA-CNN model uses the BN-Inception network [7] which is pre-trained on ImageNet [39]. Similar to the Two-stream models [18], [19], [20], [21], the STA-CNN model trains the Spatial ConvNet on RGB frames and trains the Temporal ConvNet on flow frames. Then the softmax scores of the two ConvNets are fused to classify actions. When training on RGB frames, the weights of the Spatial Attention Network are initialized from the model which is pre-trained on the flow predicting datasets. When training on flow frames, the convolutional layers of the Spatial Attention Network share weights with the Action Classification Network.

4 Experiments

4.1 Experiments Setting

Datasets: The STA-CNN deep model is evaluated on two of the most challenging video datasets, UCF-101 and HMDB-51. The UCF-101 [40] is a dataset of realistic action videos, collected from YouTube, and having 101 action categories with 13320 videos (27 hours in total). The UCF-101 dataset has a large diversity of actions and large variations in camera motion, object appearance, cluttered background, *etc*. The HMDB-51 dataset [41] is a large realistic collection of videos from movies and the web. It contains 6849 videos divided into 51 action categories. We use the raw videos without stabilization. We begin by comparing different architectures on the first split of the UCF-101 dataset. For comparison with the state-of-the-art, we follow the standard evaluation protocol and report the average accuracy over three splits on both of the datasets.



(a) Sensitive analysis of *top_n* (b) Sensitive analysis of *top_m*

Fig. 6. The sensitivity analysis for the effect of top_n and top_m on the UCF-101 dataset split1.

TABLE 2
Evaluating the effectiveness of the Spatial Attention Network, finetuned on the UCF-101 dataset split1.

Models	Multiple Segments	Auxiliary Loss	Spatial ConvNet	Temporal ConvNet
BN-Inception [20]	-	-	84.5%	87.2%
SAM-CNN1	-	+	85.2%	88.3%
SAM-CNN2	+	-	85.8%	87.7%
SAM-CNN3	+	+	**86.8%**	**89.3%**

Training: The STA-CNN model is trained with randomly selected snippets. Each frame in a snippet is resized to 256×340 and then cropped into 224×224 in the spatial domain. Horizontal flipping, corner cropping, and multi-scale cropping are used to prevent over-fitting. All models are trained by Stochastic Gradient Descent (SGD) with the momentum and weight-decay as 0.9 and 0.0005 respectively. For the Spatial ConvNet training, each snippet has two frames, and the batch size is set as 128. The base learning rate is set as 0.001 and it is divided by 10 for every 20 epochs. The training stops at the 50^{th} epoch. For the Temporal ConvNet training, each snippet has five flow frames that contain the horizontal and vertical components of the flow field. Also, the batch size of the Temporal ConvNet is set as 128. The base learning rate is set as 0.005 and it is divided by 10 for every 120 epochs. The training stops at the 300^{th} epoch.

Test: The test video is divided into 25 segments in the temporal domain. A snippet is selected from each segment. Each frame in the snippet is also resized to 256×340 and then is cropped into 224×224 in the spatial domain. The standard ten test samples, *i.e.*, one center sample and four corner samples, are cropped spatially. The previous approaches [18], [19], [20], [21] average the predictions of all segments for video-level predicting. In this paper, only the top_15 discriminative segments are used to predict the action. Then, the predictions of video segments are weighted by the discriminative confidences α_n which are computed from the corresponding predictions by Eqn. 1. If a segment is discriminative for classification, the discriminative confidence of its prediction is reasonably large, and vice versa. So the Discriminative Confidence Weighted (DCW) prediction p_{dcw} is formulated as:

$$p_{dcw} = \sum_{n=1}^{25} \beta_n \alpha_n p(s_n), \tag{11}$$

where, $\beta_n = 1[[\alpha_n \in top\_15\{\alpha_1, ..., \alpha_N\}]]$.

