SCNN: SEQUENTIAL CONVOLUTIONAL NEURAL NETWORK FOR HUMAN ACTION RECOGNITION IN VIDEOS

Hao Yang, Chunfeng Yuan**, Junliang Xing, Weiming Hu*

CAS Center for Excellence in Brain Science and Intelligence Technology;
National Laboratory of Pattern Recognition, Institute of Automation, Chinese Academy of Sciences;
University of Chinese Academy of Sciences, Beijing, China
{hao.yang, cfyuan, jlxing, wmhu}@nlpr.ia.ac.cn

ABSTRACT

Convolutional Neural Network (CNN) and Recurrent Neural Network (RNN) are two typical kinds of neural networks. While CNN models have achieved great success on image recognition due to their strong abilities in abstracting spatial information from multiple levels, RNN models have not achieved significant progress in video analyzing tasks (*e.g.* action recognition), although RNN can inherently model temporal dependencies from videos. In this work, we propose a Sequential Convolutional Neural Network, denoted as SCNN, to extract effective spatial-temporal features from videos, thus incorporating the strengths of both convolutional operation and recurrent operation. Our SCNN model extends RNN to directly process feature maps, rather than vectors flattened from feature maps, to keep spatial structures of the inputs. It replaces the full connections of RNN with convolutional connections to decrease parameter numbers, computational cost, and over-fitting risk. Moreover, we introduce asymmetric convolutional layers into SCNN to reduce parameter numbers and computational cost further. Our final SCNN deep architecture used for action recognition achieves very good performances on two challenging benchmarks, UCF-101 and HMDB-51, outperforming many state-of-the-art methods.

Index Terms— SCNN, Recurrent Neural Networks, Convolutional Neural Networks, Action Recognition

1. INTRODUCTION

Recognition of human actions in videos has long been a fundamental computer vision problem [1, 2, 3, 4, 5] with many important applications, such as intelligent surveillance, human-machine interaction, and video retrieve. Unlike object recognition in images, action recognition in videos needs to jointly model the spatial-temporal information, which makes it a very difficult problem.

In the last five years, Convolutional Neural Network (C-NN) models have exhibited amazing performance on many image related tasks like image classification [6, 7, 8], object detection [9], and pose estimation [10], to name a few. The main reason is due to its strong ability to extract discriminative spatial patterns at multiple levels. For human action recognition in video, however, CNN models are not so successful as those tasks in image, although many attempts have been made [2, 4, 5]. This is because temporal information is missed in the typical pipeline of a CNN model, *i.e.*, weights shared convolution operation on two dimensional images followed by max/average pooling operation.

To capture temporal information, Recurrent Neural Network (RNN) models [11, 12] provide an appropriate choices, since its current prediction bases on not only the current observation but also history information restored in hidden states. Due to this reason, RNN models are wildly applied into action recognition to model the dynamic motion features from videos [3, 13, 14, 15]. A general pipeline for these methods is to extract spatial features using a CNN model and then flatten the feature maps as input vectors to the RNN model for classification. Since these methods extract the spatial features and model the temporal dependencies at two independent phases, they may not extract effective spatial-temporal features for action recognition. Also, flattening the two dimensional feature maps into one dimensional feature vectors is likely to loss the spatial structure information about scene and actors in videos.

In order to simultaneously model the spatial structure information and temporal dynamic information of video, we propose a Sequential Convolutional Neural Network (SCNN) layer, which inherits the strength of convolutional operation and recurrent operation, then stack several SCNN layers with convolutional layers to construct a SCNN model for human action recognition, which is shown in Fig. 1. The most significant point of our deep architecture is the extensively adoption of our proposed SCNN model, which directly feeds the two dimensional convolved feature maps into the recurrent model. This designation makes the learning of the spatial information and temporal information into a single framework and permits better spatial-temporal feature representations.

