EFFECT OF WAVELET AND HYBRID CLASSIFICATION ON ACTION RECOGNITION

Eman Mohammadi, Q. M. Jonathan Wu, Yimin Yang, and Mehrdad Saif

Department of Electrical and Computer Engineering, University of Windsor, Ontario, Canada
Emails: {moham12b, jwu, eyyang, msaif}@uwindsor.ca

ABSTRACT

Any action dataset may contain similar classes such as running, walking and jogging. Therefore, equivalent probabilities may be provided for different classes upon action classification. In this case, the classifier cannot indubitably assign a class to a given sample. To address this problem, we propose a new hybrid classifier to automatically compress the features and classify them using SVM with polynomial or sigmoid kernels. Furthermore, we hypothesize that motion saliency detection can strength the power of motion feature extraction in the bag of visual words framework (BoVW). To this end, we evaluate the effect of 3D-discrete wavelet transform (3D-DWT), as the preprocessing step, on motion feature extraction. The experimental results show that the proposed framework achieves promising results on KTH, Weizmann, and URADL datasets, and outperforms recent state-of-the-art approaches.

Index Terms— Action recognition, Hybrid classification, Data Compression, Motion-based features

1. INTRODUCTION

Action recognition is one of the crucial research areas with multitude applications such as intelligent surveillance, human-computer interaction, and video annotation. Action classification is tremendously challenging for computers due to the complexity of video data and the subtlety of human actions [1–3]. Many of the successful action recognition frameworks employ low-level features with bag of visual word (BoVW) framework [4]. The pipeline of BoVW contains five major sections: feature extraction, feature pre-processing, codebook generation, feature encoding, and classification [5].

The most widely used local feature extraction and encoding methods are evaluated and analysed in [4–6]. The best current frameworks to human action recognition rely on dense trajectory features [7] that are then encoded by Fisher vector (FV) method. The combination of dense trajectories and Fisher vector encoding was first proposed in [8] and obtained state-of-the-art results on several action datasets. The approach was further modified and enhanced in [9] by employing the stacked FVs. Moreover, Jain et al. [10] showed that the accuracy of this approach could be improved by modelling the context of an action. Deep learning networks have also been employed to represent temporal features. For instance, Karpathy et al. [11] presented convolutional neural networks (CNNs) to train and classify of 1 million YouTube videos. In addition, Wang et al. [12] proposed a framework to combine the CNN learned features and hand-crafted features. However, creation of deep networks for video-based action recognition is considered as a challenging task since the video-based classification is different from image classification. A large portion of deep features may be irrelevant to actions. For example, deep networks may extract some features while moving the camera through capturing frames. The irregular distribution and aberrant features of an action make it difficult to directly apply deep networks for action recognition. Therefore, we further employ local features in our framework for action recognition.

Much less research has been done on pre-processing of image sequences and classification stages in the BoVW model. We hypothesis that motion segmentation, as the pre-processing step, can significantly improve the local feature performance. To this end, we employ 3D-discrete wavelet transform (3D-DWT) [13] to segment the moving objects in videos before local feature extraction. We evaluate a set of thresholding values to efficiently detect the motion saliency map from image sequences.

For the classification stage, the extreme learning machine (ELM) and linear support vector machine (SVM) are the most popular and widely used action classifiers in recent literatures [4–7, 14, 15]. However, action classification can be very challenging for similar classes in a dataset. For example, equivalent probabilities may be provided for running, jogging and walking classes while classifying the samples of KTH dataset. The classifier is not capable of making the final decision indubitably when equivalent probabilities are generated for different classes. To this end, we propose a hybrid classifier to automatically compress the encoded features and select the best SVM kernel for classification.

As depicted in Fig. 1, the proposed hybrid classifier is composed of three layers. We hypothesis that huge vector of encoded features may transfer redundant info and outliers to classifier. Thus, in case of providing equivalent probabilities in the first layer, we compress the encoded features to d dimension in the second layer, and then pass the compressed features to the third layer. The major motivation behind data compression is to extract the most useful and prominent information while reducing the dimension of data. In the third layer, the system chooses the best kernel among polynomial and sigmoid functions for SVM classifier. The experimental results of the proposed framework show a significant improvement over traditional SVM classifier for action recognition. In summary, this paper makes the following contributions.

