This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/LSP.2016.2611485, IEEE Signal Processing Letters

Beyond Frame-level CNN: Saliency-aware 3D CNN with LSTM for Video Action Recognition

Xuanhan Wang, Lianli Gao, Jingkuan Song, *Member, IEEE,* and Hengtao Shen, *Senior Member, IEEE*

***Abstract*—Human activity recognition in videos with CNN features has received increasing attention in multimedia understanding. Taking videos as a sequence of frames, a new record was recently set on several benchmark datasets by feeding frame-level CNN sequence features to Long Short-Term Memory (LSTM) model for video activity recognition. This recurrent model based visual recognition pipeline is a natural choice for perceptual problems with time-varying visual input or sequential outputs. However, the above pipeline takes frame-level CNN sequence features as input for LSTM, which may fail to capture the rich motion information from adjacent frames or maybe multiple clips. Furthermore, an activity is conducted by a subject or multiple subjects. It is important to consider attention which allows for salient features, instead of mapping an entire frame into a static representation. To tackle these issues, we propose a novel pipeline, Saliency-aware 3D CNN with LSTM (scLSTM), for video action recognition by integrating LSTM with salient-aware deep 3D CNN features on videos shots. Specifically, we first apply saliency-aware methods to generate saliency-aware videos. Then, we design an end-to-end pipeline by integrating 3D CNN with LSTM, followed by a time series pooling layer and a Softmax layer to predict the activities. Noticeably, we set a new record on two benchmark datasets, i.e., UCF101 with 13,320 videos and HMDB-51 with 6,766 videos. Our method outperforms the state-of-the-art end-to-end methods of action recognition by 3.8% and 3.2% respectively on above two datasets.**

***Index Terms*—deep learning, 3D convolution, LSTM, Saliency-aware, action recognition.**

I. Introduction

Recognizing human actions in videos has received a significant amount of attention in the research communities [1], [11], [23], [19], [21]. Different types of action recognition algorithms have been recently introduced. In this paper, we divide the video human action recognition into two categories: 1) local space-time feature designing; and 2) deep learning based techniques.

Human action recognition can be viewed as a pattern recognition problem and hand-crafted features used in image processing [27], [28], [3] have been successfully used for image recognition. Therefore, previous work has directly extended the image action algorithms with hand-crafted spatial and temporal features for recognizing human action in videos. The spatial and temporal features are used to characterize visual appearance and motion dynamics, respectively [11], [1], [23]. For example, [1] presented a behavior recognition algorithm based on spatio-temporal features, which are extracted via anchoring off the direct 3D and 2D interest points from spatio-temporally windowed data. [23] proposed the improved dense trajectories by explicitly estimating camera motion. To date, many classical image features used in computer vision [29], [30], [5], [4], [26], [17] have been generalized to video action recognition, such as 3D-SIFT [15], extended SURF, HOG3D, spatio-temporal feature [1], motion boundary histograms (MBH), histograms of optical flow(HOF) and improved dense trajectories (iDT) [23]. Among them, MBH has been shown to perform better than HOF for the reason that MBH is robust to camera motion. And improved dense trajectories gives the best results, which performs better than the combination of HOF+MBH. In general, these methods give good results in some challenging datasets (UCF-101 [18], HMDB-51 [10]) by encoding previous local spatial-time features into high dimensional space. However their performance often degrades when being applied to more realistic and complex video settings due to the large variations within action categories and other video issues [21], [19].

In order to improve the performance of action recognition, some recent efforts have been proposed to directly apply deep learning models to learn video representation for video action recognition and promising results were obtained [9], [25], [16], [8]. Compared with image action recognition, human actions in video sequences are 3D signals consisting of visual appearance that dynamically evolves over time [19]. As a result, there are some attempts to change 2D Convolutional Neural Networks (CNN) or utilize other deep network modules for encoding actions' temporal information. For instance, Ji *et al.*[7] extend the 2D CNN model into 3D domain. The proposed 3D CNN model extracts features from both spatial and temporal dimensions by performing 3D convolutions, thereby capturing the motion information encoded in multiple adjacent frames. Simonyan *et al.* [16] aimed to capture the complementary information on appearance from still frames and motion between adjacent frames. As a result, they proposed a two-stream ConvNet architecture which incorporates spatial and temporal networks.

