Video Classification via Relational Feature Encoding Networks

Yao Zhou*
SYSU & Sensetime
yoosan.zhou@gmail.com

Jiamin Ren
Sensetime
renjiamin@sensetime.com

Jingyu Li
Sensetime
lijingyu@sensetime.com

Litong Feng
Sensetime
fenglitong@sensetime.com

Shi Qiu
Sensetime
sqiu@sensetime.com

Ping Luo
CUHK
pluo@ie.cuhk.edu.hk

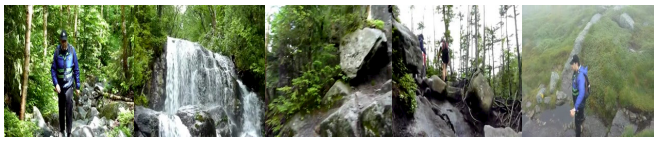
ABSTRACT

In this paper, we propose a novel Relational Feature Encoding Network for video classification. The proposed network uses a set of relational functions wired on top of a backbone convolutional neural network (ConvNet) to generate multiple complementary feature streams on the fly, which are then combined by an aggregation module to form a video-level representation for recognition. The relational functions compute new relational features by applying element-wise operations or a simple projection to pairs of raw ConvNet features, and thus encode the underlying temporal dynamics and relationship of contextual frames which are critical for recognizing video contents. In this work, we explore a number of design choices for both the relational functions and the aggregation functions, and evaluate the resulting deep model on a number of video classification benchmarks, including the extended Fudan-Columbia Video dataset, UCF101, and Kinetics. Experimental results demonstrate that our model is not only well-suited for action recognition, but also exhibits promising performance for general videos.

KEYWORDS

Video classification, Relational feature encoding, Temporal segment networks, Two-stream networks

Ground Truth: Panda
Top5 predictions: Panda, Elephant, Making sorbet, Search a sheep, Gorilla



Ground Truth: Hiking, Mountain
Top5 predictions: Hiking, Mountain, Waterfall, Forest, River

Figure 1: Two example videos from val set of the extended FCVID together with the ground truth and top-5 predictions. The model can recognize objects (first row), action events and scenes (second row) in videos.

1 INTRODUCTION

Convolutional neural networks have been extensively employed in video classification tasks [2, 4, 5, 9, 14, 19, 20]. To capture the temporal information which is critical for recognizing video contents, most state-of-the-art approaches rely on input data of multiple modalities, including RGB difference [18], optical flow [14, 19], motion vector [23]. These new input modalities, when fused with conventional RGB input, provide complementary information for video prediction and lead to improved recognition accuracy. However, employing multiple complementary modalities have some apparent drawbacks. As different modalities have to be trained in separate streams and fused at the final layer, the learned features cannot be naturally shared across modalities and assist each other. Furthermore, generating the new modalities often require considerable amount of computation. For example, pre-computing the optical flow, which proves to be the most effective modality [2, 14, 18], requires around 8 seconds to process one a 10-seconds video clip per GPU. This is unacceptable for a practical video classification system.

In this paper, we advocate the idea of using complementary information for video recognition, and go beyond by applying the idea to the deeper levels of the convolutional feature hierarchy. To this end, we propose a Relational Feature Encoding Network, which takes a single stream of RGB data as input, and generate multiple relational feature streams on the fly using a set of relational functions wired on top of a backbone convolutional neural network. A relational function takes a pair of raw ConvNet features as input, and maps them to a relational feature embedding. The relation functions take the form of element-wise operations or a simple feature projection, in order to capture the undergoing temporal dynamics and relationship between the two input features. When generating relational features at time step t, we feed into relational functions with ConvNet features extracted from two contextual frames before and after t, and thus encode the temporal relationship. These contextually relational features, along with the original ConvNet feature at time t, are then fused by an aggregation

* Work performed when the first author was an intern at Sensetime.