4.2 Evaluating the Effectiveness of the Temporal Attention Mechanism

We firstly evaluate the effectiveness of the Temporal Attention Mechanism on the UCF-101 dataset split1. As shown in the upper part of Tab. 1, there are three temporal attention models which are constructed by introducing the Temporal Attention Mechanism into the baseline model BN-Inception [20]. The BN-Inception model [20] is fed with a snippet sampled from each video, which could be regarded as that the video is first divided into one segment and then an input snippet is randomly sampled from the segment. It could be denoted by the first temporal sampling strategy $TS(N, top\_n)$, in which $N = 1$ and $top\_n = 1$. To evaluate the effectiveness of the Temporal Attention Mechanism intuitively, we set $top\_n = 1$ as the baseline model BN-Inception [20] with different temporal segments (i.e. $N = 3, 6, 9$). From the results in Tab. 1, all of the temporal attention models, TAM-$TS(3,1)$, TAM-$TS(6,1)$ and TAM-$TS(9,1)$, outperform the baseline model BN-Inception [20], especially when the DCW method is use for action prediction. In the subsequent experiments, we set $N = 9$ to balance the performance and computational complexity.

To evaluate the two temporal sampling strategies, we design two models, TAM-$TS(3 \times 3, 3 \times 1)$ and TAM-$TS(9,3)$, with same computational cost. Their experimental results are listed in the last two lines of Tab. 1. The TAM-$TS(3 \times 3, 3 \times 1)$ model outperforms the TAM-$TS(9,3)$ model. The improvement indicates that the temporal segment strategy $TS(N \times M, N \times top\_m)$ is better than the strategy $TS(N, top\_n)$, because the strategy $TS(N \times M, N \times top\_m)$ can model the long-term evolution of actions.

Then, we conduct experiments to analyze the sensitivity of the hyper-parameter top_n in Fig. 6(a). Based on the experiments above, we set $N = 9$ and set $top\_n = 1, 3, 6, 9$ to obtain the TAM-$TS(9,1)$, TAM-$TS(9,3)$, TAM-$TS(9,6)$ and TAM-$TS(9,9)$ models respectively. As shown in Fig. 6(a), TAM-$TS(9,3)$ and TAM-$TS(9,6)$ achieve better performances than the other models. These experimental results indicate that in a certain range, using more predictions of temporal segments in back-propagation improves the performance of our temporal attention model. However, the last segments with lower discriminative confidences include more noise and irrelevant information, so these segments will reduce the performance of our temporal attention model. Moreover, we design three models with $top\_m = 1, 2, 3$, denoted as TAM-$TS(3 \times 3, 3 \times 1)$, TAM-$TS(3 \times 3, 3 \times 2)$ and TAM-$TS(3 \times 3, 3 \times 3)$ respectively, to analyze the sensitivity of the hyper-parameter top_m. The results of these models are shown in Fig. 6(b). The model TAM-$TS(3 \times 3, 3 \times 1)$ achieves the best performance among these three temporal attention models. Based on the results, we set $top\_m = 1$ in the subsequent experiments.

4.3 Evaluating the Effectiveness of the Spatial Attention Mechanism

In the second experiment, the effectiveness of the Spatial Attention Mechanism is evaluated on the UCF-101 dataset split1. In the experiment, three spatial attention models are designed for comparison with the baseline BN-Inception

TABLE 3
Evaluating the effectiveness of the Spatial-Temporal Attention on the UCF-101 dataset split1.

Models	Spatial ConvNet	Temporal ConvNet	Fusion
BN-Inception	84.5%	87.2%	92.0%
STA-$TS(6,1)$	85.5%	87.9%	93.1%
STA-$TS(9,3)$	86.4%	88.7%	94.0%
STA-$TS(3\times3,3\times1)$	**86.9%**	**89.5%**	**94.2%**

model [20]. The three spatial attention models introduce the Spatial Attention Mechanism into the baseline model. The experimental results of these spatial attention models and the baseline model are listed in Tab. 2. A "+" in the column Multiple Segments indicates that the predictions of multiple temporal segments are averaged during training. A "-" in this column indicates that there is no averaging predictions of multiple temporal segments. The input video is divided into three temporal segments in these experiments. A "+" in the column Auxiliary Loss indicates that the auxiliary classification loss is added to the Spatial Attention Network. A "-" indicates that the auxiliary classification loss is not added to the Spatial Attention Network.