Some attempts have been made to design sequential convolution in speech recognition [16] and machine translation [17] tasks. The employed sequential convolution is still performed on one dimensional vector and is thus different from ours. Other attempts to fuse convolutional network and recurrent network can also be found in other domains, such as dialogue topics tracking [18], precipitation nowcasting [19], scene labeling [20], image classification [21] and supper res-

[*] Corresponding Author

978-1-5090-2175-8/17/$31.00

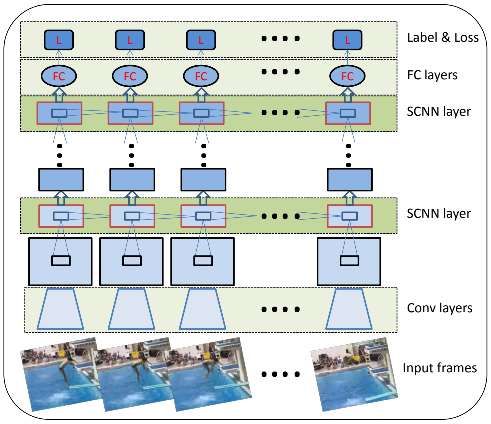

Fig. 1: The architecture of our SCNN model.

olution [22]. Also in action recognition [23, 24], GRU-RCN [23] models spatial-temporal features in two phases. Firstly, it extracts layer-wise convolutional features for each frame, and then feed them to the correspond layers of Gated Recurrent Network to model temporal dependencies. VideoLSTM [24] introduces convolutional operation into Attention LSTM to classify actions, which extracts spatial-temporal features from weighted RGB frames and the weights are computed from soft-attention LSTM network [25]. Meanwhile, the architecture of our SCNN model, as shown in Fig. 1, is different from that of the GRU-RCN and VideoLSTM models.

To summaries, the main contributions of this work are three-fold as follows:

- We propose a SCNN layer, which incorporates the advantages of convolutional operation and recurrent operation, to feed two dimensional feature maps recurrently and learn better spatial-temporal features.
- We design a "double deep" architecture both in spatial and temporal domains by stacking the SCNN layers and convolutional layers, which is end-to-end trainable and adapted to the action recognition task.
- We evaluate SCNN models on two most challenging action datesets, UCF-101 and HMDB-51, with very competitive performance compared to many state-of-the-art methods.

2. THE PROPOSED SCNN MODELS

In this section, we firstly propose a method to formulate the general SCNN layer from convolutional operation and recurrent operation. Then in order to model long-short term motion patterns from actions, we propose the Long-term Sequential Convolutional Neural Network (L-SCNN) built on the LSTM model. At last, we describe the designation of our SCNN based deep architecture for action recognition.

2.1. General SCNN Model

CNN model abstracts spatial information from local receptive fields and extracts multi-level features with better invariance

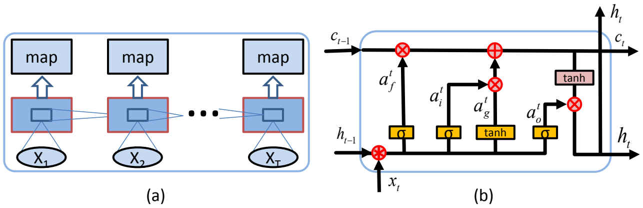

Fig. 2: (a) The general SCNN layer, which simultaneously models the spatial structure information and temporal dynamic information from videos. (b) The specific unit of L-SCNN layer, which introduces LSTM into SCNN model to extract long term motion features.

using weights sharing and localized pooling mechanism. Denote the weights of convolutional kernel as W and the bias as b. The operation of convolutional layer is defined as:

$$F(X|W,b) = W * X + b, \tag{1}$$

where $*$ represents convolution operation. Recurrent network is designed to learn temporal dependencies from sequential data by remembering the history information in hidden states. Let H_t denote the t^{th} hidden states of recurrent layer and it is computed as follows:

$$H_t = W_{xh}X + W_{h'h}H_{t-1} + b, \tag{2}$$

where $t = 1, ..., T$, W_{xh} and $W_{h'h}$ are the weights of input-hidden and hidden-hidden connections of recurrent layer.