LSVC'17, October 27, 2017, Mountain View, CA, USA

ACM ISBN 978-1-4503-5537-7/17/10…$15.00
https://doi.org/10.1145/3134263.3134265

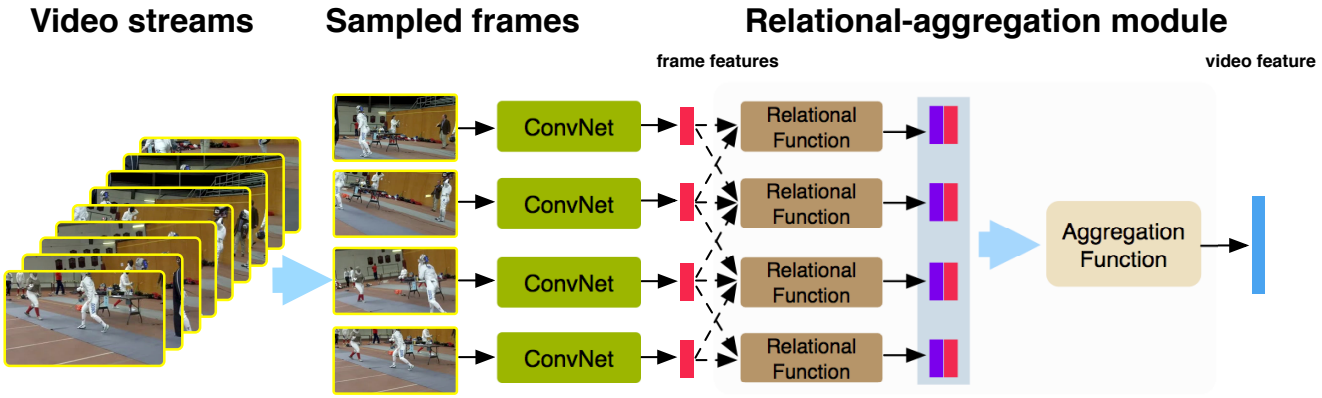


Figure 2: A graphical overview of our proposed model.

module to predict the video labels. By design, our relational feature encoding network is end-to-end trainable. The supervision signals can therefore back-propagate through relational functions into the backbone network and encourage the learning of more powerful spatio-temporal features.

We evaluate the proposed approach on three video classification benchmarks: the extended Fudan-Columbia video dataset [22], UCF101 action video dataset [15] and the Kinetics action video dataset [10]. We empirically investigate various design choices of the relational and aggregation functions. Experimental results show that employing the relational and aggregation functions not only boosts the recognition accuracy by a large margin, but also leads faster convergence during training stage. Using Inception-v3 as the backbone architecture, the proposed Relational Feature Encoding Network achieves an mAP score of 0.772 on the validation set of extended FCVID dataset.

2 THE MODEL

Figure 2 presents a graphical overview of our proposed model. The model receives video streams and outputs a single feature vector indicating the video representation. The pipeline of our video recognition system consists of four phases: 1) sampling frames from video streams, 2) extracting features with ConvNets, 3) encoding relational features and aggregating features over time and 4) decoding the video representation for classification. These phases will be detailed in the following sections.

2.1 Temporal Segment-based Sampling

It has been proven that a single RGB appearance and short-term motions can offer a promise performance for classifying trimmed action videos (around 10s) [14]. However web videos hold an arbitrary duration ranging from a few seconds to hundreds of seconds. Sampling only one frame from long video poses a risk of missing import information. Another choice is sampling as many frames as possible, which could incur excessive computational cost as well as a high redundancy of densely consecutive frames in long videos. A sparse and global temporal sampling strategy named temporal segment-based sampling [19], can not only capture information of the whole video contents but also explicitly reduce computational cost. A detailed description is presented in the following.