As shown in Tab. 2, the spatial attention models obviously outperform the baseline model. For example, the SAM-CNN3 model outperforms the BN-Inception model by 2.3% and 2.1% fed with RGB and flow frames respectively. The SAM-CNN3 model outperforms the SAM-CNN2 model fed with RGB and flow frames respectively. It indicates that adding the auxiliary classification loss module to the Spatial Attention Network in training can improve the performance obviously. The SAM-CNN3 model outperforms the SAM-CNN1 model fed with RGB and flow frames respectively. The improvement indicates that averaging multiple temporal predicting is also effective for our Spatial Attention Mechanism.

4.4 Evaluating the Effectiveness of the Spatial-Temporal Attention

In the third experiment, the temporal attention and the spatial attention are unified into a single convolutional network, referred as the STA-CNN model. This model is trained end-to-end. In the experiment, the effectiveness of the Spatial-Temporal Attention models is evaluated on the UCF-101 dataset split1. The Spatial-Temporal Attention models introduce the Temporal Attention Mechanism and the Spatial Attention Mechanism into the BN-Inception model, where the BN-Inception model is used as the baseline. The classification performances of the Spatial-Temporal Attention models and the baseline model are reported in Tab. 3.

As shown in Tab. 3, all of the Spatial-Temporal Attention models outperform the baseline model. In particular, the STA-$TS(3\times3,3\times1)$ model outperforms the BN-Inception model by over 2% fed with RGB and flow frames respectively. On comparing the fusing results of Spatial ConvNet and Temporal ConvNet, the STA-$TS(3\times3,3\times1)$ model outperforms the BN-Inception model by 2.5%. These experimental results indicate that the proposed STA-CNN deep model is very effective for action recognition.

TABLE 4
Comparing with current state-of-the-art methods.

Models	UCF-101	HMDB-51
Two-stream [18] (NIPS 2014)	88.0%	59.4%
TSN [20] (TPAMI 2018)	94.2%	69.4%
Fusion Two-stream [21] (CVPR 2016)	93.5%	69.2%
Action Visual Attention [8] (NIPS 2015)	-	42.3%
Hierarchical Attention [9] (arXiv 2016)	92.7%	64.3%
Attention VideoLSTM [10] (CVIU 2018)	91.5%	63.0%
Collaborate Two-stream [42] (CSVT 2018)	94.0%	68.7%
Pyramid Attention [43] (ECCV 2018)	95.5%	70.7%
Attention Clusters [44] (CVPR 2018)	94.6%	69.2%
Interpretable ST-attention [16] (ICCV 2019)	87.1%	53.1%
C3D [27] (ICCV 2015)	85.2%	-
LTC-CNN [30] (PAMI 2017)	92.7%	67.2%
MiCT-Net [45] (CVPR 2018)	94.7%	70.5%
ResNet-3DCNNs [46] (CVPR 2018)	90.7%	63.8%
KVM [47] (CVPR 2016)	93.1%	63.3%
CoViAR [48] (CVPR 2018)	94.9%	70.2%
ARTNet [49] (CVPR 2018)	94.3%	70.9%
PBNet-8-4+iDT [50] (TIP 2019)	95.1%	**72.5**%
STMN+iDT [51] (TIP 2019)	94.5%	70.2%
STA-CNN (RGB+flow)	95.3%	70.2%
STA-CNN (RGB+flow+warpflow)	**95.8%**	71.5%

4.5 Comparison with State-of-the-art

The STA-CNN deep model is compared with current state-of-the-art methods on the UCF-101 and HMDB-51 datasets in Tab. 4. The results of the STA-CNN model are obtained by averaging the standard three splits provided in [40], [41]. When fusing the softmax scores of Spatial ConvNet and Temporal ConvNet, the STA-CNN model achieves the state-of-the-art performance on both of the datasets. To improve the classification performance further, we fuse the softmax scores of three networks fed with RGB, flow and warped flow frames respectively. The resulting STA-CNN model achieves 95.8% on the UCF-101 dataset and 71.5% on the HMDB-51 dataset. Compared with Two-stream based methods, the STA-CNN model outperforms the Two-stream [18] by 7.8% and 11.2% on the UCF-101 and HMDB-51 datasets respectively. It also outperforms the Temporal Segment Network (TSN) model [20] by 1.6% and 2.1% on the two datasets respectively. Then compared with the visual attention based methods, our STA-CNN model significantly outperforms all the LSTM based visual attention models [8], [9], [10] on the two datasets. The STA-CNN model outperforms the recently proposed spatial-temporal attention model [16] by 8.7% and 18.4% on UCF-101 and HMDB-51 respectively. The STA-CNN model also outperforms the CNN based attention models, such as the Collaborative Two-stream model [**?**], the Pyramid Attention Network [43] and the Attention Clusters [44]. Moreover, our STA-CNN model outperforms the 3D convolutional models on both datasets, such as the C3D [27], the LTC-CNN [30] and the MiCT-Net [45]. Also, the result of the STA-CNN model is better than the results of the recent methods such as ARTNet [49], PBNet-8-4+iDT [50] and STMN+iDT [51].

