White Rose Research Online

eprints@whiterose.ac.uk
https://eprints.whiterose.ac.uk
Universities of Leeds, Sheffield and York

Deposited via The University of York.

White Rose Research Online URL for this paper:
https://eprints.whiterose.ac.uk/id/eprint/158187/

Version: Accepted Version

Proceedings Paper:
Huang, Guoxi and Bors, Adrian Gheorghe (2020) Learning Spatio-Temporal Representations with Temporal Squeeze Pooling. In: Proc. IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP). IEEE.

https://doi.org/10.1109/ICASSP40776.2020.9054200

Reuse
Items deposited in White Rose Research Online are protected by copyright, with all rights reserved unless indicated otherwise. They may be downloaded and/or printed for private study, or other acts as permitted by national copyright laws. The publisher or other rights holders may allow further reproduction and re-use of the full text version. This is indicated by the licence information on the White Rose Research Online record for the item.

Takedown
If you consider content in White Rose Research Online to be in breach of UK law, please notify us by emailing eprints@whiterose.ac.uk including the URL of the record and the reason for the withdrawal request.

LEARNING SPATIO-TEMPORAL REPRESENTATIONS WITH TEMPORAL SQUEEZE POOLING

Guoxi Huang and Adrian G. Bors

Department of Computer Science, University of York, York YO10 5GH, UK

ABSTRACT

In this paper, we propose a new video representation learning method, named Temporal Squeeze (TS) pooling, which can extract the essential movement information from a long sequence of video frames and map it into a set of few images, named Squeezed Images. By embedding the Temporal Squeeze pooling as a layer into off-the-shelf Convolution Neural Networks (CNN), we design a new video classification model, named Temporal Squeeze Network (TeSNet). The resulting Squeezed Images contain the essential movement information from the video frames, corresponding to the optimization of the video classification task. We evaluate our architecture on two video classification benchmarks, and the results achieved are compared to the state-of-the-art.

Index Terms— Convolution Neural Networks (CNN), Temporal Squeeze pooling, video representation.

1. INTRODUCTION

Video classification attracts increasing research interest, given its numerous applications. Extracting features characteristic to the movement in the scene is essential in tasks which are required in applications from video surveillance to video summarization, or when attributing the movement of an actor in a movie. We can identify two categories of approaches: using hand-crafted features, and based on deep learning.

Hand-crafted features, such as 3D histograms of gradients [1], scale-invariant spatio-temporal interest points [2, 3] dense trajectories [4] as well as the dynamics of change in movement and location [5], have been used in various video tasks. Bilen *et al.* [6] proposed dynamic images that summarize spatio-temporal information of a video clip into a single image which preserves the semantics of the scene in a compact format.

Convolution Neural Networks (CNN) have been used for learning visual representations in many applications and recently they have been employed successfully for video processing as well. The two-stream video classification model [7] contains both spatial and temporal processing pipelines. CNNs models containing 3D convolution kernels (3D CNN), such as C3D [8] and I3D [9], represent a promising way for spatio-temporal representation learning. However, 3D CNNs are prone to overfitting when trained on small datasets from scratch. Meanwhile, the training of 3D CNN on large datasets requires very large computational demands, and the model size is quadratic when compared to 2D CNNs used for video processing. Long short-term memory (LSTM) networks [10], represent a category of recursive neural networks (RNN) that can learn the long term dependency of time series data and can compensate for the shortcomings of 2D CNNs to some extent. 2D CNN+LSTM [11] was proposed to capture the spatio-temporal information from videos. However, 2D CNN+LSTM was shown to have lower performance than the two-stream model in action recognition benchmarks, [9].

In this study, a new Temporal Squeeze (TS) pooling methodology, which can be embedded into CNNs, is proposed. The proposed TS pooling approach aggregates the temporal video information into a reduced spatial dimension by means of an optimization approach which preserves the video information characteristics. In this study, TS pooling is optimized for the video classification task. TS pooling can compensate for the shortcomings of the dynamic images [6], by controlling the pooling size. In this research study, we demonstrate that the proposed TS pooling mechanism can summarize the visual representation of up to 64 video frames while dynamic images would only process 10 frames. By embedding the temporal squeeze pooling as a layer into the off-the-shelf CNNs, we design a new video classification model named Temporal Squeeze Network (TeSNet). The proposed methodology for representing video information is presented in Section 2. The experimental results are provided in Section 3 and the conclusion is drawn in Section 4.