Given a video of total N frames, first we divide it into K segments $\{S_1, S_2, \ldots, S_K\}$ on average, where each segment holds N/K frames. A snippet T_t is sampled from its corresponding segment S_t with an uniform sampling. Therefore, the temporal segment-based sampling method obtains a sequence of snippets $(T_1, T_2, \ldots, T_K)$ from raw video streams. In this work, a snippet T_t is a RGB frame sampled from S_t. By this way, we can reduce the number of computational frames from N to K and cover the visual content of whole video.

2.2 The ConvNets

It is natural to migrate the high-performance ConvNets which are well performing on image recognition task to our framework. Different from fine-grained image classification task, we argue that ConvNets used in video recognition task should balance its perception ability and parameter size. Therefore, we compared the performance of several deep ConvNets including Inception-v2 [8], Inception-v3 [3], Inception-resnet-v2 [16], resnet-50 and resnet-101 [7] on ImageNet ILSVRC2012 classification task [13], as well as the size of their parameters. We utilize Inception-v3 with batch normalization [8] as the backbone ConvNets in this work.

Recall that our approach collected K snippets by segment-based sampling at the first phase. A ConvNet encodes the t-th RGB frame to a single feature vector f_t. Thus a sequence of frame features F are extracted from the sampled frames by ConvNets.

$$F = [f_1, f_2, \ldots, f_K] = ConvNets([T_1, T_2, \ldots, T_K]). \quad (1)$$

where $f_t \in R^d$ can be interpreted as the representation of t-th image and $d = 2048$ for Inception-v3 architecture. So far, we have collected the features of sampled frames, we will discuss how to process these frame features in the next subsections.

2.3 Relational Functions

Existing methods directly aggregate the frame features over whole video, ignoring the relationship (such as difference, semantic similarity) of the contextual frames. We argue that underlying temporal relationship between two contextual frames can help capturing the

temporal dynamics and improve the video representation. Therefore, we propose a series of relational functions (**RelaFunc**) to model the relationship of two frames on top of a backbone ConvNet.

For t-th snippet T_t, we consider its two context: the previous l steps frame T_{t-l} and the following l steps frame T_{t+l}. We propose the following relational functions to compute the relational feature h_t based on contextual frame features f_{t-l} and f_{t+l}.

A simple choice of *RelaFunc* is a linear projection layer followed by a non-linear activation function, which transforms the original contextual features to a relational feature embedding.

$$\begin{aligned} \textbf{Projection} : h_t^{proj} &= CompFunc(f_{t-l}, f_{t+l}) \\ &= ReLU(W[f_{t-l}, f_{t+l}] + b) \end{aligned} \tag{2}$$

We also consider that the semantic similarity between two frames can be useful when information is propagated across frames. We use the average of euclidean and cosine distance between f_{i-1} and f_{i+1} multiply it on the element-wise sum of contextual features.

$$\begin{aligned} \textbf{Distance} : h_t^{dist} &= CompFunc(f_{t-l}, f_{t+l}) \\ &= \frac{euc(f_{t-l}, f_{t+l})}{2}(f_{t-l} + f_{t+l}) \\ &+ \frac{\cos(f_{t-l}, f_{t+l})}{2}(f_{t-l} + f_{t+l}) \end{aligned} \tag{3}$$

It is reasonable to take the difference of contextual frames for modeling temporal relationship. Instead of the difference at inputing interface such as RGB difference, we compare the difference at the frame-feature level.

$$\begin{aligned} \textbf{Subtraction} : h_t^{sub} &= CompFunc(f_{t-l}, f_{t+l}) \\ &= f_{t+l} - f_{t-l} \end{aligned} \tag{4}$$

Besides, an element-wise multiplication between two vectors is similar to cosine distance but preserves information about the original vectors.

$$\begin{aligned} \textbf{Multiplication} : h_t^{mul} &= CompFunc(f_{t-l}, f_{t+l}) \\ &= f_{t-l} \odot f_{t+l} \end{aligned} \tag{5}$$

Last we consider a combination of subtraction feature and multiplication feature. It has been proved that this feature is useful for modeling relationship of natural sentences [17, 21].

$$\begin{aligned} \textbf{Submul} : h_t^{submul} &= CompFunc(f_{t-l}, f_{t+l}) \\ &= ReLU(W[h_t^{sub}, h_t^{mul}] + b) \end{aligned} \tag{6}$$

where $W \in R^{d \times 2d}$ and $b \in R^d$ are learned parameters, $0 \leq i \leq K$ and l is set to 1 in this paper.