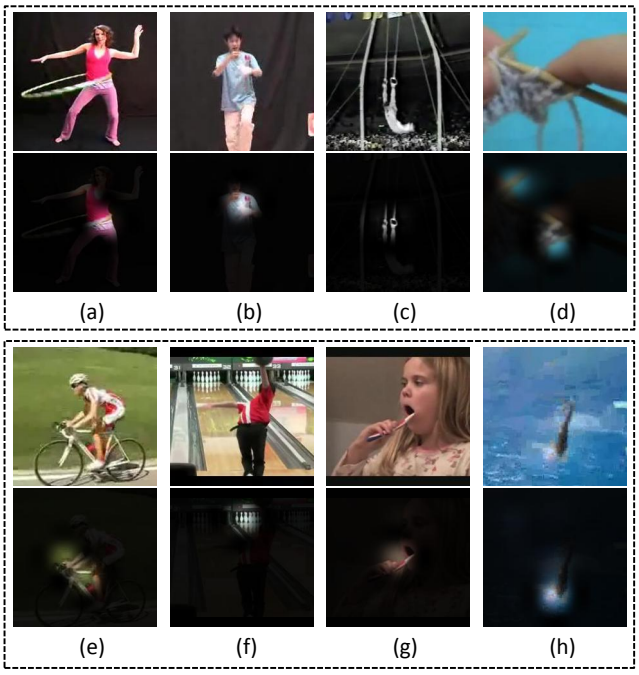


Fig. 7. Eight video examples in the UCF-101 video dataset. In each example, the top image is the original RGB frame, and the bottom one is the corresponding Spatial Attention Map enhanced frame.

4.6 Visualization

To get an intuitive understanding of the Spatial Attention Mechanism, the Spatial Attention Map generated by the Spatial Attention Network is visualized in Fig. 1. Firstly, the Spatial Attention Map is resized to the input size, 224×224. Then the resized Spatial Attention Map is used to weight each channel of the corresponding RGB frame. In Fig. 1, the top row shows four action examples from the UCF-101 dataset, and the bottom row shows the corresponding Spatial Attention Map weighted RGB frames. For example, in (a), the Spatial Attention Network pays attention on the eyelid as shown in (e). In the example (d), the attention model focuses on the soccer ball and foot as shown in (h). These examples indicate that the Spatial Attention Network locates the motion salient regions and discriminative non-motion regions accurately. More visualizing examples of the Spatial Attention Map are shown in Fig. 7.

Also, we visualize the discriminative confidences of video temporal segments in Fig. 2 to give an intuitive understanding of the Temporal Attention Mechanism. Firstly, ten video segments are sampled from each video with equal time intervals. Then, the discriminative confidence is computed for each segment by Eqn. 1. In the top row of Fig. 2, the first few segments are not suitable for classifying the action "Biking" because the rider and the bike have not yet appeared in the scene. In the bottom row of Fig. 2, the confidences of the first segments are relatively low, because a person running in a sports ground appears in "HighJump", "JavelinThrow", "PoleVault", "SkipJump" and so on. When the athlete jumps from the take-off line, the discriminative confidence clearly increases. These examples indicate that the Temporal Attention Mechanism can mine the discriminative temporal segments accurately.