2. VIDEO INFORMATION REPRESENTATION

2.1. Temporal Squeeze Pooling

The proposed approach relies on the observation that consecutive video frames usually contain repeating information, especially either the background for a still camera, or the foreground, when the camera follows a target. A temporal squeeze pooling aims to compress the dynamic information of a video clip with K frames into D frames ($D << K$), such that essential information is preserved. Consequently, repeating information is filtered out while preserving the essential, usually specific movement patterns. Let $\mathbf{X} = [\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_K]$ denote K video frames where

$\mathbf{x}_i \in \mathbb{R}^{H\times W\times C}$, $i = 1, \ldots, K$ and H, W, C represent the height, width and the number of channels (colour), respectively. The TS layer aims to find out the optimal hyperplane $\mathbf{A} \in \mathbb{R}^{K\times D}$, and map every pixel of $\mathbf{X}$ from the vector space of $\mathbb{R}^K$ onto a much smaller information defining space $\mathbb{R}^D$. The aim is to preserve the relevant dynamic information across the temporal direction into the compressed space.

In the following, the squeeze and excitation operations proposed in [12] are adopted for the TS pooling. The frame sequence $\mathbf{X}$ is initially processed by the squeeze operation, producing a frame descriptor. The squeeze operation is implemented by using global average pooling along the spatial dimensions H, W and C. Then, the squeeze operation is followed by the excitation operation, which is made up of two consecutive fully connected (FC) layers. The output of the excitation operation is reshaped to become the column space of $\mathbf{A}$, which defines a hyperplane. In the squeeze operation, the k-th element of a frame-wise $\mathbf{z} \in \mathbb{R}^K$ is calculated by:

$$z_k = F_{sq}(\mathbf{x}_k) = \frac{1}{HWC}\sum_{i=1}^{H}\sum_{j=1}^{W}\sum_{l=1}^{C} x_k(i,j,l). \quad (1)$$

In the excitation operation, the input-specific hyperplane is calculated by:

$$F_{ex}(\mathbf{z}, \mathbf{W}) = \delta_2(\mathbf{W}_2\delta_1(\mathbf{W}_1\mathbf{z})), \quad (2)$$

where δ_1 and δ_2 refer to the activation functions and $\mathbf{W}_1 \in \mathbb{R}^{K\times K}$, $\mathbf{W}_2 \in \mathbb{R}^{KD\times K}$ refer to the weights of the FC layers. Then, the output of equation (2) is reshaped into a matrix $\mathbf{A}' \in \mathbb{R}^{K\times D}$. The input-specific hyperplane for the projection is given by

$$\mathbf{A} = \Phi(\mathbf{A}'), \quad (3)$$

where Φ is a function that guarantees $\mathbf{A}$ is column independent. We flatten $\mathbf{X}$ along its H, W and C dimensions into a vector $\overline{\mathbf{X}} = [\bar{\mathbf{x}}_1, \bar{\mathbf{x}}_2, ..., \bar{\mathbf{x}}_{HWC}]$ where $\bar{\mathbf{x}}_i \in \mathbb{R}^K$, and then project it onto the hyperplane $\mathbf{A}$, resulting in a vector $\widehat{\mathbf{X}}$. The i-th element of the projection, $\widehat{\mathbf{x}}$ is calculated by:

$$\begin{aligned} \mathbf{y}_i &= (\mathbf{A}^\mathsf{T}\mathbf{A})^{-1}\mathbf{A}^\mathsf{T}\bar{\mathbf{x}}_i, \\ \widehat{\mathbf{x}}_i &= \mathbf{A}\mathbf{y}_i, \quad i = 1, \ldots, HWC \end{aligned} \quad (4)$$

where $\mathbf{y}_i \in \mathbb{R}^D$ represent the mapping coefficients. We reshape the vector $\mathbf{Y} = [\mathbf{y}_1, \mathbf{y}_2, \ldots, \mathbf{y}_{HWC}]$ into a new image sequence $\mathbf{Y}'$ of D frames of size $H \times W \times C$. The squeezed sequence of D frames can be used as a simplified, yet an information comprehensive representation, that summarizes the dynamics taking place in the given set of K video frames.

