MULTI-VIEW FUSION FOR ACTION RECOGNITION IN CHILD-ROBOT INTERACTION

Niki Efthymiou[1,3], *Petros Koutras*[1,3], *Panagiotis Paraskevas Filntisis*[1,3],
Gerasimos Potamianos[2,3], *Petros Maragos*[1,3]

[1] School of E.C.E., National Technical University of Athens, Greece
[2] E.C.E. Department, University of Thessaly, Volos, Greece
[3] Athena Research and Innovation Center, Maroussi, Greece

{nefthymiou,filby}@central.ntua.gr,{pkoutras,maragos}@cs.ntua.gr,gpotam@ieee.org

ABSTRACT

Answering the challenge of leveraging computer vision methods in order to enhance Human Robot Interaction (HRI) experience, this work explores methods that can expand the capabilities of an action recognition system in such tasks. A multi-view action recognition system is proposed for integration in HRI scenarios with special users, such as children, in which there is limited data for training and many state-of-the-art techniques face difficulties. Different feature extraction approaches, encoding methods and fusion techniques are combined and tested in order to create an efficient system that recognizes children pantomime actions. This effort culminates in the integration of a robotic platform and is evaluated under an alluring Children Robot Interaction scenario.

***Index Terms*—** multi-view fusion, action recognition, child-robot interaction

1. INTRODUCTION

Interest in human action recognition remains strong in the computer vision community due to the plethora of its applications such as video context analysis, retrieval, and surveillance. In addition, the continuous evolution in robotics and especially in Human-Robot Interaction (HRI) entails a continuous need for enhancement of perception systems, such as human action recognition. In HRI, recognition of an action ensures that the robot party will be aware of human moves and will act according to them.

As HRI has been evolving, new applications and possibilities are brought forth such as edutainment [2, 3], assisted living [4], or assistance in treating certain disorders such as autism [5, 6]. Current utilization of computer vision in HRI applications usually includes simple hand gesture recognition for robot operation [7]. However, under the context of these applications, it is arguable that there is a need for recognition of more general human actions since the scenario of the interaction is not strict and can change abruptly. Furthermore, due to the wider and more free nature of these applications, problems such as occlusions or poor camera view point can occur more often.

In light of these, in this work, we present a robust multi-view action recognition system under the HRI umbrella, designed to tackle the common limitations that occur during the operation of an assisted robot such as occlusions, reduced camera field of view, and inadequate camera view point. We examine this framework using a specially designed Child Robot Interaction (CRI) task in which children perform pantomime actions. Under this use case, children present spontaneous behavior and an informal way of communication. In addition, the same actions can be performed in a variety of ways and a wide spectrum, further complicating the recognition of actions.

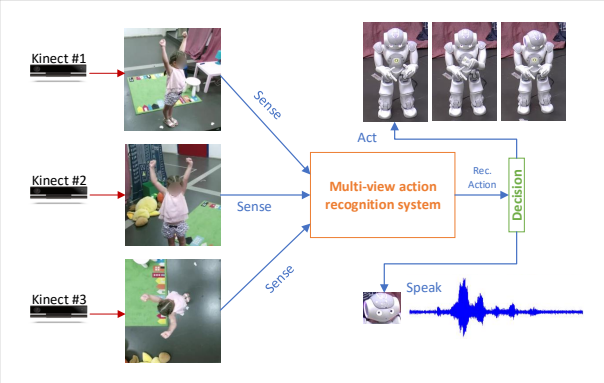

Fig. 1: Multi-view action recognition system for child-robot interaction.

Although human action recognition is a popular problem with many proposed methods [8–13], the requirements of multi-view action recognition differ significantly as it has to take into account both action recognition that results from single views and also the fusion among the resulting information from the different streams [14, 15]. In cross-view action recognition works it is attempted to share knowledge for the action among the different setup views. For example, in [16] a specific view is treated as the target domain and the other views as source domains in order to formulate a cross-view learning framework. In other approaches, the knowledge of actions is transferred from the different views in a single canonical view [17]. In [18] it is proposed to learn view-invariant features robust to view variations using deep models. In the field of multi-view action recognition, a new global representation that is called multi-view super vector has also been proposed in order to enhance recognition performance [19]. Finally, another interesting approach is presented in [20] where it is attempted to transfer the low-level features into a high-level semantic space and a multi-task learning approach for joint action modeling is examined.

