IEEE Access
Multidisciplinary | Rapid Review | Open Access Journal

Received June 27, 2018, accepted August 3, 2018, date of current version August 28, 2018.

Digital Object Identifier 10.1109/ACCESS.2018.2863943

Human Action Recognition Based on Selected Spatio-Temporal Features via Bidirectional LSTM

WENHUI LI, WEIZHI NIE, AND YUTING SU
School of Electrical and Information Engineering, Tianjin University, Tianjin 300072, China

Corresponding author: Weizhi Nie (weizhinie@tju.edu.cn)

This work was supported in part by the National Natural Science Foundation of China under Grant 6147227, Grant 61303208, and Grant 61502337, in part by the Tianjin Research Program of Application Foundation and Advanced Technology under Grant 15JCYBJC16200, and in part by the Elite Scholar Program of Tianjin University under Grant 2014XRG-0046.

ABSTRACT Recently, many deep convolutional networks approaches have been proposed for human action recognition. The challenge is to capture complementary information from still and motion frames. In contrast to previous models, which use holistic clips to model spatial information and traditional temporal information for sequential processing, in this paper, we propose a novel framework that can select the discriminative part in the spatial dimension and enrich the modeling action of motion in the temporal dimension. We utilize part selection within clips and consider the bidirectional temporal information when modeling the temporal pattern using multiple layers of a long short-term memory framework, which can learn compositional representations in space and time. Our results are evaluated on the standard benchmarks UCF101 and HMDB51 and show that the proposed architecture achieves state-of-the-art results.

INDEX TERMS Spatio-temporal part selection, LSTM, human action recognition.

I. INTRODUCTION

Recently, there has been a significant increase in human activity data on the internet, resulting in an increasing number of videos. Human action recognition aims to enable computers to automatically recognize human action in real-world videos. This is a fundamental and challenging task of computer vision. Many researchers and groups have worked on video analysis for decades and have tackled different problems, such as action recognition, event detection, and activity understanding. Considerable research progress has been made in these individual problems by using different specific solutions. However, there is still a growing need for a generic video descriptor that can help solve the task of large-scale human action recognition.

Impressive progress has been achieved via supervised convolutional neural network models for retrieval [1] and recognition tasks [2], and several extensions to process video have been recently proposed. Ideally, a video model should allow variable length input sequences to be processed. The challenge of model action patterns is how to model the motion pattern of between frames. In this paper, we propose a novel framework to mine both spatial and temporal information and combine the spatiotemporal convolutional layers and long-range temporal recursion (Figure 1). Our key insight to improve the recognition performance is to utilize discriminative cubes of clips, which contain short convolutional temporal information and a long short-term memory framework to model human actions. Research on CNN models for video processing has considered learning 3D spatiotemporal filters over raw sequence data [3], [4] and learning frame-to-frame representations, which incorporate aggregated instantaneous optical flow or trajectory-based models over fixed windows or video shot segments [5], [6]. Such models explore the two extrema of perceptual time-series representation by learning: a fully general time-varying weighting or the application of simple temporal pooling. Following the same inspiration that has motivated the current deep convolutional models, we advocate for video recognition and description models that are also deep over temporal dimensions, i.e., with the temporal recurrence of latent variables. Recurrent Neural Network (RNN) models are explicitly "deep in time" such that they can be unrolled and form implicit compositional representations in the time domain. Such "deep" models predated deep spatial convolution models in the literature [7], [8]. The use of RNNs in perceptual applications has been explored for many decades, with varying results. A significant limitation of simple RNN models, which strictly integrates state information over time,

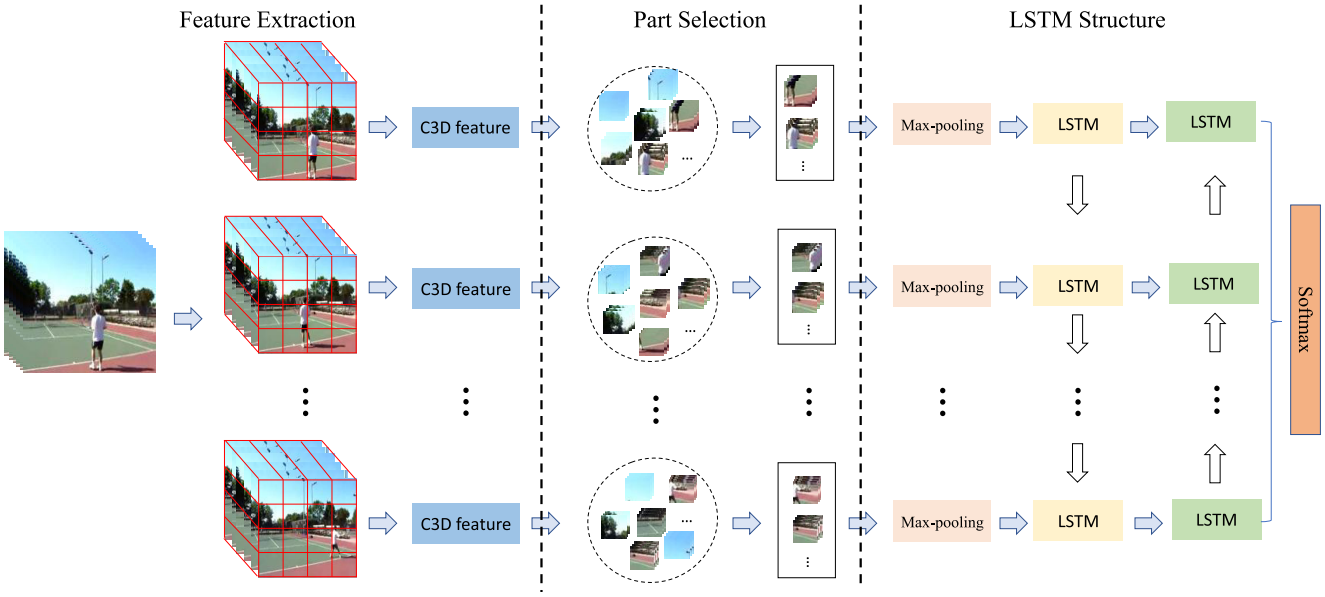


FIGURE 1. Overview of the proposed framework, the class of architectures leveraging the strengths of the rapid progress in CNNs for the visual recognition problem, and the growing desire to apply such models to time-varying inputs and outputs. We split the videos into multiple clips with fixed-length frames in the temporal direction and use the sliding windows to generate local cubes in the spatial direction. The cubes are extracted by using C3D network [9]. Then, we design the part selection method to select useful cubes and utilize the max-pooling function to generate the representation of clips. Finally, the representation of clips is fed into two layers of the LSTM network to train the multiple LSTM and the softmax layer yields the prediction by using the feature vector generated from the LSTM outputs.

is known as the "vanishing gradient" effect: the ability to backpropagate an error signal through a long-range temporal interval has become increasingly difficult in practice. Long Short-Term Memory (LSTM) units, first proposed in [10], are recurrent modules that enable long-range learning. LSTM units have hidden states that are augmented with nonlinear mechanisms to allow the state to propagate without modification, be updated, or be reset by using simple learned gating functions. LSTMs have recently been demonstrated to be capable of large-scale learning of speech recognition [11] and language translation models [12], [13].