We introduce the above relational functions which aims at not only capturing the temporal information but also encoding the relationship of consecutive frames. We concatenate the original frame feature f_t and relational feature h_t at t-th segment as the fused feature $p_t = [f_t, h_t] \in R^{2d}$. So far we obtain K fused features $P = [p_1, p_2, \ldots, p_K]$, where each column is a fused feature vector.

2.4 Aggregation Functions

Recent work has demonstrated the effectiveness of learning video representation by aggregating the frame-level features across entirely temporal snippets of the video [6, 12, 18, 19]. We study three aggregation functions: average pooling (**avg pool**) , Long-short Term Memory network (**LSTM**) and a convolution layer (**CNN**).

Average Pooling It can be found that average pooling (avg pool) is a preferable choice for aggregating features [19]. The avg pool function perform average operation on all snippets. Particularly, the feature vector r, which indicates the video representation, is computed by the following equation: $r = \frac{\sum_{i=0}^{K} U p_t}{K}$, where $U \in R^{2d \times d}$ is the learned parameters.

Recurrent Neural Networks A more satisfying approach is to employ a recurrent neural network, such as Long Short-term Memory networks (LSTM) and Gated Recurrent Units(GRU), which can encode states and capture temporal ordering and long-range dependencies, on the top of frame features. Instead of LSTM and GRU, which prove not yielding a promise performance, we use a modified version of them as the following:

$$q_t = ReLU(W p_t + U q_{t-1} + b) \odot p_t \tag{7}$$

We take the last hidden states of modified RNN as the video feature vector: $r = q_K$ is the hidden states.

Temporal Convolution Layer We last attempt to use a temporal convolution layer (CNN) as the aggregation function. It is inspired by the convolutional neural networks for text classification [11]. We compute the final feature vector using a one-layer convolutional neural networks followed by a max-over-time pooling: $r = CNN([p_1, p_2, \ldots, p_K])$, where $r \in R^d$ and kernel size set to 3, strides set to 2.

The aggregation function learn a compositional representation as a result of capturing the underlying temporal structure of a sequence of frames. It easy to adopt the learned video representation to high-level video-related tasks, such as video classification or video description generation.

2.5 Classification

Finally, We feed the video representation r to the classifier and train whole model using the cross-entropy loss.

3 EXPERIMENTS AND RESULTS

3.1 Implementation Details

All of the models are implemented in TensorFlow [1]. The parameters of ConvNet are initialized with pre-trained model from ImageNet [13]. Each convolutional layer of feature extracting is followed by a batch normalization layer and a ReLU activation function. Training on videos uses standard SGD with initial learning rate of 0.001 and momentum set to 0.9 in all cases. All models are trained with synchronous parallelization across 8 GPUs. We set a batchsize of 128 and 64 depending on number of segments.

Data preprocessing is known to be of crucial importance for the performance of deep architectures. Firstly, we process the raw video by randomly cropping a 240×240 patch and resizing it to 299×299 pixels. Data augmentation, such as random left-right flipping, was applied for each video during training. We set the

number of segments N to 5 or 7 during training and fix it to 25 during evaluation. Overall, our architecture can process around 15 videos per GPU per second.

