RECOGNIZING MICRO ACTIONS IN VIDEOS: LEARNING MOTION DETAILS VIA SEGMENT-LEVEL TEMPORAL PYRAMID

Yang Mi[‡], *Song Wang*[†,‡,*]

[†]Tianjin Univeristy, Tianjin, China [‡]University of South Carolina, Columbia, SC, USA

miy@email.sc.edu, songwang@cec.sc.edu

ABSTRACT

Recognizing micro actions from videos is a very challenging problem since they involve only subtle motions of body parts. In this paper, we propose a new deep-learning based method for micro action recognition by building a segment-level temporal pyramid to better capture the motion details. More specifically, we first temporally sample the input video for short segments and for each of the video segment, we employ a two-stream convolutional neural networks (CNNs) followed by a temporal pyramid for extracting deep features. Finally, the features derived from all the video segments are combined for action classification. We evaluate the proposed method on a micro-action video dataset, as well as a general-action video dataset, with very promising results.

Index Terms— Action recognition, micro actions, temporal pyramid

1. INTRODUCTION

Video-based human action recognition plays a central role in video surveillance. In recent years, many methods have been developed for tackling this problem by extracting and classifying the motion features present in the input videos [1, 2, 3, 4]. However, in practice, many important human actions, such as small head nodding and certain gestures, are micro actions by only involving subtle motions of body parts and it is difficult to accurately extract and represent such subtle motions from the input videos for action recognition [5], especially in the presence of other common complexities, such as camera motion, view difference, lighting change, etc. Figure 1 shows sample videos of two micro actions: "a slight shift of attention" and "small hand pointing movement", both of which are taken by moving cameras. In this paper, our main goal is to develop a new approach that can better capture the motion details for recognizing micro actions.

Video-based action recognition is usually achieved by extracting spatial-temporal features to represent the underlying human motion, followed by a classifier for recognition. While earlier methods usually use handcrafted features, such as dense trajectories, HOF, etc. [1], recently, more effective methods are developed by extracting deep features automatically by supervised learning. In this paper, we follow the general framework used in recent state-of-the-art approaches for action recognition [2, 3, 6] – temporally sampling several short video segments from the input video, and then applying convolutional neural network (CNN) on each segment to extract spatial-temporal features for classification. In this framework, we further design a temporal pyramid after the CNNs to better capture motion details in each video segment for better recognizing micro actions.

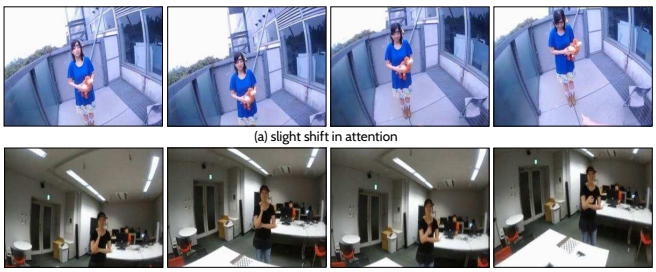

Fig. 1: Sample videos of two micro actions.

More specifically, after we sparsely sample a video for several short video segments, we further temporally partition each video segment into several evenly-shorter sub-segments. We then employ a two-stream CNN to extract the features in spatial and temporal domains, respectively, for each sub-segment. After that, we build a temporal pyramid for each video segment by pooling over the whole segment and different levels of segment partitions, respectively. The pooled features from different levels are combined to represent each segment and representations of all the segments are finally combined for action classification. In the experiments, we evaluate the proposed method on a micro-action video dataset [5], as well as a general-action dataset [7], with very promising results.

2. RELATED WORK

Both hand-crafted features, e.g., iDT [1], HOG [8], HOF [9], and MBH [10], and deep features have been used for video-

*corresponding author.

based action recognition. Deep features can be extracted using either 2D CNNs [11] or 3D CNNs [2, 12]. Two-stream CNNs have also been used for action recognition by combining appearance and motion features [3, 6, 13]. However, these methods may not handle well micro-action recognition without considering the problem of motion subtleness. Yonetani *et al* [5] propose to use the paired first-person (actor's view) and second-person (observer's view) videos for recognizing micro actions and build a micro-action video dataset for performance evaluation. However, the first-person videos are not always available in practice. In this paper, we focus on micro-action recognition by only using the second-person videos.