4.7 Computational Complexity Analysis

The attention mechanism based methods [8], [9], [10], [11], [12], [13], [14], [15], [16], [42], [43], [44] improve the performance of action recognition but at the same time increase computational complexity. However, the computational complexity of our attention method is acceptable. Firstly, Our STA-CNN model is based on 2D-CNN and follows the two-stream architectures [18], [19], [20]. It has much lower computational complexity than RNN (LSTM) based methods [8], [9], [10], [16], [52] and 3D-CNN based methods [27], [28], [29], [30]. Then, compared with the 2D-CNN based action recognition models [18], [19], [20], on the one hand, the Temporal ConvNet of our STA-CNN model does not consume additional computing resources, because the Spatial Attention Network shares weights with Action Classification Network when it is fed with optical flow fields and the Spatial Attention Network reuses the features extracted by Action Classification Network. On the other hand, the Spatial ConvNet of our STA-CNN model does need additional computing resources for the Spatial Attention Network to extract flow features from RGB images. In the future work, we will try to further reduce the computational complexity of the STA-CNN by using more efficient deep networks such as MobileNet [53] and shuffleNet [54], or using model compression such as distillation [55] and pruning [56].

5 Conclusions

This paper has proposed a Spatial-Temporal Attentive Convolutional Neural Network (STA-CNN), which incorporates the Temporal Attention Mechanism and the Spatial Attention Mechanism into a unified convolutional network for action recognition. The Temporal Attention Mechanism automatically mines the discriminative temporal segments from long and noisy videos based on the discriminative confidence of each segment. The Spatial Attention Mechanism exploits both the motion information in flow features and the discriminative information learned from classification loss to locate the informative spatial regions in video frames. The proposed STA-CNN model is able to mine the discriminative segments in the temporal domain and at the same time focus on the informative regions in the spatial domain. It has achieved the state-of-the-art performance on two of the most challenging action recognition datasets.

Acknowledgments

This work is partly supported by the 973 basic research program of China (Grant No. 2014CB349303), the Natural Science Foundation of China (Grant No. U1636218, 61472420, 61472063, 61370185, 61472421, 61672519, 2017YFB1002801, 61100099), the Strategic Priority Research Program of the CAS (Grant No. XDB02070003), and the CAS External cooperation key project.

REFERENCES

[1] S. Yi, H. Li, and X. Wang, "Pedestrian behavior modeling from stationary crowds with applications to intelligent surveillance," *IEEE Transactions on Image Processing*, vol. 25, no. 9, pp. 4354–4368, 2016.

[2] S. S. Rautaray and A. Agrawal, "Vision based hand gesture recognition for human computer interaction: a survey," *Artificial Intelligence Review*, vol. 43, no. 1, pp. 1–54, 2015.

[3] I. Rodomagoulakis, N. Kardaris, V. Pitsikalis, E. Mavroudi, A. Katsamanis, A. Tsiami, and P. Maragos, "Multimodal human action recognition in assistive human-robot interaction," in *IEEE International Conference on Acoustics, Speech and Signal Processing*, 2016, pp. 2702–2706.

[4] C. Zhuang, H. Zhou, and S. Sakane, "Learning by showing: An end-to-end imitation leaning approach for robot action recognition and generation," in *IEEE International Conference on Robotics and Biomimetics*, 2016, pp. 173–178.

[5] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "ImageNet classification with deep convolutional neural networks," in *Advances in Neural Information Processing Systems*, 2012, pp. 1097–1105.

[6] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *IEEE Conference on Computer Vision and Pattern Recognition*, 2016, pp. 770–778.

[7] C. Szegedy, W. Liu, Y. Jia, P. Sermanet, S. Reed, D. Anguelov, D. Erhan, V. Vanhoucke, and A. Rabinovich, "Going deeper with convolutions," in *IEEE Conference on Computer Vision and Pattern Recognition*, 2015, pp. 1–9.

[8] S. Sharma, R. Kiros, and R. Salakhutdinov, "Action recognition using visual attention," in *Advances in Neural Information Processing Systems Workshops*, 2015.

[9] Y. Wang, S. Wang, J. Tang, N. O'Hare, Y. Chang, and B. Li, "Hierarchical attention network for action recognition in videos," *arXiv preprint arXiv:1607.06416*, 2016.

[10] Z. Li, K. Gavrilyuk, E. Gavves, M. Jain, and C. G. Snoek, "VideoLSTM convolves, attends and flows for action recognition," *Computer Vision and Image Understanding*, vol. 166, pp. 41–50, 2018.