In this paper we develop a multi-view action recognition system suitable for CRI. The main contributions of this paper can be summarized as follows: 1) Single-view methods are explored in order to create robust action recognition models for particular users, i.e. children, under difficult tasks with few training data. 2) Methods for the fusion of information from different streams in a multi-view system are proposed to enhance action recognition during CRI. 3) The multi-view action recognition system is integrated in robotic plat-

This work was supported by EU Horizon 2020 project BabyRobot [1], under grant agreement no. 687831.

Fig. 2: Setup of the multiple RGB sensors employed during CRI experiments.

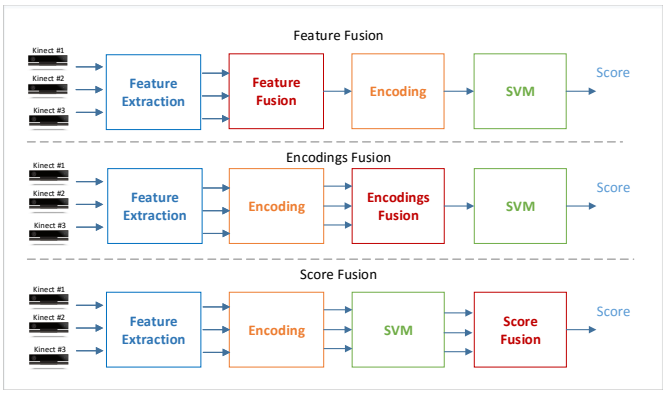

Fig. 3: Multi-view fusion approaches: 1) feature fusion, 2) encodings fusion, 3) score fusion.

forms and evaluated under an interesting CRI scenario that forced children to act spontaneously.

2. MULTI-VIEW ACTION RECOGNITION SYSTEM

In order to investigate optimal techniques for recognizing children action pantomimes during a CRI task, we have employed different combinations of features and descriptors, along with different fusion schemes of multiple RGB camera streams. As we see from Fig. 1 the visual information captured by the multiple sensors is processed by our multi-view action recognition system. Afterwards, the recognized action is forwarded to the robotic agent, i.e., a NAO robot, which subsequently interacts with the child either verbally, by announcing the recognized action, or by performing a similar action. In Fig. 2 we see the experimental setup for the multi-view CRI task.

2.1. Single-view Approaches

In order to design an efficient action recognition system we have explored the possible choices about the visual representations of a video stream. We have experimented with two state-of-the-art approaches in video processing and action recognition: the hand-crafted dense trajectories features and the 3D CNN-based features. The main challenge is to find a representation that is able to be adapted to a new task, such as pantomime actions performed by children, with few training data available.

Dense Trajectories Features: For the first pipeline of our single-view action recognition system, the state-of-the-art Dense Trajectories (DT) [21] features are combined with the Bag of Visual Words (BoVW) encoding framework. In each video frame, dense points are sampled and tracked through time based on a dense optical flow field. The features that are computed along each trajectory are: the Trajectory descriptor [21], Histograms of Oriented Gradients (HOG) [22], Histograms of Optical Flow (HOF) [22], and Motion Boundary Histograms (MBH) [21] computed on both axes (MBHx, MBHy). Encoding of the features using the BoVW and assignment to K=4000 clusters follows in order to form a representation of each video. Videos are classified based on their BoVW representation, using non-linear Support Vector Machines (SVMs) with the χ^2 kernel [23]. In addition, different types of descriptors are combined, by computing distances between their corresponding BoVW histograms and adding the corresponding kernels, see also (5). Since we face multiclass classification problems, we follow the one-against-all approach and select the class with the highest score.

