

Joint Network based Attention for Action Recognition

Yemin Shi^{1,2}, Yonghong Tian^{1,2*}, Yaowei Wang^{3*}, Tiejun Huang^{1,2}

¹ National Engineering Laboratory for Video Technology, School of EE&CS,
Peking University, Beijing, China

² Cooperative Medianet Innovation Center, China

³ School of Information and Electronics,
Beijing Institute of Technology, Beijing, China

Abstract

By extracting spatial and temporal characteristics in one network, the two-stream ConvNets can achieve the state-of-the-art performance in action recognition. However, such a framework typically suffers from the separately processing of spatial and temporal information between the two standalone streams and is hard to capture long-term temporal dependence of an action. More importantly, it is incapable of finding the salient portions of an action, say, the frames that are the most discriminative to identify the action. To address these problems, a joint network based attention (JNA) is proposed in this study. We find that the fully-connected fusion, branch selection and spatial attention mechanism are totally infeasible for action recognition. Thus in our joint network, the spatial and temporal branches share some information during the training stage. We also introduce an attention mechanism on the temporal domain to capture the long-term dependence meanwhile finding the salient portions. Extensive experiments are conducted on two benchmark datasets, UCF101 and HMDB51. Experimental results show that our method can improve the action recognition performance significantly and achieves the state-of-the-art results on both datasets.

1. Introduction

Action recognition is the key technique for many visual applications, such as security surveillance, automated driving, home-care nursing, and automatically video tag. Generally speaking, action recognition aims at categorizing the actions or behaviours of one or more persons in a video sequence. Typically, an action can be identified by its spatial (for example, football and piano) or temporal features. In [20], Simonyan *et al.* proposed the two-stream Con-

vNets for action recognition. Basically, their method extracts the spatial and temporal characteristics in one framework, and trains the standalone CNNs for two streams separately. However, it is well-known that the spatial and temporal domain are not independent from each other. Naturally, it is beneficial to train the spatial and temporal streams jointly.

In order to make better use of the two-stream framework, several studies made their efforts on fusing the two streams. Simonyan *et al.* [20] tested three ways: training a joint stack of fully-connected layers on top of two streams' features, fusing the softmax scores by averaging and fusing the softmax scores using a linear SVM. They claimed that fusion with fully-connected layers was infeasible due to over-fitting, while SVM-based fusion of softmax scores outperformed the averaging fusion. Wu *et al.* [34] tested more strategies, including SVM-based early fusion, SVM-based late fusion, multiple kernel learning [12], early fusion with neural networks, late fusion with neural networks, multimodal deep Boltzmann Machines [14, 22] and RDNN [33], and then proposed their Regularized Feature Fusion Network.

However, these two-stream networks often suffer from the so-called "one-stream-dominating-network" problem. Usually, if we use a CNN before fusing the two streams, we should train the CNNs for both spatial and temporal stream at first. Because the temporal stream will converge much slower than the spatial one, training the two streams together will always result in the spatial stream dominating the whole network. It is known that in case of one stream dominating the predictions, little information exchange happens between the two streams.

In action recognition, another important problem is the modeling and utilization of the long-term dependence. It has been proven by many works that better modeling the long-term dependence will improve the performance significantly. Karpathy *et al.* [11] found that a slow fusion in the temporal domain would produce a better result than single

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

frame, late fusion or early fusion. Yue-Hei *et al.* [35] and Donahue *et al.* [8] proposed to use recurrent networks by connecting LSTMs to CNNs, and found that RNNs were a better solution than the temporal domain fusion strategy. Shi *et al.* [19, 18] also introduced their DTD and sDTD to model the dependence on the temporal domain. However, none of them is effective enough for modeling the long-term dependence.

Recently, the attention mechanism [2, 26] was also introduced to action recognition. Sharma *et al.* [17] transferred the attention mechanism on the spatial domain to action recognition. Wu *et al.* [32] used attention as a regularization to make use of features from different layers in CNN. Unfortunately, no remarkable performance gain was achieved in both works. Obviously, by simply introducing the attention mechanism, they are totally incapable of finding the salient portions of an action, say, the frames that are most discriminative to identify the action.

