Discriminant Bag of Words based Representation for Human Action Recognition

Alexandros Iosifidis, Anastasios Tefas and Ioannis Pitas

Department of Informatics, Aristotle University of Thessaloniki
Box 451, 54124 Thessaloniki, Greece

{*aiosif,tefas,pitas*}*@aiia.csd.auth.gr*

Abstract

In this paper we propose a novel framework for human action recognition based on Bag of Words (BoWs) action representation, that unifies discriminative codebook generation and discriminant subspace learning. The proposed framework is able to, naturally, incorporate several (linear or non-linear) discrimination criteria for discriminant BoWs-based action representation. An iterative optimization scheme is proposed for sequential discriminant BoWs-based action representation and codebook adaptation based on action discrimination in a reduced dimensionality feature space where action classes are better discriminated. Experiments on five publicly available data sets aiming at different application scenarios demonstrate that the proposed unified approach increases the codebook discriminative ability providing enhanced action classification performance.

Keywords: Bag of Words, Discriminant Learning, Codebook Learning

1. Introduction

Human action recognition from videos has been intensively studied in the last two decades due to its importance in a wide range of applications, like human-computer interaction (HCI), content-based video retrieval and augmented reality, to name a few. It is, still, an active research field due to its difficulty, which is, mainly, caused because there is not a formal description of actions. Action execution style variations and changes in human body sizes among individuals, as well as different camera observation angles are some of the reasons that lead to

high intra-class and, possibly, small inter-class variations of action classes. Recently, several action descriptors aiming at action recognition in unconstrained environments have been proposed, including local sparse and dense space-time features ([23, 45, 10, 20, 34, 35]). Such descriptors capture information appearing in video frame locations that either correspond to video frame interest points which are tracked during action execution, or that are subject to abrupt intensity value variations and, thus, contain information regarding motion speed and/or acceleration, which is of interest for the description of actions. These local video frame descriptors are calculated by using the color (grayscale) video frames and, thus, video frame segmentation is not required.

After describing actions, videos depicting actions, called action videos hereafter, are usually represented by fixed size vectors. Several feature coding approaches have been proposed in order to determine compact (vectorial) representations ([14]), including Sparse Coding ([42]), Fisher Vector ([28]), Local Tangent coding ([47]) and Salience-based coding ([13]). Perhaps the most well studied and successful approach for action representation is based on the Bag of Words (BoWs) model ([4]). According to this model, each action video is represented by a vector obtained by applying (hard or soft) quantization on the features describing it and using a set of feature prototypes forming the so-called codebook. This codebook is determined by clustering the features describing training action videos. The BoWs-based action representation has been combined with several classifiers, like Support Vector Machines, Artificial Neural Networks and Discriminant Analysis based classification schemes, providing high action classification performance on publicly available data sets aiming at different application scenarios. However, due to the fact that the calculation of the adopted codebook is based on an unsupervised process, the discriminative ability of the BoWs-based action representation is limited.

In order to increase the quality of the adopted codebook, codebook adaptation processes have been proposed which adopt a generative approach. That is, the initial codebook generated by clustering the features describing training videos is adapted so as to reduce the reconstruction error of the resulted video representation ([37]). However, since this generative adaptation process does not take into account the class labels that are available for the training action videos, the discriminative ability of the optimized codebook is not necessarily increased. In order to increase the discriminative ability of the adopted codebook, researchers have begun to introduce discriminative codebook learning processes ([40, 29]). However, since the codebook calculation process is, still, disconnected from the adopted classification scheme, the obtained codebook may not be the one that is

best suited for the task under consideration, i.e., the classification of actions in our case.