Recent efforts have shown that Long Short-Term Memory (LSTM) [12], [2] is able to learn when to forget previous hidden states and when to update hidden states by integrating memory units. A new record [2] was recently set on several benchmark datasets by feeding frame-level CNN sequence features to Long Short-Term Memory (LSTM) model for video activity recognition. These recurrent model based visual recognition pipeline is a natural choice for perceptual problems with time-varying visual input or sequential outputs. LSTM has been successfully adopted to several tasks, e.g., speech recognition [6], language translation [20] and image

Lianli Gao is with the School of Computer Science and Engineering, University of Electronic Science and Technology of China, 611731 (lianli_gao@uestc.edu.cn.)

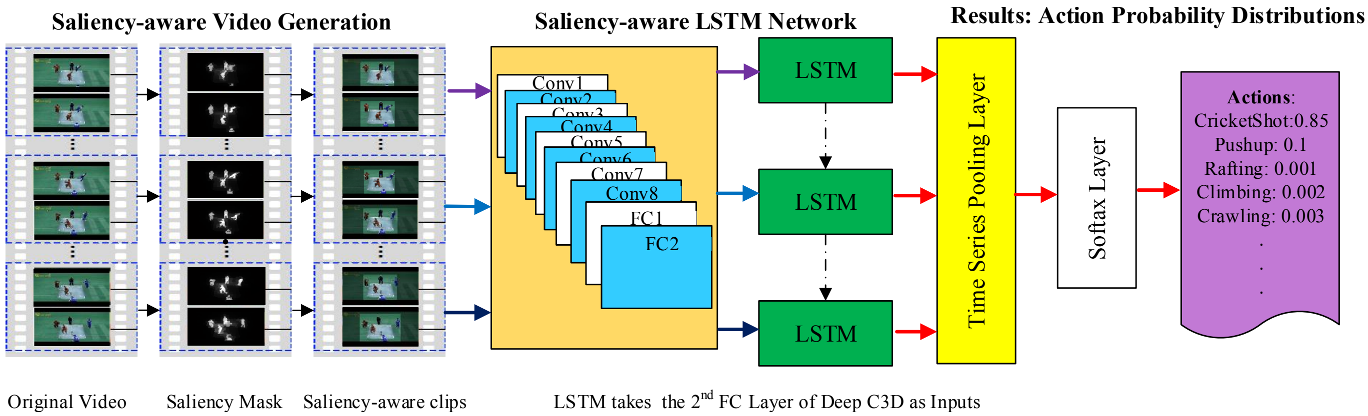

Fig. 1. The general framework of our saliency-aware 3D CNN with LSTM for video activity recognition. scLSTM consists of two phases. Firstly, we apply saliency-aware methods to generate salience-aware videos. Then, we design an end-to-end pipeline by integrating 3D CNN with LSTM, followed by a time series pooling layer and a Softmax layer to predict the activities of videos.

caption [22]. However, the above pipeline takes frame-level CNN sequence features as input for LSTM, which may fail to capture the rich motion information from adjacent frames. Furthermore, an activity is conducted by a subject or multiple subjects. It is important to consider attention which allows for salient features, instead of mapping an entire frame into a static representation.