We instantiate our proposed architecture in terms of two aspects. The firsst one is that we train the part detectors to select useful action parts by using the weights calculated from a classifier and combine the local parts and global clips to enrich the action representation. The second is to develop three LSTM settings (Figure 2) to model the temporal action patterns. First, we show that by directly connecting a visual convolutional model to deep LSTM networks, we can train video recognition models that captures temporal state dependencies. While the existing labeled video activity datasets may not have actions or activities with particularly complex temporal dynamics, we nevertheless observe significant improvements on conventional benchmarks. Second, we explore multiple layers of the LSTM structure. The spatiotemporal features are fed into the first layer of LSTM to learn the hidden states from the input feature to the output units, and the units are utilized in the input of second layer. Compared to the second setting, the difference in the third structure is the second layer of LSTM. We use the backward direction to model the high-level temporal information. Finally, we show that LSTM decoders can be driven directly from the conventional computer vision methods, which predict higher-level discriminative labels, such as the semantic video role tuple predictors in [14].

The contributions of this paper are as follows:

- Different to the exsiting methods, which directly extract feature from holistic clip, we combine the segmented frames in the temporal domain and the local key information in the spatial domain to extract the feature to represent the human action information, which can achieve more robust visual features;
- We utilize three LSTM structures to model the local temporal patterns of human action. The LSTM is utilized in this paper to model the temporal information of the selected cubes and generate global information from spatiotemporal patterns;
- The experimental results of UCF101 and HMDB51demonstrate the performance of our method and the related experiments also demonstrate the effectiveness of our framework.

This paper is organized as follows. In Section 2, we provide a brief overview of the literature related to activity recognition from feature extraction and LSTM modeling. In Section 3, we elaborate on the details of the proposed preserve local and global spatiotemporal structure for action recognition. In Section 4, we evaluate the performance of the approach and end Section 5 with the conclusion of the work.

II. RELATED WORK

Human action recognition is a very active research field, and many approaches have been proposed over the last decade.

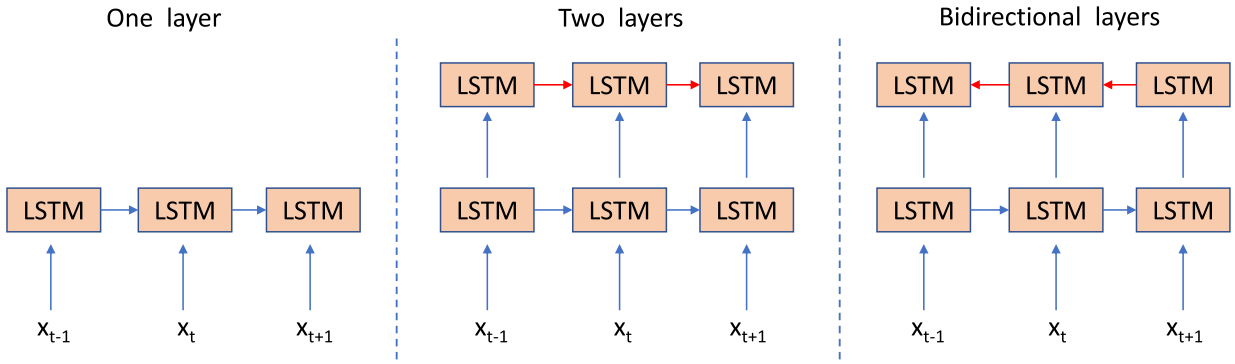


FIGURE 2. The three different structures of LSTM.

In this section, we briefly discuss the various approaches adopted to solve the problem of action recognition.

A. FEATURE EXTRACTION

1) HAND-CRAFTED FEATURE

Action recognition methods have been proposed by encoding local spatiotemporal features. Laptev *et al.* [15] proposed a spatial-temporal interest point (STIP) feature, which extended to the Harris *et al.* [16] operator from spatial images to spatiotemporal ones. Reference [17] applied to the 2D-Gaussian smoothing kernel along the spatial dimensions and 1D Gabor filters along the temporal dimensions to generate Cuboids feature. Moreover, [18] densely sampled feature points and used the optical flow to track the location of the sampled points. The features were then encoded into the Bag of Features (BoF) histograms or Fisher Vector representation and combined with an SVM classifier [19]. While typical pipelines resemble earlier methods for object recognition, the use of local motion features, Motion Boundary Histograms [18], has been found important for action recognition in practice. Explicit representations of the temporal structure of actions have rarely need used, with some exceptions, such as that found in recent work [20]. To make feature more robustness, [21] used orthogonal version of locality preserving projections to learn more discriminative feature by effectively reducing the feature dimension.

2) 2D CNN-BASED FEATURE

Learning visual representations with CNNs has shown a marked advantage over ''hand-crafted'' features for many recognition tasks in static images [22]–[24]. Compared to image data domains, there is relatively little work applying CNNs to video classification. Since all successful applications of CNNs in image domains share the availability of a large training set, the action recognition also contains big data. Extensions of the CNN representations to action recognition in video have been proposed in several recent works. Some of these methods encode single video frames with static CNN features [5], [6], [25]. Extensions to short video clips where video frames are treated as multi-frame inputs to 2D CNNs have also been investigated in [5], [6], [26], and [27]. For example, [5] finds that CNN architectures are capable of learning powerful features from weakly-labeled data that far surpass feature-based methods in terms of performance. These benefits are surprisingly robust to details of the connectivity of the architecture in time.

Many works have addressed the learning CNN representation for action recognition by raw RGB pixel input and precomputed the optical flow features. Motion-based CNNs typically outperform the CNN representations learned from RGB inputs. Zhang *et al.* [28] transfer information from optical flow algorithms to motion vector representations in compressed videos without an extra calculation. Reference [6] propose a deep video classification model that incorporates separate spatial and temporal recognition streams based on 2D convolutional networks. Similar settings can be found in [26], where training strategies were designed to evaluate the very deep two-stream ConvNets for action recognition. By implementing designed different fusion strategies, [27] found different fusion locations in ConvNet towers have a lot of influence on performance and finally show the importance of the learning fusion correspondences between the highly abstract ConvNet features both spatially and temporally.