Traditional recurrent network accepts vectors as input, which ignores the spatial structure information. Moreover, the full connections between input, hidden and output units involve too many parameters, which are costly. On the contrary, the convolutional layers share weights in local regions, which abstracts spatial information effectively and decreases the parameter numbers dramatically. To incorporate the convolutional operation and recurrent operation into a single framework, we replace full connections of recurrent network with convolutional connections, as shown in Fig. 2 (a). Based on Eq. 1 and Eq. 2, we compute the t^{th} hidden states of SCNN layer as follows:

$$\begin{aligned} H_t &= F(X|W_{xh}, b_1) + F(H_{t-1}|W_{h'h}, b_2) + b & (3)\\ &= W_{xh} * X + W_{h'h} * H_{t-1} + b'. & (4)\end{aligned}$$

The SCNN layer models the spatial and temporal information simultaneously, which enables it extract effective spatial-temporal feature representations.

2.2. Long-term SCNN Mode (L-SCNN)

General recurrent networks have difficulties in learning long-term dependencies due to the vanishing gradient problem [26]. The LSTM [12] is proposed to address this problem by adding a memory unit to decide remembering or forgetting previous hidden states. In order to let LSTM operate on two dimensional convolved feature maps directly, we introduce convolutional operation into the LSTM and propose Long-term Sequential Convolutional Neural Network (L-SCNN). A single unit of L-SCNN layer is shown in Fig. 2 (b). The

Table 1: The architectures of SCNN deep models.

SCNN-M(L1)	L-SCNN-M(L2)	L-SCNN-16
input (224 × 224 × 16 frames)		
conv (7, 2, 96)	conv (7, 2, 96)	conv (3, 1, 64) conv (3, 1, 64)
maxpool (3, 2)	maxpool (3, 2)	maxpool (3, 2)
conv (5, 2, 256)	conv (5, 2, 256)	conv (3, 1, 128) conv (3, 1, 128)
maxpool (3, 2)	maxpool (3, 2)	maxpool (3, 2)
conv (3, 1, 512)	conv (3, 1, 512)	conv (3, 1, 256) conv (3, 1, 256) conv (3, 1, 256)
		maxpool (3, 2)
conv (3, 1, 512)	L-SCNN (3, 1, 512)	conv (3, 1, 512) conv (3, 1, 512) L-SCNN (3, 1, 512)
		maxpool (3, 2)
SCNN (3, 1, 512)	L-SCNN (3, 1, 512)	conv (3, 1, 512) conv (3, 1, 512) L-SCNN (3, 1, 512)
maxpool (3, 2)	maxpool (3, 2)	maxpool (3, 2)
FC (4096)	FC (4096)	FC (4096)
FC (2048)	FC (2048)	FC (4096)
FC (101/51)		

forget gates, input gates, output gates and candidate cell states of the L-SCNN layer at t^{th} timestep are denoted as F_t, I_t, O_t and G_t respectively. Let C_t represent the cell states at the t^{th} timestep and H_t be the hidden states of L-SCNN layer. The forward pass of the L-SCNN layer is defined as follows:

$$\begin{pmatrix} F_t \\ I_t \\ O_t \\ G_t \end{pmatrix} = \begin{pmatrix} \sigma \\ \sigma \\ \sigma \\ \tanh \end{pmatrix} (W_{ih} * X_t + W_{h'h} * H_{t-1} + b), \quad (5)$$

$$C_t = F_t \odot C_{t-1} + I_t \odot G_t, \quad (6)$$

$$H_t = O_t \odot \tanh(C_t), \quad (7)$$

where $\tanh(x) = \frac{e^x - e^{-x}}{e^x + e^{-x}}$, $\sigma(x) = \frac{1}{1+e^{-x}}$, and the $\odot$ represents element-wise operation on 3D tensors, such as the gates I_t, F_t, O_t, the candidate cell states G_t, cell states C_t, C_{t-1} and hidden states H_t, H_{t-1}. The two dimensional convolutional weights W_{ih} and $W_{h'h}$ enable traditional LSTM preserve spatial structure information from successive frames.

To model complex dynamic motion features from videos, more convolutional layers in hidden-hidden connections of SCNN layer can be added. Unfortunately, it will increase computational cost dramatically because every convolutional layer in hidden-hidden connections of SCNN layer will be propagated T times. We thus introduce asymmetric convolutional layers into hidden-hidden connections of SCNN layer, which improves temporal representative ability without increasing parameter numbers and computational cost.