Temporal pyramid has also been used in video-based action recognition recently. Wang *et al* [14] use temporal pyramid pooling to convert multiple frame-level features into a fixed-length video-level representation. It is applied to the whole video. Differently, in this paper, temporal pyramid is built for each sampled video segment. Cheng *et al* [15] build spatial-temporal pyramid by applying 3D CNNs [2] on each video segment and then partition the feature map into spatiotemporal bins for pooling. Differently, we build temporal pyramid by explicitly partitioning each segment into sub-segments for pooling. In the experiments, we show that the proposed method significantly outperforms these two related methods.

3. PROPOSED METHOD

In this section, we first introduce the overall architecture of the proposed network. Then, we elaborate on describing the proposed segment-level temporal pyramid.

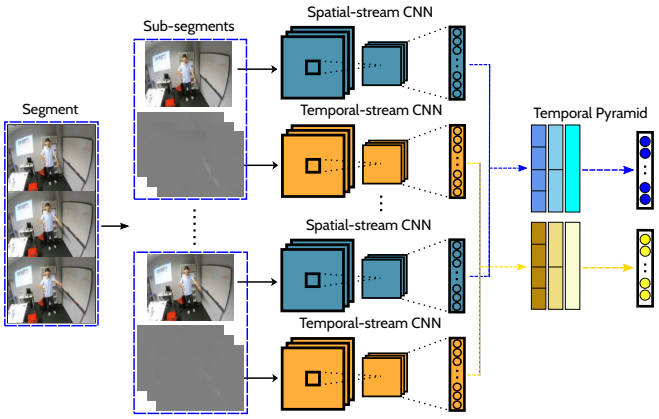

Fig. 2: An illustration of the construction of segment-level pyramid.

3.1. Overall Architecture

In this paper, we use temporal segment network (TSN) [3] as the baseline to model the long-term (video-level) temporal structure of the input video. Then we focus on capturing motion details from short-term (segment-level) dynamics in micro-action videos. The main difference between TSN and our method is that, TSN directly uses CNNs to extract features from a video segment, while we build temporal pyramid for the segment to better learn motion details.

Without loss of generality, given an input video V, we first uniformly divide it into N video clips and then randomly crop a short video segment of L frames from each clip. Then, the video V is represented by a sequence of segments $\{v_1, v_2, \cdots, v_N\}$. By removing redundant information, this sparse video-sampling scheme has been shown to be very effective in action recognition [3]. As shown in Fig. 2, for each L-frame segment, we evenly divide it into K sub-segments $\{s_1, s_2, \cdots, s_K\}$. Then, for each sub-segment, we employ two-stream CNNs to extract the appearance and motion features from the RGB frames and stacked optical flows, respectively. We employ the BN-Inception [16] as the based architecture to build the two-stream CNNs. For each stream, CNNs on all sub-segments share weights. The features from all sub-segments are built into the segment-level temporal pyramid, which contains different temporal-level representations of the segment, followed by a fully connected layer with a linear activation to produce the classification scores of all action classes for the segment. Finally, the classification scores are combined over all the segments for the video-level representation in each stream. Formally, the final prediction $\mathcal{P}$ of input video V is computed as follows:

$$\mathcal{P} = \sigma\left(\mathcal{F}\left(\mathcal{W}X_1 + b, \mathcal{W}X_2 + b, \cdots, \mathcal{W}X_N + b\right)\right) \quad (1)$$

where $X_1, X_2, \cdots, X_N$ are the representations (i.e., temporal pyramid) of N segments, $\mathcal{W}$ and b are the parameters of the last fully connected layer, $\mathcal{F}$ is an aggregation function to combine the classification scores from all the segments to obtain the video-level representation, σ is a prediction function that gives the probability of each action class for the video. Here, we choose evenly averaging as the aggregation function $\mathcal{F}$, and Softmax function as the prediction function σ. The predictions from different streams are fused via weighted averaging to combine the appearance and motion representations for the final classification performance. We will discuss details of the fusion in the experiments.

