TEMPORAL ATTENTIVE NETWORK FOR ACTION RECOGNITION

Yemin Shi[1], *Yonghong Tian*[1*], *Tiejun Huang*[1], *Yaowei Wang*[2*]

[1] National Engineering Laboratory for Video Technology, School of EE&CS,
Peking University, Beijing, China
[2] School of Information and Electronics, Beijing Institute of Technology, Beijing, China

ABSTRACT

In action recognition, one of the most important challenges is to jointly utilize the texture and motion information as well as capturing the long-term dependence of various common and action-specific postures. Motivated by this fact, this paper proposes **T**emporal **A**ttentive **N**etwork (TAN) for action recognition. The key idea in TAN is that not all postures, each of which represented by a small collection of consecutive frames, contribute equally to the successful recognition of an action. As a result, TAN incorporates two separate spatial and temporal streams into one network. Information in the two streams is partially shared so that discriminative spatiotemporal features can be extracted to characterize various postures in an action. Moreover, a temporal attention mechanism is introduced in the form of Long-Short Term Memory (LSTM) network. With this mechanism, features from the action-specific postures can be emphasized, while common postures shared by many different actions will be ignored to some extent. By jointly using such spatial and temporal information as well as attentive cues in a single network, TAN achieves impressive performance on two public datasets, HMDB51 and UCF101, with accuracy scores of 72.5% and 94.1%, respectively.

***Index Terms*—** Action Recognition, Temporal Attention, Two-stream Network, CNNs, LSTM

1. INTRODUCTION

Action recognition is one of the most challenging tasks in the areas of multimedia and computer vision, which aims at categorizing actions or behaviours of one or several persons described by a short video sequence into some predefined semantic concepts. By successfully recognizing various actions, many innovative applications such as security surveillance [1, 2], image/video captioning [3], video tagging[4] and automated driving [5, 6] can be developed to further facilitate the booming of the multimedia industry.

For the human-being, various spatial and temporal features like color, texture and motion, as well as the cognitive visual mechanisms like selective attention and working memory, can be simultaneously involved to correctly recognize an action. In classic action recognition models, an action can be identified by heuristically designed spatiotemporal features and shallow learning approaches [7, 8]. For example, Wang *et al.* [9] first extracted dense trajectories by sampling and tracking dense points from each frame at multiple scales. Hand-crafted features at each point were extracted and encoded to derive the final representation for a video to recognize the actions it contains. However, such hand-crafted features may have difficulties in representing long-range actions (*e.g.*, people hovering and slow walking).

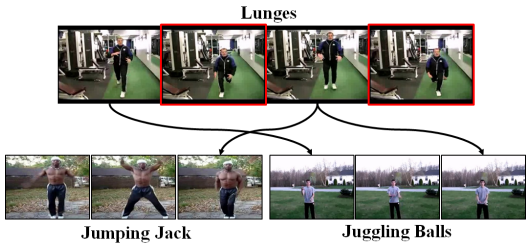

Fig. 1. An action may contain action-specific and common postures. Capturing the action-specific postures (marked in red) and ignoring the common ones may help to characterize and recognize an action.

To obtain better features, many approaches [10, 11, 10, 12, 13] adopted CNNs for learning representations and recognizing actions. In [14], Simonyan *et al.*proposed the two-stream ConvNets for action recognition. Their model first extracted the spatial and temporal features with two standalone CNNs, which were denoted as two streams. Although their model successfully recognizes some types of actions, its improvement against classic models with heuristic features are not as high as expected. This may be caused by the fact that spatial and temporal features are inherently correlated in defining an action. It may be inappropriate to utilize them separately in different models for action recognition. Although many ways [14, 10] of fusing spatial and temporal streams have been tested, the long-term dependencies between various postures, as well as their importance in recognizing an

Corresponding author: Yonghong Tian (yhtian@pku.edu.cn) and Yaowei Wang (yaoweiwang@bit.edu.cn).