3.2 Fudan-Columbia video dataset

We conduct video category classification on the extended Fudan-Columbia video dataset (FCVID), the benchmark used at Large Scale Video Classification Challenge 2017 (LSVC2017). This challenge aims at recognizing the categories of web videos, such as social events, procedural events, objects and scenes etc. This dataset has 500 classes and total 155942 videos, with 62414 for training, 15604 for validation and the remaining 77924 for testing. In this work, we only use 1fps RGB frames provided by challenge organizer owing to time constraints. We will use the raw videos to extract more RGB frames, optical flows and audios in future.

The experimental results are presented at Table 2. We first study the effect of the number of segments. We can see that the results of $K = 5$ and $K = 7$ are almost same. It can be found that the relational function gain an improvement of 0.02~0.03 mAP score comparing with the baseline model (without relational function). Performance of different relational functions are shown at the second group, we found that **submul** reaches an mAP scores of 0.752, which is better than **projection** and **distance** for capturing the temporal relationship. The last group presents the effect of different aggregation functions when the number of segments is fixed to 7 and relational function is set to submul. Experimental results indicates that **avg pool** is the best choice among three candidates. We argues that the weakness of **LSTM** and **CNN** is that the information is not well propagated when training the whole model, instead of the disability of aggregating. We will improve it later.

Model	Validation	Test
VGG19 features, pooling	0.552	0.524
Inception-v3-BN, submul + avg pool	0.772	0.763
ensemble	0.82038	

Table 1: Performance on LSVC2017 validation dataset.

K	ConvNet	RelaFunc	AggFunc	mAP
5	Inception-v3-BN	w/o	avg pool	0.722
7	Inception-v3-BN	w/o	avg pool	0.724
7	Inception-v3-BN	projection	avg pool	0.748
7	Inception-v3-BN	distance	avg pool	0.743
7	Inception-v3-BN	submul	avg pool	**0.772**
7	Inception-v3-BN	submul	LSTM	0.741
7	Inception-v3-BN	submul	CNN	0.750

Table 2: Results comparison of different setting.

We also plot the learning curves on the trains set of LSVC2017. Basic settings are K=7, ConvNet=Inception-v3-BN and AggFunc=avg pool. As presented at Figure 3, comparison to the baseline model, relational feature encoding (submul) leads a faster convergence during training. Additionally, incorporating relational function let our model reduce training loss by a lower order of magnitude.

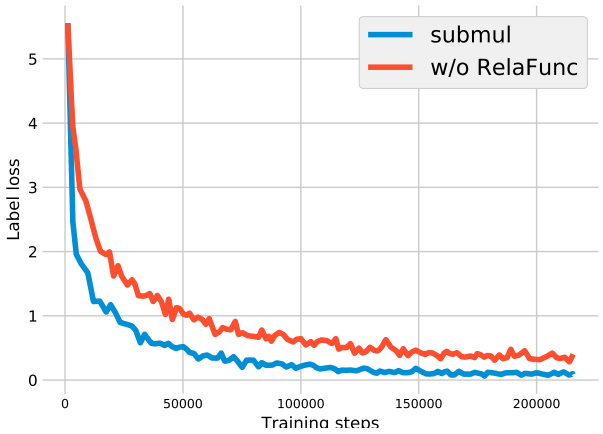


Figure 3: Learning curves on LSVC2017 train set.

3.3 UCF101 and Kinetics Dataset

Now we provide the performance on action video datasets - UCF101 and Kinetics datasets. Both of these two video datasets consists of trimmed videos describing a human action in daily life. The UCF101 dataset has 101 action categories, with train/validation splits of 9537/3783 videos. Kinetics is a very large action video dataset built by Deepmind, consisting of around 300k videos for 400 human actions classification. We also use 1fps RGB frame in our experiments for a fair comparison.

Methods		UCF101	Kinetics
RelaFunc	AggFunc	Acc.	Top-1/Top-5
w/o	avg pool	84.5%	69.1/88.2
submul	avg pool	90.1%	71.4/91.2

Table 3: Experiment results on UCF101 and Kinetics val set.