2.3. Overall Architecture

To reduce feature map size and computation cost, we stack several convolutional layers and max pooling layers in front of SCNN layers to construct SCNN deep models. The architecture of our SCNN models is illustrated in Table 1. The SCNN-M(L1) replaces the last convolutional layer of VGG-M-2048 [8] with single SCNN layer, including R-SCNN-M(L1) and L-SCNN-M(L1), using the general SCNN layer and L-SCNN layer correspondingly. The second SCNN model stacks two L-SCNN layers on convolutional layers of VGG-M-2048, referred to as L-SCNN-M(L2). To extract more representative spatial-temporal features from videos, we design a very deep SCNN model by combining VGG-16 [6] with two L-SCNN layers, referred to as L-SCNN-16.

Table 2: Evaluating the effectiveness of convolutions in SCNN layers on the UCF-101 dataset.

Models	Parameter numbers	Performance
SpatialNet [4]	90.62M	72.8
LRCN-fc6 [3]	86.24M	70.84
R-SCNN-M(L1)	92.98M	73.58
L-SCNN-M(L1)	107.13M	**73.75**

3. EXPERIMENTS

3.1. Experimental Settings

We evaluate our model on two challenging datasets, UCF-101 [27] and HMDB-51 [28] datasets. The UCF-101 dataset includes 101 action categories with 13320 videos. We report the average accuracy of the three splits. The HMDB-51 dataset contains 6849 clips divided into 51 action categories. We use the standard splits from [28]. We initialize our SCNN models from CNN models, VGG-M-2048 [8] or VGG-16 [6]. Then we pre-trained the SCNN models on the FCVID dataset to initialize the SCNN models more carefully.

We train our SCNN models by Stochastic Gradient Decent (SGD) and Back Propagation Thought Time (BPTT). We set the batch-size as 8, the momentum as 0.9, the weight-decay as 0.0005, and the clip-gradient as 5. We use initial learning rate 0.001, divide it by 10 after every 20K iterations, and stop at 50K iterations. We perform data augmentation techniques like randomly clipping, multi-scale cropping and randomly flipping to avoid over-fitting. During test, we randomly sample 10 clips from each video, and use the standard 10-views of each clip. We weight predictions from RGB and optical flow networks using weights of $\frac{1}{3}$ and $\frac{2}{3}$ respectively.

3.2. Evaluation of Sequential Convolution

In the first experiment, we test two designations of our SCNN models, R-SCNN-M(L1) and L-SCNN-M(L1), compared with LRCN-fc6 [3] and SpatialNet [4] models, to evaluate the effectiveness of convolutional operation in SCNN layers. To make a fair comparison, we modify the convolutional layers of LRCN-fc6 same as VGG-M-2048 [8] and the four models in this experiment are initialized from VGG-M-2048. We test all the models on the UCF-101 dataset and list parameter numbers and performances in Table 2.

As shown in Table 2, LRCN-fc6 [3] achieves 70.84% on the UCF101 dataset, which is weakly lower than 71.12% reported in [3], but it is much lower than 72.8% of SpatialNet, which indicates the LSTM in LRCN-fc6 cannot learn effective temporal dependencies from flattened feature vectors. The two SCNN-M(L1) models outperform both the SpatialNet and LRCN-fc6 models. It indicates: 1) our SCNN models can learn effective temporal dependencies from two dimensional feature maps, compared with SpatialNet; 2) feature

Table 3: Evaluating the fusion of deep SCNN models on UCF-101 and HMDB-51 datasets.

Models	UCF-101			HMDB-51		
	RGB	Flow	fusion	RGB	Flow	fusion
Two-stream [4]	73.0	83.7	86.9	40.5	54.6	58.0
Deeper Two [29]	78.4	87.0	91.4	-	-	-
LRCN-fc6 [3]	71.12	76.95	82.92	-	-	-
L-SCNN-M	77.53	85.14	87.76	50.12	57.45	59.34
L-SCNN-16	84.13	89.17	91.98	56.73	61.32	64.47

maps are better than flattened vectors for recurrent networks to extract spatial-temporal features, compared with LRCN-fc6, because feature maps preserve useful spatial structure information. L-SCNN-M(L1) does not show much advantages against R-SCNN-M(L1), partially because L-SCNN-M(L1) has more parameters without appropriate initialization.