In this paper, we propose a **joint network based attention (JNA) model** which aims at learning the salient portions of actions. Through several exploratory experiments, we find that the fully-connected fusion, branch selection and spatial attention mechanism are totally infeasible for action recognition. So in our joint network, the spatial and temporal branches share some information during the training stage. We also introduce an attention mechanism on the temporal domain to capture the long-term dependence meanwhile finding the salient portions. As a result, our method takes both spatial and temporal stream as input and pulls the most important parts as output. To limit the information exchange between the two streams, their connection is constrained by softmax. Only the crucial information can go through the gate propagating to the lower layers of the other stream during back propagation. Extensive experiments are carried out on two challenging datasets: HMDB51 and UCF101. The results show that the proposed method significantly improves our baseline and achieves the state-of-the-art performance.

The rest of the paper is organized as follows: In section 2, we review the related work on action recognition and attention mechanism. We explore the base model before head into the proposed model in section 3. The proposed joint network based attention model is presented in section 4. We will evaluate our JNA in section 5. Finally, section 5.5 concludes this paper.

2. Related works

Action recognition. Basically, the action recognition approaches can be categorized into two types by the way of feature extraction: hand-crafted low-level features [5, 6] and deep features [34, 19, 33, 37]. The most successful hand-crafted feature is the dense trajectories [27], which sample and track dense points from each frame in multi-

ple scales. HOG, HOF and Motion Boundary Histogram (MBH) are also extracted at each point. The combination of these features was shown to further boost the final performance. The improved version of dense trajectories [28] also takes the camera motion estimation into account and then applies the Fisher vector [16] to derive the final representation for each video.

In recent years, CNN has achieved state-of-the-art performance on various tasks (e.g. [9, 24, 25, 10]) and it has been proven that features learnt from CNN are much better than the hand-crafted features. In order to transfer CNNs to video tasks, many models [19, 37] have been proposed. **Two-stream ConvNets** [20] is the most important framework which acts the baseline (with two GRU layers) in many works. Basically, the two-stream ConvNets incorporate spatial and motion networks and pre-train these networks on the large ImageNet [7] dataset, consequently achieving the state-of-the-art performance.

Unlike these pure deep models, Wang *et al.* [29] proposed trajectory-pooled deep-convolutional descriptor (TDD), which shares the merits of both hand-crafted features and deeply-learned features. Shi *et al.* [19] proposed the deep trajectory descriptor (DTD) by converting dense trajectories into 2D images and utilizing a CNN to learn features for these images.

Attention mechanism. The attention mechanism is first introduced to neural machine translation (NMT) by Bahdanau *et al.* [2] to automatically learn the alignment between a target word and the relevant parts of the source sentence. Some works did their efforts to apply the attention mechanism to action recognition. Sharma *et al.* [17] proposed the spatial attention without modification of the mechanism. Wu *et al.* [32] proposed a more complicated model and used attention as a regularization. Unfortunately, no remarkable performance gain was achieved by the attention mechanism in both works.

Our joint network based attention (JNA) method will focus on extracting the most important parts in the temporal domain of a video. As a consequence, the improvement can be achieved by fusing the most important frames in two streams.

3. Exploratory findings

In this section, we will describe the findings from two exploratory experiments including fully-connected fusion and branch selection for fusing two streams. These methods inspired us to propose our JNA method.

3.1. Fully-connected fusion

As shown in Figure 1(a), a simplest way to fuse two streams is to add another fully-connected (FC) layer on top of them. We first train the two-stream CNNs. Then the FC

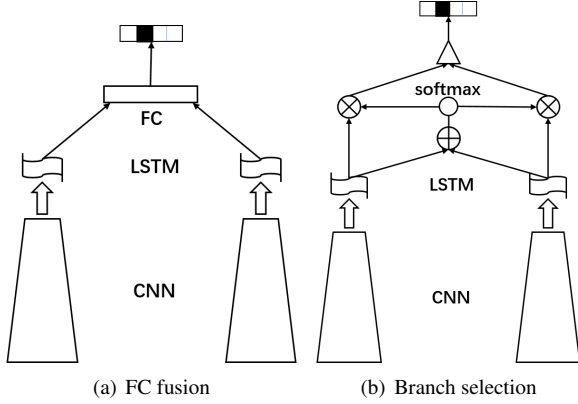


Figure 1. The illustration of two exploration fusion methods. The FC fusion uses a fully-connected layer to combine the two streams. The branch selection approach uses an attention model to decide which stream should be selected to produce prediction for the current input. So it works as an automatic weighted averager.

