DOI: 10.7641/CTA.2017.60607

基于多层卷积神经网络特征和双向长短时记忆单元的行为识别

葛 瑞[1], 王朝晖[1], 徐 鑫[1], 季 怡[1], 刘纯平[1,2,3], 龚声蓉[4,1†]

(1. 苏州大学 计算机科学与技术学院, 江苏 苏州 215000;
2. 吉林大学 符号计算与知识工程教育部重点实验室, 吉林 长春 130012;
3. 软件新技术与产业化协同创新中心, 江苏 南京 210046;
4. 常熟理工学院 计算机科学与工程学院, 江苏 常熟 215500)

摘要: 鲁棒的视频行为识别由于其复杂性成为了一项极具挑战的任务. 如何有效提取鲁棒的时空特征成为解决问题的关键. 在本文中, 提出使用双向长短时记忆单元(Bi–LSTM)作为主要框架去捕获视频序列的双向时空特征. 首先, 为了增强特征表达, 使用多层的卷积神经网络特征代替传统的手工特征. 多层卷积特征融合了低层形状信息和高层语义信息, 能够捕获丰富的空间信息. 然后, 将提取到的卷积特征输入Bi–LSTM, Bi–LSTM包含两个不同方向的LSTM层. 前向层从前向后捕获视频演变, 后向层反方向建模视频演变. 最后两个方向的演变表达融合到Softmax中, 得到最后的分类结果. 在UCF101和HMDB51数据集上的实验结果显示本文的方法在行为识别上可以取得较好的性能.

关键词: 行为识别; 卷积神经网络; 递归神经网络; 双向递归神经网络

中图分类号: TP273 **文献标识码**: A

Action recognition with hierarchical convolutional neural networks features and bi-directional long short-term memory model

GE Rui[1], WANG Zhao-hui[1], XU Xin[1], JI Yi[1], LIU Chun-ping[1,2,3], GONG Sheng-rong[4,1†]

(1. School of computer science and technolgoy, Soochow University, Suzhou Jiangsu 215000, China;
2. Key Laboratory of Symbolic Computation and Knowledge Engineering of Ministry of Education, Jilin University, Changchun Jilin 130012, China;
3. Collaborative Innovation Center of Novel Software Technology and Industrialization, Nanjing Jiangsu 210046, China;
4. School of Computer Science and Engineering, Changshu Institute of Technology, Changshu Jiangsu 215500, China)

Abstract: Robust action recognition in videos is a challenging task due to its complexity. To solve it, how to effectively capture the robust spatio-temporal features becomes very important. In this paper, we propose to exploit bi-directional long short-term memory (Bi–LSTM) model as main framework to capture bi-directional spatio-temporal features. First, in order to boost our feature representations, the traditional hand-crafted descriptors are replaced by the extracted hierarchical convolutional neural network features. The multiple convolutional layer features fuse the information of low level basic shapes and high level semantic contents to get powerful spatial features. Then, the extracted convolutional features are fed into Bi–LSTM which has two different directional LSTM layers. The forward layer captures the evolution from front to back over video time and the backward layer models the opposite directional evolution. The two directional representations of evolution are then fused into Softmax to get final classification result. The experiments on UCF101 and HMDB51 datasets show that our method can achieve comparable performance with the state of the art methods for action recognition.

Key words: action recognition; convolutional neural networks; recurrent neural networks; bi-directional recurrent neural networks

1 Introduction

Action recognition[1] is one of the most active areas due to its wide applications, such as video surveillance, virtual reality, human-computer interaction, robotics, etc. However, there are still some challenges.

The first challenge is the selection of powerful feature representations. In the last several decades, many hand-crafted features are proposed. Some of them have been proved to be able to perform very well, such as scale-invariant feature transform[2], space time interest

Received 12 August 2016; accepted 23 January 2017.
†Corresponding author. E-mail: shrgong@suda.edu.cn; Tel.: +86 18913754857.
Recommended by Associate Editor YU Zhu-liang.
Supported by National Natural Science Foundation of China (61170124, 61272258, 61301299, 61272005, 61572085), Provincial Natural Science Foundation of Jiangsu (BK20151254, BK20151260), Key Laboratory of Symbolic Computation and Knowledge Engineering of Ministry of Education, Jilin University (93K172016K08), a Prospective Joint Research Projects from Joint Innovation and Research Foundation of Jiangsu Province (BY2014-05914) and Collaborative Innovation Center of Novel Software Technology and Industrialization.

points[3] and improved dense trajectories[4]. However, all these traditional features are sensitive to noise, and cannot meet the actual application requirements. Recently, static image classification using convolutional neural networks (CNN)[5] has achieved great success. It is believed that the deep learning method is able to extract the intrinsic features from large-scale training data. Therefore, it is also an effective way to improve the feature representation when applied to action recognition.

