Computational Intelligence Approaches for Digital Media Analysis and Description

Alexandros Iosifidis, Anastasios Tefas, and Ioannis Pitas

Department of Informatics
Aristotle University of Thessaloniki
54124 Thessaloniki, Greece
{aiosif,tefas,pits}@aiia.csd.auth.gr

Abstract. This paper provides an overview of recent research efforts for digital media analysis and description. It focuses on the specific problem of human centered video analysis for activity and identity recognition in unconstrained environments. For this problem, some of the state-of-the-art approaches for video representation and classification are described. The presented approaches are generic and can be easily adapted for the description and analysis of other semantic concepts, especially those that involve human presence in digital media content.

Keywords: Digital Media Analysis, Digital Media Description, Human Action Recognition, Video Representation, Video Classification

1 Introduction

Recent advances in technological equipment, like digital cameras, smart-phones, etc., have led to an increase of the available digital media, e.g., videos, captured every day. Moreover, the amount of data captured for professional media production (e.g., movies, special effects, etc) has dramatically increased and diversified using multiple sensors (e.g., 3D scanners, multi-view cameras, very high quality images, motion capture, etc), justifying the digital media analysis as a big data analysis problem. As expected, most of these data are acquired in order to describe human presence and activity and are exploited either for monitoring (visual surveillance and security) or for personal use and entertainment. Basic problems in human centered media analysis are face recognition, facial expression recognition and human activity recognition. According to YouTube statistics[1], 100 hours of video are uploaded by the users every minute. Such a data growth, as well as the importance of visual information in many applications, has necessitated the creation of methods capable of automatic processing and decision making when necessary. This is why a large amount of research has been devoted in the analysis and description of digital media in the last two decades.

In this paper a short overview on recent research efforts for digital media analysis and description using computational intelligence methods is given. Computational intelligence methods are very powerful in analyzing, representing and

[1] http://www.youtube.com/yt/press/statistics.html

classifying digital media content through various architectures and learning algorithms. Supervised, unsupervised and semi-supervised algorithms can be used for digital media feature extraction, representation and characterization. The specific problem that will be used as a case study for digital media analysis is the human-centered video analysis for activity and identity recognition. These two problems have received considerable research study in the last two decades and numerous methods have been proposed in the literature, each taking into account several aspects of the problem, relating to the application scenario under consideration. In this paper, we focus on the recognition of human activities in unconstrained environments, a problem which is usually referred to as *human action recognition in the wild*.

A pipeline, that most of the methods proposed in the literature follow, consists of two processing steps, as illustrated in Figure 1. In the first step, a process aiming at the determination of a video representation that, hopefully, preserves information facilitating action discrimination, is performed. In the second step, the previously calculated video representations are employed for action discrimination. These two processing steps are, usually, applied to a set of (annotated) videos, forming the so-called training video database. After training, a new (unknown) video can be introduced to the method and classified to one of the known classes appearing in the training video database. In the following sections, we describe some of the most successful and effective approaches that have been proposed for the two aforementioned processing steps.

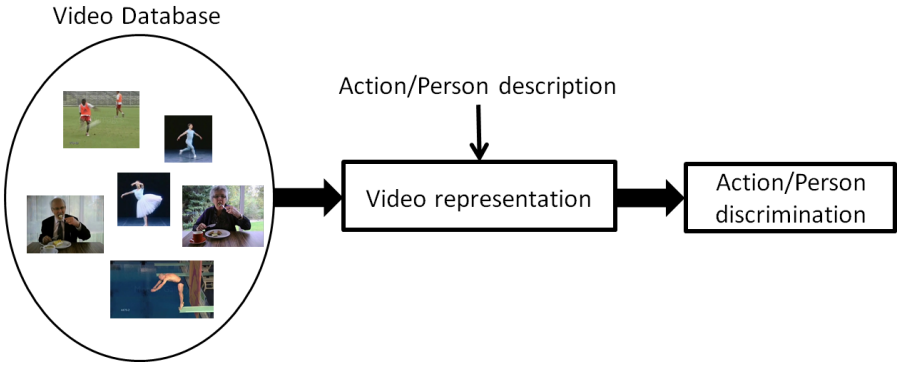
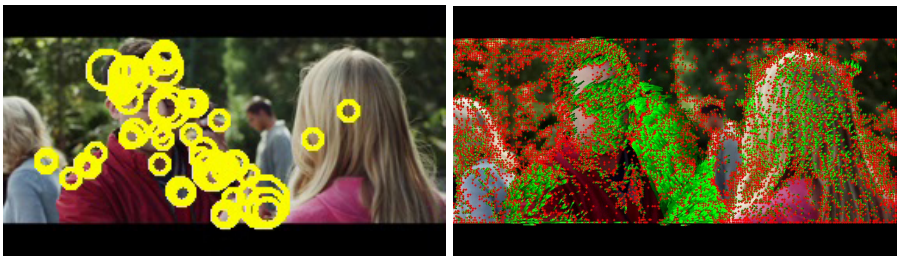