Table 1. Comparison of different fusion methods. We use GoogLeNet and two GRU layers, and train the network on UCF101 dataset.

Method	Spatial	Temporal	Fusion
FC fusion	80.3%	80.7%	81.4%
Branch selection (BS)	-	-	80.0%
BS with L2 norm	-	-	83.9%
Average	80.5%	82.8%	90.2%

layer takes the two streams as input and the output is then used to learn a classifier.

The results of FC fusion are listed in Table 1. As reported by [20] and [33], we find that FC fusion produces much worse results than averaging fusion of softmax scores. The fusion model almost has the same performance as the single spatial or temporal stream. In our opinion, the low accuracy is not only due to the over-fitting problem but also because that one stream dominates the network while the other stream only has a small effect on the final prediction. This assumption is also confirmed by the following branch selection approach.

3.2. Branch selection

To begin the discussion, we will first revisit the attention mechanism. In sequence-to-sequence tasks, we have two separate LSTMs (one to encode the sequence of input words A_i and another to produce or decode the output symbols B_i). Let $(h_1, h_2, \dots, h_{T_A})$ denote the hidden states of the encoder while $(d_1, d_2, \dots, d_{T_B})$ for those of the decoder. To compute the attention vector at each output time t over the

input words, we define:

$$\begin{aligned}
 e_i^t &= v^T \tanh(W'_1 h_i + W'_2 d_{t-1}) \\
 \alpha_i^t &= \frac{\exp(e_i^t)}{\sum_{k=1}^{T_A} \exp(e_k^t)} \\
 d'_t &= \sum_{i=1}^{T_A} \alpha_i^t h_i
 \end{aligned}$$

where vector v and matrices W'_1, W'_2 are the parameters of the model. The vector u^t assigns a weight for each encoder hidden state h_i , indicating how much attention should be put on h_i . These attention weights are normalized by softmax to create the attention mask a^t over the encoder hidden states.

In the two-stream framework, we have to fuse two predictions to get the final result. However, if a video can be predicted correctly by the final prediction, this video is also likely to be correctly predicted by one of the streams. Is it possible to select such a stream for a video?

As illustrated in Figure 1(b), we modify the attention mechanism to take two streams as input and compute attention weights for each stream, as follows:

$$\begin{aligned}
 s &= W'_1 x_1 + W'_2 x_2 \\
 e_i &= v^T \tanh(s + W'_3 x_i) \\
 \alpha_i &= \frac{\exp(e_i)}{\sum_{k=1}^2 \exp(e_k)} \\
 o_i &= \sum_{k=1}^2 \alpha_i x_i
 \end{aligned}$$

where x_1, x_2 are the spatial and temporal features respectively. In this model, the output of the branch we are looking at is used twice so that the attention model can generate different weights for two branches. Because that two streams are in different feature space, we apply L2 norm to the inputs:

$$x_1 = \|x_1\|_2 \quad x_2 = \|x_2\|_2$$

As shown in Table 1, the pure branch selection produces a very similar performance to one stream. Even after applying L2 norm, the performance gain is still small.

4. Joint network based attention

According to previous proposed models, we find that when trying to fuse two streams, we should avoid one stream dominating the output. In this section, we find that applying attention to spatial domain is not effective. However, it works better by joint training two streams with temporal attention model.

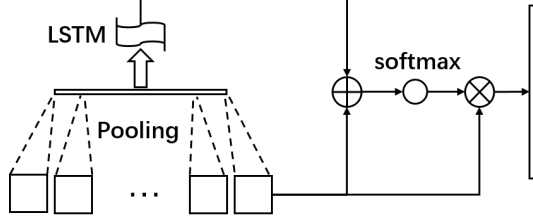


Figure 2. Illustrating our spatial attention method.

4.1. Spatial attention

In order to make use of the attention mechanism, we also propose the spatial attention model which is designed to replace the last pooling layer.