The second challenge is how to model the dynamic temporal structure. Simonyan et al.[1] take video action recognition as an image classification problem by splitting the videos into single frames. They predict each video by averaging the output scores of each frame. As a result, the relationship between video frames is ignored. Besides, some approaches may pool all the frames into a global bag-of-word (BoW)[6] vector to represent the video. Since these approaches heavily rely on static features, they would inevitably lose too much useful temporal information, leading to poor performance. In order to capture the dynamic temporal information of the sequences, recurrent neural networks (RNN)[7] is well designed to learn from the time series and has been exploited in many applications, such as speech recognition[8] and handwriting recognition[9]. Recent papers[10–11] applied long short-term memory (LSTM) to action recognition and have achieved cutting-edge results.

For the first challenge, we utilize hierarchical CNN features. We extract hierarchical abstract representations from the raw input by deep learning. These representations which fuse multiple CNN layer features, are powerful for human activities. In order to deal with the second challenge, we use bi-directional LSTM (Bi--LSTM)[12] to model the temporal revolution of actions to capture bi-directional information of sequences. The developed model is therefore more discriminative and robust than single directional LSTM.

2 Related work

In this section, we give more details related to the latest research on action recognition. We mainly focus on two main new-emerging trends: the action recognition researches using convolutional neural networks and recurrent neural networks.

Traditional methods focus on designing robust low-level feature descriptor, such as histogram of gradient (HOG)[13] and histogram of optical flow (HOF)[14]. However, these descriptors are single frame representations and can't model the temporal evolution of video sequence. Wang et al.[4] proposed an improved dense trajectories (iDT) method which extracted the trajectories of interest points. Cai et al.[15] extended iDT to multiple views. However, these traditional features are vulnerable to noise and the room for improvement is limited.

Inspired by the success of convolutional neural networks on ImageNet large-scale visual recognition challenge[5], many works have tried to process the video sequences using the deep learning method. Simonyan et al.[1] proposed to use a two-stream framework for action recognition. The authors utilized two standalone convolutional networks, including spatial stream and temporal stream. The whole framework achieves the great improvement on action recognition comparing to traditional single stream convolutional networks. Wang et al.[16] exploited trajectory-pooled deep-convolutional descriptor (TDD) to fuse hand-crafted features with Two-Stream network features. However, all those methods are not capable to capture the temporal information for action recognition.

In order to exploit temporal information, Wang et al.[17] decomposed the action into action-lets, but it is complex to model the body model. Haoi et al.[18] used structural SVM to model the relationship between the frames and predict the happening of the whole event when observing part of it. However, these models can't model the revolution of video sequences effectively. Baccouche et al.[19] proposed to use LSTM for action recognition, but their LSTMs are based on the extracted hand-crafted features instead of the original sequences. Therefore this leads to lose some useful information of original frames. Ng et al.[20] presented to capture the motion information using LSTM in the forward video sequences. Srivastava et al.[21] explored composite LSTMs for action recognition. Unfortunately, these methods can only model the changes of the sequences on single direction.

Instead of designing the temporal descriptors, we hope to model the evolution between two consecutive sequence frames. LSTM cell can only capture the single direction changes. However, relevant to past and future status, the information of two directions can be very useful in successful learning. To solve this problem, we exploit the Bi–LSTM model[12]. Besides, from low level raw input to high level semantic cells, different layers of convolutional neural networks encode different information. For strong representation, we select features extracted from different CNN layers and average the scores to get the final results.

The flow chart of our proposed model is shown in Fig.1. For input video, we execute the two-stream framework[1]. The RGB frames and optical flow are extracted and saved by tools in advance. After collecting the CNN features from different layers of spatial network and temporal network, we feed these features into Bi–LSTM model[12]. Finally, the spatial and temporal scores are fused to get the classification result.