3.2. Segment-level Temporal Pyramid

Given one video segment v containing K sub-segments, we denote x_k as the feature extracted by the spatial or temporal stream network from the k-th sub-segment, where $k = 1, 2, \cdots, K$. For the segment v, we construct a T-level temporal pyramid. In the l-th level, the segment is evenly divide into 2^{l-1} parts. For the i-th part of the l-th level, we generate a pooled feature

$$f_i^{\,l} = \frac{1}{n} \sum_{k=(i-1)\cdot n+1}^{i\cdot n} x_k \quad (2)$$

where $n = \frac{K}{2^{l-1}}$, i.e., the number of sub-segments in this part. The feature ${f_i}^l$ is computed by average pooling over the features of all sub-segments in this part. Then we obtain the overall segment-level representation by concatenating the features from all parts in all levels as $X = Concat\left({f_i}^l\right)$, where $i = 1, 2, \cdots, 2^{l-1}$ and $l = 1, 2, \cdots, T$. Feature at level 1 provides a global representation of the entire segment, while higher levels provide more local representations by dividing the segment into more parts. This way, the proposed method can better capture motion details. As shown in Fig. 3, we randomly pick up 4 action classes from micro-action video dataset [5] and use Guided Grad-CAM [17] to visualize the class-discrimination map of both spatial and temporal CNNs with 3-level temporal pyramid. We can see that the proposed method can better capture details of the micro human motion, when compared to the baseline method, e.g., the proposed method can better capture motion details of arm movement for recognizing *negative* action.

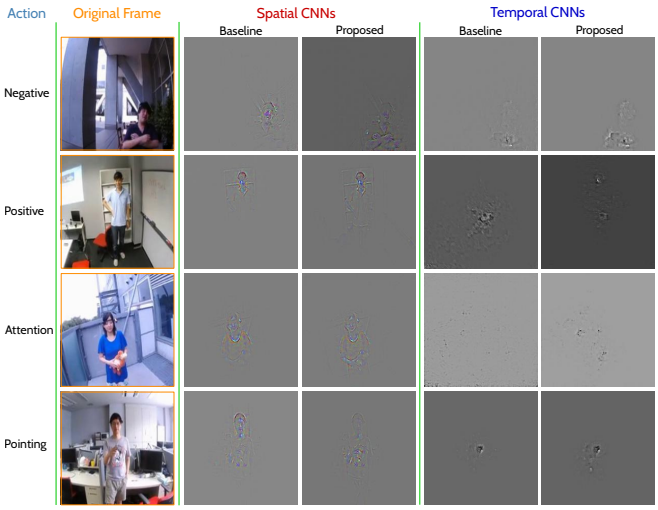

Fig. 3: Visualization of CNN models for micro-action recognition using Guided Grad-CAM. Figure best viewed in zoom-in mode.

3.3. Loss Function

The proposed network is trained with the standard categorical cross-entropy loss in an end-to-end fashion. Specifically, the network takes an input video V with the ground-truth labels $\{y_1, y_2, \cdots, y_M\}$ regarding M action classes, as a training sample. For each sample, the loss function is defined as

$$\mathcal{L} = -\sum_{p=1}^{M} y_p \left(F_p - \log \sum_{q=1}^{M} \exp\left(F_q\right) \right) \tag{3}$$

where $F = \mathcal{F}\left(\mathcal{W}X_1 + b, \mathcal{W}X_2 + b, \cdots, \mathcal{W}X_N + b\right)$ is the video-level representation of the sample.

4. EXPERIMENTS ON MICRO-ACTION VIDEOS

In this section, we first introduce the evaluation dataset, and then describe the experimental setup. Finally, we report the experimental results on this dataset.