As shown in Figure 2, we assume the last pooling layer is P, the layer before P is layer C and the feature map size of C is $K \times K$. We add a LSTM layer L after P. C has K^2 positions and $C_{i,j}$ is a vector constructed by the value at (i, j) in all feature maps of C. We compute the attention weight for each position by:

$$e_{i,j} = v^T \tanh(W'_1 L + W'_2 C_{i,j})$$

$$\alpha_{i,j} = \frac{\exp(e_{i,j})}{\sum_{k=1}^K \sum_{l=1}^K \exp(e_{k,l})}$$

Then the output of the spatial attention is the weighted average of C by α .

Unlike most existing spatial attention approaches, our model employs another LSTM layer to help remember the history information. However, according to Table 2, both soft attention or our complicated spatial attention are not able to improve our baselines. These failures prove the infeasibility of spatial attention methods for action recognition.

4.2. JNA

To overcome the “one-stream-dominating-the-network” problem and design a good joint network, we should limit the information exchanging between two streams. This means we must avoid fusion of final features or predictions during training. To satisfy this rule, each stream should have their own softmax and loss layers. On the other hand, two branches should share some layers so that the most crucial information is able to go across two streams during the back propagation. We restrict the information flow by using softmax layer as gate so that information can go through it but only important information can back propagate through this gate.

Our proposed attention model is shown in Figure 3. In JNA, we have two branches, spatial and temporal branches, which have at least one LSTM as the last layers. The outputs of the last LSTM layer in spatial branch are denoted

(h_1, h_2, \dots, h_T) and the outputs of the last LSTM layer of temporal branch are denoted (g_1, g_2, \dots, g_T) . To compute the attention weight for each frame (video frame and optical flow fields), we define:

$$e_{ij} = v^T \tanh(W'_1 h_i + W'_2 g_j)$$

$$f_{ji} = u^T \tanh(W'_3 g_j + W'_4 h_i)$$

$$\alpha_{ij} = \frac{\exp(e_{ij})}{\sum_{k=1}^T \exp(e_{kj})}$$

$$\beta_{ji} = \frac{\exp(f_{ji})}{\sum_{k=1}^T \exp(f_{jk})}$$

$$o_j^h = \sum_{i=1}^T \alpha_{ij} h_i$$

$$o_i^g = \sum_{j=1}^T \beta_{ji} g_j$$

where vector v, u and matrices W'_1, W'_2, W'_3, W'_4 are the parameters. Every 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. In this way, the number of output feature vectors in B is equal to the number of input feature vectors in A. The o_j^h and o_i^g are the output of spatial branch and temporal branch respectively and followed by fully-connected layers to learn classifiers.

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 gradients of these important inputs can back propagate to e and f , and finally impact both branches. Because that there is no other layer is shared by two branches, α and β are the gates to control the information flow which can share across two branches and is called sharing gates.

5. Experiments

This section will first introduce the detail of datasets and their corresponding evaluation schemes. Then, we describe the implementation details of our model. Finally, we report the experimental results and compare JNA with the state-of-the-art methods.

5.1. Datasets

To verify the effectiveness of our methods, we conduct experiments on two public datasets: HMDB51 [13] and UCF101 [21].

The HMDB51 dataset is a large collection of realistic videos from various sources, including movies and web videos. It is composed of 6,766 video clips from 51 action

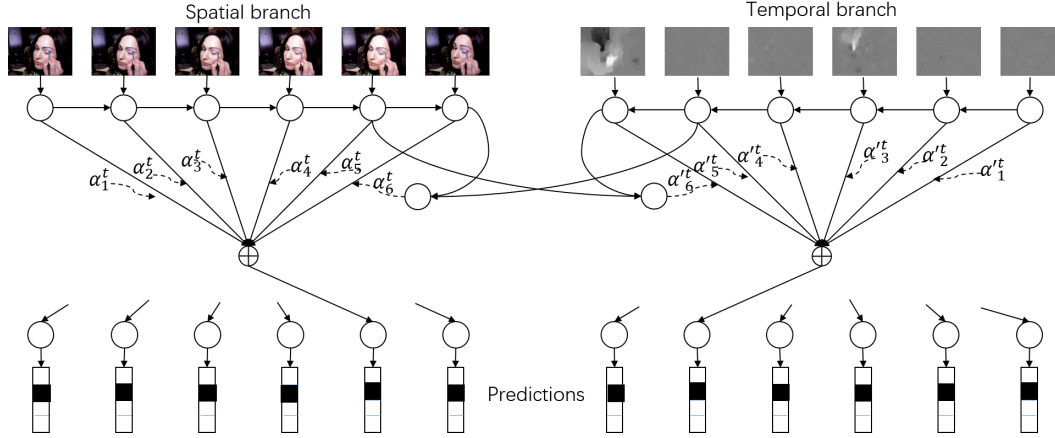


Figure 3. Illustration of our joint network based attention (JNA).

categories, with each category containing at least 100 clips. Our experiments follow the original evaluation scheme, but only adopt the first training/testing split. In this split, each action class has 70 clips for training and 30 clips for testing.