In order to deal with above two issues, we propose a novel pipeline, called saliency-aware 3D CNN and LSTM (scLSTM) for video action recognition by integrating the Long Short-Term Memory (LSTM) with salient action motion detection. It is worth highlighting the following contributions: 1) We propose an end-to-end pipeline by integrating LSTM with 3D CNN for video action recognition. The 3D CNN features on videos shots contains richer motion information than frame-level CNN features, and LSTM can explore the temporal relationship between video shots; 2) Saliency is further introduced to capture important subjects from video shots, which will improve the performance of 3D CNN features; and 3) Our method set a new record on two benchmark datasets, i.e., UCF101 with 13,320 videos and HMDB-51 with 6,766 videos. It outperforms the counterparts by 3.8% and 3.2% respectively.

II. THE PROPOSED APPROACH

In this section, we introduce our method scLSTM which consists of two phases (See Fig.1). Firstly, we apply saliency-aware methods to generate salience-aware videos. Then, we design an end-to-end pipeline by integrating 3D CNN with LSTM, followed by a time series pooling layer and a Softmax layer to predict the activities.

A. Salience-aware Video Generation

Image region segmentation has shown to benefit many specific visual tasks and applications, such as object detection and action recognition[13]. Recently, several methods have managed to generate considerable object proposals in every frame and transfer the task of object segmentation into an object region selection problem by utilizing both motion and appearance information to calculate the objectness scores. For example, [24] has proposed a saliency-aware based video object segmentation method, which performs better than the state-of-the-art. Inspired by the success of saliency methods, in this paper, we aim to integrate saliency technique into our framework. Given a video, we process the video frame by frame. Firstly, we use method in [24] to generate a saliency-aware map $\mathbf{M}$ for each frame. Then, the corresponding frame saliency mask is computed by binarizing the $\mathbf{M}$. If $m_{i,j} < mean(\mathbf{M})$ then $m_{i,j} = 0$, otherwise $m_{i,j} = 1$. Next, we weaken the background regions to improve the importance of foreground objects (i.e. ,subject salience information). This is conducted by halving the RGB values of background regions where $m_{i,j} = 0$. As a result, a salience-aware video is generated and denoted as $\mathbf{V}$.

B. Saliency-aware 3D CNN with LSTM

As mentioned in the previous section, a salience-aware video is represented as $\mathbf{V}$. In this section we integrate the deep 3D CNN with LSTM to analyze $\mathbf{V}$ for action recognition. Suppose $\mathbf{V}$ has N frames, and we denote $\mathbf{V} = (v_1, v_2, \cdots, v_N)$. Firstly, we divide the video $\mathbf{V}$ into T splits and $\mathbf{V} = (\mathbf{v}_1^s, \mathbf{v}_2^s, \cdots, \mathbf{v}_t^s, \cdots, \mathbf{v}_T^s)$, where $\mathbf{v}_t^s$ is the t-th split of the $\mathbf{V}$, $T = \frac{N}{K}$, and K is the length of the split. Next, we encode each split with a 3D CNN network, thus a sequence of video shots is generated as $\mathbf{X} = (\mathbf{x}_1, \cdots, \mathbf{x}_T)$. Finally, a sequence model takes $\mathbf{X}$ as input to recognize actions.

3D signals encoding. Given a video split $\mathbf{v}_t^s$, we propose to model the saliency-aware video at the level of the temporal features $\mathbf{X} = (\mathbf{x}_1, \mathbf{x}_2, \cdots, \mathbf{x}_t, \cdots, \mathbf{x}_T)$ that are extracted by the encoder. Specifically, we propose to use a 3D CNN which has recently been demonstrated to capture well temporal dynamics in video splits [21]. In this paper, we use a 3D CNN to build the higher-level representations that preserve and summarize the local motion of a video clip. According to [21], a deep 3D CNN network contains 8 3D convolution