3.3. Evaluation of SCNN Architecture

In the second experiment, we design deeper SCNN models by two techniques. Firstly, we stack more L-SCNN layers on convolutional layers, which is referred to as L-SCNN-M(L2). Secondly, we use two asymmetric convolutional layers, 5×1 and 1×5, in the hidden-hidden connections of L-SCNN layers, referred to as L-SCNN-M(L2+asy). All the models are initialized from VGG-M-2048. After that, the SCNN models are pre-trained on the FCVID video dataset. We evaluate these SCNN models on the UCF-101 dataset.

In Fig. 3, all the L-SCNN-M variants outperform SpatialNet [4]. Especially, the L-SCNN-M(L2+asy) surpasses SpatialNet around 2%. This indicates the L-SCNN-M models learn effective temporal dependencies from videos. The performance of L-SCNN-M(L2) is weakly lower than L-SCNN-M(L1) without pre-training because the L-SCNN-M(L2) has more parameters and higher risk of over-fitting. However, with better initialization by pre-training on the FCVID dataset, The L-SCNN-M(L2) outperforms L-SCNN-M(L1) over 1%. Comparing with L-SCNN-M(L2), the L-SCNN-M(L2+asy) with asymmetric convolutional layers improves the performance a lot. So we employ asymmetric convolutional layers and two L-SCNN layers in the next experiments.

In the third experiment, we evaluate the very deep SCNN model and fusion prediction of two networks fed with RGB and optical flow respectively. In Table 3, we compare our models, L-SCNN-M and L-SCNN-16, with Two-stream [4], Deeper Two-stream [29] and LRCN-fc6 [3] models on UCF-101 and HMDB-51 datasets. Our L-SCNN-M model outperforms Two-stream by 1.34% on HMDB-51 dataset and outperforms LRCN-fc6 about 5% on UCF-101 dataset. When fed with RGB only, L-SCNN-M outperforms Two-stream over 9% on HMDB-51 dataset and outperforms LRCN-fc6 over 6% on UCF-101 dataset. The L-SCNN-16 and Deeper Two-stream are both extended from VGG-16. And the L-SCNN-16 outperforms Deeper Two-stream by 5.63%, 2.22% and 0.58% for RGB, optical flow and fusion accuracies on the UCF-101 dataset respectively. From the last two rows of Table 3, we can conclude that increasing the spatial depth of the SCNN models can improve the performance of action recognition significantly.

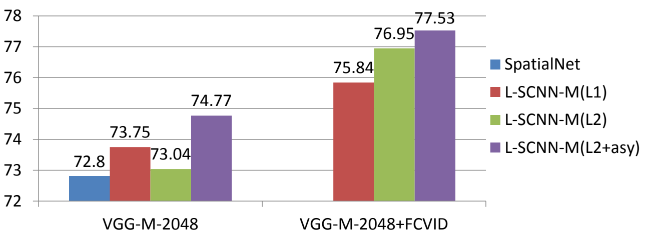

Fig. 3: Evaluating several variants of L-SCNN-M models on UCF-101 dataset with different initialization.

Table 4: Comparing with current state-of-the-art methods.

UCF-101		HMDB-51	
Wang et al. [1]	85.9	Wang et al. [1]	57.2
Peng et al. [30]	87.9	Peng et al. [30]	61.1
Simonyan et al. [4]	88.0	Simonyan et al. [4]	59.4
Sun et al. [31]	87.9	Sun et al. [31]	58.6
Wang et al. [29]	91.4	Wang et al. [32]	63.4
Christoph et al. [33]	**92.5**	Wu et al. [34]	56.4
Donahua et al. [3]	82.3	Zhu et al. [35]	63.3
Ballas et al. [23]	90.8	Wang et al. [36]	63.2
L-SCNN-16	92.0	L-SCNN-16	**64.5**