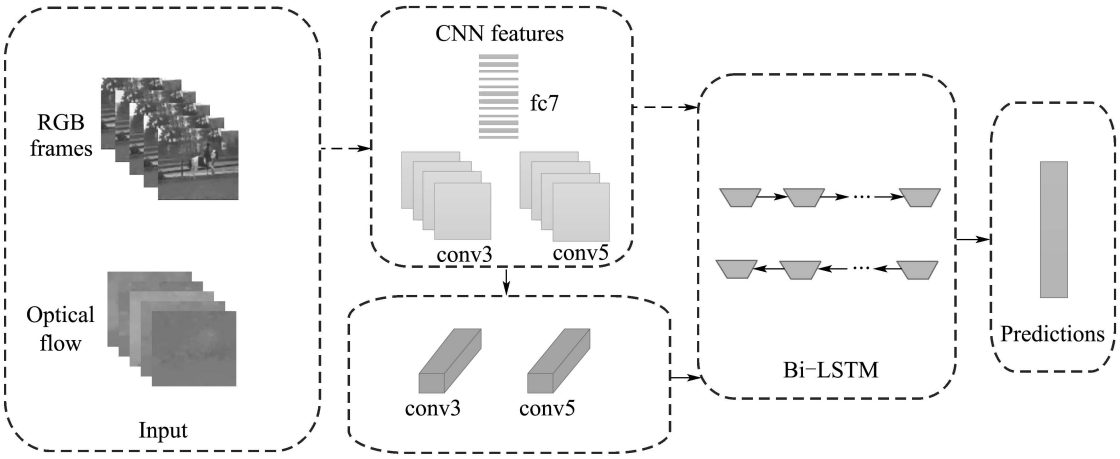


Fig. 1 Overview of our proposed model for action recognition

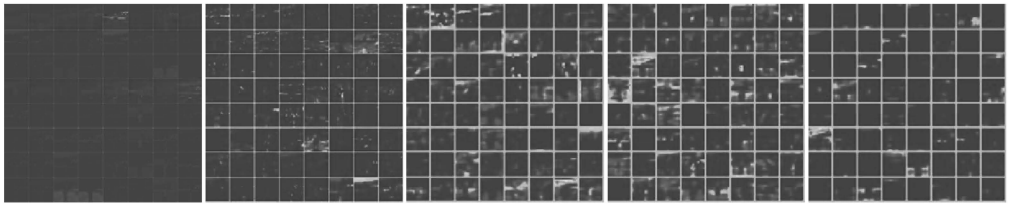
3 Methods

In this section, we will introduce our action recognition method. Firstly, we describe our convolutional neural network settings and the method of extracting CNN features. Then, we introduce RNN and LSTM cell architecture. Finally, we show our Bi–LSTM architecture for action recognition.

3.1 Convolutional neural networks and features

To extract strong CNN features, firstly, we need to train a deep convolutional neural network. We setup the Caffe framework which is simple and effective for deep learning. We pre-train the model on the ImageNet dataset which has a large number of pictures. Then we transfer the model to our own dataset for fine-tuning. Because the ImageNet contains tens of millions images, it makes the convolutional networks more general. We adapt the architecture similar to two-stream ConvNets[1]. RGB frames and optical flow of the video sequences are extracted and fed into spatial and temporal networks respectively.

The abstracts of different CNN layers are different according to their levels. The early layers have more low level information, such as edge information. The outputs of last layers are more abstract and encode the semantic information of videos. Fig.2 is the visualization of the different layer filters. From the illustration, we can find that the different layers provide different information. They trigger different responses for the same input image. Besides, convolutional layers have more space information than fully connected layer. Hence, we choose conv3, conv5, fc7 layers as our strong fusion representation by experiments discussed in Section 4. The features of conv1 and conv2 have a little generalization ability and make no significant contribution to our results.



(a) conv1 (b) conv2 (c) conv3 (d) conv4 (e) conv5

Fig. 2 The comparison of different convolutional layer filters. We select the first 64 filters for each layer to visualize

After extracting feature maps, we pool the feature maps into a fixed-size vector. Assume that we get the feature map F_{m}^{t} of m-th layer at time t. The feature map is $H_{\mathrm{f}} \times W_{\mathrm{f}} \times C \times T$ dimension, H_{f} is height of feature map, W_{f} is width of feature map, C is the channels of feature map (here is 512), T is the total number of video frames. We first pool the feature maps along time domain by following formula:

$$Desc = \sum_{j=1}^{N} F_{\mathrm{m}}^{t_j}(x, y), \tag{1}$$

where j is the pooling frames, N is the number of temporal pooling extent. Afterwards, as in [16], we treat each 512 dimensions channel feature as a latent descriptor. Finally, we apply spatial pyramid max-pooling to get fixed sized vector for each frame. For more details, refer to Section 4.2.