4.1. Dataset and Experiment Setup

Dataset. We conduct experiments on a micro-action video dataset [5] to evaluate the effectiveness of the proposed method. This dataset contains 911 pairs of the first-person (actor's view) and the second-person (observer's view) videos, each of which includes one micro action performed by a person A and viewed by another person B. As mentioned above, the proposed method focuses on recognizing micro actions presented in a video. Hence, we only use the 911 second-person videos for evaluation. There are 7 micro actions: *pointing*, *attention*, *positive*, *negative*, *passing*, *receiving*, and *gesture*, and each action has 182, 97, 159, 40, 119, 143, 171 videos, respectively. Sample videos of the micro actions are illustrated in Fig. 4.

Network input. For training, we uniformly divide a sample video into $N = 3$ video clips and randomly choose a segment with $L = 20$ consecutive frames from each clip. Each segment is evenly divided into $K = 4$ sub-segments. All video frames are resized to a spatial resolution of 340×256. For each sub-segment, the spatial stream and temporal stream networks take the input of a single RGB frame and optical flow stacks respectively, where optical flow is computed via TVL1 optical flow algorithm [18] implemented in OpenCV [19]. We also use warped optical flow stacks for temporal stream network, to compensate the camera motion. Specifically, the warped optical flow is computed between two adjacent frames by using their estimated homography matrix as in [1].

Network training. We initialize the parameters of the proposed network with the weights pre-trained on ImageNet [20]. To train the network, we use the stochastic gradient descent algorithm, where the mini-batch size is set to 32, and momentum set to 0.9. The initial learning rate is 0.001. For the spatial stream network, the learning rate is decreased by a factor of $\frac{1}{10}$ after 30 and 60 epochs. The training process stops after 80 epochs. For the temporal stream network with the input modality of optical flow or warped optical flow stacks, the learning rate is decreased by a factor of $\frac{1}{10}$ after 110 and 250 epochs, and the training stops after 320 epochs. To effectively train the network, we follow the good practice in [3]: (1) regularization: partial BN and extra dropout layer after the global pooling layer in BN-Inception; (2) data augmentation: corner cropping and scale jittering.

Network testing. Following the testing scheme of the original two-stream CNNs [6], we sample each testing video into 25 segments. To fuse the predictions from different input modalities, we use weighted averaging scheme to combine

Fig. 4: Exemplar videos for the 7 different micro actions, where each column shows sample frames of one video.

the classification scores. When combining the RGB based and optical flow based modalities, we give more credits to temporal stream by setting its weight as 1.5 and that of spatial stream as 1, due to the fact that motion-based input (optical flow) usually contributes more than appearance-based input (RGB) for action recognition. When both optical flow based modalities are used, warped optical flow stacks is served as complementary modality, and the weights of temporal stream is set to 1.5 for optical flow stacks and 1 for warped optical flow stacks.

Evaluation scheme. Following the original evaluation scheme in [5], we conduct a three-fold cross validation on all the 911 second-person videos in the micro-action video dataset. Specifically, the dataset is randomly split into three subsets of similar size: two subsets are used for training and the remaining subset is used for testing. We calculate the average accuracies over all actions as the recognition performance. All experiments are run on Pytorch [21].

4.2. Results

We study the impact of the segment-level temporal pyramid by varying the number of pyramid levels. We conduct experiments on different input modalities, in comparison with the baseline method TSN, which uses 3 five-frame segments to directly construct the video-level temporal structure [3]. Notice that, the one-level pyramid would be the same as the baseline method but using 12 segments (subsegments constructed in this paper), each of which contains five frames. The recognition accuracies are shown in Table 1. For convenience, the input modalities of RGB frames, optical flow stacks, and warped optical flow stacks, are denoted as RGB, OF, and WOF, respectively. We can see that, the proposed method outperforms the baseline method on each input modality, and achieves the highest accuracies when constructing the 3-level pyramid. These results verify the effectiveness of the proposed method, which can better learn motion details for recognizing micro actions.

For the use of different input modalities, although WOF tries to compensate camera motions, it leads to lower accuracies than OF. One reason might be the use of homography transform in WOF: homography transform could not reflect the camera motion well when the background in not on a planar surface [22]. Fusing the predictions from different input modalities on spatial and/or temporal stream network, can produce better performance than using a single modality. When combining RGB, OF and WOF with 3-level pyramid, we achieve the highest accuracy of 81.7%. Thus, we use these three input modalities with 3-level pyramid for the remaining experiments.