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/LSP.2016.2611485, IEEE Signal Processing Letters

layers, 5 3D max-pooling layers, and 2 fully connected layers. For simplicity, we refer 3D convolution as $C(k, d, f, s_t, s_p)$ and pooling kernels as $P(d, f, s_t, s_p)$, where k is the number of kernels, d is temporal depth, f is the spatial size, s_t is the temporal stride and s_p is the spatial stride. Using shorthand notations, the 3D CNN that is used in our method can be represented as: *Conv(64,3,3,1,1)*, *Pool(1,2,1,2)*, *Conv(128,3,3,1,1)*, *Pool(2,2,2,2)*, *Conv(256,3,3,1,1)*, *Conv(256,3,3,1,1)*, *Pool(2,2,2,2)*, *Conv(512,3,3,1,1)*, *Conv(512,3,3,1,1)*, *Pool(2,2,2,2)*, *Conv(512,3,3,1,1)*, *Conv(512,3,3,1,1)*, *Pool(2,2,2,2)*, *FC(4096)*, *FC(4096)*. where $FC(n)$ is full connected layer with 4096 output units.

Sequence-to-sequence model. Recent advances in machine translation has shown that recurrent Neural Network, especially Long Short-Term Memory (LSTM) has potential to efficiently map sequences to sequences by incorporating memory units [2].

The main idea to handle the relationship between several video clips is to first encode the salience-aware video clips with 3D CNN, one at a time, representing the video using a set of latent vector representations, and then decode from that representations to action names. Let us briefly introduce the basic LSTM unit, which consists of a single memory cell, an input activation function, and three gates (input i_t, forget f_t and output o_t). i_t allows incoming signal to alter the state of the memory cell or block it. f_t controls what to be remembered and what to be forgotten by the cell and somehow can avoid the gradient from vanishing or exploding when back propagating through time. Finally, o_t allows the state of the memory cell to have an effect on other neurons or prevent it. These additions to the single memory cell enable LSTM to capture extremely complex and long-term temporal dynamics and to overcome the vanishing gradients problems. Based on the LSTM unit, for an input x_t at time step t, the LSTM computes a hidden/control state h_t and a memory cell state c_t which is an encoding of everything the cell has observed until time t:

$$\begin{aligned}
&i_t = \sigma\left(W_{xi}x_t + W_{hi}h_{t-1} + b_i\right)\\
&f_t = \sigma\left(W_{xf}x_t + W_{hf}h_{t-1} + b_f\right)\\
&o_t = \sigma\left(W_{xo}x_t + W_{ho}h_{t-1} + b_o\right)\\
&g_t = \sigma\left(W_{xg}x_t + W_{hg}h_{t-1} + b_g\right)\\
&c_t = f_t \odot c_{t-1} + i_t \odot g_t\\
&h_t = o_t \odot \phi\left(c_t\right)
\end{aligned}$$

where σ is the sigmoidal non-linearity, ϕ is the hyperbolic tangent non-linearity, $\odot$ is the element-wise product with the gate value, W_{ij} is the weight matrices and b_j is the bias, $\sigma(x)$ is the logistic sigmoid non-linearity and $\phi(x)$ is the hyperbolic tangent activation function.

LSTM with time series pooling. Temporal feature pooling can be incorporated directly as a layer and has been extensively used for video classification [14], [12]. This allows us to implement with the location of the temporal pooling layer with respect to the LSTM network architecture. By exploring various types of temporal pooling, we consider using both mean-pooling and max-pooling, which has several desirable properties shown in [12]. Next, we concatenate the mean-pooling and max-pooling features into a vector $\mathbf{Z}$ as the final video-level descriptor and feed it into a classifier loss layer.