The UCF101 dataset contains 13,320 video clips from 101 action classes and there are at least 100 video clips for each class. We tested our model on the first training/testing split in the experiments.

Compared with the very large dataset used for image classification, the dataset for action recognition is relatively smaller. Therefore, we pre-train our model on the ImageNet dataset [7]. As UCF101 is larger than HMDB51, we also use UCF101 to train our joint model initially, and then transfer the learnt model to HMDB51.

5.2. Implementation details

We use the TensorFlow [1] to implement our model and the CNN in every branch is implemented with GoogLeNet [25] structure. We use GRU [4] as our LSTM implementation.

The network weights are learnt using the mini-batch stochastic gradient descent with momentum (set to 0.9). The batch size for training CNN is 128 and the batch size for training joint network is 64. When training or testing the joint model, we read 16 frames/flows with a stride of 5 from each video as one sample for the GRU. We resize all input images to 340×256 , and then use the fixed-crop strategy [30] to crop a 224×224 region from images or their horizontal flip. Because the 16 consecutive samples are needed in the GRU, we also force images from the same video to crop the same region. In the test phase, we sample 4 corners and the center from each image and its horizontal flip, and 25 samples are extracted from each video.

In order to fully train the CNN feature of spatial and temporal branches, we first train two CNN separately. We initialize the CNN with the pre-trained ImageNet model and

Table 2. Comparison with existing attention methods on HMDB51 and UCF101.

Model	HMDB51	UCF101
Soft attention [17]	41.3%	-
Multi-branch attention [32]	61.7%	90.6%
Spatial attention (SA)	-	81.95%
SA + pre-train CNN	-	88.47%
JNA	66.9%	91.2%

train CNN classifier like two-stream ConvNets [20]. The trained CNN weights are used to initialize the CNN part of our joint network. Then we train two branches with our JNA method jointly.

For CNN, the learning rate starts from 0.01 and is divided by 10 at iteration $20K$, $30K$ and $35K$, and training is stopped at $40K$ iterations. For joint network, the learning rate is initially set as 10^{-3} and divide by 10 at iteration $25K$, $45K$ and $60K$, and training is stopped at $65K$. For the temporal stream, we choose the TVL1 optical flow algorithm [36] and the warped TVL1 optical flow field [31].

In the remainder of the paper, we use spatial stream and temporal stream to indicate the streams in two-stream framework, and each stream is a GoogLeNet followed by two GRU layers. We use spatial branch and temporal branch to indicate the branches in JNA network, and the structure of the two branches is the same as two streams. We use warped spatial branch and warped temporal branch to indicate the two branches in JNA network whose temporal branch input is warped TVL1 optical flow fields, and the JNA is called warped JNA.

5.3. Evaluation of JNA

Comparison with exploratory experiments. Our JNA is an extension of fully-connected fusion, branch selection and spatial attention methods. We use joint network structure like FC fusion and branch selection while avoid their one-

Table 3. Performance of different modules on HMDB51 and UCF101.

Module	HMDB51	UCF101
Spatial stream	46.2%	80.5%
Temporal stream	50.3%	82.8%
Spatial branch	50.2%	81.6%
Temporal branch	56.9%	82.5%
Warped spatial branch	50.3%	81.2%
Warped temporal branch	56.9%	79.1%
Two streams	58.4%	90.2%
JNA	66.9%	91.2%
Warped JNA	66.3%	90.0%

stream-dominating-network problem. JNA learns attention weights on temporal domain while spatial attention learns attention weights on spatial domain. Even though these methods are using similar solution, JNA is the only one which can improve baseline performance and outperform average softmax score fusion. According to Table 1 and 2, JNA is 1% better than average fusion and other trails are worse than average fusion.