3.2 Recurrent neural networks

The characteristic of recurrent neural networks makes it easy to model temporal sequences. For the video sequences, current output depends on the current input and the previous status. More generally, suppose given input sequences denoted by $x = \{x_1, x_2, \cdots, x_t, \cdots, x_T\}$, where t represents tth frame, and there are totally T frames. We get the for-

mulation as following:

$$h_t = \sigma_{\rm h}(W_{\rm xh}x_t + W_{\rm hh}h_{t-1} + b_{\rm h}), \tag{2}$$

where h_t denotes the output of the hidden layer at time t, $W_{\rm xh}$ stands for the corresponding weight matrices from input layer to hidden layer, $W_{\rm hh}$ is the weight matrices from hidden layer to hidden layer, $b_{\rm h}$ is the bias for the hidden layer, and $\sigma_{\rm h}$ is activation function. Finally, we can get the output through following formulation:

$$y_t = \sigma_{\rm y}(W_{\rm ho}h_t + b_{\rm o}), \tag{3}$$

where y_t denotes the predict label of t-th sequence, $W_{\rm ho}$ stands for the weight matrices from hidden layer to output, $b_{\rm o}$ is the bias for the output, $\sigma_{\rm y}$ denotes the activation function.

The major problem of RNN is that it can only model the short time sequences because the error gradients vanish quickly as the networks become deeper. To solve this problem, LSTM introduces three gates to keep the status. As in Fig.3, there are 3 gates, including input gate (i_t), forget gate (f_t), and output gate (o_t). Where i_t and o_t control information that flows in or out the network, f_t controls the influence of previous sequences. Details are formulated as follows:

$$\begin{cases} i_t = \sigma(W_{\rm xi}x_t + W_{\rm hi}h_{t-1} + W_{\rm ci}c_{t-1} + b_{\rm i}), \\ f_t = \sigma(W_{\rm xf}x_t + W_{\rm hf}h_{t-1} + W_{\rm cf}c_{t-1} + b_{\rm f}), \\ o_t = \sigma(W_{\rm xo}x_t + W_{\rm ho}h_{t-1} + W_{\rm co}c_{t-1} + b_{\rm o}), \\ c_t = f_t \odot c_{t-1} + i_t \odot \tanh(W_{\rm xc}x_t + W_{\rm hc}h_{t-1} + b_{\rm c}), \\ h_t = o_t \odot \tanh\, c_t. \end{cases} \tag{4}$$

where c_t denotes the memory cell of time t, h_t represents the output of hidden layer, b_α denotes the bias of α with $\alpha \in \{i, f, c, o\}$, $W = \{W_{\rm xi}, W_{\rm xo}, W_{\rm xf}, W_{\rm ci}, W_{\rm co}, W_{\rm cf}, W_{\rm hi}, W_{\rm ho}, W_{\rm hf}\}$ denote the weighted parameters and are jointly learned by back propagation through time (BPTT)[22]. For more details about LSTM, please read Graves' excellent paper[7].

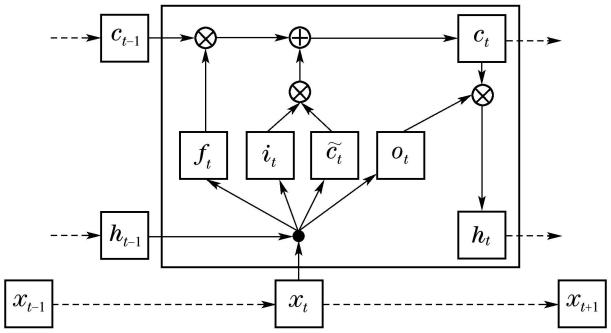


Fig. 3 The LSTM architecture

3.3 Our Bi–LSTM architecture

Although the LSTM can capture the long time series information, it considers only one direction. That is to say, LSTM assumes that the current frame is only affected by previous frames, but the following frames are also relevant to the current status. We hope to strengthen this relationship to bi-directions. It means that we also consider the next frame when processing the current frame. Bi–LSTM[12] is well suitable for this problem. Our Bi–LSTM model depicts in Fig.4. The first layer is forward LSTM, and the second layer is backward LSTM. The final output can be calculated by the following formulations:

$$\begin{aligned} h_t &= \alpha h_t^{\rm f} + \beta h_t^{\rm b}, \\ y_t &= \sigma(h_t), \end{aligned} \tag{5}$$

where $h_t^{\rm f}$ represents the output of forward LSTM layer which takes sequences from x_1 to x_T as input, $h_t^{\rm b}$ stands for the output of backward LSTM which takes sequences from x_T to x_1, α and β control the importance of forward LSTM and backward LSTM ($\alpha + \beta = 1$), h_t denotes the element-wise sum of two single directional LSTMs at time t, σ here is the softmax function, y_t is the predict label. As Bi –LSTM can model the bi-directional temporal structure, it can capture more structural information and performs better than single directional LSTM.

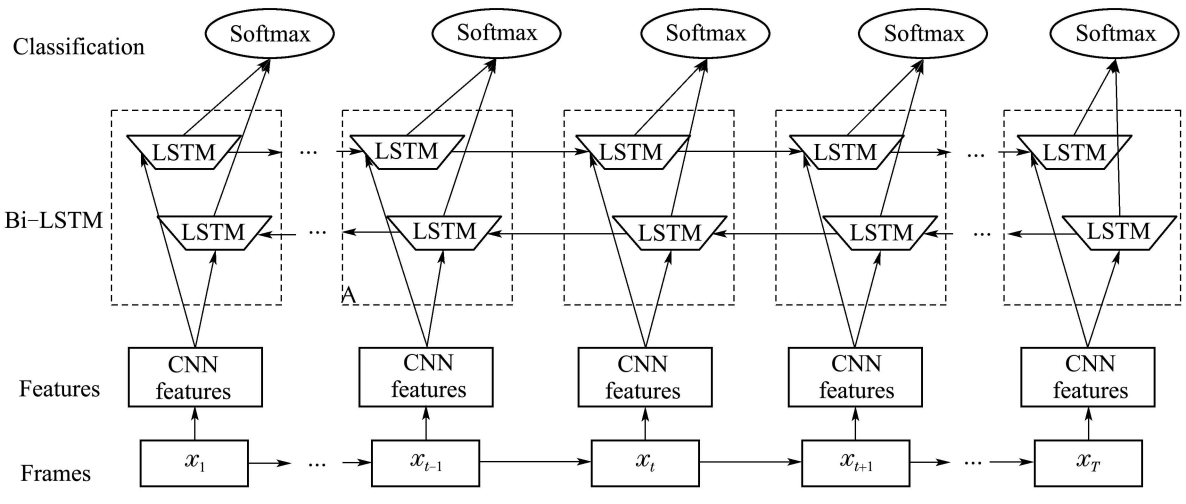


Fig. 4 Our Bi–LSTM architecture

4 Experiments

In this section, we first introduce our evaluation dataset and scheme. Then we give the details of our implementation. Finally, we report our experimental results and compare with the state of the art methods.

4.1 Dataset

Our proposed approach for action recognition has been evaluated on the UCF101[23] and HMDB51[24] dataset. Both are popular and considered as benchmarks in this area.

The UCF101 dataset contains 13,320 videos and 101 action classes. It is composed of realistic web videos under unconstrained condition. It has three split settings to separate the dataset into training and testing videos. We test our approach on each split and average each result to get the final results.

The HMDB51 dataset is collected from various sources, such as movies or web videos. The dataset is composed of 6,766 video clips organized as 51 action classes. It is more challenging than other datasets as it has more complex backgrounds and context environment. The dataset is also split into three training/testing split. For final results, we evaluate on each split and get the average accuracy.