1) Applying 3D-DWT on videos to efficiently extract motion saliency maps. Different thresholding values are evaluated to extract the best motion saliency map for local feature extraction. The effect of 3D-DWT on motion-based features is evaluated in this paper.

2) Proposing a hybrid classifier to automatically compress the extracted features and select the best SVM kernel for action classification.

The remainder of this paper is organized as follows. Section 2 presents the architecture, formulation, and implementation of our framework for action recognition in detail. The experiments and results are described in Section 3. Finally, we conclude the paper in Section 4.

978-1-5090-2175-8/17/$31.00 ©2017 IEEE

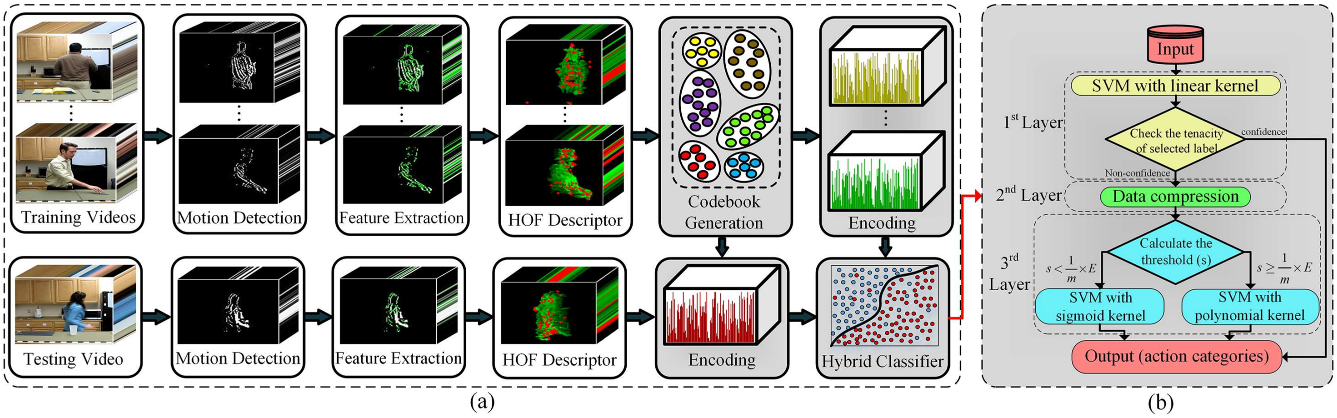

Fig. 1. (a) Proposed framework using motion detection and hybrid classification for action recognition. (b) Structure of the proposed hybrid classifier. s denotes $\max_1(P) - \max_2(P)$ where P is the set of probabilities which are generated by linear SVM in the first layer. m denotes the number of classes and E indicates the constant threshold in the third layer of hybrid classifier.

2. PROPOSED FRAMEWORK

In this paper, we employ the 3D-discrete wavelet transform (3D-DWT), as a preprocessing step, in the BoVW model. Moreover, we propose a hybrid classification system to confidently classify human actions. The proposed framework is depicted in Fig. 1, and discussed in the following subsections.

2.1. Preprocessing and Feature Extraction

The 3D-DWT can be considered as a combination of three 1D-DWT in the x, y and t directions [13]. It is composed of high-pass and low-pass filters that perform a convolution of filter coefficients and input pixels. After a one-level of 3D discrete wavelet transform, the volume of image sequences is decomposed into 8 sub-signals. We employ the sub-band which is generated with high-pass filters in three directions. We first resize the image sequences to 500×500 and then apply 3D-DWT on resized videos. The extracted sub-signal, which is composed of high-pass filters to each direction, is converted to video with 10 frames per second. Then, the motion saliency map is generated by applying a threshold of (θ) on created pixels. We evaluate different thresholding values to provide the best motion saliency map.

We hypothesize that only the motion features can provide enough information to recognize actions from the videos which are captured with static cameras. To this end, the dense trajectory features [7] are extracted from preprocessed videos and then described by histogram of optical flow (HOF). The HOF describes the local motion by defining a grid around the encompassing space-time area and computing a histogram of optical flow for each cell of the grid [16]. The HOF description is performed faster on motion saliency maps compare to the raw videos. The described features are encoded by Fisher vector [17], and then fed to the proposed hybrid classifier. The dimension of the encoded features is $2DK$ where D is the dimension of the initial features and K is the codebook size while encoding the features. We further call the extracted motion features as W-HOF since the wavelet is employed to extract the motion saliency maps before HOF extraction.