Fig. 1: Action recognition and person identification pipelineL.

2 Problem Statement

Before describing the various approaches proposed for video representation and classification, we provide an overview of the problem. Let us assume that the training video database $\mathcal{U}$ consists of N_T videos depicting persons performing

actions. Such videos will be noted as action videos hereafter. We employ the different actions appearing in $\mathcal{U}$ in order to form an action class set $\mathcal{A}$. Similarly, the persons appearing in $\mathcal{U}$ are employed in order to form a person ID class set $\mathcal{P}$. Let us assume that the N_T action videos have been manually annotated, i.e., they have been characterized according to the performed action and/or the ID of the persons appearing in them. Thus, they are followed by an action class and a person ID label, α_i and h_i, $i = 1, \ldots, N_T$, respectively. We would like to employ the videos in $\mathcal{U}$, and the corresponding labels α_i, h_i in order to train an algorithm that will be able to automatically perform action recognition and/or person identification, i.e., to classify a new (unknown) video to an action and/or person ID class appearing in the action class set $\mathcal{A}$ and/or the person ID class set $\mathcal{P}$, respectively.



(a) (b)

Fig. 2: *Local video locations of interest: a) STIPs and b) video frame interest points tracked in consecutive video frames.*

3 Action Video Representation

Video representations proposed for action recognition and person identification problems exploit either global body information, e.g. binary silhouettes corresponding to the human body video locations [1–3], or shape and motion information appearing in local video locations of interest [4–6]. In the first case, videos are usually described by sets of binary images depicting the human body silhouettes during action execution. Such silhouettes are obtained by applying video frame segmentation techniques, like background subtraction or chroma keying. Due to this preprocessing step, such representations set several assumptions, like a relatively simple background and static cameras. However, such requirements are unrealistic in most cases. For example, most of the videos uploaded in video sharing websites (like Youtube) have been recorded by non-experts (users) in scenes containing cluttered backgrounds by using moving cameras. Another example can be given for movie productions, where the leading actor may perform an action in a scene containing several extras performing the same or different action.

Video representations belonging to the second category are able to operate in the above mentioned cases, since they are evaluated directly on the color (grayscale) video frames and do not require video segmentation. Perhaps the most successful and effective local video representations have been designed around the Bag of Features model. According to this model, a video is represented by one or multiple vectors denoting the distribution of local shape and/or motion descriptors. These descriptors are calculated on local video locations corresponding either to Space Time Interest Points (STIPs) [7], or to video frame interest points that are tracked in successive video frames [5, 6], or to video frame pixels belonging to a pre-defined grid [8]. Example video frame locations of interest belonging to the first two categories are illustrated in Figure 2. STIPs determination is usually performed by applying extended versions of interest point detectors, like the Harris and Hessian ones.

The adopted descriptor types may be either handcrafted, or learned directly from data. Popular handcrafted descriptors include the Histogram of Oriented Gradients (HOG), the Histogram of Optical Flow (HOF) [9], the Motion Boundary Histogram (MBH) [5] and the Relative Motion Descriptor (RMD) [10]. Regarding data derived video representations exploiting local video information, a popular choice is to use overlapping 3D blocks, where the third dimension refers to time, in order to learn representative 3D blocks (filters) describing local shape and motion information. This is achieved by applying Deep Learning techniques, like the Independent Subspace Analysis (ISA) algorithm which has been proposed in the context of human action recognition [8]. Example filters learned by applying ISA on video frames depicting traditional Greek dances are illustrated in Figure 3.