3) 3D SPATIOTEMPORAL FEATURE

Most of the current CNN methods use only architecture with 2D spatial convolutions, but the invariance to translations in time is also important for action recognition because the beginning and end actions are generally unknown. Recently, several works proposed 3D spatiotemporal convolutional structures to address this issue and provided a natural extension of 2D CNNs to video. Unlike the 2D filters used in image classification, 3D convolutional filters can learn temporal relationships from continuous RGB frames and enrich the representation information. Reference [4] modeled structural features from both the spatial and temporal dimensions by performing 3D convolutions ($7\times7\times4$). The developed deep architecture generates multiple channels of information from adjacent input frames and perform convolution and

sub-sampling separately in each channel. The final feature representation is obtained by combining the information from all channels. Another work [9] learns 3D ConvNets on a limited temporal support of 16 consecutive frames with all filter kernels of size 3×3×3 to model appearance and motion simultaneously. Then, they empirically show that these learned features with a simple linear classifier can yield good performance on various video analysis tasks.

Recent work showed that using temporal structure in video can enrich the discriminative information for video representation. These methods, however, learn video temporal representations. They only use the LSTM structure to handle full temporal scales. For example, [29] trained a network that combines CNNs and LSTMs for activity recognition. In this work, we extend 3D CNNs to significantly longer temporal convolutions that enable action representation at their full temporal scale by combining the local (spatiotemporal convolutional filters) and longer temporal (LSTM) information, We also explore the impact to adopt optical flow frames as input.

B. LONG SHORT-TERM MEMORY NEURAL NETWORKS

Standard recurrent neural networks (RNN) are a natural generalization of feedforward neural networks and model temporal dynamics by computing the hidden unit $h = (h_1, h_2, \ldots, h_T)$ from the input sequences $x = (x_1, x_2, \ldots, x_T)$ to output vector sequence $z = (z_1, \ldots, z_T)$ via the following equations:

$$h_t = g(W_{xh}x_t + W_{hh}h_{t-1} + b_h) \quad (1)$$
$$z_t = g(W_{hz}h_t + b_z) \quad (2)$$

where the W denotes the weight metrics, g is usually an elementwise application of a sigmoid function, h_t is the hidden state with N hidden units, and b is the bias term. Although RNNs can capture long-distance dependencies and are applied successfully to tasks such as text generation [30], machine translation [12], and speech recognition [31]. Thus, it is difficult to learn long-term dynamics because of gradient vanishing and exploding [32]. However, LSTMs [10] are known to be able to learn long-range temporal dependencies by incorporating explicitly controllable memory units. The most important part of LSTM model is using this memory cell c modulated by three gates (ranging [0,1]), input gates control in g whether the LSTM considers its current input, forget gates allowing the LSTM to forget its previous memory and output gates that determine how much of the information to transfer to the hidden state. This unit controls whether the network will learn when to ''forget'' previous hidden states (if the layer is evaluated as 0) and when to update hidden states (if it is evaluated as 1), given new information.

As research on LSTM has progressed, varying connections of the hidden units in the memory cell have been proposed. We use the LSTM unit as described in [32]. A slight simplification of the one described in [11]. The LSTM updates for time step t given inputs x_t,h_{t-1},and c_{t-1} are defined as follows:

$$i_t = \sigma(W_{xi}x_t) + W_{hi}h_{t-1} + b_i) \quad (3)$$
$$f_t = \sigma(W_{xf}x_t) + W_{hf}h_{t-1} + b_f) \quad (4)$$
$$c_t = f_t c_{t-1} + i_t tanh(W_{xc}x_t + W_{hc}h_{t-1} + b_c) \quad (5)$$
$$o_t = \sigma(W_{xo}x_t + W_{ho}h_{t-1} + W_{co}c_t + b_o) \quad (6)$$
$$h_t = o_t \odot tanh(c_t) \quad (7)$$

where $x \odot y$ denotes the element-wise product of vector x and y and σ is the sigmoidal non-linearity. Recently, analogous to CNNs, LSTMs have attracted attention because they allow end-to-end fine-tuning. For example, [11] trained a deep bidirectional LSTM that maps spectrogram inputs to text to eliminate the need for complex multi-step pipelines in speech recognition. Even with no language models or pronunciation dictionaries, the model produces convincing text translations. Reference [12] and [13] adopted a multilayer LSTM encoder and decoder to solve the language translation task. They mapped the source language to the hidden state by using encoding LSTM and mapped the hidden state to the target language by using decoding LSTM. Such an encoder-decoder scheme allows an input sequence of arbitrary length to be mapped to an output sequence of different length. Reference [33] use LSTM networks to learn representations of video sequences. An LSTM encoder is adopted to map an input sequence to a fixed-length representation, and the representation is decoded using single or multiple decoder LSTMs to perform different tasks, such as reconstructing the input sequence and predicting the future sequence. Reference [25] combined the CNN architecture and LSTM pipeline to represent the video and showed that the models provide improved recognition on conventional video activity challenges and enable a novel end-to-end optimizable mapping from image pixels to sentence-level natural language descriptions. Specifically, they found that convolutional neural networks with LSTM units are generally applicable to visual time-series modeling. In addition to the above method, much work has focused on using attention-based encoder-decoder framework computer vision tasks. Reference [34] used the informative joints in the action sequence with the assistance of global contextual information. They proposed a recurrent attention mechanism to iteratively improve attention performance that can selectively focus on the informative joints in the action sequence with the assistance of global contextual information. Reference [35] integrated hierarchical LSTMs, temporal attention and an attention mechanism to automatically determine when to use visual information or sentence context information.

The advantages of LSTMs for modeling temporal sequential data are apparent, greatly solving the gradient vanishing and exploding and using the memory cell to model dynamic information. However these methods consider the frame-based LSTM structure, which uses only the LSTM to model the global temporal dynamics in each frame. In this work, we extracted 3D spatiotemporal convolutional neural features to preserve the local temporal information and adopted part

selection strategy to select the important parts, then used the multiple layers of LSTM to model the global action dynamics. Both extensions show clear advantages in our experimental comparison to previous methods. The focus of our paper is to accelerate action recognition with LSTM structure while preserving the spatial and temporal information for better performance.

III. THE PROPOSED ARCHITECTURE

In this section, we first present feature extraction part in the architecture as illustrated in Figure 1. Then, we introduce the scheme of part selection. We finally provide details on the procedures to how to construct the multiple LSTM layers.