2.2. Hybrid Classification

Equivalent probabilities may be provided for similar action categories while classifying a given sample. In this case, the classifier cannot confidently categorize the actions. In order to address this problem, we propose a novel hybrid classifier to use the appropriate SVM kernel when equivalent probabilities are generated by linear SVM for multiple classes. Our approach is composed of the following three layers.

First layer. The encoded features are fed to the SVM classifier with linear kernel. The one-vs-all strategy is followed and the linear SVM is trained for multiple classes. For a given sample, the set of generated probabilities, $P = [p_1, \ldots, p_m]$, is checked and evaluated to figure out whether the maximum probability is confidently assigned to its class. The thresholding value τ is obtained as

$$\tau = \max_1(P) - \max_2(P) + \tfrac{1}{m} \qquad (1)$$

where m is the number of classes and P is the set of probabilities, generated in one-vs-all mode. The $\max_1(P)$ and $\max_2(P)$ are the first and second maximum values in set of probabilities which are generated by linear SVM. In case of providing $\tau \leq \max_1(P)$, the result of linear SVM is considered as non-confident and the features are passed to the second layer. Otherwise, the final decision is made based on the $\max_1(P)$ which is generated by linear SVM.

Second layer. The double-layer net with sub-network nodes (DL-SNN) [18] is employed to generate the compressed version of encoded features. The major motivation behind the usage of DL-SNN is to extract the most useful features and remove the redundant info from data. The features are compressed to d dimension and then transferred to the third layer.

Third layer. The experiments demonstrate that SVM with sigmoid and polynomial kernels obtain different recognition performances based on the compressed features. Therefore, in the third layer, the SVM classifier with polynomial or sigmoid kernels is adopted to classify the compressed features which are inherited from the second layer. The sigmoid and polynomial kernels are selected based on the following conditions:

$$\begin{array}{ll} Sigmoid: & \max_1(P) - \max_2(P) < \frac{1}{m} \times E \\ Polynomial: & \max_1(P) - \max_2(P) \geq \frac{1}{m} \times E \end{array} \qquad (2)$$

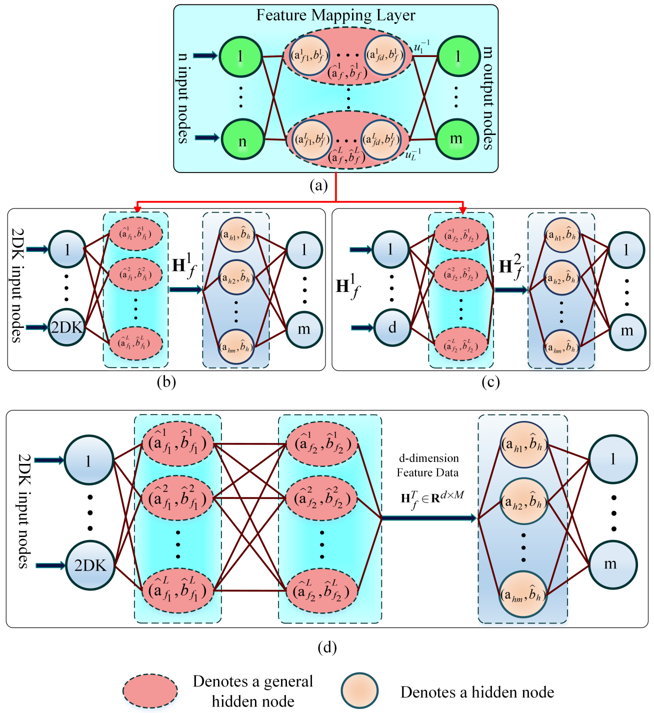

Fig. 2. Structure of DL-SNN for compressing the encoded features. (a) demonstrates the feature mapping layer. (b) and (c) show the first and second networks for compressing the original data in two stages. (d) shows the combination of the first and second stages in the multi-layer network including two feature mapping layers.

where E denotes the constant threshold in classifying the samples of three employed datasets. It is worth pointing out that we train three models during the training stage. the first model is created by SVM with linear kernel over the original encoded features. The second and third models are created by SVM with polynomial and sigmoid kernels over the compressed encoded features. The three models are trained and evaluated by libsvm library [19]. During training of three models, the parameter C, as the cost, is set to 100 and the rest of the parameters remain as defaults in libsvm package [19].