Input Modality	Baseline	1-Level	2-level	3-level	4-level
RGB	54.4%	58.1%	59.1%	60.0%	59.4%
OF	74.1%	77.2%	78.1%	79.8%	78.5%
WOF	74.0%	74.2%	76.1%	76.6%	76.2%
RGB+OF	75.9%	78.5%	79.3%	80.6%	79.6%
OF+WOF	76.2%	78.7%	80.7%	81.4%	80.9%
RGB+OF+WOF	77.7%	78.9%	81.3%	**81.7**%	81.4%

Table 1: Performance of the proposed method with different number of pyramid levels on different input modalities.

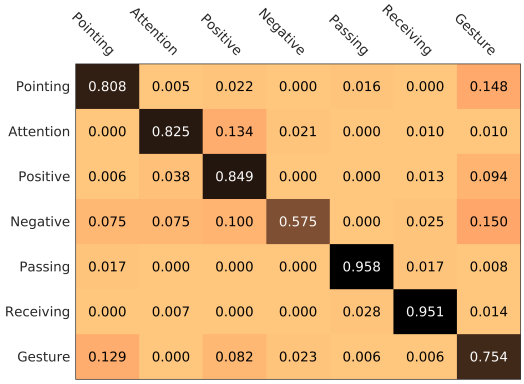

Fig. 5: Confusion matrix of the proposed method on the micro-action video dataset.

Figure 5 shows the confusion matrix of the proposed method on the micro-action video dataset. We can see that, the actions *passing* and *receiving* show high accuracy, because their features are more discriminative than other action classes. Notice that, recognition accuracy of *negtive* action is low, and the videos of this action are likely to be classified as *gesture* action. The main reason is that both actions involve

hand gestures, which makes their motion patterns difficult to distinguish.

We choose four existing methods for comparison. The first one is a handcrafted feature based method [1], which detects the improved dense trajectories (iDT), HOG, HOF and MBH features, with linear SVM (support vector machine) as the classifier. The second one is proposed by R. Yonetani *et al* [5], where multiple point-of-view features (MPOV) from paired of first-person and second-person videos are used. Specifically, they use cumulative displacement [23] encoded by pooled time series [24] from first-person videos and iDT from second-person videos, followed by a standard linear decision function to describe the relative importance of the two point-of-view features. The third one is proposed by Tran *et al* [2], where deep 3D convolutional neural networks (C3D) are used to learn features from fixed-length video clips, followed by using linear SVM as the classifiers. The fourth one is a two-stream deep-learning method [3], which develops temporal segment network (TSN) to model long-term temporal structure. Notice that, we use TSN as the baseline architecture to build the proposed network. For iDT and MPOV, we use the results reported by R. Yonetani *et al* [5]. For C3D and TSN, we use their released codes with default settings and parameters, and make sure that the loss converges during the training process.

Method	Pretraining Dataset	Accuracy
iDT [1]	None	43.0%
C3D [2]	Sports-1M	53.8%
MPOV [5]	None	69.0%
TSN [3]	ImageNet	77.7%
Proposed	ImageNet	**81.7**%

Table 2: Comparison results against several existing action recognition methods on the micro-action video dataset.

From Table 2, we can see that, both C3D and TSN perform better than iDT, which verifies the effectiveness of the deeply learned features. MPOV performs much better than iDT. This is because that, MPOV uses iDT as the features from the second-person videos, and further combine iDT with the features from the first-person videos to improve performance. MPOV also outperforms C3D, which further shows the advantage of using additional information from the first-person videos. TSN performs better than C3D and MPOV, due to the power of modelling the long-term temporal structure. Notice that, the proposed method significantly outperforms MPOV, while only using the second-person videos. The proposed method outperforms all the comparison methods, and improves the recognition accuracy by 4% to 81.7%, when compared to TSN. This verifies the effectiveness of the proposed segment-level temporal pyramid, which can better learn motion details from short-term dynamics for better recognizing micro actions.