A method aiming at simultaneously learning both a discriminative codebook and a classifier is proposed in ([43]) for image classification. This method consists of two iteratively repeated steps. The first one involves training images representation by a set of class-specific histograms of visual words at the bit level and multiple binary classifiers, one for each image category, training by using the obtained histograms. Based on the performance of each classifier, the set of training histograms is updated in the second step. While this approach has lead to increased image classification performance, its extension in other classification tasks, e.g., action recognition, is not straightforward. Another approach has been proposed in ([25]), where a two-class linear SVM-based codebook adaptation scheme is formulated. The adoption of a two class formulation generates the drawback that $C(C-1)/2$ two-class codebooks have to be learned (C being the number of classes) and used in the test phase along with an appropriate fusion strategy. In addition, such an approach is not able to exploit inter-class correlation information appearing in multi-class problems, which may facilitate class discrimination.

In this paper, we build on the BoWs-based action video representation by introducing discriminative criteria on the codebook learning process. The proposed approach integrates codebook learning and action class discrimination in a multi-class optimization process in order to produce a discriminant BoWs-based action video representation. Two processing steps are iteratively repeated to this end. The first one, involves the calculation of BoWs-based representation of the training action videos using a codebook of representative features and learning of an optimal mapping of the obtained BoWs-based action video representations to a discriminant feature space where action classes are better discriminated. In the second step, based on an action class discrimination criterion in the obtained feature space, the adopted codebook is adapted in order to increase action classes discrimination. In order to classify a new, unknown, action video, it is represented by employing the optimized codebook and the obtained BoWs-based action video representation is mapped to the discriminant feature space determined in the training phase. In this discriminant space, classification can be performed by employing several classifiers, like K-Nearest Neighbors (K-NN), Support Vector Machine (SVM) or Artificial Neural Networks (ANN). Here it should be noted that the proposed approach is not aiming at increasing the representation power of the adopted codebook. Instead, it aims at increasing its discrimination power for the classification task under consideration, i.e., the discrimination of the action classes involved in the classification problem at hand.

The rest of the paper is structured as follows. The proposed approach for integrated discriminant codebook and discriminant BoWs-based action representation learning is described in Section 2. Experiments conducted on publicly available data sets aiming at different application scenarios are illustrated in Section 3. Finally, conclusions are drawn in Section 4.

2. Discriminant Codebook Learning for BoWs-based Action Representation

In this section we describe in detail the proposed integrated optimization scheme for discriminant BoWs-based action representation. Let $\mathcal{U}$ be a video database containing N_T action videos followed by action class labels $l_i,\ i = 1, \ldots, N_T$ appearing in an action class set $\mathcal{A} = \{\alpha\}_{\alpha=1}^{C}$. Let us assume that each action video i is described by N_i feature vectors $\mathbf{p}_{ij}, \in \mathbb{R}^D,\ i = 1, \ldots, N_T,\ j = 1, \ldots, N_i$, which are normalized in order to have unit l_2 norm. We employ the feature vectors $\mathbf{p}_{ij}$ and the action class labels l_i in order to represent each action video i by two discriminant feature vectors $\mathbf{s}_i \in \mathbb{R}^D$ and $\mathbf{z}_i \in \mathbb{R}^d$, $d < D$, in the feature space determined by the adopted codebook and the discriminant space, respectively.

2.1. Standard BoWs-based action representation

Let us denote by $\mathbf{V} \in \mathbb{R}^{D \times K}$ a codebook formed by codebook vectors $\mathbf{v}_k \in \mathbb{R}^D,\ k = 1, \ldots, K$. This codebook is calculated by clustering the feature vectors $\mathbf{p}_{ij},\ i = 1, \ldots, N_T,\ j = 1, \ldots, N_i$ without exploiting the available labeling information for the training action videos. Several clustering techniques can be employed to this end. K-Means has been widely adopted for its simplicity and fast operation. The codebook vectors $\mathbf{v}_k$ are, usually, determined to be the cluster mean vectors. After determining the codebook $\mathbf{V}$, the standard BoWs-based action representation of action video i is obtained by applying hard or soft vector quantization on the feature vectors $\mathbf{p}_{ij},\ j = 1, \ldots, N_i$. In the first case, the BoWs-based representation of action video i is a histogram of features, calculated by assigning each feature vector $\mathbf{p}_{ij}$ to the cluster of the closest codebook vector $\mathbf{v}_k$. In the second case, a distance function, usually the Euclidean one, is used in order to determine N_i distance vectors, each denoting the similarity of feature vector $\mathbf{p}_{ij}$ to all the codebook vectors $\mathbf{v}_k$, and the BoWs-based representation of action video i is determined to be the mean normalized distance vector ([15]).