978-1-5386-1737-3/18/$31.00

action, are not put into consideration. Actually, such dependencies and importance of gestures contain useful cues that depict what is unique in an action and how to recognize it.

To address this problem, many approaches introduce the cognitive visual mechanisms such as visual attention [15, 16, 17] and working memory [18, 19] into the action recognition models. For example, Wu *et al.* [20] used spatial attention to regularize the usage of features from different layers in CNNs. Yue-Hei *et al.* [21] and Donahue *et al.* [22] proposed their own recurrent networks respectively by connecting Long-Short Term Memory network (LSTM) [18] to CNNs. However, all postures are equally treated in their networks, while we show in Fig. 1 some action-specific postures may be more important than the common postures shared by many actions in distinguishing one action from the others. In other words, action recognition needs to incorporate both the temporal attention and the working memory mechanism so as to not only describes the long-term dependencies but also estimates their importance to the recognition process.

Inspired by the pros and cons of previous works, this paper proposes the **T**emporal **A**ttentive **N**etwork (TAN) that aims at jointly training the spatial and temporal streams with the assistant of temporal attention mechanism. In TAN, two separate spatial and temporal streams are integrated into one network. Information in the two streams is partially shared, which, after the training process, can extract discriminative spatiotemporal features that can well characterize various postures in an action. Moreover, we explore several ways of incorporating the attention mechanisms into action recognition, in which we find the temporal attention, introduced by implementing a Long-Short Term Memory (LSTM) network, may be an appropriate way. With the temporal attention mechanism, features from the action-specific postures can be emphasized, while common postures shared by many different actions will be ignored to some extent. By jointly using such spatial and temporal information as well as emphasizing the most discriminative frames (postures) in a single network, TAN achieves impressive performance on two public datasets.

Our main contributions are summarized as follows: 1) We propose a temporal attentive network for action recognition, which can extract effective spatiotemporal features from the most discriminative postures; 2) We investigate several ways to incorporate attention into action recognition and find that temporal attention implemented by LSTM may be among the most appropriate choices. Such a finding can be useful to the future development of such attentive networks; 3) We conduct extensive experiments to validate the effectiveness of the proposed approach, in which TAN achieve state-of-the-art performances on two benchmark dataset.

2. OUR APPROACH

In this section, we give detailed descriptions of performing action recognition with the proposed Temporal Attentive Networks. Specifically, we first introduce the basic concepts in the framework of Temporal Attentive Networks. Then, we study the good practices in learning joint network. Finally, we describe the testing details of the learned joint network.

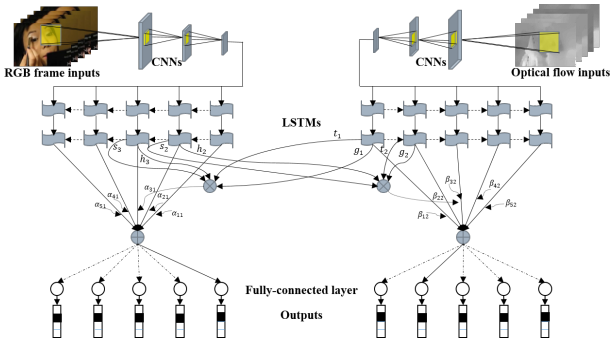

Fig. 2. The framework of the proposed Temporal Attentive Network.

2.1. Temporal Attentive Network

The proposed network is shown in Fig. 2. In TAN, there are two branches which aim at processing frame and optical flow fields. The input images will pass through CNNs to learn texture representation. The outputs of CNNs are then fed into LSTMs to learn the temporal description. The outputs of the last LSTM layer in spatial branch are denoted $(h_1, h_2, ..., h_T)$ and the outputs of the last LSTM layer of temporal branch are denoted $(g_1, g_2, ..., g_T)$. First, we use two small networks to get the global state. We define:

$$m_{ij} = w'_1 h_i + w'_2 s_i + w'_3 g_j + w'_4 t_j \quad (1)$$

$$n_{kl} = w'_5 g_k + w'_6 t_k + w'_7 h_l + w'_8 s_l \quad (2)$$

where matrices $w'_1, w'_2, \cdots, w'_8$ are the trainable parameters, s_i is the cell state of spatial branch LSTM and t_j is the cell state of temporal branch LSTM. Both networks take current outputs and cell states of two branches as inputs. In this way, m_{ij} and n_{kl} are able to see history state, texture state and motion state.