2.2. Optimization

In this study, the TS pooling is optimized with respect to the video classification task. In order to ensure that the projection $\widehat{\mathbf{X}}$ retains as much meaningful spatio-temporal information as possible from the original video sequence $\mathbf{X}$, $\widehat{\mathbf{X}}$ should be close to the original video data $\overline{\mathbf{X}}$. This relies onto finding the optimal hyperplane $\mathbf{A}$ fitting $\mathbf{X}$ aiming to minimize the residuals of projections. Let us denote the mean absolute error (MAE) on projections by l_{proj}, calculated as:

$$l_{proj} = \frac{1}{HWC}\sum_{i}^{HWC} \left\|\bar{\mathbf{x}}_i - \widehat{\mathbf{x}}_i\right\|, \quad (5)$$

where $\|\cdot\|$ represents the standard L^2 norm in the K-dimensional Euclidean space $\mathbb{R}^K$.

2.3. Temporal Squeeze Network

The Temporal Squeeze pooling can process not just video frames but also the outputs of convolution layers of a CNN. When it is plugged into off-the-shelf CNNs, it forms a new architecture, named Temporal Squeeze Network (TeSNet). We choose the Inception-ResNet-V2 [13] as the backbone CNN embedding the TS pooling block. In order to form an end-to-end training, we add the loss term l_{proj} from (5) to the classification loss used in the original network, resulting in the following loss function:

$$l_{final} = l_{classif} + \beta\sum_{i=1}^{M} l^i_{proj} + \lambda l_{L2}, \quad (6)$$

where $l_{classif}$ is the cross-entropy loss of the classification [13], l_{L2} is the L2 normalization term of all the trainable weights in the architecture, λ is the weight decay, β is the weight for the TS loss component l_{proj}, where the projection residuals are summed up for all M TS layers.

TS layers can be embedded in different sections of the backbone CNN. We design our model by following the principle of decreasing the number of mapped frames D when embedding into a deeper network layer position. In this case, the model represents a pyramidal video processing scheme. The first TS layer should be configured with a relatively larger D generating more frames, and therefore the loss of temporal information caused by successive pyramidal projections would be reduced. We adopt the two-stream architecture [7], including an RGB image frame stream and an Optical Flow (OF) stream. For the OF stream we use the TV-L1 optical flow algorithm [14] and its output is stored as JPEG images, where the colour encodes the optical flow vectors.

3. EXPERIMENTAL RESULTS

3.1. Dataset and Implementation Details

We conduct experiments on two human activity classification benchmarks, UCF101 [15] and HMDB51 [16]. UCF101 contains 13,320 real video sequences labelled int 101 classes, collected from YouTube [15], while HMDB51 contains 7,000 video clips distributed in 51 action classes [16]. During the experiments we follow the "three train/test splits" rule, and report the final performance by averaging the top-1 accuracy over the three splits. Our model is pre-trained on ImageNet [17]. To evaluate our model, we reimplement the Temporal Segment Network (TSN) [18] with our backbone network. We set the dropout as 0.5 to prevent overfitting and adopt

(a) Selection of 10 consecutive frames from video sequence. (b) Output of TS

Fig. 1. Visualizing the input and the corresponding output of the TS layer with $K = 10$, $D = 2$.

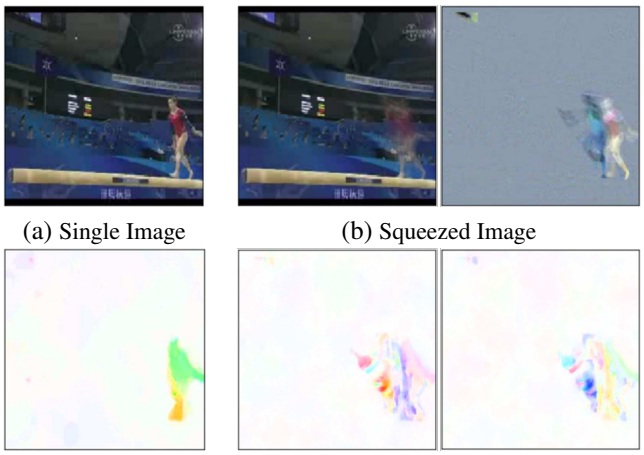
the same data augmentation techniques as in [18] for network training. The size of the input frames is set to 299×299, which is randomly cropped from the resized images, and K consecutive frames are randomly selected from each video sequence. We use Stochastic Gradient Descent for optimizing the network parameters in which the batch size is set to 32, momentum of 0.9, weight decay $\lambda = 4e^{-5}$, $\beta = 10$. The initial learning rate is set to 0.001 for the image stream and at 0.005 for the Optical Flow stream. We train the model for 30 epochs, with a ten times reduction for the learning rate when the validation accuracy saturates.