Since the above mentioned descriptors contain complementary information, multiple descriptor types are usually employed for video representation. Let us denote by $\mathbf{x}_{ij}^d$, $j = 1, \ldots, N_i$, $d = 1, \ldots, D$ the descriptors (of type d) calculated for the i-th video in $\mathcal{U}$. D is the number of adopted descriptor types. We employ $\mathbf{x}_{ij}^d$, $i = 1, \ldots, N_T$, $j = 1, \ldots, N_i$ in order to determine a set of K_d descriptor prototypes forming the so-called codebook. This is achieved by clustering $\mathbf{x}_{ij}^d$, usually applying the K-Means algorithm [11], in K_d clusters and using the cluster mean vectors $\mathbf{v}_k^d$, $k = 1, \ldots, K_d$ as codebook vectors. After the determination of $\mathbf{v}_k^d$, each video is represented by D vectors obtained by quantizing $\mathbf{x}_{ij}^d$ according to $\mathbf{v}_k^d$. We denote by $\mathbf{s}_i^d \in \mathbb{R}^{K_d}$ the D vectors representing action video i. We would like to employ the action vectors $\mathbf{s}_i^d$ and the class labels α_i, h_i in order to train a classifier that will be able to automatically classify action videos to one of the classes appearing in $\mathcal{A}$ and/or $\mathcal{P}$.

Fig. 3: *Filters learned by the ISA algorithm when trained on video frames depicting traditional Greek dances.*

4 Action Video Classification

After applying the above described process, each action video in $\mathcal{U}$ is represented by D vectors $\mathbf{s}_i^d$. By employing $\mathbf{s}_i^d$, $i = 1, \ldots, l$ and the corresponding class labels α_i (h_i), supervised learning techniques can be employed in order to discriminate the classes appearing in $\mathcal{A}$ ($\mathcal{P}$). This is usually achieved by training N_A (N_P) nonlinear Support Vector Machine (SVM) classifiers in an one-versus-rest scheme. In order to fuse the information captured by different descriptor types d, a multi-channel RBF-χ^2 kernel function is used, which has been shown to outperform other kernel function choices in BoF-based classification [12]:

$$[\mathbf{K}]_{i,j} = exp\left(-\frac{1}{A^d}\sum_{k=1}^{K_d}\frac{(s_{ik}^d - s_{jk}^d)^2}{s_{ik}^d + s_{jk}^d}\right). \quad (1)$$

A_d is a parameter scaling the χ^2 distances between the d-th action video representations and is set equal to the mean χ^2 distance between the training vectors $\mathbf{s}_i^d$.

Except SVM classifiers, Neural Networks (NNs) have been proven effective for the classification of action videos. Single-hidden Layer Feedforward Neural (SLFN) networks have been adopted for action recognition and person identification in [13–16]. A SLFN network consists of K_d input (equal to the dimensionality of $\mathbf{s}_i^d$), L hidden and N_A (N_P) (equal to the number of classes forming the classification problem) output neurons. In order to perform fast and efficient network training, the Extreme Learning Machine (ELM) algorithm has been employed in [14]. Typically, D NNs are trained, each for a different action video representation d, and network output combination is subsequently performed.

In ELM, the network's input weights $\mathbf{W}_{in}^d$ and the hidden layer bias values $\mathbf{b}$ are randomly assigned, while the output weights $\mathbf{W}_{out}^d$ are analytically calculated. Let us denote by $\mathbf{v}_j$ the j-th column of $\mathbf{W}_{in}^d$ and by $\mathbf{w}_k$ the k-th column of $\mathbf{W}_{out}^d$. For a given activation function $\Phi(\cdot)$, the output $\mathbf{o}_i^d = [o_1^d, \ldots, o_{N_A}^d]^T$ of the ELM network corresponding to training action vector $\mathbf{s}_i$ is calculated by:

$$o_{ik}^d = \sum_{j=1}^{L} \mathbf{w}_k^T \, \Phi(\mathbf{v}_j, b_j, \mathbf{s}_i^d), \; k = 1, ..., N_A. \quad (2)$$

By storing the hidden layer neurons outputs in a matrix $\mathbf{\Phi}_d$, i.e.:

$$\mathbf{\Phi} = \begin{bmatrix} \Phi(\mathbf{v}_1, b_1, \mathbf{s}_1^d) & \cdots & \Phi(\mathbf{v}_1, b_1, \mathbf{s}_{N_T}^d) \\ \cdots & \ddots & \cdots \\ \Phi(\mathbf{v}_L, b_L, \mathbf{s}_1^d) & \cdots & \Phi(\mathbf{v}_L, b_L, \mathbf{s}_{N_T}^d) \end{bmatrix}, \quad (3)$$