The results are presented at Table 2. Note that we haven't made an empirical study on different relational and aggregation functions, but show the boosted performance when incorporating relational feature encoding to capture the contextual information and temporal relationship. Both of two experiments hold the same training setting: N=3 and ConvNet=Inception-v3-BN. It can be observed that the relational function **submul** gain an improvement of 5.6% accuracy on UCF101 split-1 validation set, and an average error decrease of 0.09 on Kinetics validation set. The average error rate (avg error) is the average of top-1 error and top-5 error.

4 CONCLUSION

Most video recognition approaches nowadays require pre-computing multiple input modalities to achieve the state-of-the-art performance. In this work, we propose a Relational Feature Encoding Network that takes as input a single stream of RGB data, and generate multiple complementary feature streams on the fly using

high-level feature representations extracted by a backbone convolutional neural network. Each feature stream is computed by a relational function that operates on pairwise raw ConvNet features. The generated feature streams are combined by an aggregation function to form a holistic representation for classifying videos. Through experiments on multiple video recognition dataset, we empirically investigate the design choices for relational and aggregation functions, and validate the effectiveness of the proposed relational feature encoding network. We achieves an mAP score of 0.772 on the extended FCVID dataset.

ACKNOWLEDGMENTS

The authors would like to thank Ruimao Zhang and Zhanglin Peng for helpful discussion.

REFERENCES

[1] Martín Abadi, Ashish Agarwal, Paul Barham, Eugene Brevdo, Zhifeng Chen, Craig Citro, Gregory S. Corrado, Andy Davis, Jeffrey Dean, Matthieu Devin, Sanjay Ghemawat, Ian J. Goodfellow, Andrew Harp, Geoffrey Irving, Michael Isard, Yangqing Jia, Rafal Józefowicz, Lukasz Kaiser, Manjunath Kudlur, Josh Levenberg, Dan Mané, Rajat Monga, Sherry Moore, Derek Gordon Murray, Chris Olah, Mike Schuster, Jonathon Shlens, Benoit Steiner, Ilya Sutskever, Kunal Talwar, Paul A. Tucker, Vincent Vanhoucke, Vijay Vasudevan, Fernanda B. Viégas, Oriol Vinyals, Pete Warden, Martin Wattenberg, Martin Wicke, Yuan Yu, and Xiaoqiang Zheng. 2015. TensorFlow: Large-Scale Machine Learning on Heterogeneous Distributed Systems. *CoRR* abs/1603.04467 (2015).

[2] Joao Carreira and Andrew Zisserman. 2017. Quo Vadis, Action Recognition? A New Model and the Kinetics Dataset. *CoRR* abs/1705.07750 (2017).

[3] Sergey Ioffe Jonathon Shlens Zbigniew Wojna Christian Szegedy, Vincent Vanhoucke. 2015. Rethinking the Inception Architecture for Computer Vision. *arXiv preprint arXiv:1512.00567* (2015).

[4] Jeff Donahue, Lisa Anne Hendricks, Sergio Guadarrama, Marcus Rohrbach, Subhashini Venugopalan, Kate Saenko, and Trevor Darrell. 2015. Long-Term Recurrent Convolutional Networks for Visual Recognition and Description. *2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)* (2015), 2625–2634.

[5] Christoph Feichtenhofer, Axel Pinz, and Andrew Zisserman. 2016. Convolutional Two-Stream Network Fusion for Video Action Recognition. *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)* (2016), 1933–1941.

[6] Rohit Girdhar, Deva Ramanan, Abhinav Gupta, Josef Sivic, and Bryan C. Russell. 2017. ActionVLAD: Learning spatio-temporal aggregation for action classification. *CoRR* abs/1704.02895 (2017).

[7] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. 2015. Deep Residual Learning for Image Recognition. *arXiv preprint arXiv:1512.03385* (2015).