2.2.1. Compression Stage

The DL-SNN is composed of general nodes formed by several hidden nodes to compressing features (see Fig. 2). The number of general nodes and output dimension are independent while the number of hidden nodes in each general neuron must be equal to the dimension of outputs (m). The optimal general parameters are generated in feature mapping layer using the inverse of the activation function. The following five steps are performed to provide the optimal feature set [18].

1) Randomly generate the initial general node of the feature mapping layer , by setting $j = 1$, as

$$\mathbf{H}_f^j = \mathbf{g}\left(\hat{\mathbf{a}}_f^j . \mathbf{x} + \hat{\mathrm{b}}_f^j\right), \left(\hat{\mathbf{a}}_f^j\right)^T . \hat{\mathbf{a}}_f^j = \mathbf{I}, \left(\hat{\mathrm{b}}_f^j\right)^T . \hat{\mathrm{b}}_f^j = 1 \qquad (3)$$

where $\mathbf{H}_f^j$ is the current feature data, and $\hat{\mathbf{a}}_f^j \in \mathbf{R}^{d \times 2DK}$, $\hat{\mathrm{b}}_f^j \in \mathbf{R}$ are the orthogonal random weight and bias of feature mapping layer.

2) Calculate the parameters in the learning layer based on the sigmoid activation function ($\mathbf{g}$) for any continuous desired outputs ($\mathbf{y}$),

$$\begin{aligned} \hat{\mathbf{a}}_h &= \mathbf{g}^{-1}(u_{2DK}(\mathbf{y})) \cdot \left(\mathbf{H}_f^j\right)^{-1} \quad, \hat{\mathbf{a}}_h^j \in \mathbf{R}^{d \times m} \\ \hat{b}_h &= \sqrt{\mathrm{mse}\left(\hat{\mathbf{a}}_h^j \cdot \mathbf{H}_f^j - \mathbf{g}^{-1}(u_{2DK}(\mathbf{y}))\right)} \ , \hat{b}_{2DK}^j \in \mathbf{R} \\ \mathbf{g}^{-1}(\cdot) &= -\log(\frac{1}{(\cdot)} - 1) \quad \text{if } \mathbf{g}(\cdot) = 1/(1 + e^{-(\cdot)}) \end{aligned} \qquad (4)$$

where $\mathbf{H}^{-1} = \mathbf{H}^T(\frac{C}{\mathbf{I}} + \mathbf{H}\mathbf{H}^T)^{-1}$ while C is a positive value, u_{2DK} is a normalized function $u_{2DK}(\mathbf{y}) : \mathbf{R} \to (0, 1]$, and $\mathbf{g}^{-1}$ and u_{2DK}^{-1} represent reverse functions.

3) Update the output error as $\mathbf{e}_j = \mathbf{y} - u_{2DK}^{-1}\mathbf{g}(\mathbf{H}_f^j, \hat{\mathbf{a}}_h, \hat{b}_h)$, and obtain the error feedback data as

$$\mathbf{P}_j = \mathbf{g}^{-1}(u_{2DK}(\mathbf{e}_j)) \cdot (\hat{\mathbf{a}}_h)^{-1} \qquad (5)$$

4) Update the feature data as $\mathbf{H}_f^j = \sum_{l=1}^{j} u_l^{-1}\mathbf{g}(\mathbf{x}, \hat{\mathbf{a}}_f^l, \hat{b}_f^l)$ by setting $j = j+1$ and adding a new general node $\hat{\mathbf{a}}_f^j, \hat{b}_f^j$ in the feature mapping layer by

$$\begin{aligned} \hat{\mathbf{a}}_f^j &= \mathbf{g}^{-1}(u_j(\mathbf{P}_{j-1})) \cdot \mathbf{x}^{-1} \quad, \hat{\mathbf{a}}_f^j \in \mathbf{R}^{d \times 2DK} \\ \hat{b}_f^j &= \sqrt{\mathrm{mse}(\hat{\mathbf{a}}_f^j \cdot \mathbf{x} - \mathbf{P}_{j-1})} \ , \hat{b}_f^j \in \mathbf{R} \end{aligned} \qquad (6)$$