Based on the current global state, we want the network to learn the importance of each frame.

$$e_{ij} = v^T tanh(m_{ij}) \quad , \quad f_{kl} = u^T tanh(n_{kl}) \quad (3)$$

where vectors v and u are the trainable parameters. We apply hyperbolic tangent nonlinearity to the global state. Then we use v and u to project the global state into a scalar which represents how important current frame is.

It is important that we want to get the relative importance for each frame among a sequence of frames. In the same time, we need a gate to control the information exchange in m_{ij} and n_{kl}. In this paper, we use softmax to get the relative impor-

tance score as follows:

$$\alpha_{ij} = \frac{exp(e_{ij})}{\sum_{d=1}^{T} exp(e_{dj})} \quad , \quad \beta_{kl} = \frac{exp(f_{kl})}{\sum_{d=1}^{T} exp(f_{dl})} \tag{4}$$

In this formulation, the information flow is controlled by α and β. After applying softmax, most of α or β are 0 and only the most important inputs have positive weights. This ensures that only the information of these important inputs can back propagate to e and f, and finally impact both branches. Because that there is no other layer shared by two branches, α and β are the gates to control the information flow which can be shared across two branches and is called sharing gates.

Finally, the importance scores are used to weighted average all features.

$$o_j^s = \sum_{d=1}^{T} \alpha_{dj} h_d \quad , \quad o_l^t = \sum_{d=1}^{T} \beta_{dl} g_d \tag{5}$$

where o_j^s and o_l^t are the outputs of spatial branch and temporal branch respectively and followed by fully-connected layers to learn classifiers. Each input feature vector in one branch (denote as A) will be used to compute a group of weights for the other branch (denote as B), and the weights are then used to get weighted average of input feature vectors in B.

2.2. Training Details

Many works [14, 23] have shown that the stack of 10 optical flow fields will get better performance than single optical flow field. However, the more optical flow fields we are using, the more parameters there will be in the first layer. In the same time, the input itself will also consume much more memory. In order to speed up the training and testing, we use single optical flow field as input. In our implementation, the flow-x, flow-y and their quadratic mean are stacked to get a three channel image which is then used to train the temporal branch. It should be noted that single optical flow field will produce relatively worse performance than 10.

Pre-training has turn out to be an effective way to initialize deep networks when the target dataset is not big enough. As the spatial branch takes video frame as input, which is actually a RGB image, it is natural to initialize it with ImageNet [24] pre-trained model. For the temporal branch, we transform the original optical flow fields into images so that we can train it like spatial branch. We discretize the optical flow fields into interval of $[0, 255]$ by a linear transformation and save them as images. As discussed before, we stack flow-x, flow-y and their quadratic mean to get a three channel image. These steps make optical flow fields to be the same with RGB images. Then we are able to initialize the temporal branch with ImageNet pre-trained model.

In order to further improve the performance of TAN, we investigate how to train the model on small dataset. After going through no pre-training, pre-train CNNs on target dataset, pre-train CNNs on big dataset and pre-train TAN on big dataset, we find that pre-train TAN will improve the most (see the Experiment Section).

Table 1. Accuracy of TAN under different pre-training settings on the first split of HMDB51 and UCF101.

Dataset	HMDB51		UCF101	
Pre-train	TAN	TAN+†	TAN	TAN+
CNN on ImageNet	60.5%	69.0%	86.2%	91.8%
CNN‡	68.6%	73.0%	92.0%	**93.8%**
CNN on UCF101	70.3%	73.7%	-	-
TAN on UCF101	70.2%	**73.9%**	-	-

† TAN + MIFS.
‡ Pre-train the CNN part on the target dataset.