Thus, given an input sequence $\mathbf{X} = (\mathbf{x}_1, \cdots, \mathbf{x}_T)$ in the encoding phase, the LSTM computes a sequence of hidden states $(h_1, h_2, \cdots, h_L)$. A probability distribution of an action category $P_{U,W}(y)$ is calculated by taking a softmax over the output $\mathbf{Z}$ of the time series pooling layer. U is the parameters for the C3D network. Finally, The distribution is computed over a space C (action categories) by the following equation:

$$P_{U,W}\left(y = c|\mathbf{x}_1, \mathbf{x}_2, \cdots, \mathbf{x}_t, \cdots, \mathbf{x}_T\right) = \frac{\exp\left(S_{zc}\mathbf{Z} + b_c\right)}{\sum\limits_{c' \in C}\exp\left(S_{zc'}\mathbf{Z} + b_{c'}\right)}$$

where S_{zc},$S_{zc'}$, b_c and $b_{c'}$ are the parameters for the softmax layer.

C. Optimization.

In our framework, the weight parameters of scLSTM can be learned jointly by using mini-batch SGD algorithm to minimize the negative log likelihood $L(U, W, S) = \sum\limits_{i \in D} -\log\left(P_{U,W}(y)\right)$. The batch-size is set to 30. Learning rate is set to $1e^{-4}$. The optimization is finished after 80K iterations. To mitigate the risk of over-fitting, we use the pretrained C3D model trained on the large-scale Sports-1M dataset in [21] to initialize 3D CNN and then fine tune it on the target dataset. One of the most appealing aspects of the described approach is the ability to learn the parameters ”end-to-end”.

III. Experiments

We conduct experiments with two goals. First, we study the influence of LSTM and saliency in our algorithm. Second, we compare our results with other state-of-the-art algorithms.

Datasets. In this experiment, we use two datasets: UCF101 dataset [18] and HMDB-51[10]. Specifically, UCF101 consists of 13,320 videos of 101 human action categories. It is one of two most challenging data sets to date. Moreover, the HMDB-51 dataset has 6,766 videos organized as 51 distinct action categories. They are collected from a wide range of sources. This dataset is quite challenging for it contains complex context environments and a small amount of training videos. In addition, both UCF-101 and HMDB-51 have 3 split settings, thus we report the mean accuracy over three splits.

Baselines and evaluation metrics. We compare our method with a few baselines: 1) The current best hand-crafted feature: improved dense trajectories (iDT) that normalize histogram of trajectories, HOG, HOG and MBH features to form a 25,000 dimensional feature vector for a video; 2) Deep 2D CNN and 3D CNN features are separately extracted and then input to a linear SVM; and 3) End-to-end methods taking only RGB frames as inputs, such as deep networks [8], Spatial stream network [16], Long-term recurrent convolutional network (LRCN) [2] and factorized spatio-temporal convolutional networks (FstCN) [19]. We report the mean accuracy for each of these methods.

Comparison of network architectures. We first study the effect of the submodule of the scLSTM framework and compare different architectures on the split 1 of the UCF101