[8] Sergey Ioffe and Christian Szegedy. 2015. Batch Normalization: Accelerating Deep Network Training by Reducing Internal Covariate Shift. In *ICML*.

[9] Shuiwang Ji, Wei Xu, Ming Yang, and Kai Yu. 2010. 3D Convolutional Neural Networks for Human Action Recognition. *IEEE Transactions on Pattern Analysis and Machine Intelligence* 35 (2010), 221–231.

[10] Will Kay, Joao Carreira, Karen Simonyan, Brian Zhang, Chloe Hillier, Sudheendra Vijayanarasimhan, Fabio Viola, Tim Green, Trevor Back, Apostol Natsev, Mustafa Suleyman, and Andrew Zisserman. 2017. The Kinetics Human Action Video Dataset. *CoRR* abs/1705.06950 (2017).

[11] Yoon Kim. 2014. Convolutional Neural Networks for Sentence Classification. In *EMNLP*.

[12] Antoine Miech, Ivan Laptev, and Josef Sivic. 2017. Learnable pooling with Context Gating for video classification. *CoRR* abs/1706.06905 (2017).

[13] Olga Russakovsky, Jia Deng, Hao Su, Jonathan Krause, Sanjeev Satheesh, Sean Ma, Zhiheng Huang, Andrej Karpathy, Aditya Khosla, Michael S. Bernstein, Alexander C. Berg, and Li Fei-Fei. 2015. ImageNet Large Scale Visual Recognition Challenge. *International Journal of Computer Vision* 115 (2015), 211–252.

[14] Karen Simonyan and Andrew Zisserman. 2014. Two-Stream Convolutional Networks for Action Recognition in Videos. In *NIPS*.

[15] Khurram Soomro, Amir Roshan Zamir, and Mubarak Shah. 2012. UCF101: A Dataset of 101 Human Actions Classes From Videos in The Wild. *CoRR* abs/1212.0402 (2012).

[16] Christian Szegedy, Sergey Ioffe, Vincent Vanhoucke, and Alexander A. Alemi. 2017. Inception-v4, Inception-ResNet and the Impact of Residual Connections on Learning. In *AAAI*.

[17] Kai Sheng Tai, Richard Socher, and Christopher D. Manning. 2015. Improved Semantic Representations From Tree-Structured Long Short-Term Memory Networks. In *ACL*.

[18] Limin Wang, Yuanjun Xiong, Zhe Wang, Yu Qiao, Dahua Lin, Xiaoou Tang, and Luc Van Gool. 2017. Temporal Segment Networks for Action Recognition in Videos. *CoRR* abs/1705.02953 (2017).

[19] Limin Wang, Yuanjun Xiong, Zhe Wang, Yu Qiao, Dahua Lin, Xiaoou Tang, and Luc Val Gool. 2016. Temporal Segment Networks: Towards Good Practices for Deep Action Recognition. In *ECCV*.

[20] Zuxuan Wu, Yu-Gang Jiang, Xi Wang, Hao Ye, and Xiangyang Xue. 2016. Multi-Stream Multi-Class Fusion of Deep Networks for Video Classification. In *ACM Multimedia*.

[21] Cong Liu Yao Zhou and Yan Pan. 2016. Modelling Sentence Pairs with Tree-structured Attentive Encoder. In *The International Conference on Computational Linguistics*.

[22] Jun Wang Xiangyang Xue Shih-Fu Chang Yu-Gang Jiang, Zuxuan Wu. 2015. Exploiting Feature and Class Relationships in Video Categorization with Regularized Deep Neural Networks. *arXiv preprint arXiv:1502.07209* (2015).

[23] Bowen Zhang, Limin Wang, Zhe Wang, Yu Qiao, and Hanli Wang. 2016. Real-Time Action Recognition with Enhanced Motion Vector CNNs. *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)* (2016), 2718–2726.