Benefits from JNA. The performances of different modules are shown in Table 3. JNA 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, JNA outperforms two-stream model 1% on UCF101 and 8.5% on HMDB51. The significantly improvement proves that two-stream framework can benefit a lot from JNA.

Comparison with exist video attention. As shown in Table 2, our JNA also achieves much better accuracy than exist attention methods. Although JNA is much simpler than multi-branch attention [32], 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.

Variation of the model’s attention. The visualization curve of the attention weights (before softmax) in JNA are shown in Figure 4. Because that our models employ two GRUs, latter frames are able to predict based on history information hence more important than previous frames, and JNA gives bigger weights on the latter frames. This also agrees with our intuition that LSTM like GRU is able to memorize information. However, the variation of attention weights also proves that LSTM can not store all information and methods like JNA can help LSTM improve the performance. The circled points have negative attention weights, which means JNA is able to detect the beginning of action

Table 4. Comparison of JNA to the state-of-the-art methods on HMDB51 and UCF101.

Module	HMDB51	UCF101
DT+MVSV [3]	55.9%	83.5%
iDT+HSV [15]	61.1%	88.0%
Two-stream model [30]	59.4%	88.0%
F_{ST} -CN [23]	59.1%	88.1%
TDD+iDT+FV [29]	65.9%	91.5%
Multi-branch attention [32]	61.7%	90.6%
JNA	66.9%	91.2%
Warped JNA	66.3%	90.0%
JNA + Warped JNA	68.8%	91.5%

and automatically filter out the irrelevant frames.

5.4. Comparison with the state-of-the-art methods

Table 4 compares our results with several state-of-the-art methods on HMDB51 and UCF101 datasets. The performance of JNA significantly surpasses these methods on HMDB51 and outperforms most methods on the UCF101 dataset. The superior performance of our method demonstrates the effectiveness of joint network based attention and justifies the importance of long-term temporal dependence.

5.5. Conclusion

In this paper, we propose a joint network based attention (JNA) for action recognition, which aims to make two streams benefit from each other and learn to focus on the most discriminative frames of a video. As demonstrated by the experimental results on two challenging datasets, our JNA model can improve the two-stream framework remarkably and achieve state-of-the-art performance. Compared with other methods, JNA is easy to implement while maintaining a similar computational cost.

References

- [1] M. Abadi, A. Agarwal, P. Barham, E. Brevdo, Z. Chen, C. Citro, G. S. Corrado, A. Davis, J. Dean, M. Devin, et al. Tensorflow: Large-scale machine learning on heterogeneous distributed systems. *arXiv preprint arXiv:1603.04467*, 2016. 5
- [2] D. Bahdanau, K. Cho, and Y. Bengio. Neural machine translation by jointly learning to align and translate. *arXiv preprint arXiv:1409.0473*, 2014. 2
- [3] Z. Cai, L. Wang, X. Peng, and Y. Qiao. Multi-view super vector for action recognition. In *Proceedings of IEEE conference on Computer Vision and Pattern Recognition*, pages 596–603, 2014. 6
- [4] K. Cho, B. Van Merriënboer, C. Gulcehre, D. Bahdanau, F. Bougares, H. Schwenk, and Y. Bengio. Learning phrase representations using rnn encoder-decoder for statistical machine translation. *arXiv preprint arXiv:1406.1078*, 2014. 5