Equation (2) can be written in a matrix form as $\mathbf{O}_d = \mathbf{W}_{out}^{d\,T}\mathbf{\Phi}_d$. Finally, by assuming that the network's predicted outputs $\mathbf{O}_d$ are equal to the network's desired outputs, i.e., $\mathbf{o}_i^d = \mathbf{t}_i$, and using linear activation function for the output neurons, $\mathbf{W}_{out}^d$ can be analytically calculated by $\mathbf{W}_{out}^d = \mathbf{\Phi}_d^\dagger \, \mathbf{T}^T$, where $\mathbf{\Phi}_d^\dagger =$

$\left(\mathbf{\Phi}_d\mathbf{\Phi}_d^T\right)^{-1}\mathbf{\Phi}_d$ is the Moore-Penrose generalized pseudo-inverse of $\mathbf{\Phi}_d^T$ and $\mathbf{T} = [\mathbf{t}_1, \ldots, \mathbf{t}_{N_T}]$ is a matrix containing the network's target vectors.

A regularized version of the ELM algorithm has, also, been used in [13, 15]. According to this, the network output weights $\mathbf{W}_{out}$ are calculated by solving the following optimization problem:

$$\textbf{Minimize:} \quad L_P = \frac{1}{2}\|\mathbf{W}_{out}^{d\,T}\|_F + \frac{c}{2}\sum_{i=1}^{N_T}\|\boldsymbol{\xi}_i\|_2^2 \tag{4}$$

$$\textbf{Subject to:} \quad \boldsymbol{\phi}_i^{d\,T}\mathbf{W}_{out}^d = \mathbf{t}_i^T - \boldsymbol{\xi}_i^T, \;\; i = 1, ..., N_T, \tag{5}$$

where $\boldsymbol{\xi}_i$ is the training error vector corresponding to action vector $\mathbf{s}_i^d$, $\boldsymbol{\phi}_i^d$ denotes the i-th column of $\mathbf{\Phi}_d$, i.e., the $\mathbf{s}_i^d$ representation in the ELM space, and c is a parameter denoting the importance of the training error in the optimization problem. By substituting the condition (5) in (4) and solving for $\frac{\vartheta L_P}{\vartheta \mathbf{W}_{out}^d} = 0$, $\mathbf{W}_{out}^d$ can be obtained by:

$$\mathbf{W}_{out}^d = \left(\mathbf{\Phi}_d\mathbf{\Phi}_d^T + \frac{1}{c}\mathbf{I}\right)^{-1}\mathbf{\Phi}_d\mathbf{T}^T, \tag{6}$$

or

$$\mathbf{W}_{out}^d = \mathbf{\Phi}_d\left(\mathbf{\Phi}_d^T\mathbf{\Phi}_d + \frac{1}{c}\mathbf{I}\right)^{-1}\mathbf{T}^T. \tag{7}$$

where $\mathbf{I}$ is the identity matrix.

Exploiting the fact that the ELM algorithm can be considered to be a non-linear data mapping process to a high dimensional feature space followed by linear projection and classification, the Minimum Variance ELM (MVELM) and the Minimum Class Variance ELM (MCVELM) algorithms have been proposed in [16, 17] for action recognition. These two algorithms aim at simultaneously minimizing the network output weights norm and (within-class) variance of the network outputs. The network output weights $\mathbf{W}_{out}^d$ are calculated by solving the following optimization problem:

$$\textbf{Minimize:} \quad L_P = \frac{1}{2}\|\mathbf{S}_d^{1/2}\mathbf{W}_{out}^{d\,T}\|_F + \frac{c}{2}\sum_{i=1}^{N_V}\|\boldsymbol{\xi}_i\|_2^2 \tag{8}$$

$$\textbf{Subject to:} \quad \boldsymbol{\phi}_i^{d\,T}\mathbf{W}_{out}^d = \mathbf{t}_i^T - \boldsymbol{\xi}_i^T, \;\; i = 1, ..., N_T, \tag{9}$$

and the network output weights are given by:

$$\mathbf{W}_{out}^d = \left(\mathbf{\Phi}_d\mathbf{\Phi}_d^T + \frac{1}{c}\mathbf{S}_d\right)^{-1}\mathbf{\Phi}_d\mathbf{T}^T. \tag{10}$$