A. SPATIOTEMPORAL FEATURE EXTRACTION

Activity recognition is an instance of sequential learning tasks that is described in our framework as follows: each video is in a length T sequence and is split into a set of N clips $C = \{c_1, c_2, \ldots, c_N\}$ by extracting m sequential frames, where $length(c_i) = m$. Each clip is the input to a spatiotemporal convolutional network. The network has 5 convolutional layers, with 64, 128, 256, 512 and 512 response maps, followed by 3 fully connected layers with sizes 4096, 4096 and the number of classes. Following [9], we use $3 \times 3 \times 3$ as spatiotemporal filter size for all convolutional layers. Each convolutional layer is followed by a rectified linear unit (ReLU) and a space-time max pooling layer. The pooling filter size in the first layer is $2 \times 2 \times 1$ to keep the temporal information in the early stage, and it is $2 \times 2 \times 2$ in the other layers. The filter stride for all dimensions is 1 for convolution and 2 for the pooling stage. The size of convolution output is kept constant by using a pad of 1 pixel in all three dimensions. We use dropout(0.5) for the two fully connected layers. Fully connected layers are followed by ReLU layers. After learning the feature transformation $\phi(i)$, the clip sets C produce fixed-length (D) vectors $X = \{x_1, x_2, \ldots, x_N\}$, $x_i = \phi(i_1)\mathbb{R}^D$, $X \in \mathbb{R}^{DxN}$.

During training, videos are resized to 240×320, and we use 227×227 crops and mirror the frames in one clip to augment the dataset. To enrich the video information, we use both RGB and optical flow frames as inputs to our recognition system. Optical flow is generated, following [36], and transformed into a flow image by scaling and shifting the x and y flow values to a range from -128 to 128.

B. DISCRIMINATIVE PART SELECTION

Traditional spatiotemporal feature extraction treats a clip as holistic information. Compared to these methods, we want to explore the most useful parts within the clips and remove the background influence. Given N clips $C = \{c_1, c_2, \ldots, c_N\}$, we use a S×S×T size with a $S_p \times S_p \times T_p$ sampling interval from clip to generate cubes $c_i = \{v_1, v_2, \ldots, v_M\}$, where M is the number of cubes. These cubes contain a large amount of information, and our goal is to learn a compact collection of local part detectors that can distinguish a particular action category from others and remove the redundant background information.

To achieve the goals and select useful parts, we first extract features of the S×S×T cubes described in the next subsection and discover the n candidate part detectors $D_{original} = \{d_1, d_2, \ldots, d_n\}$ by applying a cross-validation strategy and an exemplar-linear discriminative analysis (Exemplar-LDA) [37]. Some background cubes are also considered candidates because we treat all clusters of spatiotemporal cubes as potential parts. Then, we select fewer detectors $D_s \subset D_{original}$, which contain less background noise and more of the action patterns. Given a part detector d_j, we run it on one video V and get the video representation $V_{d_j} = Max\left\{f_{d_j}^N\right\}$, and $f_{d_j} = \{s_1, s_2, \ldots, s_M\}$; s_i indicates the resulting detection score. Finally, the video for all detectors is indicated by $F_v = \left\{V_{d_1}, V_{d_2}, \ldots, V_{d_n}\right\}$.

1) MODELING PART SELECTION

For a specific action category k, we select all the cube samples of this category as a positive set and randomly select samples from all the samples of other categories as the negative set.The proportion of positive and negative samples was 1:1. The $D_c^k \subset D_{original}$ denotes the part set that we selected from a positive set of k category and all the positive videos represented by $F_v^+ = \left\{V_{d_1}, V_{d_2}, \ldots, V_{d_q}\right\}$; In contrast negative videos, which do not belong to this category, use $F_v^- = \left\{V_{d_1}, V_{d_2}, \ldots, V_{d_q}\right\}$. Now, we can use linear SVM to train the appropriate model for each action category, the objective function is

$$\phi(F) = WF + b \tag{8}$$

where W represents the weights and b is a bias. Finally, we use the weight vector of the classifier to rank the part detectors and the top 100 detectors are selected for each class.

2) PART SELECTION SCHEME

After training the model of the part selection by using the training data, we get the part detectors $D_s \subset D_{original}$, where D_r indicates the removed detectors and $D_s \cup D_r = D_{original}$. For the clip $c_i = \{v_1, v_2, \ldots, v_M\}$, we define the selection rule as follows $c_i^{new} = []$:

$$c_i^{new} = \begin{cases} c_i^{new} \cup v_i, & v_i \in D_s \\ c_i^{new}, & v_i \in D_r \end{cases} \tag{9}$$

When the cube v_i belongs to the set centered by one of the detectors $d_i \in D_s$, this cube is retained; Otherwise, the cube is removed. Because we need to guarantee the continuity of the temporal dimension in LSTM structure, if $\text{length}(c_i^{new}) = 0$ when using Eq. (9), which denotes that clip c_i does not have cubes that belongs to the positive detectors. Then, the cube v with the largest distance within the d_i will be added to c_i^{new}. The cube with largest distance is always the clear background, so if we choose this cube, it has little influence when we model the temporal pattern by using LSTM. After selecting cubes, we use max-pooling to unite the features of cubes

in one clip and combined the holistic clip feature to feed them into the LSTM structure which is introduced in next subsection.

C. TWO-LAYER LSTM STRUCTURE

In this section, we elaborate on the different settings of LSTM to model the long-term temporal information of human actions. In its most general form, a sequence model parametrized by W maps an input $\phi(i_1)$ and a previous timestep hidden state h_{t1} to an output z_T and updates the hidden state h_t. Therefore, the hidden states are sequentially computed in order: $h_1 = W(x_1 1, 0)$, $h_2 = W(x_1, h_1), \ldots, h_T = W(x_T, h_{T-1})$. This procedure is described in detail in Section II-B.

1) ONE LAYER

This setting of LSTM adopts the traditional configuration (See the left of Fig 2), and we provide the ordered clip features of one video as the input representation at each time step to the LSTM to model the temporal dynamics. We named this setting $LSTM_1$.

2) TWO LAYERS

Different the one layer setting, this configuration uses two layers LSTM (See the middle of Fig 2). The most straightforward way to construct the LSTM network is to stack multiple LSTM layers on top of each other. The first layer is represented by $h_1^1 = W^1(x_1, 0), h_2^1 = W^1(x_1, h_1^1), \ldots, h_T^1 = W^1(x_T, h_{T-1}^1)$. The second layer uses the output of first LSTM layer as the input to learn the high-level mapping: $h_1^2 = W^2(h_1^1, 0), h_2^2 = W^2(h_2^1, h_1^2), \ldots, h_T^2 = W^2(h_T^1, h_{T-1}^2)$. This stacked LSTM network can combine multiple levels with a long range context. We named this setting $LSTM_2$.