[11] G. Chéron, I. Laptev, and C. Schmid, "P-CNN: Pose-based CNN features for action recognition," in *IEEE International Conference on Computer Vision*, 2015, pp. 3218–3226.

[12] W. Du, Y. Wang, and Y. Qiao, "Rpan: An end-to-end recurrent pose-attention network for action recognition in videos," in *IEEE Conference on Computer Vision and Pattern Recognition*, 2017, pp. 3725–3734.

[13] F. Baradel, C. Wolf, and J. Mille, "Human action recognition: Pose-based attention draws focus to hands," in *IEEE International Conference on Computer Vision Workshop*, 2017.

[14] G. Gkioxari, R. Girshick, and J. Malik, "Contextual action recognition with R*CNN," in *IEEE International Conference on Computer Vision*, 2015, pp. 1080–1088.

[15] A. Mallya and S. Lazebnik, "Learning models for actions and person-object interactions with transfer to question answering," in *European Conference on Computer Vision*. Springer, 2016, pp. 414–428.

[16] L. Meng, B. Zhao, B. Chang, G. Huang, W. Sun, F. Tung, and L. Sigal, "Interpretable spatio-temporal attention for video action recognition," in *IEEE International Conference on Computer Vision Workshop*, 2019, pp. 1–10.

[17] S. Hochreiter and J. Schmidhuber, "Long short-term memory," *Neural computation*, vol. 9, no. 8, pp. 1735–1780, 1997.

[18] K. Simonyan and A. Zisserman, "Two-stream convolutional networks for action recognition in videos," in *Advances in Neural Information Processing Systems*, 2014, pp. 568–576.

[19] L. Wang, Y. Xiong, Z. Wang, and Y. Qiao, "Towards good practices for very deep Two-stream ConvNets," *arXiv preprint arXiv:1507.02159*, 2015.

[20] L. Wang, Y. Xiong, Z. Wang, Y. Qiao, D. Lin, X. Tang, and L. Van Gool, "Temporal segment networks for action recognition in videos," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 41, no. 11, pp. 2740–2755, 2018.

[21] C. Feichtenhofer, A. Pinz, and A. Zisserman, "Convolutional Two-stream network fusion for video action recognition," in *IEEE Conference on Computer Vision and Pattern Recognition*, 2016, pp. 1933–1941.

[22] B. Zhou, A. Khosla, A. Lapedriza, A. Oliva, and A. Torralba, "Learning deep features for discriminative localization," in *IEEE Conference on Computer Vision and Pattern Recognition*, 2016, pp. 2921–2929.

[23] Z. Bolei, A. Khosla, A. Lapedriza, A. Oliva, and A. Torralba, "Object detectors emerge in deep scene CNNs," in *International Conference on Learning Representations*, 2015.

[24] K. Simonyan and A. Zisserman, "Very deep convolutional networks for large-scale image recognition," in *International Conference on Learning Representations*, 2015, pp. 1–14.

[25] G. Huang, Z. Liu, L. van der Maaten, and K. Q. Weinberger, "Densely connected convolutional networks," in *IEEE Conference on Computer Vision and Pattern Recognition*, 2017, pp. 4700–4708.

[26] A. Karpathy, G. Toderici, S. Shetty, T. Leung, R. Sukthankar, and F.-F. Li, "Large-scale video classification with convolutional neural networks," in *IEEE Conference on Computer Vision and Pattern Recognition*, 2014, pp. 1725–1732.

[27] T. Du, L. Bourdev, R. Fergus, and L. Torresani, "Learning spatiotemporal features with 3D convolutional networks," in *IEEE International Conference on Computer Vision*, 2015, pp. 4489–4497.

[28] S. Ji, W. Xu, M. Yang, and K. Yu, "3D convolutional neural networks for human action recognition," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 35, no. 1, pp. 221–231, 2013.

[29] J. Carreira and A. Zisserman, "Quo vadis, action recognition? a new model and the kinetics dataset," in *IEEE Conference on Computer Vision and Pattern Recognition*, 2017, pp. 4724–4733.

[30] G. Varol, I. Laptev, and C. Schmid, "Long-term temporal convolutions for action recognition," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. PP, no. 99, pp. 1–8, 2017.