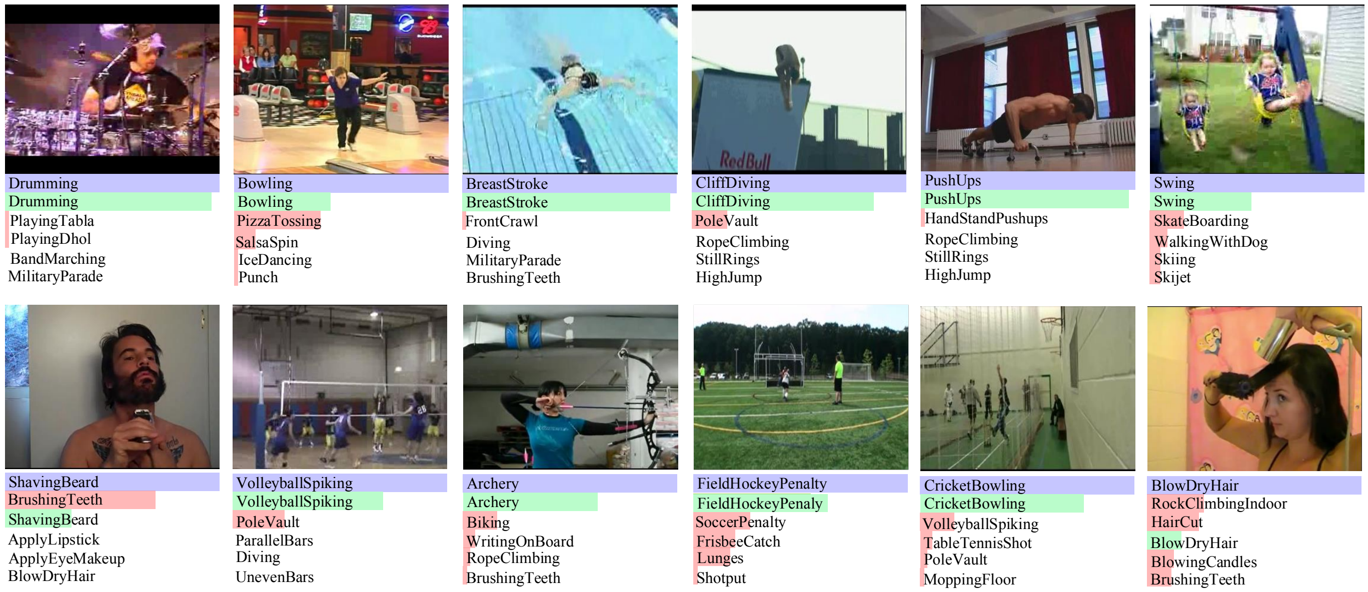

Fig. 2. Predictions on UCF-101 test data. Purple (first row) indicates ground truth label and the bars below show model predictions sorted in decreasing confidence. Green and red distinguish correct and incorrect predictions, respectively.

dataset. In experiments, we fix the value of K is 16 and the max number of T is 10. Results in Table.I shows that 3D CNN+LSTM outperforms 3D CNN. The increased accuracy is probably due to the advances in considering the relationship between video clips, which is also depicted in Long-term recurrent convolutional network (LRCN) [2]. From the results, we find that time series pooling over the output of LSTM provides better performance than LSTM without time series pooling. Moreover, integrating salience-aware mechanism with time series pooling based LSTM provides the best performance 84.7%. This suggests that the performance of action recognition largely depends on the relationship between clips and the salience regions.

TABLE I
RESULTS OF DIFFERENT SETTINGS ON THE UCF101(SPLIT 1)

Training setting	Accuracy(%)
3D CNN	80.0
3D CNN + LSTM	82.8
3D CNN + LSTM+Time_Pooling	83.6
Saliency+3D CNN + LSTM+Time_Pooling	**84.7**

Results. We show that the comparison of our approach with the baselines in Table.II and Fig.2. Firstly, the experimental results on HMDB51 show that our approach outperforms the state-of-the-art methods. Specifically, our proposed method outperform the currently best approach (Deep 3D CNN) on mean accuracy by 3.2%. Secondly, we have several observations from results on UCF101: 1) iDT performs better than Imagenet, but the dimension of the iDT (i.e, 25K) is extremely higher than Imagenet (i.e., 4096); 2) Deep 3D CNN performs better than iDT features and 2D CNN features; 3)Based on the only RGB and end-to-end setting, our approach outperform the best state-of-the-art methods by 3.8%. Thirdly, the quantitative examples in Fig. 2 shows that our approach can achieve promising performance in practice.

TABLE II
ACTION RECOGNITION RESULTS ON UCF101 AND HMDB51 OVER 3 SPLITS: SPLIT@1, SPLIT@2 AND SPLIT@3. OUR METHOD IS COMPARED WITH BASELINES AND CURRENT STATE-OF-THE-ART METHODS. TOP: THE BEST HANDCRAFTED FEATURES WITH LINEAR SVM; MIDDLE: 2D CNN AND 3D CNN FEATURES WITH LINEAR SVM (NONE END-TO-END); BOTTOM: END-TO-END METHODS TAKING ONLY RGB FRAMES AS INPUTS.