During testing, we uniformly sample 20 clips from each video clip and perform spatially fully convolutional inference for all clips, and the video-level score is obtained by averaging all the clip prediction scores of a video. For the proposed TeSNet, we set $\Phi(\cdot) = I$ in (3), resulting in $\mathbf{A}' = \mathbf{A}$, while the column independent $\mathbf{A}$ is properly initialized. We consider LeakyReLU for $\delta_2(\cdot)$ and the Sigmoid activation function for $\delta_1(\cdot)$ in equation (2) and these choices are crucial for the performance of the model.

3.2. Visualization Analysis

We explore how the temporal squeeze pooling represents the spatio-temporal information within the video clips by visualizing its outputs. In Fig 1, we show the output of the TS layer with $K = 10$, $D = 2$ resulting in 2 squeezed images. The clip, shown on the first row in Fig 1a display a clear salient movement, and we can observe that its corresponding output of TS summarizes the spatio-temporal information, as shown on the first row in Fig 1b. The other clip, shown on the second row, does not contain any obvious movement. When there is no movement present in a video clip, the TS layer captures the characteristic static information about the scene, as shown in the last two images from the second row of Fig 1b.

Fig 2 depicts the outputs of the TS layer with $K = 10$, $D = 2$. The output of the TS layer with RGB frames is shown in Fig. 2b, and the output of the TS layer of optical flow images is shown in Fig. 2d. We observe that the output of the TS layer tends to preserve the still information and the motion information separately. This indicates that by considering a single image we may not be able to represent the underlying spatio-temporal information from the video. Moreover, when considering $D = 3$, the classification accuracy is higher than for $D = 1$, according to the results from Table 1. This result further demonstrates that summarizing the dynamics of a long video clip into a single image would lose essential spatio-temporal information. A dynamic image [6] attempts to summarize the entire information from a video clip into a single image, which can explain why they fail to properly represent long video clips.



(a) Single Image (b) Squeezed Image

(c) Optical Flow (d) Squeezed Optical Flow

Fig. 2. Given input video frames, flow images and the corresponding outputs for the TS layers $K = 10$, $D = 2$.

3.3. Embedding the TS layer into the network

In the following, we explore where and how to embed TS layers into the CNN. The results are shown in Table 1, where the second column indicates the location for inserting a TS layer with the corresponding D indicated in the third column. A single TS layer, $M = 1$ is embedded in the settings No. 1 and 2, while $M = 2$ for the settings No. 3, 4 and 5. The model from No. 1 setting, which embeds a TS layer directly after the inputs of the network, achieves the best result in all settings. However, the model with No. 5 setting which embeds two TS layers into the backbone network requires less computation and has almost the same performance as the No. 1 setting. When a lower level of computation complexity is required, then No. 5 setting is preferable to be used. When inserting the TS layers into the middle section of the backbone network it leads to worse performance. One possible explanation is that the network was initially pretrained on ImageNet and then the inserted TS layers did not fit well with

the settings of these pretrained kernels and resulted in poor performance. To avoid this problem, the model including its TS layers has to be pretrained on a large video dataset.

No.	Location of TS layer	Number of squeezed frames (D_i)	Top-1 (%)
1	Input	$D_1 = 3$	85.4
2	Input	$D_1 = 1$	83.1
3	Conv2d_1a_3x3 Conv2d_4a_3x3	$D_1 = 3$ $D_2 = 1$	81.7
4	Conv2d_1a_3x3 Block A	$D_1 = 3$ $D_2 = 1$	84.9
5	Conv2d_1a_3x3 Block B	$D_1 = 3$ $D_2 = 1$	85.3

Table 1. Evaluating the accuracy when embedding the TS layer at different depths of the CNN.