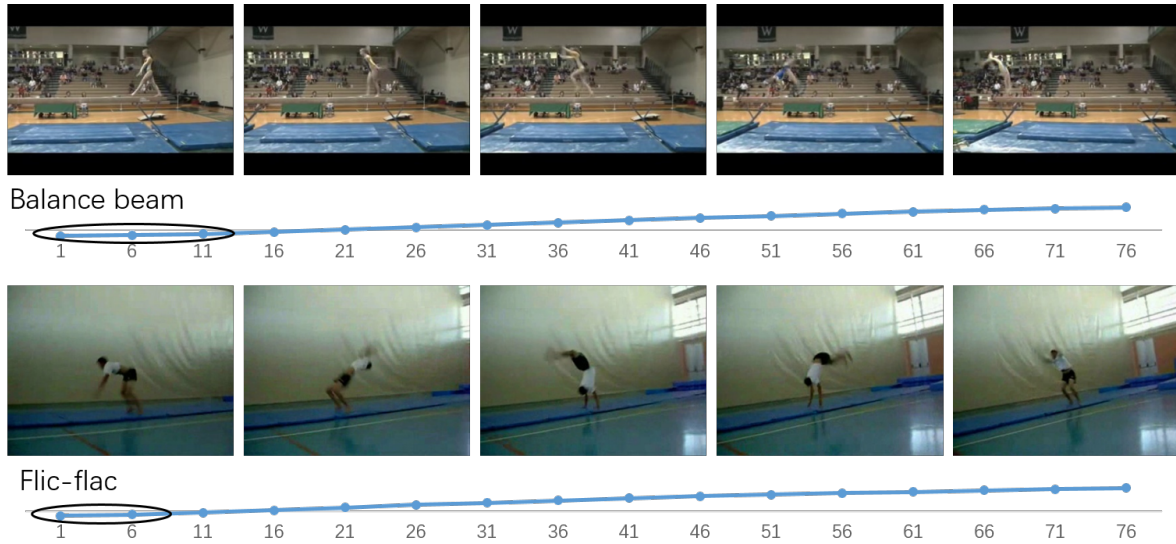


Figure 4. Visualization of the attention weights (before softmax) in JNA on two videos from UCF101 and HMDB51. The x-axis and y-axis are frame number and attention weight respectively. A higher attention weights means the frame is more salient. The circled parts are the frames which have negative attention weights.

- [5] N. Dalal and B. Triggs. Histograms of oriented gradients for human detection. In *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition*, volume 1, pages 886–893, 2005. 2
- [6] N. Dalal, B. Triggs, and C. Schmid. Human detection using oriented histograms of flow and appearance. In *Proceedings of European Conference on Computer Vision*, pages 428–441. Springer, 2006. 2
- [7] J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, and L. Fei-Fei. Imagenet: A large-scale hierarchical image database. In *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition*, pages 248–255, 2009. 2, 5
- [8] J. Donahue, L. Anne Hendricks, S. Guadarrama, M. Rohrbach, S. Venugopalan, K. Saenko, and T. Darrell. Long-term recurrent convolutional networks for visual recognition and description. In *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition*, pages 2625–2634, 2015. 1
- [9] K. He, X. Zhang, S. Ren, and J. Sun. Deep residual learning for image recognition. *arXiv preprint arXiv:1512.03385*, 2015. 2
- [10] K. He, X. Zhang, S. Ren, and J. Sun. Delving deep into rectifiers: Surpassing human-level performance on imagenet classification. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 1026–1034, 2015. 2
- [11] A. Karpathy, G. Toderici, S. Shetty, T. Leung, R. Sukthankar, and L. Fei-Fei. Large-scale video classification with convolutional neural networks. In *Proceedings of IEEE conference on Computer Vision and Pattern Recognition*, pages 1725–1732, 2014. 1
- [12] M. Kloft, U. Brefeld, S. Sonnenburg, and A. Zien. Lp-norm multiple kernel learning. *Journal of Machine Learning Research*, 12(Mar):953–997, 2011. 1
- [13] H. Kuehne, H. Jhuang, E. Garrote, T. Poggio, and T. Serre. Hmdb: a large video database for human motion recognition. In *Proceedings of IEEE International Conference on Computer Vision*, pages 2556–2563, 2011. 4
- [14] J. Ngiam, A. Khosla, M. Kim, J. Nam, H. Lee, and A. Y. Ng. Multimodal deep learning. In *Proceedings of the 28th international conference on machine learning (ICML-11)*, pages 689–696, 2011. 1
- [15] X. Peng, L. Wang, X. Wang, and Y. Qiao. Bag of visual words and fusion methods for action recognition: Comprehensive study and good practice. *Computer Vision and Image Understanding*, 2016. 6
- [16] F. Perronnin, J. Sánchez, and T. Mensink. Improving the fisher kernel for large-scale image classification. In *Proceedings of European Conference on Computer Vision*, pages 143–156. Springer, 2010. 2
- [17] S. Sharma, R. Kiros, and R. Salakhutdinov. Action recognition using visual attention. *arXiv preprint arXiv:1511.04119*, 2015. 2, 5, 6
- [18] Y. Shi, T. Yonghong, Y. Wang, and T. Huang. Sequential deep trajectory descriptor for action recognition with three-stream cnn. *arXiv preprint arXiv:1609.03056*, 2016. 2
- [19] Y. Shi, W. Zeng, T. Huang, and Y. Wang. Learning deep trajectory descriptor for action recognition in videos using deep neural networks. In *Proceedings of IEEE International Conference on Multimedia and Expo*, pages 1–6, 2015. 2
- [20] K. Simonyan and A. Zisserman. Two-stream convolutional networks for action recognition in videos. In *Advances in Neural Information Processing Systems*, pages 568–576, 2014. 1, 2, 3, 5
- [21] K. Soomro, A. R. Zamir, and M. Shah. Ucf101: A dataset of 101 human actions classes from videos in the wild. *arXiv preprint arXiv:1212.0402*, 2012. 4