$\mathbf{S}_d$ in (8), (10) is either the within-class scatter matrix $\mathbf{S}_w^d$ of the network hidden layer outputs, i.e., the representation of $\mathbf{s}_i^d$ in the so-called ELM space, or the

total scatter matrix $\mathbf{S}_T^d$ in the ELM space. In the case of unimodal action classes in the ELM space, $\mathbf{S}_w^d$ is of the form:

$$\mathbf{S}_w^d = \sum_{j=1}^{N_A} \sum_{i=1}^{N_T} \frac{\beta_{ij}}{N_j} (\boldsymbol{\phi}_i^d - \boldsymbol{\mu}_j^d)(\boldsymbol{\phi}_i^d - \boldsymbol{\mu}_j^d)^T. \tag{11}$$

In (11), β_{ij} is an index denoting if training action vector $\mathbf{s}_i^d$ belongs to action class j, i.e., $\beta_{ij} = 1$, if $c_i = j$ and $\beta_{ij} = 0$ otherwise, and $N_j = \sum_{i=1}^{N_T} \beta_{ij}$ is the number of training action vectors belonging to action class j. $\boldsymbol{\mu}_j^d = \frac{1}{N_j} \sum_{i=1}^{N_T} \beta_{ij} \boldsymbol{\phi}_i^d$ is the mean vector of class j in the ELM space.

In the case of multi-modal action classes, $\mathbf{S}_w$ is of the form:

$$\mathbf{S}_{w,CDA}^d = \sum_{j=1}^{N_A} \sum_{k=1}^{b_j} \sum_{i=1}^{N_T} \frac{\beta_{ijk} (\boldsymbol{\phi}_i^d - \boldsymbol{\mu}_{jk}^d)(\boldsymbol{\phi}_i^d - \boldsymbol{\mu}_{jk}^d)^T}{N_{jk}}. \tag{12}$$

Here, it is assumed that class j consists of b_j clusters, containing N_{jk}, $j = 1, ..., N_A$, $k = 1, ..., b_j$ action vectors each. β_{ijk} is an index denoting if action vector $\mathbf{s}_i^d$ belongs to the k-th cluster of action class j and $\boldsymbol{\mu}_{jk}^d = \frac{1}{N_{jk}} \sum_{i=1}^{N_T} \beta_{ijk} \boldsymbol{\phi}_i^d$ denotes the mean vector of the k-th cluster of class j in the ELM space.

Finally, $\mathbf{S}_T$ is given by:

$$\mathbf{S}_T^d = \sum_{i=1}^{N_T} (\boldsymbol{\phi}_i^d - \boldsymbol{\mu}^d)(\boldsymbol{\phi}_i^d - \boldsymbol{\mu}^d)^T, \tag{13}$$

where $\boldsymbol{\mu}^d = \frac{1}{N_T} \sum_{i=1}^{N_T} \boldsymbol{\phi}_i^d$ is the vector of the entire training set in the ELM space.

The information captured by different descriptor types is fused either by combining the network outputs corresponding to different descriptor types, e.g., by calculating the mean network output, or by optimally weighting the contribution of each NN according to combination weights $\boldsymbol{\gamma} \in \mathbb{R}^D$ by solving the following optimization problem:

$$\textbf{Minimize: } \mathcal{J} = \frac{1}{2} \sum_{d=1}^{D} \|\mathbf{W}_{out}^d\|_F^2 + \frac{c}{2} \sum_{i=1}^{N} \|\boldsymbol{\xi}_i\|_2^2 \tag{14}$$

$$\textbf{Subject to: } \left(\sum_{d=1}^{D} \gamma_d \mathbf{W}_{out}^{d\,T} \boldsymbol{\phi}_i^d \right) - \mathbf{t}_i = \boldsymbol{\xi}_i, \;\; i = 1, ..., N, \tag{15}$$

$$\|\boldsymbol{\gamma}\|_2^2 = 1, \tag{16}$$

An iterative optimization process consisting of two convex optimization problems has been proposed in [18] to this end.