[31] D. Bahdanau, K. Cho, and Y. Bengio, "Neural machine translation by jointly learning to align and translate," in *International Conference on Learning Representations*, 2015.

[32] V. Mnih, N. Heess, A. Graves *et al.*, "Recurrent models of visual attention," in *Advances in Neural Information Processing Systems*, 2014, pp. 2204–2212.

[33] J. Ba, V. Mnih, and K. Kavukcuoglu, "Multiple object recognition with visual attention," in *International Conference on Learning Representations*, 2015.

[34] K. Xu, J. Ba, R. Kiros, K. Cho, A. Courville, R. Salakhudinov, R. Zemel, and Y. Bengio, "Show, attend and tell: Neural image caption generation with visual attention," in *International Conference on Machine Learning*, 2015, pp. 2048–2057.

[35] S. Vladyslav, A. Karteek, and S. Cordelia, "Focused attention for action recognition," in *British Machine Vision Conference*, 2019, pp. 1–13.

[36] A. Tran and L.-F. Cheong, "Two-stream flow-guided convolutional attention networks for action recognition," *IEEE International Conference on Computer Vision*, pp. 3110–3119, 2017.

[37] E. Park, X. Han, T. L. Berg, and A. C. Berg, "Combining multiple sources of knowledge in deep CNNs for action recognition," in *IEEE Winter Conference on Applications of Computer Vision*, 2016, pp. 1–8.

[38] A. Dosovitskiy, P. Fischer, E. Ilg, P. Hausser, C. Hazirbas, V. Golkov, P. van der Smagt, D. Cremers, and T. Brox, "FlowNet: Learning optical flow with convolutional networks," in *IEEE International Conference on Computer Vision*, 2015, pp. 2758–2766.

[39] J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, and F.-F. Li, "ImageNet: A large-scale hierarchical image database," in *IEEE Conference on Computer Vision and Pattern Recognition*, 2009, pp. 248–255.

[40] K. Soomro, A. R. Zamir, and M. Shah, "UCF101: A dataset of 101 human actions classes from videos in the wild," *arXiv preprint arXiv:1212.0402*, 2012.

[41] H. Kuehne, H. Jhuang, E. Garrote, T. Poggio, and T. Serre, "HMDB: a large video database for human motion recognition," in *IEEE International Conference on Computer Vision*, 2011, pp. 2556–2563.

[42] Y. Peng, Y. Zhao, and J. Zhang, "Two-stream collaborative learning with spatial-temporal attention for video classification," *IEEE Transactions on Circuits and Systems for Video Technology*, 2018.

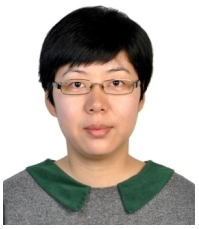
[43] Y. Du, C. Yuan, B. Li, L. Zhao, Y. Li, and W. Hu, "Interaction-aware spatio-temporal pyramid attention networks for action classification," in *European Conference on Computer Vision*. Springer, 2018, pp. 388–404.

[44] X. Long, C. Gan, G. de Melo, J. Wu, X. Liu, and S. Wen, "Attention clusters: Purely attention based local feature integration for video classification," in *IEEE Conference on Computer Vision and Pattern Recognition*, 2018, pp. 7834–7843.

[45] Y. Zhou, X. Sun, Z.-J. Zha, and W. Zeng, "MiCT: Mixed 3d/2d convolutional tube for human action recognition," in *IEEE Conference on Computer Vision and Pattern Recognition*, 2018, pp. 449–458.