2.3. Using TAN for Action Recognition

For each time step, spatial and temporal branch will generate attention weights for each other. Because we use the cell states and outputs to compute attention weights, it is easy to understand that the last time step can predict based on the whole video history hence producing the best attention weights. Therefore, we only keep the last prediction for each sample video. Unlike many other papers [14, 23] which sample 25 RGB frames or optical flow stacks from each original video, we only use 5 samples to reduce complexity, and we find that this setting only decrease the performance slightly. Meanwhile, we crop 4 corners and 1 center, and their horizontal flipping from each sample to evaluate TAN. Finally, all predictions are averaged to get the final label.

Many works [12, 19] have proved the complementation between deep features and hand-crafted features. In order to further improve the performance, we merge TAN and MIFS [25] by late fusion in the test stage.

3. EXPERIMENTS

This section will first introduce experimental settings. Then, we describe the implementation details of our model. We will then compare TAN with several baseline methods. Finally, we report the experimental results and compare TAN with the state-of-the-art methods.

3.1. Experimental Settings

To verify the effectiveness of our methods, we conduct experiments on two benchmark datasets: HMDB51 [26] and UCF101 [27].

The HMDB51 dataset is composed of 6,766 video clips from 51 action categories, with each category containing at least 100 clips. Our experiments follow the original evaluation scheme, and average accuracy over the three train-test

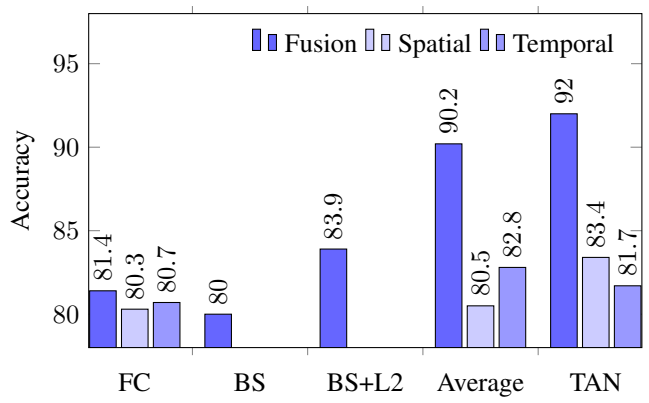

Fig. 3. Comparison of different fusion methods on the first split of UCF101.

Table 2. The accuracies of TAN on all splits of HMDB51 and UCF101.

Split	1	2	3	Mean
HMDB51	73.9%	71.7%	71.9%	72.5%
UCF101	93.8%	94.5%	94.0%	94.1%

splits is reported. UCF101 is one of the most popular action recognition datasets. It contains 13,320 video clips (27 hours in total) from 101 action classes and there are at least 100 video clips for each class. We conduct evaluations using 3 train/test splits and report the average classification accuracy.

3.2. Implementation details

We use TensorFlow [28] to implement our model. To simplify the training process and highlight the contribution of TAN, we use a very simple CNN architecture, GoogLeNet [13]. We use GRU [29] as our LSTM implementation. To extract optical flow, we choose the TVL1 optical flow algorithm [30] and use the OpenCV GPU implementation.

In the remainder of the paper, we use spatial stream and temporal stream to indicate the streams in two-stream framework, and use spatial branch and temporal branch to indicate the branches in TAN network.

3.3. Exploration study

In order to prove the effectiveness of TAN, we propose three other methods which share similar techniques with TAN. The architectures of three methods are described in the appendix. The TAN can be considered as an extension of FC fusion, branch selection and spatial attention methods. We use joint network structure like FC fusion and branch selection while avoid their "one-stream-dominating-network" problem. TAN learns attention weights on temporal domain while spatial attention learns attention weights on spatial domain. Even

Table 3. Comparison with our baseline two-stream model on the first split of HMDB51.