4.2 Implementation details

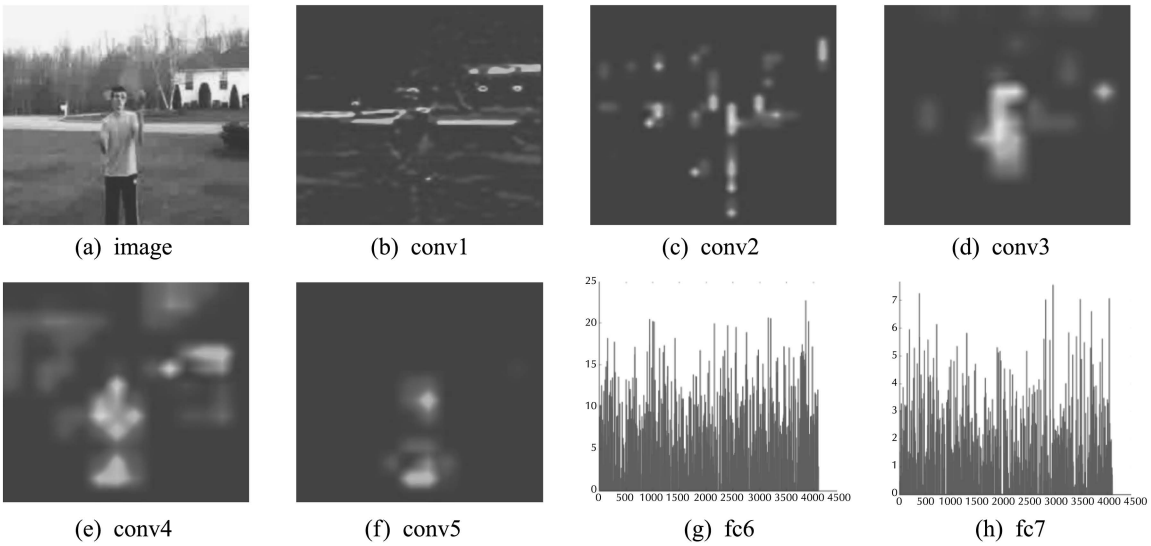
As the first main step, we develop our ConvNets based on popular Caffe framework. We fine-tune the networks on action recognition dataset using the model pre-trained on ImageNet. The mini-batch is set to 256 and momentum to 0.9. Spatial network takes 224×224 sized region of RGB frames as one channel input, while temporal network takes optical flow extracted from sequences as another channel input. After fine-tuning is done, the different layer features are cached for further usage.

As the next step, we conduct pooling on extracted convolutional features. The feature map features are in high dimensions. Thus, temporal pooling and spatial pooling are conducted separately. The pyramid level is set to 3 for spatial pyramid max-pooling. Such yields 21 pooling areas and 10752 dimension features for convolution layers. For fully connected layer, it is a 4096 dimensional vector with fixed size. For all the chosen layers, we feed the features to Bi–LSTM cell and then train Softmax classifiers.

In order to classify actions, Softmax is connected after Bi–LSTM. The output size of Softmax is the number of action classes. The whole recurrent network is trained by BPTT[22]. Especially, batch is set to 64 and momentum to 0.9. The learning rate starts at 0.01. After every 20,000 iterations, the rate is divided by 10. Training converges after 50,000 iterations. Afterwards, the test video features are fed into trained LSTM and the score matrix will be output. As the final step, we follow[10] to get the weighted average value of RGB and flow through cross validation. The weights of RGB stream and flow stream are set to 1 and 1.5 separately. It illustrates that the flow features perform better than RGB features for dynamic action recognition.

4.3 Experimental results

In order to choose the useful CNN layers, we compare the results of different layers on the UCF101 dataset. Fig.5 shows that different layers have different responses for the same input image. The early layers have more fine-grained details and the later layers are more abstract. However, conv1 and conv2 have much more useless noise which will make no obvious contribution to our results.



(a) image (b) conv1 (c) conv2 (d) conv3

(e) conv4 (f) conv5 (g) fc6 (h) fc7

Fig. 5 Visualization of different layers

Next step, we conduct a quantitative analysis. For easy and quick comparison, we down-sample the sequence data heavily. We sample only 10 frames from each video. The short clip provides effective information, and reduces our experimental time and memory space. The results illustrated in Table 1 are conducted on spatial networks by RGB frames. From the results, we conclude that several earlier layers may be harmful to our results for its poor generalization ability.

Moreover, we explore the performance from different combinations of later layers. We can achieve better results by multi-layer fusions than single layer. It proves that the multi-layer CNN features are more discriminative. In experiments, we get the best results by fusing conv3, conv5, fc7 (as in Table 2).