3) BIDIRECTIONAL LAYERS

This model structure setting is shown in Fig 2(right). In the second layer, the direction of LSTM is in the opposite direction as the first layer. The description is provided by the characterization $h_1^2 = W^2(h_T^1, 0)$, $h_2^2 = W^2(h_{T-1}^1, h_1^2), \ldots, h_T^2 = W^2(h_1^1, h_{T-1}^2)$. The structure of the two revised direction layers enriches the video representations from multiple levels and hybrid directions. We named this setting $LSTM_3$.

The features extracted in Section III-A are fed into the LSTM structure to model the global temporal information, and the model generates the outputs z_t. The final step in predicting the distribution is to take a soft-max over the outputs z_t of the sequential model. This allows the LSTM to spend its modeling capacity on more complex and longer term interactions instead of maintaining a summary of the recent frames in case it may be useful for the next few clips.

IV. EXPERIMENT

In this section, we evaluate our architecture using two popular human action datasets: UCF101 and HMDB51. In this experiment, we focus on multiple input configuration and different temporal modeling by using multiple LSTM layers. We also compare the proposed method against several popular methods, such as Improvement Dense Trajectory with Fisher Vector, C3D, LRCN, LSTM, Two-stream architecture, and we use the mean average precision as the evaluation metric.

A. DATASETS AND EXPERIMENTAL SETTING

UCF101 [38] is a widely used benchmark and consists of 13,320 videos that are categorized into 101 human action classes. The clips are 320 × 240 pixels, and they have a 25 fps frame rate that lasts 7 seconds, on average. The UCF101 dataset is split into three parts, with just under 8,000 videos in the training set for each split. We divide the first training split of UCF101 into a smaller training set and a validation set to explore different variants. HMDB51 [39] contains 6766 videos that have been annotated for 51 actions. The clips have a 320x240 resolution and a 30 fps frame rate. For both datasets, we evaluate both the RGB and optical flow images (Figure 3) that are generated by [36].

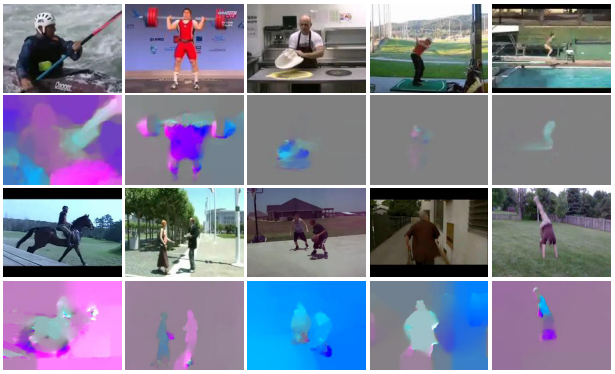


FIGURE 3. Sample action frames and optical flow images in the UCF101 (first two rows) and HMDB51 (last two rows) datasets.

1) FEATURE EXTRACTION AND PART SELECTION SETTING

Following the setting of [9], we use a pre-trained model from Spoirt-1M dataset to fine-tune the model on UCF101 and HDBM51 to extract the fc6 feature, which has 4096 dimensions. The size of cubes is 80 × 60 × 16 with a 40 × 30 × 8 sampling interval. For the spatial scale used to extract features, we resize all the frames from 80 × 60 to 320 × 240. There are no more than 300 selected parts for each action category. The final features of the clip are composed by the selected partial feature (4096 dimensions) and holistic clip (4096 dimensions) feature, we concatenate them as the clip representation, which is used as the input of LSTM layers and explores various settings for the multiple-layer LSTM activity recognition architecture.

2) LSTM STRUCTURE SETTING

According to Figure 4, the longest video lengths are over 256 frames in UCF101, but most have 64 to 96 frames. The different video lengths influence the input of LSTM,

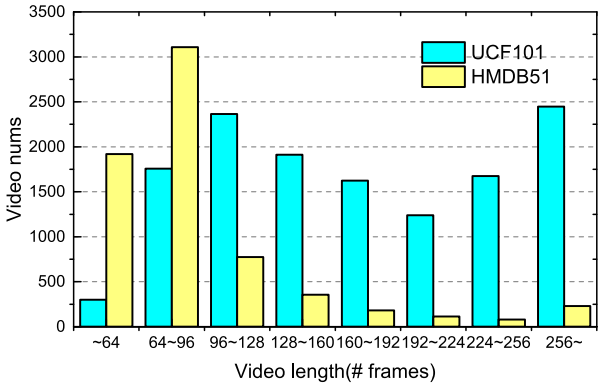


FIGURE 4. Statistics of frame numbers in the UCF101 (blue) and HMDB51 (yellow) datasets.

so we adopted 12 as the length of input clips for UCF101 (nearly half of the videos' lengths are longer than $12 \times 16 = 196$ frames). When the video length contains fewer than 160 frames, we pad the input feature of LSTM with zeros. Otherwise, if the video length is greater than 160 frames, we discard the extra part. The number of hidden states is evaluated from 256 to 4096 in Section IV-B2.

B. EVALUATION

In the following, we compared the influential hyper-parameters, including different input configurations, feature combinations, numbers of layers and LSTM directions. We evaluate our different settings on the UCF-101 and HMDB-51 action recognition benchmarks by using two types of input: RGB images and Optical flow images, which are shown from Table 1 to Table 5.

1) PERFORMANCE WITH PART SELECTION

The C3D structure uses holistic clips to learn the spatiotemporal features. We design the part selection scheme to select the important parts of every clip and combine the parts and holistic clips to extract the spatiotemporal features that represent the clip. The results are shown in Table 1 and Table 2. When we only use the selected parts to represent an action, we get improvement on UCF101 by 2.7% and on HMDB51 by 2.1% with RGB type. When we concatenated the holistic and part cubes as the input of LSTM, the performance of the design outperforms the original C3D setting by 5% on HMDB51 and 3.7% on UCF101 with RGB type. The concatenation of the holistic clips and selected parts is termed Combined-C3D. With optical flow images, the performance also has the improvement on the performance. From Figure 3, we find that the optical flow has a significant response when actions occur and that the background is nearly smooth and the partial selection selects the important parts which contain

TABLE 1. The accuracy of different feature settings on HMDB51.

Input Type	Model	Acc(%)
RGB	C3D	51.9
	Selected C3D	54.0
	Combined-C3D	56.9
Optical flow	C3D	49.1
	Selected C3D	52.8
	Combined-C3D	54.0

TABLE 2. The accuracy of different feature settings on UCF101.