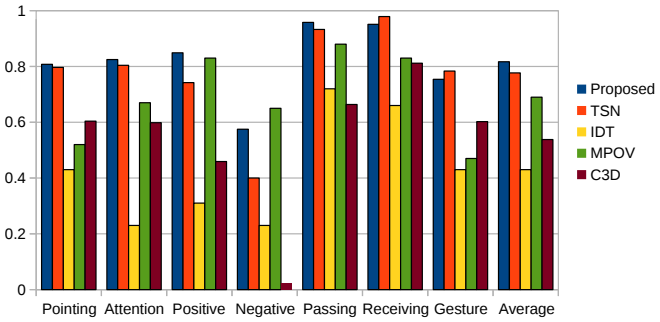

Fig. 6: Per-class action recognition accuracy of the proposed method and comparison methods on the micro-action video dataset.

In Fig. 6, we report the performance for the proposed and comparison methods on each action class. We can see that, the proposed method outperforms the comparison methods on four out of seven action classes, and achieves the second highest performance on the other three action classes. We notice that all the methods show low accuracies on the *negtive* action, due to the complexity of mixing micro movements from different parts of human body.

5. EXPERIMENTS ON GENERAL-ACTION VIDEOS

To further demonstrate the effectiveness of the proposed method, we also evaluate our method on a large-size general-action video dataset HMDB51 [7]. This dataset is a well-established benchmark for general-action recognition. It contains 6,766 videos from 51 action categories. We follow the original evaluation scheme using three training/testing splits and report average accuracy over these splits. The remaining experimental setup is the same as the one described in Section 4.1.

We compare the proposed method with several state-of-art approaches, including both hand-crafted feature based methods such as iDT [1], and deep-learning based methods such as TSN [3]. We use the results reported in these papers for comparison. As we can see from Table 3, the proposed method significantly outperforms the two previous temporal-pyramid based methods – STPP Net [14] and TPP Net [15]. The proposed method also outperforms all the comparison methods that are pre-trained on ImageNet, and further improves the recognition accuracy by 3.2% to 72.6%, when compared to TSN. These results verify that, the proposed method can also effectively learn motion details for recognizing general actions. The performance of the proposed method is comparable to the state-of-art method R(2+1)D-Two Stream, which is pre-trained on Sports-1M, a large dataset designed for video classification.

Method	Pretraining Dataset	Accuracy
iDT [1]	None	57.2%
MoFAP [25]	None	61.7%
STPP Net [14]	Kinetic	56.7%
TPP Net [15]	ImageNet	61.8%
Two Stream [6]	ImageNet	59.4%
Two-Stream Fusion [26]	ImageNet	65.4%
Spatiotemp. MultiNet [13]	ImageNet	68.9%
TSN [3]	ImageNet	69.4%
TAN [27]	ImageNet	72.5%
Proposed	ImageNet	72.6%
R(2+1)D-Two Stream [12]	Sports-1M	**72.7**%

Table 3: Comparison results against several existing action recognition methods on the HMDB51 dataset.

6. CONCLUSION

This paper introduced a new approach to recognize micro human actions by effectively learning motion details via segment-level temporal pyramid. In this method, we first sample a video into several segments and then for each segment, we utilize two-stream CNNs to extract visual features, followed by building temporal pyramids to model the segment. The segment-level representations are combined for video-level classification in each stream. Finally, the classification scores of different streams are combined via weighted average fusion. We tested the proposed method on a micro-action video dataset and a general-action video dataset, with very promising results.

7. REFERENCES

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

[2] D. Tran, L. Bourdev, R. Fergus, L. Torresani, and M. Paluri, "Learning spatiotemporal features with 3d convolutional networks," in *ICCV*, 2015.

[3] Y. Wang, L.and Xiong, Z. Wang, Y. Qiao, D. Lin, X. Tang, and L. Van Gool, "Temporal segment networks: Towards good practices for deep action recognition," in *ECCV*, 2016.