Table 1 The performance of different single layers on UCF101

Layer	conv1	conv2	conv3	conv4	conv5	fc6	fc7
Accuracy/%	33.5	44.7	68.2	68.9	69.1	62.8	63.4

Table 2 The performance of composite layers on UCF101

Composite layers	Accuracy/%
conv3 + conv4	70.1
conv4 + conv5	69.2
conv3 + conv5	70.5
conv3 + conv5 + fc6	71.2
conv3 + conv5 + fc7	71.7

Subsequently, we compare our Bi–LSTM model with average model and single directional LSTM model. In average model, we directly average the scores of Softmax to yield the final result. We use one single LSTM layer in our LSTM model. The Bi –LSTM model is described in Section 3.3. All three models take extracted fc7 features as input. From results in Table 3, we can conclude that the Bi-LSTM outperforms the average model and single directional LSTM.

Table 3 The performance of different model on UCF101

Model	Average model	LSTM	Bi–LSTM
Accuracy/%	63.4	64.2	64.8

Finally, we compare the computation cost and accuracy of various popular methods. Table 4 shows the computation cost of main methods on our computer with GPU of NVIDIA Tesla K20, 64GB memory, 6 Intel(R) Xeon(R) CPU @ 2.10GHz cores. The accuracy of different methods on UCF101 and HMDB51 is presented in Table 5. From these two tables, we find that our approach outperforms iDT, two-stream and LSTM-related methods (LRCN[10], BSS[20] and ComLSTM[21]), which have similar computation cost as our method. On the other hand, although TDD[16] has similar performance as our method, it requires longer computation time and more storage space than ours. Because TDD is based on iDT and Two-Stream, the computation process is much more complex than our methods. In conclusion, our approach balances accuracy, computation time and storage and has better overall performance.

Table 4 The performance of different methods on UCF101

Method	Accuracy/ %	Time/ day	Storage/ GB
iDT+FV[4]	85.9	15	529
Two-Stream[1]	88.0	10	81
TDD[16]	90.3	20	701
LRCN[10]	82.9	12	81
Ours	88.9	14	272

Table 5 The accuracy of different methods on UCF101 and HMDB51 dataset

Methods	UCF101/%	HMDB51/%
STIP+BovW[23]	43.9	23.0
Motionlets[17]	–	42.1
DT+MVSV[15]	83.5	55.9
iDT+FV[4]	85.9	57.2
iDT+HSV[11]	87.9	61.1
Two-Stream[1]	88.0	59.4
TDD[16]	90.3	63.2
LRCN[10]	82.9	–
BSS[20]	88.6	–
ComLSTM[21]	84.3	–
Our results	88.9	62.3

5 Conclusions

In this paper, we have presented a new framework which combines hierarchical CNN features and Bi –LSTM model to recognize human actions. In one aspect, the hierarchical CNN features are powerful representations of human activities. In the other aspect, Bi–LSTM model can effectively model the bi-directional temporal evolution of individual actions. The experimental results demonstrate that our method can achieve comparable accuracy to the state of the art methods.

References:

[1] SIMONYAN K, ZISSERMAN A. Two-stream convolutional networks for action recognition in videos [C] *//Advances in Neural Information Processing Systems*. Montreal: Neural information processing systems foundation, 2014: 568 – 576.

[2] LOWE D G. Distinctive image features from scale-invariant keypoints [J]. *International Journal of Computer Vision*, 2004, 60(2): 91 – 110.

[3] LAPTEV I. On space-time interest points [J]. *International Journal of Computer Vision*, 2005, 64(2/3): 107 – 123.

[4] WANG H, SCHMID C. Action recognition with improved trajectories [C] //*Proceedings of the IEEE International Conference on Computer Vision*. Sydney: IEEE, 2013: 3551 – 3558.

[5] KRIZHEVSKY A, SUTSKEVER I, HINTON G E. Imagenet classification with deep convolutional neural networks [C] //*Advances in Neural Information Processing Systems*. Lake Tahoe: Neural information processing system foundation, 2012: 1097 – 1105.

[6] SIVIC J, ZISSERMAN A. Video Google: A text retrieval approach to object matching in videos [C] //*International Conference on Computer Vision*. Nice: IEEE, 2003: 1470 – 1477.

[7] GRAVES A. Generating sequences with recurrent neural networks [J]. arXiv preprint arXiv:1308.0850, 2013.