Input Type	Model	Acc(%)
RGB	C3D	85.2
	Selected C3D	87.9
	Combined-C3D	88.9
Optical flow	C3D	77.1
	Selected C3D	79.0
	Combined-C3D	79.8

TABLE 3. The accuracy of different LSTM model structures on HMDB51.

Input Type	Model	Acc(%)
RGB	Combined-C3D+$LSTM_1$	62.4
	Combined-C3D+$LSTM_2$	64.7
	Combined-C3D+$LSTM_3$	68.2
Optical flow	Combined-C3D+$LSTM_1$	56.9
	Combined-C3D+$LSTM_2$	58.3
	Combined-C3D+$LSTM_3$	60.2
RGB+Optical flow	Combined-C3D+$LSTM_3$	70.4

TABLE 4. The accuracy of different LSTM model structures on UCF-101.

Input Type	Model	Acc(%)
RGB	Combined-C3D+$LSTM_1$	90.4
	Combined-C3D+$LSTM_2$	91.1
	Combined-C3D+$LSTM_3$	92.1
Optical flow	Combined-C3D+$LSTM_1$	81.1
	Combined-C3D+$LSTM_2$	82.7
	Combined-C3D+$LSTM_3$	84.3
RGB+Optical flow	Combined-C3D+$LSTM_3$	94.2

TABLE 5. Comparison of recognition results in percentage (%) based on UCF101 and HMDB51 datasets (Mean classification accuracy).

Method	UCF101	HMDB51
IDT+FV [18]	85.9	57.2
C3D [9]	85.2	-
Spatiotemporal Convnet [5]	65.4	-
Temporal Stream [6]	83.7	54.6
Factorized ConvNet [42]	88.1	59.1
LRCN [25]	82.7	-
ST-Resnet [44]	93.4	66.4
TDD [45]	91.5	65.9
TSN [46]	94.0	68.5
Two-Stream+LSTM [41]	88.6	-
Two-Stream (avg. fusion) [43]	86.9	58.0
Two-Stream(VGG16)[43]	91.4	58.5
Two-Stream Model [6]	88.0	59.4
Our method	94.2	70.4

action pattern to suppress the noise. The performance on the optical flow images has improved by 4.9% on HMDB51 and 2.7% on UCF101 dataset.

2) PERFORMANCE WITH DIFFERENT LSTM SETTING

This part shows the influence of different LSTM settings, including number of layers, different hidden states of LSTM and the reverse direction of two layers. Two classic LSTM structures are utilized in this part.

- Combined-C3D+$LSTM_1$ [25] indicates the popular settings that feed the features of temporal clips into the LSTM structure as input and learn the hidden states between the features and a specific category.
- Combined-C3D+$LSTM_2$ [40]:The stack LSTMs use the hidden state units to learn the first layer as the input for the second layer, and the second layer connects the class layers. This structure adds one more layer to adequately exploit the hidden information between the feature vector and category.

The $LSTM_2$ contains two layers that both follow the direction of actions. Thus, we design $LSTM_3$, which has the opposite direction to that of the two LSTM layers. The experiment results shown in Table 3 and Table 4 give the following observations.

- The performance of using $LSTM_1$ is higher than the average pooling temporal clips by 5.5% (RGB) and 2.9% (Optical flow) on HMDB51 and by 1.5% (RGB) and 1.3% (Optical flow) on UCF101. This occurs because when using average pooling, the sequence temporal information is directly merged together while considering the temporal variation in LSTM.
- The performance of $LSTM_2$ increases by 7.8% (RGB) and 4.3% (Optical flow) on HMDB51 and by 2.2% (RGB) and 2.9% (Optical flow) on UCF101. The results show that the two-layer LSTM structure models the hierarchical hidden information and compensates for the one-layer structure, which has a straight-forward relation between the features and action classes. The result with the different setting of hidden state number from 256 to 4096 on UCF101 dataset(RGB) is shown in Fig 5. The performance of $LSTM_3$ gets the best performance with the 2048 hidden state on two layers of LSTM structure. Without loss of generality, we use this setting in all the LSTM structure.
- The performance of $LSTM_3$ obtains the highest accuracies in our three LSTM structure settings, with 68.2%(RGB) and 60.2%(Optical flow) and 92.1% (RGB) and 84.3% (Optical flow) on HMDB51 and UCF101, respectively. This occurs because $LSTM_3$ utilizes both forward and backward temporal information and can indeed enrich the modeling of the temporal action pattern.
- Finally, we use the score fusion to merge the classifier scores of $LSTM_3$ for RGB and optical flow data and obtain better results,which are benefited from the different domain data of action.

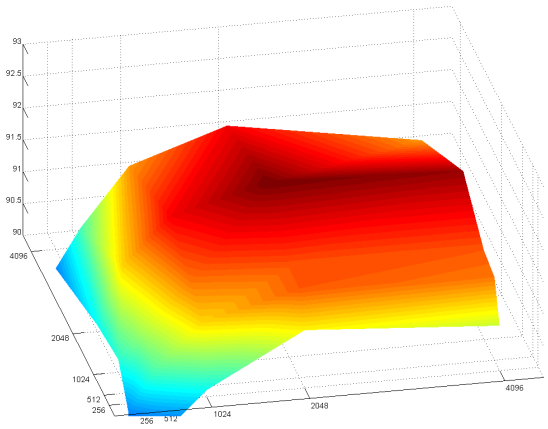


FIGURE 5. Performance of different hidden state numbers of LSTM3 structure on UCF101 datasets.

3) COMPARISON WITH THE STATE OF THE ART

Table 5 presents the overall classification accuracy of our method compared to those of the state-of-the-art methods on the UCF101 and HMDB51 datasets. Our proposed method outperforms the state-of-the-arts methods. We have several observations in two aspects:

- Feature representation. IDT+FV [18] is the typical hand-crafted feature for evaluating action recognition and uses the trajectory to model the spatiotemporal pattern. C3D [9] uses the 3D convolutional kernel and pooling strategy to learn the spatiotemporal features. Our proposed combined-C3D explores the clip representation by combining the selected parts and holistic clip information as the input for SVM. By doing so, it improves the performance by 13.2% and 8.3% compared to IDT+FV on HMDB51 and UCF101, respectively, and by 9.0% compared to the original C3D on UCF101.
- Temporal modeling. Our proposal outperforms the most temporal modeling methods. Especially, LRCN [25] uses the 2D convolutional framework to extract 2D features of single frame clips and then feeds the frame-based feature into a single-layer LSTM to learn the temporal pattern. Our proposal out-performs it by 11.5% and LSTM [41] by 5.6% on the UCF101 dataset. On HMDB51, our proposal improves upon the Temporal stream [6], Factorized ConvNet [42], by 15.8% and 11.3%, respectively.
- Two-Stream modeling. Our proposal uses the two-stream structure because of combining the two modality data, RGB and optical flow information. From the Table 5, we can find that our method gets best results and outperforms Two-Stream+LSTM [41], Two-Stream (avg. fusion) [27], Two-Stream(VGG16) [43] and Two-Stream Model [6] by 5.6%, 7.3%, 2.8% and 6.2% on UCF101, respectively. The similar improvement can been found on HDBM51 dataset.