Method	HMDB51	UCF101
iDT w/BoW + linear SVM [23]	52.1	76.2
Imagenet + linear SVM [21]	-	68.8
Deep 3D CNN+linear SVM [21]	-	82.3
Deep networks [8]	-	65.4
LRCN [2]	-	71.1
Spatial Stream Network [16]	40.5	73.0
FstCN (only 1 path) [19]	49.3	76.0
Deep 3D CNN [21]	51.9	81.2
scLSTM	**55.1**	**84.0**

IV. CONCLUSIONS

In this paper, we propose a general framework for video action recognition. We firstly utilize saliency-aware method to generate a video that enhances the importance of foreground regions. Next, we integrate C3D net, LSTM and time pooling for extracting the most representative features for videos. Experiments on the UCF101 and HMDB51 demonstrate the superiority of our scLSTM compared to others.

V. ACKNOWLEDGMENTS

This work is supported by the Fundamental Research Funds for the Central Universities (Grant no. ZYGX2014J063) and the National Natural Science Foundation of China (Grant no. 61502080).

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/LSP.2016.2611485, IEEE Signal Processing Letters

References

[1] P. Dollar, V. Rabaud, G. Cottrell, and S. Belongie. Behavior recognition via sparse spatio-temporal features. In *ICCCN*, pages 65–72, 2005.

[2] J. Donahue, L. A. Hendricks, S. Guadarrama, M. Rohrbach, S. Venugopalan, K. Saenko, and T. Darrell. Long-term recurrent convolutional networks for visual recognition and description. In *CVPR*, 2015.

[3] L. Gao, J. Song, X. Liu, J. Shao, J. Liu, and J. Shao. Learning in high-dimensional multimedia data: the state of the art. *Multimedia Systems*, pages 1–11, 2015.

[4] L. Gao, J. Song, F. Nie, Y. Yan, N. Sebe, and H. T. Shen. Optimal graph learning with partial tags and multiple features for image and video annotation. In *2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 4371–4379, June 2015.

[5] L. Gao, J. Song, F. Nie, F. Zou, N. Sebe, and H. T. Shen. Graph-without-cut: An ideal graph learning for image segmentation. In *Proceedings of the Thirtieth AAAI Conference on Artificial Intelligence, February 12-17, 2016, Phoenix, Arizona, USA.*, pages 1188–1194, 2016.

[6] A. Graves and N. Jaitly. Towards end-to-end speech recognition with recurrent neural networks. In T. Jebara and E. P. Xing, editors, *Proceedings of the 31st International Conference on Machine Learning (ICML-14)*, pages 1764–1772. JMLR Workshop and Conference Proceedings, 2014.

[7] S. Ji, W. Xu, M. Yang, and K. Yu. 3d convolutional neural networks for human action recognition. *IEEE Trans. Pattern Anal. Mach. Intell.*, 35(1):221–231, 2013.

[8] A. Karpathy, G. Toderici, S. Shetty, T. Leung, R. Sukthankar, and F. Li. Large-scale video classification with convolutional neural networks. In *CVPR*, pages 1725–1732, 2014.

[9] A. Krizhevsky, I. Sutskever, and G. E. Hinton. Imagenet classification with deep convolutional neural networks. In F. Pereira, C. J. C. Burges, L. Bottou, and K. Q. Weinberger, editors, *Advances in Neural Information Processing Systems 25*, pages 1097–1105. Curran Associates, Inc., 2012.

[10] H. Kuehne, H. Jhuang, E. Garrote, T. Poggio, and T. Serre. HMDB: a large video database for human motion recognition. In *Proceedings of the International Conference on Computer Vision (ICCV)*, 2011.

[11] I. Laptev, M. Marszalek, C. Schmid, and B. Rozenfeld. Learning realistic human actions from movies. In *CVPR*, 2008.