[8] GRAVES A, JAITLY N. Towards end-to-end speech recognition with recurrent neural networks [C] //*Proceedings of the 31st International Conference on Machine Learning*. Beijing: IMLS, 2014: 1764 – 1772.

[9] GRAVES A, LIWICKI M, FERNáNDEZ S, et al. A novel connectionist system for unconstrained handwriting recognition [J]. *IEEE Transactions on Pattern Analysis And Machine Intelligence*, 2009, 31(5): 855 – 868.

[10] DONAHUE J, HENDRICKS L A, GUADARRAMA S, et al. Long-term recurrent convolutional networks for visual recognition and description [C] //*Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. Boston: IEEE, 2015: 2625 – 2634.

[11] PENG X, WANG L, WANG X, et al. Bag of visual words and fusion methods for action recognition: Comprehensive study and good practice [J]. *Computer Vision and Image Understanding*, 2016, 150: 109 – 125.

[12] SCHUSTER M, PALIWAL K K. Bidirectional recurrent neural networks [J]. *IEEE Transactions on Signal Processing*, 1997, 45(11): 2673 – 2681.

[13] DALAL N, TRIGGS B. Histograms of oriented gradients for human detection [C] //*Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. San Diego: IEEE, 2005: 886 – 893.

[14] LAPTEV I, MARSZALEK M, SCHMID C, et al. Learning realistic human actions from movies [C] //*Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. Anchorage: IEEE, 2008: 1 – 8.

[15] CAI Z, WANG L, PENG X, et al. Multi-view super vector for action recognition [C] //*Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. Columbus: IEEE, 2014: 596 – 603.

[16] WANG L, QIAO Y, TANG X. Action recognition with trajectory-pooled deep-convolutional descriptors [C] //*Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. Boston: IEEE, 2015: 4305 – 4314.

[17] WANG L M, QIAO Y, TANG X. Motionlets: mid-level 3d parts for human motion recognition [C] //*Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. Portland: IEEE, 2013: 2674 – 2681.

[18] HOAI M, DE LA TORRE F. Max-margin early event detectors [C] //*Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. Providence: IEEE, 2012: 2863 – 2870.

[19] BACCOUCHE M, MAMALET F, WOLF C, et al. Action classification in soccer videos with long short-term memory recurrent neural networks [C] //*International Conference on Artificial Neural Networks*. Thessaloniki: Springer, 2010: 154 – 159.

[20] NG Y H, HAUSKNECHT M, VIJAYANARASIMHAN S, et al. Beyond short snippets: deep networks for video classification [C] //*Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. Boston: IEEE, 2015: 4694 – 4702.

[21] SRIVASTAVA N, MANSIMOV E, SALAKHUTDINOV R. Unsupervised learning of video representations using LSTMs [C] //*International Conference on Machine Learning*. Lille: IMLS, 2015: 843 – 852.

[22] MOZER M C. A focused backpropagation algorithm for temporal pattern recognition [J]. *Complex Systems*, 1995, 3(4): 349 – 381.

[23] SOOMRO K, ZAMIR A R, SHAH M. Ucf101: A dataset of 101 human actions classes from videos in the wild [J]. arXiv preprint arXiv:1212.0402, 2012.

[24] KUEHNE H, JHUANG H, GARROTE E, et al. HMDB: a large video database for human motion recognition [C] //*Proceedings of the IEEE International Conference on Computer Vision*. Barcelona: IEEE, 2011: 2556 – 2563.

作者简介:

葛 瑞 (1991–), 男, 硕士, 目前研究方向为计算机视觉等, E-mail: forgerui@163.com;

王朝晖 (1967–), 女, 副教授, 目前研究方向为图像处理与分析等, E-mail: zhhwang@suda.edu.cn;

徐 鑫 (1992–), 男, 硕士, 目前研究方向为计算机视觉等, E-mail: 20144227042@stu.suda.edu.cn;

季 怡 (1973–), 女, 博士，副教授, 目前研究方向为计算机视觉、机器学习等, E-mail: jiyi@suda.edu.cn;

刘纯平 (1971–), 女, 博士, 教授, 目前研究方向为图像处理与分析、模式识别等, E-mail: cpliu@suda.edu.cn;

龚声蓉 (1966–), 男, 博士, 教授, 博士生导师, 目前研究方向为计算机视觉、机器学习等, E-mail: shrgong@suda.edu.cn.