The experiment results indicate that the strategy of combining local and global clip information enriches the

representation of human actions and uses the two different LSTM directional layers to help model the rich temporal patterns. On the challenging UCF101 and HMDB51 datasets, we obtain comparable results, which demonstrates the effectiveness and robustness of the proposed method.

V. CONCLUSION

In this paper, we present a novel framework for human action recognition by enriching the spatiotemporal action information to model human action patterns. Typically, we combine the selected spatial parts and traditional holistic clips to enrich the clip representation and the common temporal order. We consider the reverse temporal information by using multiple layers of long short-term memory framework to model the temporal pattern, which can learn compositional representations in space and time. Our approach is extensively evaluated on two widely used datasets, and our experimental results demonstrate the effectiveness of the proposed method in comparison with several state-of-the-art methods. Future work will focus on devising more powerful features, such as using the attention strategy. More structural information will also be considered to achieve a better characterization of human actions.

REFERENCES

[1] M. Luo, X. Chang, Z. Li, L. Nie, A. G. Hauptmann, and Q. Zheng, "Simple to complex cross-modal learning to rank," *Comput. Vis. Image Understand.*, vol. 163, pp. 67–77, Oct. 2017.

[2] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Las Vegas, NV, USA, Jun. 2016, pp. 770–778.

[3] M. Baccouche, F. Mamalet, C. Wolf, C. Garcia, and A. Baskurt, "Sequential deep learning for human action recognition," in *Proc. Int. Workshop Hum. Behav. Understand.*, Amsterdam, The Netherlands, 2011, pp. 29–39.

[4] S. Ji, W. Xu, M. Yang, and K. Yu, "3D convolutional neural networks for human action recognition," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 35, no. 1, pp. 221–231, Jan. 2013.

[5] A. Karpathy, G. Toderici, S. Shetty, T. Leung, R. Sukthankar, and L. Fei-Fei, "Large-scale video classification with convolutional neural networks," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Columbus, OH, USA, Jun. 2014, pp. 1725–1732.

[6] K. Simonyan and A. Zisserman, "Two-stream convolutional networks for action recognition in videos," in *Proc. Adv. Neural Inf. Process. Syst.*, Montreal, QC, Canada, 2014, pp. 568–576.

[7] D. E. Rumelhart, G. E. Hinton, R. J. Williams, *Learning Internal Representations by Error Propagation*. Parallel Distributed Processing, 1988, pp. 318–362.

[8] R. J. Williams and D. Zipser, "A learning algorithm for continually running fully recurrent neural networks," *Neural Comput.*, vol. 1, no. 2, pp. 270–280, Jun. 1989.

[9] D. Tran, L. D. Bourdev, R. Fergus, L. Torresani, and M. Paluri, "Learning spatiotemporal features with 3D convolutional networks," in *Proc. IEEE Int. Conf. Comput. Vis. (ICCV)*, Santiago, Chile, 2015, pp. 4489–4497.

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

[11] A. Graves and N. Jaitly, "Towards end-to-end speech recognition with recurrent neural networks," in *Proc. Int. Conf. Mach. Learn.*, Jan. 2014, pp. 1764–1772.

[12] I. Sutskever, O. Vinyals, and Q. V. Le, "Sequence to sequence learning with neural networks," in *Proc. Adv. Neural Inf. Process. Syst.*, Montreal, QC, Canada, 2014, pp. 3104–3112.

[13] K. Cho, B. van Merrienboer, D. Bahdanau, and Y. Bengio, "On the properties of neural machine translation: Encoder-decoder approaches," in *Proc. 8th Workshop Syntax, Semantics Struct. Stat. Transl.*, Doha, Qatar, 2014, pp. 103–111.

[14] M. Rohrbach, W. Qiu, I. Titov, S. Thater, M. Pinkal, and B. Schiele, "Translating video content to natural language descriptions," in *Proc. IEEE Int. Conf. Comput. Vis. (ICCV)*, Sydney, NSW, Australia, Dec. 2013, pp. 433–440.

[15] I. Laptev, M. Marszalek, C. Schmid, and B. Rozenfeld, "Learning realistic human actions from movies," in *Proc. IEEE Comput. Soc. Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Anchorage, AK, USA, Jun. 2008, pp. 1–8.

[16] C. Harris and M. Stephens, "A combined corner and edge detector," in *Proc. Alvey Vis. Conf. (AVC)*, Manchester, U.K., 1988, pp. 1–5.

[17] P. Dollàr, V. Rabaud, G. Cottrell, S. Belongie, "Behavior recognition via sparse spatio-temporal features," in *Proc. IEEE Int. Workshop Vis. Surveill. Perform. Eval. Tracking Surveill.*, Oct. 2005, pp. 65–72.

[18] H. Wang and C. Schmid, "Action recognition with improved trajectories," in *Proc. IEEE Int. Conf. Comput. Vis. (ICCV)*, Sydney, NWS, Australia, Dec. 2013, pp. 3551–3558.

[19] F. Perronnin, J. Sánchez, and T. Mensink, "Improving the Fisher kernel for large-scale image classification," in *Proc. Eur. Conf. Comput. Vis.*, Heraklion, Greece, 2010, pp. 143–156.

[20] B. Fernando, E. Gavves, M. J. Oramas, A. Ghodrati, and T. Tuytelaars, "Modeling video evolution for action recognition," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Boston, MA, USA, Jun. 2015, pp. 5378–5387.

[21] R. Wang, F. Nie, R. Hong, X. Chang, X. Yang, and W. Yu, "Fast and orthogonal locality preserving projections for dimensionality reduction," *IEEE Trans. Image Process.*, vol. 26, no. 10, pp. 5019–5030, Oct. 2017.

[22] B. Zhou, A. Lapedriza, J. Xiao, A. Torralba, and A. Oliva, "Learning deep features for scene recognition using places database," in *Proc. Adv. Neural Inf. Process. Syst.*, Montreal, QC, Canada, 2014, pp. 487–495.