Module	Spatial	Temporal	Fusion
Two-stream CNNs+LSTMs	46.2%	50.3%	58.4%
TAN	51.4%	60.3%	70.2

though these methods are using similar solution, TAN is the only one which can improve baseline performance and outperform average softmax score late fusion. According to Figure 3 and Table 4, TAN is 1.8% better than average late fusion and other trails are worse than average late fusion on the first split of UCF101. As shown in Table 3, TAN is 11.8% better than average fusion on the first split of HMDB51.

As shown in Table 1, we test TAN under multiple pre-training setting on the first split of HMDB51 and UCF101. When there is no pre-training, TAN get lower accuracies than two-stream framework. However, compared with FC fusion or branch selection, there is clearly no "one-stream-dominating-network" problem and the fused model is much better than each branch. As discussed before, when training joint network, pre-training CNNs is very important because temporal branch is converging slower than spatial branch. After pre-training CNNs on the target dataset, TAN is able to achieve significantly better performance on both datasets.

For the fact that HMDB51 is smaller than UCF101, we also try to pre-train CNN or TAN on UCF101, then transfer pre-trained model to HMDB51. Pre-training on UCF101 has more impact on the spatial branch. It is possibly because spatial branch shares the same modality with ImageNet and is very easy to converge which makes it easy to be over-fitting on small dataset. Both pre-training setting generate better results than pre-train on target dataset. However, pre-train CNN or TAN produce similar performance. Because pre-training TAN on UCF101 will make it converge faster on HMDB51, in the rest of the paper, we will use the model which pre-trains TAN on UCF101 to compare with other methods.

3.4. Evaluation of TAN

In this section, we will test TAN on two datasets and report the accuracies. We will also explore several experiments to see why and how TAN works.

Benefits from TAN. The performances of two-stream framework with average late fusion and TAN are shown in Figure 3 and Table 3. TAN can not improve single branch markedly on UCF101, but improves significantly on HMDB51. This may be because that (1) frames in one video of UCF101 do not vary too much and can be well classified by single frame; (2) video lengths of UCF101 are shorter than HMDB51 and selecting important frames for a longer video is much more useful. When considering the final fused model, TAN outperforms two-stream model 1.8% on UCF101 and 11.8% on

Table 4. Comparison with existing attention methods on HMDB51 and UCF101.

Model	HMDB51	UCF101
Soft attention [17]	41.3%	-
Multi-branch attention [20]	61.7%	90.6%
Spatial attention (SA)	-	81.95%
SA + pre-train CNN	-	88.47%
TAN	**68.3%**	**92.1%**

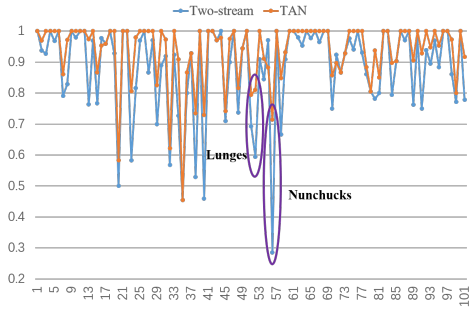

Fig. 4. The class-level improvement from two-stream to TAN on UCF101. TAN improves a lot to the classes which have many postures shared with other action classes.

HMDB51. The improvement is because that TAN is able to select the best frames from spatial and temporal sequences. When fusing two streams, we can always benefit from using the best predictions. The significantly improvement proves that two-stream framework can benefit a lot from TAN.

Comparison with existing video attention. As shown in Table 4, our TAN achieves much better accuracy than existing attention methods. TAN outperforms soft attention [17] 27% on HMDB51 dataset and outperforms multi-branch attention [20] 6.6% on HMDB51 and 1.5% on UCF101 respectively. Although TAN is much simpler than multi-branch attention [20], our performance outperforms it a lot, especially on HMDB51 dataset. The inefficiency of soft attention [17] also confirms the experiment result of our spatial attention.