[12] J. Y.-H. Ng, M. Hausknecht, S. Vijayanarasimhan, O. Vinyals, R. Monga, and G. Toderici. Beyond short snippets: Deep networks for video classification. In *Computer Vision and Pattern Recognition (CVPR), 2015 IEEE Conference on*, pages 4694–4702, June 2015.

[13] S. Ren, K. He, R. Girshick, and J. Sun. Faster R-CNN: Towards real-time object detection with region proposal networks. In *Advances in Neural Information Processing Systems (NIPS)*, 2015.

[14] M. S. Ryoo, B. Rothrock, and L. Matthies. Pooled motion features for first-person videos. In *CVPR*, 2015.

[15] P. Scovanner, S. Ali, and M. Shah. A 3-dimensional sift descriptor and its application to action recognition. In *ACM Multimedia*, pages 357–360, 2007.

[16] K. Simonyan and A. Zisserman. Two-stream convolutional networks for action recognition in videos. In *NIPS*, pages 568–576, 2014.

[17] J. Song, L. Gao, F. Nie, H. Shen, Y. Yan, and N. Sebe. Optimized graph learning with partial tags and multiple features for image and video annotation. *IEEE Transactions on Image Processing*, PP(99):1–1, 2016.

[18] K. Soomro, A. R. Zamir, and M. Shah. UCF101: A dataset of 101 human actions classes from videos in the wild. *CoRR*, abs/1212.0402, 2012.

[19] L. Sun, K. Jia, D. Yeung, and B. E. Shi. Human action recognition using factorized spatio-temporal convolutional networks. *CoRR*, abs/1510.00562, 2015.

[20] I. Sutskever, O. Vinyals, and Q. V. Le. Sequence to sequence learning with neural networks. In Z. Ghahramani, M. Welling, C. Cortes, N. D. Lawrence, and K. Q. Weinberger, editors, *Advances in Neural Information Processing Systems 27*, pages 3104–3112. Curran Associates, Inc., 2014.

[21] D. Tran, L. Bourdev, R. Fergus, L. Torresani, and M. Paluri. Learning spatiotemporal features with 3d convolutional networks. In *2015 IEEE International Conference on Computer Vision (ICCV)*, pages 4489–4497, Dec 2015.

[22] O. Vinyals, A. Toshev, S. Bengio, and D. Erhan. Show and tell: A neural image caption generator. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2015.

[23] H. Wang and C. Schmid. Action recognition with improved trajectories. In *IEEE International Conference on Computer Vision*, Sydney, Australia, 2013.

[24] W. Wang, J. Shen, and F. Porikli. Saliency-aware geodesic video object segmentation. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2015.

[25] M. D. Zeiler and R. Fergus. Visualizing and understanding convolutional networks. In *ECCV*, pages 818–833, 2014.

[26] S. Zhang, D. Cheng, M. Zong, and L. Gao. Self-representation nearest neighbor search for classification. *Neurocomputing*, 195:137–142, 2016.

[27] X. Zhu, X. Li, and S. Zhang. Block-row sparse multiview multilabel learning for image classification. *IEEE Transactions on Cybernetics*, 46(2):450–461, Feb 2016.

[28] X. Zhu, X. Li, S. Zhang, C. Ju, and X. Wu. Robust joint graph sparse coding for unsupervised spectral feature selection. *IEEE Transactions on Neural Networks and Learning Systems*, PP(99):1–13, 2016.

[29] X. Zhu, L. Zhang, and Z. Huang. A sparse embedding and least variance encoding approach to hashing. *IEEE Transactions on Image Processing*, 23(9):3737–3750, Sept 2014.

[30] X. Zhu, S. Zhang, Z. Jin, Z. Zhang, and Z. Xu. Missing value estimation for mixed-attribute data sets. *IEEE Transactions on Knowledge and Data Engineering*, 23(1):110–121, Jan 2011.