[23] R. B. Girshick, J. Donahue, T. Darrell, and J. Malik, "Rich feature hierarchies for accurate object detection and semantic segmentation," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Columbus, OH, USA, Jun. 2014, pp. 580–587.

[24] Y. Taigman, M. Yang, M. Ranzato, and L. Wolf, "DeepFace: Closing the gap to human-level performance in face verification," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Columbus, OH, USA, Jun. 2014, pp. 1701–1708.

[25] J. Donahue *et al.*, "Long-term recurrent convolutional networks for visual recognition and description," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Boston, MA, USA, Jun. 2015, pp. 2625–2634.

[26] L. Wang, Y. Xiong, Z. Wang, and Y. Qiao. (2015). "Towards good practices for very deep two-stream convnets." [Online]. Available: https://arxiv.org/abs/1507.02159

[27] C. Feichtenhofer, A. Pinz, and A. Zisserman, "Convolutional two-stream network fusion for video action recognition," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Las Vegas, NV, USA, Jun. 2016, pp. 1933–1941.

[28] B. Zhang, L. Wang, Z. Wang, Y. Qiao, and H. Wang, "Real-time action recognition with enhanced motion vector CNNs," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Las Vegas, NV, USA, Jun. 2016, pp. 2718–2726.

[29] S. Yeung, O. Russakovsky, N. Jin, M. Andriluka, G. Mori, and L. Fei-Fei. (2017). "Every moment counts: Dense detailed labeling of actions in complex videos." [Online]. Available: https://arxiv.org/abs/1507.05738

[30] I. Sutskever, J. Martens, and G. E. Hinton, "Generating text with recurrent neural networks," in *Proc. 28th Int. Conf. Mach. Learn.*, Bellevue, WA, USA, 2011, pp. 1017–1024.

[31] O. Vinyals, S. V. Ravuri, and D. Povey, "Revisiting recurrent neural networks for robust ASR," in *Proc. IEEE Int. Conf. Acoust., Speech Signal Process. (ICASSP)*, Kyoto, Japan, Mar. 2012, pp. 4085–4088.

[32] W. Zaremba and I. Sutskever. (2015). "Learning to execute." [Online]. Available: https://arxiv.org/abs/1410.4615

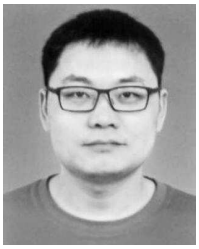
[33] N. Srivastava, E. Mansimov, and R. Salakhutdinov, "Unsupervised learning of video representations using LSTMs," in *Proc. 32nd Int. Conf. Mach. Learn. (ICML)*, Lille, France, 2015, pp. 1–10.

[34] J. Liu, G. Wang, P. Hu, L. Duan, and A. C. Kot, "Global context-aware attention LSTM networks for 3D action recognition," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Honolulu, HI, USA, Jul. 2017, pp. 3671–3680.

[35] J. Song, L. Gao, Z. Guo, W. Liu, D. Zhang, and H. T. Shen, "Hierarchical LSTM with adjusted temporal attention for video captioning," in *Proc. 26th Int. Joint Conf. Artif. Intell. (IJCAI)*, Melbourne, VIC, Australia, 2017, pp. 2737–2743.

[36] T. Brox, A. Bruhn, N. Papenberg, and J. Weickert, "High accuracy optical flow estimation based on a theory for warping," in *Proc. Eur. Conf. Comput. Vis.*, Prague, Czech Republic, 2004, pp. 25–36.
[37] S. Singh, A. Gupta, and A. A. Efros, "Unsupervised discovery of mid-level discriminative patches," in *Proc. 12th Eur. Conf. Comput. Vis.*, Italy, Rome, 2012, pp. 73–86.
[38] K. Soomro, A. R. Zamir, and M. Shah. (2012). "UCF101: A dataset of 101 human actions classes from videos in the wild." [Online]. Available: https://arxiv.org/abs/1212.0402
[39] H. Kuehne, H. Jhuang, E. Garrote, T. Poggio, and T. Serre, "HMDB: A large video database for human motion recognition," in *Proc. IEEE Int. Conf. Comput. Vis., (ICCV)*, Barcelona, Spain, Nov. 2011, pp. 2556–2563.
[40] S. Venugopalan, M. Rohrbach, J. Donahue, R. J. Mooney, T. Darrell, and K. Saenko, "Sequence to sequence—Video to text," in *Proc. IEEE Int. Conf. Comput. Vis. (ICCV)*, Santiago, Chile, 2015, pp. 4534–4542.
[41] J. Y. Ng, M. J. Hausknecht, S. Vijayanarasimhan, O. Vinyals, R. Monga, and G. Toderici, "Beyond short snippets: Deep networks for video classification," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Boston, MA, USA, Jun. 2015, pp. 4694–4702.
[42] L. Sun, K. Jia, D.-Y. Yeung, and B. E. Shi, "Human action recognition using factorized spatio-temporal convolutional networks," in *Proc. IEEE Int. Conf. Comput. Vis. (ICCV)*, Santiago, Chile, Dec. 2015, pp. 4597–4605.
[43] N. Ballas, L. Yao, C. Pal, and A. C. Courville. (2016). "Delving deeper into convolutional networks for learning video representations." [Online]. Available: https://arxiv.org/abs/1511.06432
[44] C. Feichtenhofer, A. Pinz, and R. P. Wildes, "Spatiotemporal residual networks for video action recognition," in *Proc. Adv. Neural Inf. Process. Syst.*, Barcelona, Spain, 2016, pp. 3468–3476.
[45] L. Wang, Y. Qiao, and X. Tang, "Action recognition with trajectory-pooled deep-convolutional descriptors," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Boston, MA, USA, Jun. 2015, pp. 4305–4314.
[46] L. Wang *et al.*, "Temporal segment networks: Towards good practices for deep action recognition," in *Proc. Eur. Conf. Comput. Vis.*, Amsterdam, The Netherlands, 2016, pp. 20–36.

WENHUI LI is currently the Ph.D. degree with the School of Electrical and Information Engineering, Tianjin University. He was an Intern Student with the SeSaMe Center, National University of Singapore. His research interests are in the field of computer vision, machine learning, and 3-D model retrieval.



WEIZHI NIE received the M.S. and Ph.D. degrees in electronic engineering from Tianjin University, China. He was with the School of Computer, National University of Singapore, in 2016 and 2017, respectively. His research interests include multiple object tracking, computer vision, and 3-D model retrieval.

YUTING SU received the M.S. and Ph.D. degrees in electronic engineering from Tianjin University, China. His research interests include multiple object tracking, computer vision, location-based social network, and 3-D model retrieval.