Evaluation on benchmark datasets. The accuracies of TAN on two benchmark datasets are shown in Table 2. For the HMDB51 dataset, TAN performs good on the first split while relatively worse on the other two splits. And the overall performance is not as good as in UCF101. We think this should be related to the small training set size (3570 videos). But TAN is still outstanding on HMDB51 when comparing to other methods. The TAN performs good on all three splits of UCF101. In most cases, the temporal branches achieves better performance than the spatial branches. This proves the importance of motion information and inspires us to focus on temporal domain.

In order to explore the effect of TAN on class level, we show the per-class accuracies of the 101 classes in Figure 4.

Table 5. Comparison of TAN to the state-of-the-art methods on HMDB51 and UCF101.

Module	HMDB51	UCF101
MIFS [25]	65.1%	89.1%
Two-stream ConvNets [14]	59.4%	88.0%
TDD+FV [12]	63.2%	90.3%
Multi-branch attention [20]	61.7%	90.6%
sDTD [31]	65.2%	92.2%
Conv Fusion [32]	65.4%	92.5%
TSN (Inception-BN) [23]	69.4%	**94.2%**
Ours	**72.5%**	**94.1%**

The results agree with our motivation and TAN has great effects on the classes which have many postures shared with other action classes. The TAN is able to improve the accuracy of "Lunges" from 59.4% to 81% and improve from 28.5% to 71% for "Nunchucks".

3.5. Comparison with the State-of-the-art

Table 5 compares our results with several state-of-the-art methods on HMDB51 and UCF101 datasets. Unlike most of existing methods, who use a stack of 10 optical flow fields as one sample, all of our experiments use 1 group of optical flow fields to reduce complexity. Therefore, we get relatively worse performance on the temporal stream. But we can still achieve the state-of-the-art performance after applying TAN.

Compared to the two-stream ConvNets [14], which is the most famous baseline, we get around 13.1% and 6.1% improvements on HMDB51 and UCF101 datasets, respectively. Compared to TSN [23], TAN achieves better performance on HMDB51 while slightly worse on UCF101. However, TSN use Inception-BN as the CNN network while we use GoogLeNet, and Inception-BN is proved to achieve 3% [23] better result than GoogLeNet on UCF101.

4. CONCLUSION

In this paper, we propose the Temporal Attentive Network (TAN) on action recognition, which aims to make two streams benefit from each other and learn to focus on the most discriminative frames of a video sequence in the temporal domain. We also explore several fusion and attention models. As demonstrated by the experimental results on two benchmark datasets, our TAN model can improve the two-stream framework remarkably and achieve state-of-the-art performance. Compared with other methods, TAN is easy to implement while maintaining a similar computational cost.

Acknowledgement. This work is partially supported by grants from the National Key R&D Program of China under grant 2017YFB1002401, the National Natural Science

Foundation of China under contract No. U1611461, No. 61390515, No. 61425025, No. 61471042 and No. 61650202.

5. REFERENCES

[1] Jeroen van Rest, FA Grootjen, Marc Grootjen, Remco Wijn, Olav Aarts, ML Roelofs, Gertjan J Burghouts, Henri Bouma, Lejla Alic, and Wessel Kraaij, "Requirements for multimedia metadata schemes in surveillance applications for security," *Multimedia tools and applications*, 2014.

[2] Sarvesh Vishwakarma and Anupam Agrawal, "A survey on activity recognition and behavior understanding in video surveillance," *The Visual Computer*, 2013.

[3] Subhashini Venugopalan, Marcus Rohrbach, Jeffrey Donahue, Raymond Mooney, Trevor Darrell, and Kate Saenko, "Sequence to sequence-video to text," in *ICCV*, 2015.

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

[5] Jiasen Lu, Jason J Corso, et al., "Human action segmentation with hierarchical supervoxel consistency," in *CVPR*, 2015.

[6] Stéphanie Lefèvre, Dizan Vasquez, and Christian Laugier, "A survey on motion prediction and risk assessment for intelligent vehicles," *Robomech Journal*, 2014.