5) Repeat steps 2 to 4 for $L-1$ times. It is worth to mention that a new general node is added to the existing network when repeating steps 2 to 4 once. The parameters $\{\hat{\mathbf{a}}_f^j, \hat{b}_f^j\}_{j=1}^{L}$ are *optimal projecting parameters* and the feature data $\mathbf{H}_f^L = \sum_{j=1}^{L} u_j^{-1}\mathbf{g}(\mathbf{x}, \hat{\mathbf{a}}_f^j, \hat{b}_f^j) = \mathbf{H}_f^*$ are the *optimal feature data*.

The DL-SNN can be used as a multi-layer network. The multi-layer network provides a better general performance than double-layer structure. In the multi-layer strategy, the input data is transformed into multi-layers, and the input raw data is converted into d-dimensional space using multitude feature mapping layers. As depicted in Fig. 2(d), given a training set $\{(\mathbf{x}_i, \mathbf{y}_i)\}_{i=1}^{M} \subset \mathbf{R}^{2DK} \times \mathbf{R}^m$, the compressed features are represented as $\mathbf{H}_f^T = \sum_{i=1}^{L} \mathbf{g}(\mathbf{H}_f^T \cdot \hat{\mathbf{a}}_f^i + \hat{b}_f^i)$ where $\mathbf{H}_f^T$ is the output of the second layer in the multi-layer network.

3. EXPERIMENTS AND RESULTS

This section describes the employed datasets, effect of the 3D-DWT and hybrid classification on action recognition, and the experimental results of our proposed approach compared with the recent state-of-the-are methods.

3.1. Datasets

Weizmann dataset [20]. This dataset contains 90 videos and 10 classes of simple actions. The evaluation of Weizmann is performed by leave one out cross validation.

URADL dataset [21]. This is a high resolution dataset of complicated actions. It contains 10 classes and 150 videos. The 10-fold cross validation has been employed to evaluate this dataset.

KTH dataset [22]. This dataset contains six types of human actions. The evaluation of KTH dataset is perfromed based on 192 training and 216 testing samples as described in [22].

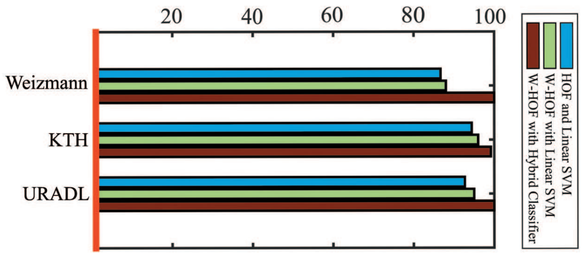

Fig. 3. The recognition performance of HOF and W-HOF based on the linear SVM and proposed hybrid classifier on three employed datasets.

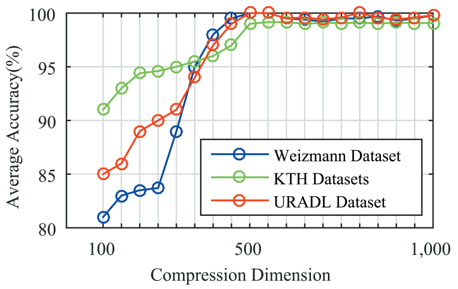

Fig. 4. Evaluation of a set of dimensions for compressing the features at the second layer of hybrid classifier.

All the videos of employed datasets are captured with static cameras and homogeneous backgrounds. The KTH and Weizmann datasets contain simple actions such as running and walking. However, the URADL dataset contains more complex actions such as writing on board and drinking water. The same experimental settings are kept for training and testing stages over each dataset.

3.2. Effect of 3D-DWT on motion features

As depicted in Fig. 3, the W-HOF provides a good performance on three employed datasets. We evaluate the W-HOF performance using linear SVM and proposed hybrid classifier. In both cases, the results are boosted compare to the HOF features which are trained by linear SVM. It shows that the described features by W-HOF delivers advanced information to the classification stage. Thus, 3D-DWT can be considered as a powerful option to extract motion saliency maps before HOF description.

We evaluate a set of thresholding values to extract motion saliency maps from the transformed data. Based on the experiments, 200 is considered as the best thresholding value to provide optimal motion saliency maps for HOF description.