By exploiting the fast and efficient ELM algorithm for SLFN network training, a dynamic classification scheme has been proposed for human action recognition in [19]. It consists of two iteratively repeated steps. In the first step, a

non-linear mapping process for both the training action vectors and the test sample under consideration is determined by training a SLFN network. In the second step, test sample-specific training action vectors selection is performed by exploiting the obtained network outputs corresponding to both the training action vectors and the test sample under consideration. These two steps are performed in multiple levels. At each level, by exploiting only the more similar to the test sample training action vectors, the dynamic classification scheme focuses the classification problem on the classes that should be able to discriminate. Considering the fact that after performing multiple data selections for a level $l > 1$ the cardinality of the training action vector set that will be used for SLFN network training will be very small compared to the dimensionality of the ELM space, the regularized version of ELM algorithm (6) has been employed in [21]. By using (7), the network output vector corresponding to $\mathbf{s}_i^d$ is obtained by:

$$\mathbf{o}_i^d = \mathbf{W}_{out}^{d\,T}\,\phi_i^d = \mathbf{T}\left(\mathbf{\Omega}_d + \frac{1}{c}\mathbf{I}\right)^{-1}\mathbf{K}_i^d, \tag{17}$$

where $\mathbf{K}_i^d = \mathbf{\Phi}_d^T\phi_i^d$, $\mathbf{\Omega}_d = \mathbf{\Phi}_d^T\mathbf{\Phi}_d$ are the kernel matrices corresponding to $\mathbf{s}_i^d$ and the entire SLFN training set, respectively. Thus, in this case the ELM space dimensionality is inherently determined by exploiting the kernel trick [24] and needs not be defined in advance. This ELM network training formulation also has the advantage that combined kernel matrices of the form (1) can be exploited.

The semi-supervised ELM (SELM) algorithms [23] has also been proposed for dynamic action classification in [22]. SELM solves the following optimization problem:

$$\textbf{Minimize: } \mathcal{J} = \|\mathbf{W}_{out}^{d\,T}\mathbf{\Phi}_d - \mathbf{T}\|_F \tag{18}$$

$$\textbf{Subject to: } \sum_{i=1}^{N_T}\sum_{j=1}^{N_T} w_{ij}\left(\mathbf{W}_{out}^{d\,T}\boldsymbol{\phi}_i^d - \mathbf{W}_{out}^{d\,T}\boldsymbol{\phi}_j^d\right)^2 = 0, \tag{19}$$

where w_{ij} is a value denoting the similarity between $\boldsymbol{\phi}_i^d$ and $\boldsymbol{\phi}_j^d$. $\mathbf{W}_{out}^d$ is given by:

$$\mathbf{W}_{out}^d = \left(\left(\mathbf{J} + \lambda\mathbf{L}_d^T\right)\mathbf{\Phi}\right)^{\dagger}\mathbf{J}\mathbf{T}^T, \tag{20}$$

where $\mathbf{J} = diag(1, 1, \ldots, 0, 0)$ with the first l diagonal entries as 1 and the rest 0 and $\mathbf{L}_d$ is the Graph Laplacian matrix [20] encoding the similarity between the training vectors $\boldsymbol{\phi}_i^d$.

In the dynamic classification scheme proposed in [22], test sample-specific training action vectors selection is performed by calculating the Euclidean distances between a given test sample and the training action vectors. The l training action vectors closest to the test sample are employed in order to form the labeled set of the SELM algorithm, while the remaining ones are used as unlabeled. Finally, the test sample under consideration is classified to the class corresponding to the highest SELM output.

Experimental results in real video data using all the previously presented methods can be found in the corresponding references. The results indicate that

computational intelligence techniques can be used for solving difficult tasks, such as video analysis and semantic information extraction in digital media.

5 Conclusion

In this paper a survey on recent research efforts for digital media analysis and description based on computational intelligence methods has been presented. The specific problem that has been used as a case study is the human centered video analysis for activity and identity recognition. The presented approaches are generic and can be easily adapted for the description and analysis of other semantic concepts, especially those that involve human presence in digital media content.

Acknowledgment

The research leading to these results has received funding from the European Union Seventh Framework Programme (FP7/2007-2013) under grant agreement number 316564 (IMPART). This publication reflects only the authors views. The European Union is not liable for any use that may be made of the information contained therein.