[7] Navneet Dalal and Bill Triggs, "Histograms of oriented gradients for human detection," in *CVPR*, 2005.

[8] Navneet Dalal, Bill Triggs, and Cordelia Schmid, "Human detection using oriented histograms of flow and appearance," in *ECCV*. Springer, 2006.

[9] Heng Wang, Alexander Kläser, Cordelia Schmid, and Cheng-Lin Liu, "Action recognition by dense trajectories," in *CVPR*, 2011.

[10] Zuxuan Wu, Xi Wang, Yu-Gang Jiang, Hao Ye, and Xiangyang Xue, "Modeling spatial-temporal clues in a hybrid deep learning framework for video classification," in *ACM MM*, 2015.

[11] Yemin Shi, Wei Zeng, Tiejun Huang, and Yaowei Wang, "Learning deep trajectory descriptor for action recognition in videos using deep neural networks," in *ICME*, 2015.

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

[13] Christian Szegedy, Wei Liu, Yangqing Jia, Pierre Sermanet, Scott Reed, Dragomir Anguelov, Dumitru Erhan, Vincent Vanhoucke, and Andrew Rabinovich, "Going deeper with convolutions," in *CVPR*, 2015.

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

[15] Dzmitry Bahdanau, Kyunghyun Cho, and Yoshua Bengio, "Neural machine translation by jointly learning to align and translate," *arXiv:1409.0473*, 2014.

[16] Oriol Vinyals, Lukasz Kaiser, Terry Koo, Slav Petrov, Ilya Sutskever, and Geoffrey Hinton, "Grammar as a foreign language," in *NIPS*, 2015.

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

[18] Sepp Hochreiter and Jürgen Schmidhuber, "Long short-term memory," *Neural computation*, 1997.

[19] Yemin Shi, Yonghong Tian, Yaowei Wang, Wei Zeng, and Tiejun Huang, "Learning long-term dependencies for action recognition with a biologically-inspired deep network," in *ICCV*, 2017.

[20] Jialin Wu, Gu Wang, Wukui Yang, and Xiangyang Ji, "Action recognition with joint attention on multi-level deep features," *arXiv:1607.02556*, 2016.

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

[22] 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.

[23] 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*. Springer, 2016.

[24] Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei, "Imagenet: A large-scale hierarchical image database," in *CVPR*, 2009.

[25] Zhengzhong Lan, Ming Lin, Xuanchong Li, Alex G Hauptmann, and Bhiksha Raj, "Beyond gaussian pyramid: Multi-skip feature stacking for action recognition," in *CVPR*, 2015.

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

[27] Khurram Soomro, Amir Roshan Zamir, and Mubarak Shah, "Ucf101: A dataset of 101 human actions classes from videos in the wild," *arXiv:1212.0402*, 2012.

[28] Martın Abadi, Ashish Agarwal, Paul Barham, Eugene Brevdo, Zhifeng Chen, Craig Citro, Greg S Corrado, Andy Davis, Jeffrey Dean, Matthieu Devin, et al., "Tensorflow: Large-scale machine learning on heterogeneous distributed systems," *arXiv:1603.04467*, 2016.

[29] Kyunghyun Cho, Bart Van Merriënboer, Caglar Gulcehre, Dzmitry Bahdanau, Fethi Bougares, Holger Schwenk, and Yoshua Bengio, "Learning phrase representations using rnn encoder-decoder for statistical machine translation," *arXiv:1406.1078*, 2014.

[30] Christopher Zach, Thomas Pock, and Horst Bischof, "A duality based approach for realtime tv-l 1 optical flow," in *PR*. Springer, 2007.

[31] Yemin Shi, Yonghong Tian, Yaowei Wang, and Tiejun Huang, "Sequential deep trajectory descriptor for action recognition with three-stream cnn," *TMM*, 2017.

[32] Christoph Feichtenhofer, Axel Pinz, and Andrew Zisserman, "Convolutional two-stream network fusion for video action recognition," in *CVPR*, 2016.