3.3. Evaluation of hybrid classifier

The encoded features are compressed to d dimension at the second layer of hybrid classifier. We evaluate different compression dimensions to efficiently compress the data. As shown in Fig. 4, compressing the features to 500 is considered as the best option for three datasets. It should be noted that the double-layer net with subnetwork nodes (DL-SNN) is not sensitive to the parameters of the networks. Thus, we can select the parameters randomly without affecting the generalization performance in the learning process.

Table 1. Comparison of our results to the state-of-the-arts

Dataset	Method	Recognition Rate
Weizmann	Cao et al. [23]	99.6%
	Lei et al. [24]	89.2%
	Samanta et al. [25]	90.0%
	Sushma et al. [26]	95.55
	Proposed Framework	**100.00%**
KTH	Cao et al. [23]	92.0%
	Lei et al. [24]	93.97%
	Samanta et al. [25]	94.7%
	Barrett et al. [27]	94.9%
	Proposed Framework	**98.00%**
URADL	Prest et al. [28]	92%
	Bilibski et al. [29]	94.7%
	Wang et al. [7]	96%
	Eman et al. [30]	96.6%
	Proposed Framework	**100.00%**

The optimized thresholding constant (E) in Eq. 2, is considered as 2.5 for three employed datasets. As shown in Fig. 3, the proposed hybrid classifier outperforms the traditional SVM while classifying the W-HOF features. For the URADL and Weizmann datasets, all the samples are automatically fed to the SVM with sigmoid kernel in the third layer of hybrid classifier. However, for the KTH dataset, some of the samples which provide equivalent probabilities are automatically classified by sigmoid or polynomial kernels in the third layer. Moreover, the classification of boxing and hand-waving samples is performed confidently with linear SVM at the first layer of hybrid classifier.

3.4. Results

In Table 1, we further compare our results with several state-of-the-art approaches. The proposed framework achieves 100%, 98%, and 100% accuracy for Weizmann, KTH, and URADL datasets, and outperforms the state-of-the-art methods. The obtained results demonstrate that the compression of encoded features can enhance the recognition performance in the hybrid classifier. This is due to the automatical usage of polynomial or sigmoid kernels in the third layer of hybrid classifier. The sigmoid and polynimial kernels are mainly very effective while training a data with moderate feature dimenssion.

We conclude that the encoded features may contain outliers and redundant info which make the classification more challenging and time-consuming. And the employed DL-SNN provides optimal compressed features for action recognition in our hybrid classifier.

4. CONCLUSION

This paper evaluates the effect of 3D-DWT on motion features and proposes an efficient hybrid classifier for action recognition. The experimental results show that motion saliency maps, which are obtained by 3D-DWT, are capable of maturing the motion feature extraction for action recognition. Furthermore, it is shown that the proposed hybrid classifier is capable of leveraging the linear, sigmoid and polynomial kernels in SVM classifier. The results show that the compression of encoded features can enhance the recognition performance in hybrid classifier. The experimental results demonstrate that the proposed framework achieves promising results compared with the state-of-the-art approaches.

5. REFERENCES

[1] C. Feichtenhofer, A. Pinz, and R. P. Wildes, "Dynamically encoded actions based on spacetime saliency," in *Proc. CVPR*, 2015, pp. 2755–2764.

[2] B. Fernando, E. Gavves, J. M. Oramas, A. Ghodrati, and T. Tuytelaars, "Modeling video evolution for action recognition," in *Proc. CVPR*, 2015, pp. 5378–5387.

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

[4] X. Wang, L. Wang, and Y. Qiao, "A comparative study of encoding, pooling and normalization methods for action recognition," in *Proc. ACCV*, 2012, pp. 572–585.

[5] 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.

[6] H. Wang, M. Muneeb, A. Klser, I. Laptev, and C. Schmid, "Evaluation of local spatio-temporal features for action recognition," in *Proc. BMVC*, 2009, pp. 124.1–124.11.

[7] H. Wang, A. Klser, C. Schmid, and L. Cheng-Lin, "Action recognition by dense trajectories," in *Proc. CVPR*, Jun. 2011, pp. 3169–3176.

[8] D. Oneata, J. Verbeek, and C. Schmid, "Action and event recognition with fisher vectors on a compact feature set," in *Proc. ICCV*, 2013, pp. 1817–1824.