Another variation of the above pipeline employs a combination of DT features and Vector of Locally Aggregated Descriptors (VLAD) [24] encoding of features. In the VLAD approach, each trajectory is assigned to the closest cluster of a vocabulary of $K = 256$. For each of the K clusters, the deviation between the features of the visual word and the features that have been assigned to it, is accumulated. The encoded features that result from VLAD are classified employing linear SVMs. Finally, each video is classified in the highest scored class, as in the BoVW pipeline.

CNN-based Features: The second pipeline of the single-view action recognition system includes feature extraction from a 3D convolutional neural network (CNN) [25, 26].

In a 3D CNN the input to the network is a 3D volume of image frames (a video), while the output of the network is the probabilities of the target classes. This end-to-end schema is used for training the network. Afterwards, we use the network for feature extraction. Between the input and the output of the network intervene: convolutional layers where the input is convolved with 3D kernels, pooling layers where the input is subsampled, and fully connected layers which correspond to the final features used for classification in the final layer. CNN-based features are extracted from these intermediate layers and then fed into an SVM for the final classification (instead of using the probabilities in the output). In this work, we use the network architecture that appeared in [25] and is presented in Fig. 5. Usually, the features that are employed in classification are extracted from the final fully connected layers (FC6 and FC7). However it has also been proposed to extract features from the final pooling or convolutional layer in order to leverage the spatial information included in these layers which is lost in the fully connected layers. In [27, 28] CNN descriptors are extracted from the intermediate layers that contain this spatial information.

The downside of using a C3D network is the large amount of data that is required to avoid overfitting. Corpora used as a benchmark usually include a large amount of data - the ActivityNet database [29] contains 15,410 videos of 200 classes for training, Sports1M [30] includes over 1 million videos of 487 classes, and UCF101 [31] contains 13,320 videos from 101 classes. In order to avoid overfitting, in our case we employ transfer-learning and use a pretrained model on the Sports1M corpus, which we then fine-tune to classify the actions in our database. In addition, since we have limited data, we split each of our videos in 16-long frame clips with the 15 frames overlapping and use these to fine-tune the network, employing a leave-one-out approach with learning rate 0.0001. At feature extraction time, for each 16-clip video we extract features from the FC6, FC7, pool5, and conv5b layers, and average over each clip in order to obtain a descriptor for the whole video.

(a) Digging a hole (b) Swimming

Fig. 4: Two example actions for the collected multi-view database.

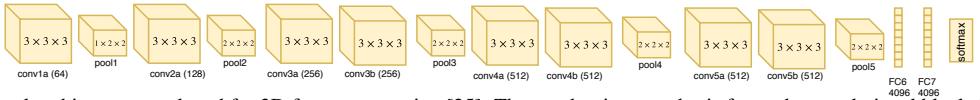

Fig. 5: 3D convolutional architecture employed for 3D feature extraction [25]. The number in parenthesis for each convolutional block denotes the number of filters while the number inside the block denotes the size of the convolution kernel. Fully connected layers both consist of 4096 neurons.

2.2. Multi-view Fusion

In this part we explore different approaches for the fusion of the visual information obtained by the multiple sensors: 1) feature fusion, 2) encodings fusion, and 3) score fusion. We modify the general frameworks of BoVW and VLAD in order to deal with our proposed multi-view approach for action recognition. In Fig. 3 we see the different employed approaches for multi-view fusion.

Feature Fusion: In this method we fuse the visual information in an early stage where we have only low-level D-dimensional feature descriptors $\mathbf{x}_m^i \in \mathbb{R}^D$, i.e., local descriptors alongside $m = 1,\ldots,M_i$ dense trajectory, from each different sensor $i = 1,\ldots,S$. Even though S sensors recorded exactly the same actions, the number of the sample points in each video tracked for getting the trajectories isn't constant as it depends on the optical flow. Thus, we cannot apply a simple concatenation of the feature descriptors and form a new descriptor vector $\tilde{\mathbf{x}}_m$ of size $D \cdot S$. So, we modify the codebook generation approach, which is based on the k-means algorithm, in order to deal with the multi-view data. Given a set of feature descriptors $\mathbf{x}_m^i$, our goal is to partition the feature set into K clusters $\mathbf{D} = [\mathbf{d}_1,\ldots,\mathbf{d}_K]$, where $\mathbf{d}_k \in \mathbb{R}^D$ is the centroid of the k-th cluster. These $\mathbf{d}_k$ are shared between the features of all sensors. Using the notation of [32], if descriptor $\mathbf{x}_m^i$ is assigned to cluster k, then the indicator value $r_{m,i,k} = 1$ and $r_{m,i,\ell} = 0$ for $\ell \neq k$. The optimal $\mathbf{d}_k$ can be found by minimizing the objective function:

$$\min_{\mathbf{d}_k, r_{m,i,k}} \sum_{k=1}^{K} \sum_{i=1}^{S} \sum_{m=1}^{M_i} r_{m,i,k} \|\mathbf{x}_m^i - \mathbf{d}_k\|_2^2. \tag{1}$$

Then we modify the encoding procedures for both the BoVW and VLAD method in order to be applied to multi-view data. For a set of feature descriptors $\mathbf{X}^i = [\mathbf{x}_1^i,\ldots,\mathbf{x}_{N_j}^i]$, which is extracted from the j-th video captured by sensor i, the encoding of $\mathbf{x}_{n_j}^i$ for the BoVW approach using the codebook $\mathbf{D}$ is given by:

$$\mathbf{s}_{n_j}^i(k) = 1, \text{ if } k = \underset{\ell}{\operatorname{argmin}} \|\mathbf{x}_{n_j}^i - \mathbf{d}_\ell\|_2^2, \text{ s.t. } \|\mathbf{s}_{n_j}^i\|_0 = 1. \tag{2}$$

In the case of VLAD, where we keep first order statistics, the encoding of $\mathbf{x}_{n_j}^i$ is:

$$\mathbf{s}_{n_j}^i(k) = [\mathbf{0},\ldots,\mathbf{x}_{n_j}^i - \mathbf{d}_k,\ldots,\mathbf{0}], \quad k = \underset{\ell}{\operatorname{argmin}} \|\mathbf{x}_{n_j}^i - \mathbf{d}_\ell\|_2^2. \tag{3}$$

The global representation $\mathbf{h}$ of the multi-view video using a sum pooling scheme is given by:

$$\mathbf{h} = \sum_{i=1}^{S} \sum_{n_j=1}^{N_j} \mathbf{s}_{n_j}^i. \tag{4}$$

Finally, for the BoVW approach we apply a $L2$ normalization scheme [33] while for the VLAD we follow the intra-normalization strategy proposed in [34].

Encodings Fusion: In this approach we have a different global vector $\mathbf{h}^i$ for each sensor i. This representation could be either an encoding of the dense trajectory features using a different codebook $\mathbf{D}^i$ for each sensor or a feature vector obtained by a different C3D network. For the BoVW encodings we apply the multi-view fusion by adding the χ^2 kernels:

$$K(\mathbf{h}_j, \mathbf{h}_q) = \sum_{i=1}^{S} \sum_{c=1}^{N_c} \exp\left(-\frac{1}{A_c} L\left(\mathbf{h}_j^{c,i}, \mathbf{h}_q^{c,i}\right)\right), \tag{5}$$

where $\mathbf{h}_j^{c,i}$ denotes the BoVW representation of the c-th descriptor of the j-th video captured by sensor i, and A_c is the mean value of χ^2 distances $L(\mathbf{h}_j^{c,i}, \mathbf{h}_q^{c,i})$ between all pairs of training samples from a specific sensor i. On the other hand for the VLAD encodings and the C3D we apply a simple concatenation of the vectors that correspond to the different sensors: $\mathbf{h} = [\mathbf{h}^1,\ldots,\mathbf{h}^S]$.

Score Fusion: For a given sensor i we train a different SVM for all employed classes and obtain the probabilities $\mathbf{P}^i$ as described in [35]. Then we apply a softmax normalization to each sensor's SVM probabilities. Alternatively, in the case that we employ an end-to-end C3D network these probabilities could be obtained from the last softmax layer. For the fusion of the different sensor output probabilities we simply apply an average fusion: $\mathbf{P} = \frac{1}{S}\sum_{i=1}^{S} \mathbf{P}^i$. Finally, we select the class with the highest fused score, following the same approach as in the single sensor case.