3.4. Comparisons with State-of-the-arts

In Table 4, we list the results of current state-of-the-art methods for action recognition on UCF-101 and HMDB-51 benchmarks. The L-SCNN-16 model outperforms all the traditional methods [1, 30] on two datasets. It also outperforms many deep learning based methods. On the UCF-101 dataset, our model outperforms Two-stream [4] by 4% and outperforms GRU-RCN [23] by 1.2%, and the L-SCNN-16 model is comparable with Deeper Two-stream [29] models and Fusion Two-stream [33]. On the HMDB-51 dataset, our SCNN model outperforms Two-stream [4] over 5%, FstCN [31] by 5.9% and Action-Transformation [32] over 1%. All these results demonstrate the effectiveness of our proposed model.

4. CONCLUSION

In this work, we have incorporated the convolutional operation and recurrent operation to propose SCNN models, which are end-to-end trainable and "double deep" in spatial and temporal domains. SCNN model permits feeding two dimensional convolved feature maps directly and extracting effective spatial-temporal features for action recognition. We have introduced asymmetric convolutional layers into hidden-hidden connections of SCNN layers to decrease the parameter numbers and improve the performance of action recognition further. And our SCNN models have demonstrated good performance on the two challenging datasets.

5. ACKNOWLEDGEMENT

This work is partly supported by the 973 basic research program of China (Grant No. 2014CB349303), the Natural Science Foundation of China (Grant No. U1636218, 61472420, 61472063, 61370185, 61472421, 61672519), the Strategic Priority Research Program of the CAS (Grant No. XD-B02070003), and the CAS External cooperation key project.

6. REFERENCES

[1] Heng Wang and Cordelia Schmid, "Action recognition with improved trajectories," in *ICCV*, 2013.

[2] Andrej Karpathy, George Toderici, Sanketh Shetty, Thomas Leung, Rahul S, and Li Feifei, "Large-scale video classification with convolutional neural networks," in *CVPR*, 2014.

[3] Jeffrey Donahue, Lisa Anne Hendricks, Sergio Guadarrama, Marcus Rohrbach, Subhashini Venugopalan, Kate Saenko, and Trevor Darrell, "Long-term recurrent convolutional networks for visual recognition and description," in *CVPR*, 2015.

[4] Karen Simonyan and Andrew Zisserman, "Two-stream convolutional networks for action recognition in videos," in *NIPS*, 2014.

[5] Limin Wang, Yuanjun Xiong, Zhe Wang, Yu Qiao, Dahua Lin, Xiaoou Tang, and Luc Van Gool, "Temporal segment networks: towards good practices for deep action recognition," in *ECCV*, 2016.

[6] Karen Simonyan and Andrew Zisserman, "Very deep convolutional networks for large-scale image recognition," *arXiv preprint arXiv:1409.1556*, 2014.

[7] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun, "Deep residual learning for image recognition," *arXiv preprint arXiv:1512.03385*, 2015.

[8] Ken Chatfield, Karen Simonyan, Andrea Vedaldi, and Andrew Zisserman, "Return of the devil in the details: Delving deep into convolutional nets," *arXiv preprint arXiv:1405.3531*, 2014.

[9] Ross Girshick, "Fast R-CNN," in *ICCV*, 2015.

[10] Alexander Toshev and Christian Szegedy, "Deeppose: Human pose estimation via deep neural networks," in *CVPR*, 2014.

[11] Jeffrey L Elman, "Finding structure in time," *Cognitive science*, vol. 14, no. 2, pp. 179–211, 1990.

[12] Sepp Hochreiter and Jürgen Schmidhuber, "Long short-term memory," *Neural Computation*, vol. 9, no. 8, pp. 1735–1780, 1997.

[13] Moez Baccouche, Franck Mamalet, Christian Wolf, Christophe Garcia, and Atilla Baskurt, "Action classification in soccer videos with long short-term memory recurrent neural networks," in *ANN*, 2010.

[14] Moez Baccouche, Franck Mamalet, Christian Wolf, Christophe Garcia, and Atilla Baskurt, "Sequential deep learning for human action recognition," in *HBU*, 2011.