[9] X. Peng, L. Wang, Z. Cai, and Y. Qiao, "Action and gesture temporal spotting with super vector representation," in *Workshop at ECCV*, 2014, pp. 518–527.

[10] M. Jain, J. C. van Gemert, and C. G. Snoek, "What do 15,000 object categories tell us about classifying and localizing actions?" in *Proc. CVPR*, 2015, pp. 46–55.

[11] 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, pp. 1725–1732.

[12] L. Wang, Y. Qiao, and X. Tang, "Action recognition with trajectory-pooled deep-convolutional descriptors," in *Proc. CVPR*, 2015, pp. 4305–4314.

[13] M. R. Tripathy, K. Sachdeva, and R. Talhi, "3d discrete wavelet transform vlsi architecture for image processing," in *Proc. PIERS*, 2009, pp. 1569–1573.

[14] H. Wang and C. Schmid, "Action recognition with improved trajectories," in *Proc. ICCV*, Dec. 2013, pp. 3551–3558.

[15] R. Minhas, A. A. Mohammed, and Q. J. Wu, "Incremental learning in human action recognition based on snippets," *IEEE Trans. Circ. Sys. for Vid. Tech.*, vol. 22, no. 11, pp. 1529–1541, 2012.

[16] I. Laptev, M. Marszaek, C. Schmid, and B. Rozenfeld, "Learning realistic human actions from movies," in *Proc. CVPR*, Aug. 2008, pp. 1–8.

[17] H. Jegou, M. Douze, C. Schmid, and P. Perez, "Aggregating local descriptors into a compact image representation," in *Proc. CVPR*, Jun. 2010, pp. 3304–3311.

[18] Y. Yang and Q. M. J. Wu, "Multilayer extreme learning machine with subnetwork nodes for representation learning," *IEEE Trans. Cybern.*, vol. 46, no. 11, pp. 2570–2583, Nov. 2016.

[19] C.-C. Chang and C.-J. Lin, "LIBSVM: A library for support vector machines," *ACM Trans. Intel. Sys. Tech.*, vol. 2, pp. 27:1–27:27, 2011.

[20] L. Gorelick, M. Blank, E. Shechtman, M. Irani, and R. Basri, "Actions as space-time shapes," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 29, no. 12, pp. 2247–2253, 2007.

[21] M. Ross, P. Chris, and K. Henry, "Activity recognition using the velocity histories of tracked keypoints," in *Proc. ICCV*, 2009.

[22] C. Schuldt, I. Laptev, and B. Caputo, "Recognizing human actions: a local svm approach," in *Proc. ICPR*, 2004, pp. 32–36.

[23] X.-Q. Cao and Z.-Q. Liu, "Type-2 fuzzy topic models for human action recognition," *IEEE Trans. Fuzzy Sys.*, vol. 23, no. 5, pp. 1581–1593, 2015.

[24] J. Lei, G. Li, J. Zhang, Q. Guo, and D. Tu, "Continuous action segmentation and recognition using hybrid convolutional neural network-hidden markov model model," *IET Comp. Vis.*, 2016.

[25] S. Samanta and B. Chanda, "Space-time facet model for human activity classification," *IEEE Trans. Multimedia*, vol. 16, no. 6, pp. 1525–1535, 2014.

[26] S. Bomma and N. M. Robertson, "Joint classification of actions with matrix completion," in *Proc. ICIP*, 2015, pp. 2766–2770.

[27] D. P. Barrett and J. M. Siskind, "Action recognition by time-series of retinotopic appearance and motion features," *IEEE Trans. Circ. Sys. Vid. Tech.*, vol. 26, no. 12, pp. 2250–2263, 2016.

[28] A. Prest, V. Ferrari, and C. Schmid, "Explicit modeling of human-object interactions in realistic videos," *Pat. Recog.*, vol. 35, no. 4, pp. 835–848, Apr. 2013.

[29] P. Bilinski and F. Bremond, "Video covariance matrix logarithm for human action recognition in videos," in *Proc. IJCAI*, Jul. 2015, pp. 2140–2147.

[30] E. Mohammadi, Q. M. J. Wu, and M. Saif, "Human activity recognition using an ensemble of support vector machines," in *Proc. HPCS*, Jul. 2016, pp. 549–554.