3. SYSTEM EVALUATION

3.1. Multi-view Pantomime Actions Dataset

In order to evaluate the methods described in this paper, we collected an in-house database. Our experimental setup consists of a room that was designed as a child's room and three Kinect V2 sensors,

Dense Trajectories (DT) Features							CNN Features			
Enc. / Desc.	Bag-Of-Visual-Words			VLAD			C3D Layers	C3D Network		
	Kinect #1	Kinect #2	Kinect #3	Kinect #1	Kinect #2	Kinect #3		Kinect #1	Kinect #2	Kinect #3
Traj.	63.08	48.62	45.54	60.31	48.61	46.46	conv5b	54.46	52.00	40.92
HOG	39.69	32.00	27.69	39.69	38.15	34.46	pool5	57.23	54.15	42.46
HOF	68.31	56.31	48.62	69.85	63.08	50.46	FC6	**59.38**	54.46	42.77
MBH	70.77	60.92	61.85	72.92	68.62	60.00	FC7	57.85	52.92	42.15
Comb.	73.85	63.38	60.00	**74.15**	69.23	58.46	Comb.	56.92	54.46	44.31
							end-to-end	58.03	52.05	41.87

Table 1: Evaluation of single-view children action recognition with Dense Trajectories and CNN features, using leave-one-out cross-validation.

Dense Trajectories (DT) Features							CNN Features			
Fusion / Desc.	Feature Fusion		Encodings Fusion		Score Fusion		Fusion / C3D Feats.	Feature Fusion	Encodings Fusion	Score Fusion
	BoVW	VLAD	BoVW	VLAD	BoVW	VLAD				
Traj.	59.38	54.15	65.85	64.62	63.02	65.23	conv5b	58.77	61.23	62.46
HOG	43.08	45.54	44.00	50.15	42.60	45.54	pool5	60.31	61.23	63.08
HOF	62.46	65.84	68.31	70.77	67.16	70.15	FC6	60.31	63.08	62.46
MBH	73.85	75.38	74.77	76.31	73.08	74.46	FC7	63.08	63.08	62.15
Comb.	74.46	**77.54**	74.46	75.69	73.08	75.08	Comb.	60.31	61.23	**63.69**
							end-to-end	-	-	61.72

Table 2: Evaluation of multi-view children action recognition using Dense Trajectories and CNN features (leave-one-out cross-validation), using the three different fusion schemes.

		Action Recognition -Training scheme	
Test Set		Adults	Children
Adults	Kinect #1	**86.26**	71.98
	Kinect #2	85.71	65.93
	Kinect #3	76.92	59.89
Children	Kinect #1	57.69	**73.85**
	Kinect #2	54.44	63.38
	Kinect #3	43.96	60.00

Table 3: Evaluation of the single-view module (DT, BoVW) with children and adult action data.

one located at the ceiling facing down to the interaction area and the remaining two at each side of it, as depicted in Fig. 2.

Using this experimental setup we have recorded a total of 39 subjects performing 12 actions: painting a wall, cleaning a window, driving a bus, swimming, dancing, working out, playing the guitar, digging a hole, wiping the floor, ironing a shirt, hammering a nail, reading a book. More specifically, 25 children, from six to ten years old, and 14 adults have been asked to play a pantomime game with the NAO robot. During the game, the human and the robot took turns in performing a pantomime that was depicted on a computer screen while the opposing party was asked to recognize this pantomime. The participants were also asked to perform a random movement to gather sufficient data for background movements. Fig. 4 shows two example actions from the database.

3.2. Single-view and Multi-view Evaluation Results

Table 1 presents average accuracy results (%) for the 12 pantomime actions and the background model performed by the children of the database, using the single-view approach and leave-one-out cross-validation. The results indicate that the combination of the DT features performs slightly better for both BoVW and VLAD encodings. Additionally, the VLAD vector further improves performance, since it encodes rich information about the visual words distribution. Regarding the performance of the CNN features we see that they have a degraded performance compared to the hand-crafted features. The reason is that children actions are very different from the actions that are included in the state-of-the-art databases, e.g., UCF101, Sports1M, and so the simple fine-tuning of pretrained networks did not help. In addition, the end-to-end training of a CNN network requires a huge amount of children data which is not a realistic scenario. However, we see that the best result in most cases is achieved employing the feature from the FC6 layer.