[46] K. Hara, H. Kataoka, and Y. Satoh, "Can spatiotemporal 3D CNNs retrace the history of 2D CNNs and ImageNet," in *IEEE Conference on Computer Vision and Pattern Recognition*, 2018, pp. 18–22.
[47] W. Zhu, J. Hu, G. Sun, X. Cao, and Y. Qiao, "A key volume mining deep framework for action recognition," in *IEEE Conference on Computer Vision and Pattern Recognition*, 2016, pp. 1991–1999.
[48] C.-Y. Wu, M. Zaheer, H. Hu, R. Manmatha, A. J. Smola, and P. Krähenbühl, "Compressed video action recognition," in *IEEE Conference on Computer Vision and Pattern Recognition*, 2018, pp. 6026–6035.
[49] L. Wang, W. Li, W. Li, and L. Van Gool, "Appearance-and-relation networks for video classification," in *IEEE Conference on Computer Vision and Pattern Recognition*, 2018, pp. 6546–6555.
[50] W. Huang, L. Fan, M. Harandi, L. Ma, H. Liu, W. Liu, and C. Gan, "Toward efficient action recognition: Principal backpropagation for training two-stream networks," *IEEE Transactions on Image Processing*, vol. 28, no. 4, pp. 1773–1782, 2019.
[51] C. Li, B. Zhang, C. Chen, Q. Ye, J. Han, G. Guo, and R. Ji, "Deep manifold structure transfer for action recognition," *IEEE Transactions on Image Processing*, vol. 28, no. 9, pp. 4646–4658, 2019.
[52] J. Donahue, L. Anne Hendricks, S. Guadarrama, M. Rohrbach, S. Venugopalan, K. Saenko, and T. Darrell, "Long-term recurrent convolutional networks for visual recognition and description," in *IEEE Conference on Computer Vision and Pattern Recognition*, 2015, pp. 2625–2634.
[53] A. G. Howard, M. Zhu, B. Chen, D. Kalenichenko, W. Wang, T. Weyand, M. Andreetto, and H. Adam, "Mobilenets: Efficient convolutional neural networks for mobile vision applications," *arXiv preprint arXiv:1704.04861*, 2017.
[54] X. Zhang, X. Zhou, M. Lin, and J. Sun, "ShuffleNet: An extremely efficient convolutional neural network for mobile devices," in *IEEE conference on computer vision and pattern recognition*, 2018, pp. 6848–6856.
[55] G. Hinton, O. Vinyals, and J. Dean, "Distilling the knowledge in a neural network," in *Advances in Neural Information Processing Systems Workshop*, 2014.
[56] T.-J. Yang, Y.-H. Chen, and V. Sze, "Designing energy-efficient convolutional neural networks using energy-aware pruning," in *IEEE Conference on Computer Vision and Pattern Recognition*, 2017, pp. 5687–5695.

Hao Yang received the Ph.D degree from the Institute of Automation, Chinese Academy of Sciences (CASIA), Beijing, China, in 2019. Currently, he is working as an algorithm research fellow in the R&D Center of Artificial Intelligent, Nuctech Company Limited, Beijing, China. His research interests include deep learning, motion analyses and action recognition.



Chunfeng Yuan received the PH.D. degree from the Institute of Automation, Chinese Academy of Sciences (CASIA), Beijing, China, in 2010. She was a visiting scholar at University of Adelaide, Australia in 2010, and in the Internet Media Group and the Media Computing Group at Microsoft Research Asia in 2016. She is currently a associate professor at the CASIA. Her research interests and publications range from statistics to computer vision, including sparse representation, deep learning, action recognition, and event detection.

Li Zhang received her academic degrees from Tsinghua University and China Institute of Atomic Energy. Currently she is a full professor in Department of Engineering Physics, Tsinghua University. She has published more than 120 papers on international journals and conferences and has been granted more than 100 patents worldwide. Her research interests include radiation imaging theory, 3D reconstruction and visual analysis.

Yunda Sun received the Ph.D degree from Beijing Jiaotong University, China in 2006. He has been granted more than 40 patents worldwide. Currently, he is working as a team leader in the R&D Center of Artificial Intelligence, Nuctech Company Limited. His research interests include visual analysis and object recognition.

Weiming Hu received the Ph.D. degree from the Department of Computer Science and Engineering, Zhejiang University in 1998. From 1998 to 2000, he was a postdoctoral research fellow with the Institute of Computer Science and Technology, Peking University. Currently, he is a full professor in the Institute of Automation, Chinese Academy of Sciences. He has published more than 200 papers on international journals and conferences. His research interests include visual motion analysis and recognition of web objectionable information.

Stephen J. Maybank received the BA degree in mathematics from Kings College Cambridge in 1976, and the Ph.D. degree in computer science from Birkbeck College, University of London in 1988. He is currently a professor in the Department of Computer Science and Information Systems, Birkbeck College. His research interests include the geometry of multiple images, camera calibration, visual surveillance, etc. He is a fellow of the IEEE.