We also explore how the length of the input video clip affects the performance of our model. we consider a rather small batch size of 8 and a maximum clip length of 64 because of the GPU memory limitation. For a clip length of 64, we adopt the setting No. 5 from Table 1 but consider $D_1 = 16$ and $D_2 = 4$. When considering clip lengths of 10 or 16, we use the first setting from Table 1. The results are shown in Table 2. It can be observed that when increasing the length of the video clip, the performance is improving as well.

clip length	Classif. (%)
10	85.3
16	86.2
64	87.8

Table 2. Comparing the effect of various clip length of videos on RGB stream on the split 1 of UCF101 database.

Architecture	length	RGB	OF	RGB + OF
Baseline	1	83.5	85.4	92.5
TSN	3	85.0	85.1	92.9
TeSNet	64	87.8	88.2	95.2

Table 3. Performance of different architectures with two-stream on the split 1 of UCF101 database.

We also evaluate TeSNet by comparing with the baseline and TSN [18] whose the backbone network is Inception-ResNet-v2. The results provided by different architectures and streams are shown in Table 3. We can observe that the fusion of the RGB and OF streams with TeSNet successfully boosts up the top-1 accuracy from 92.5% to 95.2% on the split 1 of UCF101, and outperforms that of TSN (Inception-ResNet-v2) by 2.3% which demonstrates the effectiveness of the proposed method.

Method	UCF101	HMDB51
iDT+Fisher vector [19]	84.8	57.2
iDT+HSV [20]	87.9	61.1
C3D+iDT+SVM [8]	90.4	-
Two-Stream (fusion by SVM) [7]	88.0	59.4
Two-Stream Fusion+iDT [21]	93.5	69.2
TSN (BN-Inception) [18]	94.2	69.4
Two-Stream I3D [9]	93.4	66.4
TDD+iDT [22]	91.5	65.9
Dynamic Image Network [6]	95.5	72.5
Temporal Squeeze Network	95.2	71.5

Table 4. Temporal squeeze network compared with other methods on UCF101 and HMDB51, in terms of top-1 accuracy, averaged over three splits.

3.4. Comparison with the state-of-the-art

For fare comparisons, we only consider those models that are pre-trained on ImageNet [17]. The results are provided in Table 4. The proposed TeSNet achieves 95.2% top-1 accuracy on UCF101 and 71.5% on HMDB51, which outperforms TSN (BN-Inception) by 1% and 2.1% on UCF101 and HMDB51, respectively. As the dynamic image network fuses the prediction scores of four streams using a better backbone network architecture, while the proposed model only uses two-streams, and therefore the results are not directly comparable. Nevertheless, the advantage of our proposed method is that we can control the number of frames for the output of the TS layer, while the dynamic image method [6] can only summarize a part of the spatio-temporal information into a single image. The proposed TeSNet method can represent the information through TS pooling from as many as 64 frames, unlike in [6], where the dynamic image method would show performance degradation when processing more than 20 frames.

4. CONCLUSION

In this paper, we propose a new video representation scheme, while aiming to improve video classification tasks, called Temporal Squeeze (TS) pooling. By embedding the TS layer into off-the-shelf CNNs, the network learns spatio-temporal features, characteristic to discriminating classes of video sequences. We have investigated various locations in the structure of the CNN network for embedding the TS layers. TS layers are also used on the optical flow data stream extracted from the video which are then combined with the image stream. Experiments have been performed on both UCF101 and HMDB51 datasets and the results indicate that the temporal squeezed representations are compact and meaningful for improving the video classification performance. The proposed temporal squeeze layers can be embedded into a wide range of CNN networks, leading to a video summarization which is optimized with respect to the video classification task.

5. REFERENCES

[1] A. Klaser, M. Marszałek, and C. Schmid, "A spatio-temporal descriptor based on 3d-gradients," in *Proc. of the British Machine Vision Conf. (BMVC)*, 2008, pp. 995–1004.

[2] I. Laptev, "On space-time interest points," *Int. Journal of Comp. Vision*, vol. 64, no. 2-3, pp. 107–123, 2005.