For the further evaluation of the single-view system, we have trained separate models using as training sets: a) the children action data, and b) the adult action data. From Table 3 we can observe that when training and test data come from the same age group, the recognition accuracy is high. Note that the performance in children data is significantly lower even if we use the same approach (DT features) and train on data of the same task performed by adults. This result backs up our previous remark about the performance of fine-tuned CNN features.

In Table 2 we present the evaluation results of the employed multi-view fusion methods. We can see that the fusion schemes improve the performance of the corresponding single-view method in almost all cases. Our best result 77.54% is achieved by the VLAD encoded DT features under the feature fusion scheme. In general, this early fusion scheme achieved the best performance when we used the DT features, while the score fusion has the best performance in the case we employ the C3D features. This could be explained by the fact that dense trajectories capture spatio-temporal local information, which is further encoded to form a global representation, while C3D captures a global representation of a 16-frame clip which is then averaged along the whole video. However, we mention that the employed fusion schemes achieve to improve the performance of the original single-view C3D approach in most cases.

4. CONCLUSIONS

In this work we addressed the problem of action recognition in CRI environments. We proposed a mutli-view approach that improves the performance of the single-view methods in most cases. Moreover, we explored different feature extraction approaches and in our experimental evaluation we observed that traditional dense trajectories can perform much better than 3D CNN features in cases where we have very different tasks (children actions) than those in large state-of-the-art databases. Finally we integrated and evaluated the proposed system according to a proposed CRI scenario, achieving a recognition accuracy of 77.54% for children pantomime actions. As future work we intend to investigate alternative ways to efficiently transfer deep learning knowledge from large datasets to specific and challenging problems in HRI.

5. REFERENCES

[1] "BabyRobot project," http://babyrobot.eu.

[2] M. Fridin and M. Belokopytov, "Acceptance of socially assistive humanoid robot by preschool and elementary school teachers," *Computers in Human Behavior*, vol. 33, pp. 23–31, 2014.

[3] R. Ros and Y. Demiris, "Creative dance: An approach for social interaction between robots and children," in *Proc. Workshop on Human Behavior Understanding*, 2013.

[4] A. Zlatintsi, I. Rodomagoulakis, V. Pitsikalis, P. Koutras, N. Kardaris, X. Papageorgiou, C. Tzafestas, and P. Maragos, "Social human-robot interaction for the elderly: two real-life use cases," in *Proc. ACM HRI*, 2017.

[5] B. Robins, K. Dautenhahn, and P. Dickerson, "From isolation to communication: a case study evaluation of robot assisted play for children with autism with a minimally expressive humanoid robot," in *Proc. ACHI*, 2009.

[6] P. G. Esteban, B. Paul, B. Tony, B. Erik, C. Haibin, C. Hoang-Long, C. Mark, C. Cristina, D. Daniel, A. De Beir, Y. Fang, Z. Ju, J. Kennedy, H. Liu, A. Mazel, A. Pandey, K. Richardson, S. Senft, S. Thill, G. Van de Perre, B. Vanderborght, D. Vernon, Y. Hui, and Y. Ziemke, "How to build a supervised autonomous system for robot-enhanced therapy for children with autism spectrum disorder," *Paladyn: Journal of Behavioral Robotics*, pp. 18–38, 2017.

[7] G. Canal, S. Escalera, and C. Angulo, "A real-time human-robot interaction system based on gestures for assistive scenarios," *Computer Vision and Image Understanding*, vol. 149, pp. 65–77, 2016.

[8] K. Simonyan and A. Zisserman, "Two-stream convolutional networks for action recognition in videos," in *Proc. NIPS*, pp. 568–576. 2014.