[4] Y. Mi, K. Zheng, and S. Wang, "Recognizing actions in wearable-camera videos by training classifiers on fixed-camera videos," in *ICME*, 2018.

[5] R. Yonetani, K. Kitani, and Y. Sato, "Recognizing micro-actions and reactions from paired egocentric videos," in *CVPR*, 2016.

[6] K. Simonyan and A. Zisserman, "Two-stream convolutional networks for action recognition in videos," in *NIPS*, 2014.

[7] H. Kuehne, H. Jhuang, E. Garrote, T. Poggio, and T. Serre, "Hmdb: A large video database for human motion recognition," in *ICPR*, 2011.

[8] N. Dalal and B. Triggs, "Histograms of oriented gradients for human detection," in *CVPR*, 2005.

[9] R. Chaudhry, A. Ravichandran, G. Hager, and R. Vidal, "Histograms of oriented optical flow and binet-cauchy kernels on nonlinear dynamical systems for the recognition of human actions," in *CVPR*, 2009.

[10] H. Wang, A. Kläser, C. Schmid, and C. Liu, "Dense trajectories and motion boundary descriptors for action recognition," *IJCV*, 2013.

[11] A. Karpathy, G. Toderici, S. Shetty, T. Leung, R. Sukthankar, and F. Li, "Large-scale video classification with convolutional neural networks," in *CVPR*, 2014.

[12] D. Tran, H. Wang, L. Torresani, J. Ray, Y. LeCun, and M. Paluri, "A closer look at spatiotemporal convolutions for action recognition," in *CVPR*, 2018.

[13] C. Feichtenhofer, A. Pinz, and R. Wildes, "Spatiotemporal multiplier networks for video action recognition," in *CVPR*, 2017.

[14] P. Wang, Y. Cao, C. Shen, L. Liu, and H. Shen, "Temporal pyramid pooling-based convolutional neural network for action recognition," *IEEE TCSVT*, 2017.

[15] C. Cheng, P. Lv, and B. Su, "Spatiotemporal pyramid pooling in 3d convolutional neural networks for action recognition," in *ICIP*, 2018.

[16] S. Ioffe and C. Szegedy, "Batch normalization: Accelerating deep network training by reducing internal covariate shift," *arXiv preprint arXiv:1502.03167*, 2015.

[17] R. Selvaraju, M. Cogswell, A. Das, R. Vedantam, D. Parikh, and D. Batra, "Grad-cam: Visual explanations from deep networks via gradient-based localization," in *ICCV*, 2017.

[18] C. Zach, T. Pock, and H. Bischof, "A duality based approach for realtime tv-l 1 optical flow," in *Joint Pattern Recognition Symposium*, 2007.

[19] G. Bradski, "The OpenCV Library," *Dr. Dobb's Journal of Software Tools*, 2000.

[20] J. Deng, W. Dong, R. Socher, L. Li, K. Li, and F. Li, "Imagenet: A large-scale hierarchical image database," in *CVPR*, 2009.

[21] A. Paszke, S. Gross, S. Chintala, G. Chanan, E. Yang, Z. DeVito, Z. Lin, A. Desmaison, L. Antiga, and A. Lerer, "Automatic differentiation in pytorch," in *NIPS*, 2017.

[22] R. Hartley and A. Zisserman, *Multiple view geometry in computer vision*, Cambridge university press, 2003.

[23] Y. Poleg, C. Arora, and S. Peleg, "Temporal segmentation of egocentric videos," in *CVPR*, 2014.

[24] M. Ryoo, B. Rothrock, and L. Matthies, "Pooled motion features for first-person videos," in *CVPR*, 2015.

[25] L. Wang, Y. Qiao, and X. Tang, "Mofap: A multi-level representation for action recognition," *IJCV*, 2016.

[26] C. Feichtenhofer, A. Pinz, and A. Zisserman, "Convolutional two-stream network fusion for video action recognition," in *CVPR*, 2016.

[27] T. Shi, Y. Tian, T. Huang, and Y. Wang, "Temporal attentive network for action recognition," in *ICME*, 2018.