[3] G. Willems, T. Tuytelaars, and L. Van Gool, "An efficient dense and scale-invariant spatio-temporal interest point detector," in *Proc. Eur. Conf. on Comp. Vision (ECCV), Vol. LNCS 5303*, 2008, pp. 650–663.

[4] H. Wang and C. Schmid, "Action recognition with improved trajectories," in *Proc. of IEEE Int. conf. on Comp. vision*, 2013, pp. 3551–3558.

[5] K. Stephens and A. G. Bors, "Modelling of interactions for the recognition of activities in groups of people," *Digital Signal Processing*, vol. 79, pp. 34–46, 2018.

[6] H. Bilen, B. Fernando, E. Gavves, and A. Vedaldi, "Action recognition with dynamic image networks," *IEEE Trans. on Pattern Analysis and Machine Intelligence*, vol. 40, no. 12, pp. 2799–2813, 2018.

[7] K. Simonyan and And. Zisserman, "Two-stream convolutional networks for action recognition in videos," in *Proc. Advances in Neural Inf. Proc. Systems (NIPS)*, 2014, pp. 568–576.

[8] D. Tran, L. Bourdev, R. Fergus, L. Torresani, and M. Paluri, "Learning spatiotemporal features with 3D convolutional networks," in *Proceedings of IEEE Int. Conf. on Comp. Vision (ICCV)*, 2015, pp. 4489–4497.

[9] J. Carreira and A. Zisserman, "Quo vadis, action recognition? a new model and the kinetics dataset," in *Proc. of IEEE Comp. Vision and Pattern Recognition (CVPR)*, 2017, pp. 4724–4733.

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

[11] J. Yue-Hei Ng, M. Hausknecht, S. Vijayanarasimhan, O. Vinyals, R. Monga, and G. Toderici, "Beyond short snippets: Deep networks for video classification," in *Proc. of the IEEE Conf. on Comp. Vision and Pattern Recognition (CVPR)*, 2015, pp. 4694–4702.

[12] J. Hu, L. Shen, and G. Sun, "Squeeze-and-excitation networks," in *Proc. IEEE Conf. on Comp. Vision and Pattern Recog. (CVPR)*, 2018, pp. 7132–7141.

[13] C. Szegedy, S. Ioffe, V. Vanhoucke, and A. A Alemi, "Inception-v4, inception-resnet and the impact of residual connections on learning," in *Proc. AAAI Conf. on Artificial Intelligence*, 2017, pp. 4278–4284.

[14] C. Zach, T. Pock, and H. Bischof, "A duality based approach for realtime TV-L1 optical flow," in *Proc. Joint Pattern Recog. Symp., vol. LNCS 4713*, 2007, pp. 214–223.

[15] K. Soomro, Amir R. Zamir, and M. Shah, "UCF101: A dataset of 101 human actions classes from videos in the wild," *arXiv preprint arXiv:1212.0402*, 2012.

[16] H. Kuehne, H. Jhuang, E. Garrote, T. Poggio, and T. Serre, "HMDB: a large video database for human motion recognition," in *Proc. IEEE Int. Conf. on Comp. Vision (ICCV)*, 2011, pp. 2556–2563.

[17] J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, and L. Fei-Fei, "Imagenet: A large-scale hierarchical image database," in *Proc. IEEE Comp. Vision and Pattern Recognition (CVPR)*, 2009, pp. 248–255.

[18] 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," in *Proc. European Conf. on Computer Vision (ECCV), vol LNCS 9912*, 2016, pp. 20–36.

[19] X. Peng, C. Zou, Y. Qiao, and Q. Peng, "Action recognition with stacked fisher vectors," in *Proc. European Conf. on Computer Vision (ECCV), vol LNCS 8693)*, 2014, pp. 581–595.

[20] 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*, vol. 150, pp. 109–125, 2016.

[21] C. Feichtenhofer, A. Pinz, and A. Zisserman, "Convolutional two-stream network fusion for video action recognition," in *Proc. of the IEEE Conf. on Comp. Vision and Pattern Recognition (CVPR)*, 2016, pp. 1933–1941.

[22] L. Wang, Y. Qiao, and X. Tang, "Action recognition with trajectory-pooled deep-convolutional descriptors," in *Proc. of IEEE Conf. on Comp vision and Pattern Recognition (CVPR)*, 2015, pp. 4305–4314.