[9] L. Wang, Y. Qiao, and X. Tang, "Action recognition with trajectory-pooled deep-convolutional descriptors," in *Proc. CVPR*, 2015.

[10] A. Diba, V. Sharma, and L. Van Gool, "Deep temporal linear encoding networks," in *Proc. CVPR*, 2017.

[11] N. Kardaris, V. Pitsikalis, E. Mavroudi, and P. Maragos, "Introducing temporal order of dominant visual word sub-sequences for human action recognition," in *Proc. ICIP*, 2016.

[12] Y. Peng, Y. Zhao, and J. Zhang, "Two-stream collaborative learning with spatial-temporal attention for video classification," *IEEE Trans. on Circuits and Systems for Video Technology*, 2018.

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

[14] A. B. Sargano, P. Angelov, and Z. Habib, "Human action recognition from multiple views based on view-invariant feature descriptor using support vector machines," *Applied Sciences*, vol. 6, no. 10, pp. 309, 2016.

[15] T. D. Le, T. O. Nguyen, and T. H. Tran, "Improving multi-view human action recognition with spatial-temporal pooling and view shifting techniques," in *Proc. SoICT*, 2017.

[16] W. Nie, A. Liu, W. Li, and Y. Su, "Cross-view action recognition by cross-domain learning," *Image and Vision Computing*, vol. 55, pp. 109–118, 2016.

[17] H. Rahmani and A. Mian, "Learning a non-linear knowledge transfer model for cross-view action recognition," in *Proc. CVPR*, 2015.

[18] Y. Kong, Z. Ding, J. Li, and Y. Fu, "Deeply learned view-invariant features for cross-view action recognition," *IEEE Trans. Image Processing*, vol. 26, no. 6, pp. 3028–3037, 2017.

[19] Z. Cai, L. Wang, X. Peng, and Y. Qiao, "Multi-view super vector for action recognition," in *Proc. CVPR*, 2014.

[20] T. Hao, D. Wu, Q. Wang, and J. Sun, "Multi-view representation learning for multi-view action recognition," *Journal of Visual Communication and Image Representation*, vol. 48, pp. 453–460, 2017.

[21] H. Wang, A. Klaser, C. Schmid, and C.L. Liu, "Action recognition by dense trajectories," in *Proc. CVPR*, 2011.

[22] I. Laptev, M. Marszalek, C. Schmid, and B. Rozenfeld, "Learning realistic human actions from movies," in *Proc. CVPR*, 2008.

[23] H. Wang, M. M. Ullah, A. Kläser, I. Laptev, and C. Schmid, "Evaluation of local spatio-temporal features for action recognition," in *Proc. BMVC*, 2009.

[24] H. Jégou, M. Douze, C. Schmid, and P. Pérez, "Aggregating local descriptors into a compact image representation," in *Proc. CVPR*, 2010.

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

[26] D. Tran, J. Ray, Z. Shou, S. F. Chang, and M. Paluri, "Convnet architecture search for spatiotemporal feature learning," *arXiv preprint arXiv:1708.05038*, 2017.

[27] X. Peng and C. Schmid, "Encoding feature maps of CNNs for action recognition," in *Proc. CVPR, THUMOS Challenge 2015 Workshop*, 2015.

[28] Z. Xu, Y. Yang, and A. G. Hauptmann, "A discriminative CNN video representation for event detection," in *Proc. CVPR*, 2015.

[29] F. C. Heilbron, V. Escorcia, B Ghanem, and J. C. Niebles, "ActivityNet: A large-scale video benchmark for human activity understanding," in *Proc. CVPR*, 2015.

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

[31] 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.

[32] 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.

[33] F. Perronnin, J. Sánchez, and T. Mensink, "Improving the Fisher kernel for large-scale image classification," in *Proc. ECCV*, 2010.

[34] R. Arandjelovic and A. Zisserman, "All about VLAD," in *Proc. CVPR*, 2013.

[35] C.C. Chang and C. J. Lin, "LIBSVM: a library for support vector machines," *ACM Trans. Intelligent Systems and Technology*, vol. 2, no. 3, pp. 27, 2011.