References

1. Ji, X., Liu, H.: Advances in view-invariant human motion analysis: a review. IEEE Transactions on Systems Man and Cybernetics, Part-C, 40(1), 13–24 (2010)
2. Iosifidis, A., Tefas, A., Nikolaidis, N., Pitas, A.: Multi-view Human Movement Recognition based on Fuzzy Distances and Linear Discriminant Analysis, Computer Vision and Image Understanding, 116, 347–360 (2012)
3. Iosifidis, A., Tefas, A., Pitas, A.: View-invariant action recognition based on Artificial Neural Networks, IEEE Transactions on Neural Networks, 23(3), 412–424 (2012)
4. Wang, H., Ullah, M., Klaser, A., Laptev, I.: Evaluation of local spatio-temporal features for action recognition, British Machine Vision Conference, (2009)
5. Wang, H., Klaser, A., Schmid, C., Laptev, I.: Dense trajectories and motion boundary descriptors for action recognition, International Journal of Computer Vision, 103(1), 60–79 (2013)
6. Guha, T., Ward, R.: Learning Sparse Representations for Human Action Recognition, IEEE Transactions on Pattern Analysis and Machine Intelligence, 34, 1576–1588 (2011)
7. Laptev, I. : On space-time interest points, International Journal of Computer Vision, 64(2), 107–123 (2005)
8. Le, Q.V., Zou, W.Y., Yeung, S.Y., Ng, A.Y.: Learning hierarchical invariant spatio-temporal features for action recognition with independent subspace analysis, Computer Vision and Pattern Recognition (2011)
9. Laptev, I., MarszalSchmid, C., Rozenfeld, B.: Learning realistic human actions from movies, Computer Vision and Pattern Recognition (2008)

10. Oshin, O., Gilbert, A., Bowden, R.: Capturing the relative distribution of features for action recognition, Face and Gesture Workshop (2011)
11. Theodoridis, S., Koutroumbas, K.: Pattern Recognition, Academic Press (2008)
12. Zhang, J., Marszalek, M., Lazebnik, M., Schmid, C.: Local features and kernels for classification of texture and object categories: A comprehensive study, International Journal of Computer Vision, 73(2), 213–238 (2007)
13. Iosifidis, A. and Tefas, A. and Pitas, I.: Person Identification from Actions based on Artificial Neural Networks, Symposium Series on Computational Intelligence (2013)
14. Minhas, R. and Baradarani, S. and Seifzadeh, S. and Wu, Q.J..: Human action recognition using extreme learning machine based on visual vocabularies, Neurocomputing, 73(10–12), 1906–1917 (2010)
15. Iosifidis, A. and Tefas, A. and Pitas, I.: Multi-view Human Action Recognition under Occlusion based on Fuzzy Distances and Neural Networks, European Signal Processing Conference (2012)
16. Iosifidis, A. and Tefas, A. and Pitas, I.: Minimum Class Variance Extreme Learning Machine for Human Action Recognition, IEEE Transactions on Circuits and Systems for Video Technology, 23(11), 1968–1979 (2013)
17. Iosifidis, A. and Tefas, A. and Pitas, I.: Minimum Variance Extreme Learning Machine for Human Action Recognition, International Conference on Accoustics, Speech and Signal Processing (2014)
18. Iosifidis, A. and Tefas, A. and Pitas, I.: Multi-view Regularized Extreme Learning Machine for Human Action Recognition, Hellenic Conference on Artificial Intelligence (2014)
19. Iosifidis, A. and Tefas, A. and Pitas, I.: Dynamic action recognition based on Dynemes and Extreme Learning Machine, Pattern Recognition Letters, 34, 1890–1898 (2013)
20. Belkin, M. and Niyogi, P. and Sindhawani, V.: Manifold Regularization: A Geometric Framework for Learning from Labeled and Unlabeled Examples, Journal of Machine Learning Research, 7, 2399–2434 (2006)
21. Iosifidis, A. and Tefas, A. and Pitas, I.: Dynamic Action Classification based on Iterative Data Selection and Feedforward Neural Networks, European Signal Processing Conference (2013)
22. Iosifidis, A. and Tefas, A. and Pitas, I.: Active Classification for Human Action Recognition, IEEE International Conference on Image Processing (2013)
23. Liu, J., Cheng, Y., Liu, M., Zhao, Z.: Semi-supervised ELM with application in sparse calibrated locations estimation, Neurocomputing, 74, 2566-2572 (2011)
24. Scholkpf, B., Smola, A.: Learning with kernels, MIT Press, (2001)