- [22] N. Srivastava and R. R. Salakhutdinov. Multimodal learning with deep boltzmann machines. In *Advances in neural information processing systems*, pages 2222–2230, 2012. 1
- [23] L. Sun, K. Jia, D.-Y. Yeung, and B. E. Shi. Human action recognition using factorized spatio-temporal convolutional networks. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 4597–4605, 2015. 6
- [24] I. Sutskever, O. Vinyals, and Q. V. Le. Sequence to sequence learning with neural networks. In *Advances in neural information processing systems*, pages 3104–3112, 2014. 2
- [25] C. Szegedy, W. Liu, Y. Jia, P. Sermanet, S. Reed, D. Anguelov, D. Erhan, V. Vanhoucke, and A. Rabinovich. Going deeper with convolutions. In *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition*, pages 1–9, 2015. 2, 5
- [26] O. Vinyals, Ł. Kaiser, T. Koo, S. Petrov, I. Sutskever, and G. Hinton. Grammar as a foreign language. In *Advances in Neural Information Processing Systems*, pages 2773–2781, 2015. 2
- [27] H. Wang, A. Kläser, C. Schmid, and C.-L. Liu. Action recognition by dense trajectories. In *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition*, pages 3169–3176, 2011. 2
- [28] H. Wang and C. Schmid. Action recognition with improved trajectories. In *Proceedings of IEEE International Conference on Computer Vision*, pages 3551–3558, 2013. 2
- [29] L. Wang, Y. Qiao, and X. Tang. Action recognition with trajectory-pooled deep-convolutional descriptors. In *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition*, pages 4305–4314, 2015. 2, 6
- [30] L. Wang, Y. Xiong, Z. Wang, and Y. Qiao. Towards good practices for very deep two-stream convnets. *arXiv preprint arXiv:1507.02159*, 2015. 5, 6
- [31] L. Wang, Y. Xiong, Z. Wang, Y. Qiao, D. Lin, X. Tang, and L. Van Gool. Temporal segment networks: Towards good practices for deep action recognition. *arXiv preprint arXiv:1608.00859*, 2016. 5
- [32] J. Wu, G. Wang, W. Yang, and X. Ji. Action recognition with joint attention on multi-level deep features. *arXiv preprint arXiv:1607.02556*, 2016. 2, 5, 6
- [33] Z. Wu, Y.-G. Jiang, J. Wang, J. Pu, and X. Xue. Exploring inter-feature and inter-class relationships with deep neural networks for video classification. In *Proceedings of the 22nd ACM International Conference on Multimedia*, pages 167–176, 2014. 1, 2, 3
- [34] Z. Wu, X. Wang, Y.-G. Jiang, H. Ye, and X. Xue. Modeling spatial-temporal clues in a hybrid deep learning framework for video classification. In *Proceedings of the 23rd ACM international conference on Multimedia*, pages 461–470, 2015. 1, 2
- [35] J. Yue-Hei Ng, M. Hausknecht, S. Vijayanarasimhan, O. Vinyals, R. Monga, and G. Toderici. Beyond short snippets: Deep networks for video classification. In *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition*, pages 4694–4702, 2015. 1
- [36] C. Zach, T. Pock, and H. Bischof. A duality based approach for realtime tv-l1 optical flow. In *Pattern Recognition*, pages 214–223. Springer, 2007. 5
- [37] S. Zha, F. Luisier, W. Andrews, N. Srivastava, and R. Salakhutdinov. Exploiting image-trained cnn architectures for unconstrained video classification. *arXiv preprint arXiv:1503.04144*, 2015. 2