[15] Yuehei Ng Joe, Matthew Hausknecht, Sudheendra Vijayanarasimhan, Oriol Vinyals, Rajat Monga, and George Toderici, "Beyond short snippets: Deep networks for video classification," in *CVPR*, 2015.

[16] Ngoc Thang Vu, "Sequential convolutional neural networks for slot filling in spoken language understanding," 2016.

[17] Gil Keren and Björn Schuller, "Convolutional rnn: an enhanced model for extracting features from sequential data," *arXiv preprint arXiv:1602.05875*, 2016.

[18] Seokhwan Kim, Rafael E Banchs, and Haizhou Li, "Exploring convolutional and recurrent neural networks in sequential labelling for dialogue topic tracking," in *ACL*, 2016.

[19] Xingjian Shi, Zhourong Chen, Hao Wang, Dit-Yan Yeung, Wai-kin Wong, and Wang-chun Woo, "Convolutional lstm network: A machine learning approach for precipitation nowcasting," in *NIPS*, 2015.

[20] Liang Ming, Hu Xiaolin, and Zhang Bo, "Convolutional neural networks with intra-layer recurrent connections for scene labeling," in *NIPS*, 2015.

[21] Ming Liang and Xiaolin Hu, "Recurrent convolutional neural network for object recognition," in *CVPR*, 2015.

[22] Yan Huang, Wei Wang, and Liang Wang, "Bidirectional recurrent convolutional networks for multi-frame super-resolution," in *NIPS*, 2015.

[23] Nicolas Ballas, Li Yao, Chris Pal, and Aaron Courville, "Delving deeper into convolutional networks for learning video representations," *arXiv preprint arXiv:1511.06432*, 2015.

[24] Zhenyang Li, Efstratios Gavves, Mihir Jain, and Cees G. M. Snoek, "VideoLSTM convolves, attends and flows for action recognition," *arXiv preprint arXiv:1607.01794v1*, 2016.

[25] Shikhar Sharma, Ryan Kiros, and Ruslan Salakhutdinov, "Action recognition using visual attention," *arXiv preprint arXiv:1511.04119*, 2015.

[26] J Kolen and S Kremer, "Gradient flow in recurrent nets: the difficulty of learning long-term dependencies," vol. 28, no. 2, pp. 237–243, 2003.

[27] Khurram Soomro, Amir Roshan Zamir, and Mubarak Shah, "UCF101: A dataset of 101 human actions classes from videos in the wild," *arXiv preprint arXiv:1212.0402*, 2012.

[28] Hildegard Kuehne, Hueihan Jhuang, Estíbaliz Garrote, Tomaso Poggio, and Thomas Serre, "HMDB: a large video database for human motion recognition," in *ICCV*, 2011.

[29] Limin Wang, Yuanjun Xiong, Zhe Wang, and Yu Qiao, "Towards good practices for very deep two-stream convnets," *arXiv preprint arXiv:1507.02159*, 2015.

[30] Xiaojiang Peng, Limin Wang, Xingxing Wang, and Yu Qiao, "Bag of visual words and fusion methods for action recognition: Comprehensive study and good practice," *CVIU*, vol. 150, pp. 109–125, 2016.

[31] Lin Sun, Kui Jia, Dit-Yan Yeung, and Bertram E Shi, "Human action recognition using factorized spatio-temporal convolutional networks," in *ICCV*, 2015.

[32] Xiaolong Wang, Farhadi Ali, and Gupta Abhinav, "Actions transformations," in *CVPR*, 2016.

[33] Christoph Feichtenhofer, Axel Pinz, and Andrew Zisserman, "Convolutional two-stream network fusion for video action recognition," *arXiv preprint arXiv:1604.06573*, 2016.

[34] Jianxin Wu, Yu Zhang, and Weiyao Lin, "Towards good practices for action video encoding," in *CVPR*, 2014.

[35] Wangjiang Zhu, Jie Hu, Gang Sun, Xudong Cao, and Yu Qiao, "A key volume mining deep framework for action recognition," in *CVPR*, 2016.

[36] Limin Wang, Yu Qiao, and Xiaoou Tang, "Action recognition with trajectory-pooled deep-convolutional descriptors," in *CVPR*, 2015.