2.2. Discriminant BoWs-based action representation

The proposed discriminant BoWs-based representation exploits a generalization of the Euclidean distance, i.e.,:

$$d_{ijk} = \|\mathbf{v}_k - \mathbf{p}_{ij}\|_2^{-g}. \quad (1)$$

The use of a parameter value $g = 1.0$ leads to a BoWs-based representation based on soft vector quantization, while a parameter value $g \gg 1.0$ leads to a BoWs-based representation based on hard vector quantization. By using the above distance function, each feature vector $\mathbf{p}_{ij}$ is mapped to the so-called membership vector $\mathbf{u}_{ij} \in \mathbb{R}^K$, encoding the similarity of $\mathbf{p}_{ij}$ to all the codebook vectors $\mathbf{v}_k$. Membership vectors $\mathbf{u}_{ij} \in \mathbb{R}^K$ are obtained by normalizing the distance vectors $\mathbf{d}_{ij} = [d_{ij1} \dots d_{ijK}]^T$ in order to have unit l_1 norm, i.e.:

$$\mathbf{u}_{ij} = \frac{\mathbf{d}_{ij}}{\|\mathbf{d}_{ij}\|_1}. \quad (2)$$

The BoWs-based representation of action video i is obtained by calculating the mean membership vector:

$$\mathbf{q}_i = \frac{1}{N_i} \sum_{j=1}^{N_i} \mathbf{u}_{ij}. \quad (3)$$

Finally, the mean membership vectors $\mathbf{q}_i$ are normalized in order to produce the so-called action vectors:

$$\mathbf{s}_i = \frac{\mathbf{q}_i}{\|\mathbf{q}_i\|_2}. \quad (4)$$

The adopted similarity function as well as the mean similarity to codebook vectors for action video representation allow for better diffusion of the similarity along the codebook vectors. This is more effective, especially for small feature sets, where the resulting standard BoW-based representations are rather sparse, as is the case of STIP-based extracted features.

After calculating the action vectors representing all the action videos, they are normalized in order to have zero mean and unit standard deviation, resulting to the normalized action vectors $\mathbf{x}_i \in \mathbb{R}^K$. In order to map the normalized action vectors $\mathbf{x}_i$ to a new feature space in which action classes are better discriminated, an optimal linear transformation $\mathbf{W}^*$ is obtained by solving the *trace ratio* optimization problem:

$$\mathbf{W}^* = \arg\min_{\mathbf{W}} \frac{trace\{\mathbf{W}^T\mathbf{A}\mathbf{W}\}}{trace\{\mathbf{W}^T\mathbf{B}\mathbf{W}\}}, \quad (5)$$

where $\mathbf{A}$, $\mathbf{B}$ are matrices encoding properties of interest for the training normalized action vectors $\mathbf{x}_i$, $i = 1, \dots, N_T$. That is, they are functions of the training

normalized action vectors, i.e., $\mathbf{A}(\mathbf{x}_i)$, $\mathbf{B}(\mathbf{x}_i)$. Finally, the discriminant action vectors $\mathbf{z}_i,\ i = 1, \ldots, N_T$ are obtained by:

$$\mathbf{z}_i = \mathbf{W}^{*T}\mathbf{x}_i. \tag{6}$$

The optimization problem in (5) is, usually, approximated by solving the *ratio trace* optimization problem $\mathbf{A}\mathbf{w} = \lambda\mathbf{B}\mathbf{w},\ \lambda \neq 0$ ([36]), which can be solved by performing eigenanalysis to the matrix $\mathbf{B}^{-1}\mathbf{A}$ in the case where $\mathbf{B}$ is invertible, or $\mathbf{A}^{-1}\mathbf{B}$ in the case where $\mathbf{A}$ is invertible. In the case where both $\mathbf{A}$, $\mathbf{B}$ are singular, i.e., when $N_T < K$, the strictly diagonally dominant criterion for nonsingular matrices is exploited and a regularized version of $\tilde{\mathbf{A}} = \mathbf{A} + r\mathbf{I}$ is employed ([26]), where r is a small positive value. However, as has been shown in ([36, 17]), the original *trace ratio* problem can directly be solved by solving the equivalent *trace difference* optimization problem:

$$\mathbf{W}^* = \underset{\mathbf{W}^T\mathbf{W}=\mathbf{I}}{argmax}\ Tr\left[\mathbf{W}^T\left(\mathbf{B} - \lambda^*\mathbf{A}\right)\mathbf{W}\right], \tag{7}$$

where $\lambda^* \geq 0$ is the trace ratio $\lambda^* = \frac{Tr(\mathbf{W}^T\mathbf{B}\mathbf{W})}{Tr(\mathbf{W}^T\mathbf{A}\mathbf{W})}$. The trace ratio value λ^* can be calculated by using an efficient algorithm based on Newton-Raphson method ([17]). After determining λ^*, the trace difference optimization problem is solved by performing eigenanalysis to the matrix $\mathbf{S} = \mathbf{B} - \lambda^*\mathbf{A}$. The optimal transformation matrix $\mathbf{W}^*$ is formed by the eigenvectors corresponding to the non zero eigenvalues of $\mathbf{S}$.

By following the above described procedure for discriminant action vectors calculation, several discriminant spaces can be obtained for action representation. As has been shown in ([41]), a wide range of (linear and non-linear) discriminant learning techniques can be obtained by solving the trace ratio optimization problem (5) and employing the graph embedding framework, including Principal Components Analysis (PCA) ([6]), Linear Discriminant Analysis (LDA) ([6]), ISOMAP ([33]), Locally Linear Embedding (LLE) ([3]) and Laplacian Eigenmaps (LE) ([1]).

After determining the above described discriminant action video representation, a codebook adaptation process is performed in order to increase the codebook discriminative ability based on action class discrimination in the obtained discriminant space. The procedure followed to this end is described in the following.

2.3. Codebook Adaptation

By observing that the normalized action vectors $\mathbf{x}_i$ are functions of the adopted codebook $\mathbf{V}$, as described in subsection 2.2, it can be seen that the optimization

problem (5) is a function of both the transformation matrix $\mathbf{W}$ and the codebook $\mathbf{V}$. Based on this observation, we propose to minimize the trace ratio criterion with respect to both $\mathbf{W}$ and $\mathbf{V}$, in order to simultaneously increase the codebook discriminative ability and to obtain the optimal transformation matrix for action classes discrimination:

$$\mathcal{J}(\mathbf{W}, \mathbf{V}) = \frac{trace\{\mathbf{W}^T \mathbf{A}(\mathbf{V}) \mathbf{W}\}}{trace\{\mathbf{W}^T \mathbf{B}(\mathbf{V}) \mathbf{W}\}} \tag{8}$$

Since the minimization of (8), for given $\mathbf{V}$, can be readily computed by solving the trace ratio problem ([17]), we propose an iterative optimization scheme consisting of two steps. In the first step, for a given codebook, training normalized action vectors $\mathbf{x}_i$ are employed in order to determine the optimal projection matrix $\mathbf{W}_t^*$ by solving the trace ratio problem (8). In the second step, codebook vectors $\mathbf{v}_{k,t}$ are adapted, in the direction of the gradient of (8), by using the obtained $\mathbf{W}_t^*$. Here, we have introduced the index t denoting the iteration of the iterative optimization process. The adaptation of $\mathbf{v}_{k,t}$ is performed by following the gradient of $\mathcal{J}$ with respect to $\mathbf{v}_{k,t}$:

$$\mathbf{v}_{k,t+1} = \mathbf{v}_{k,t} - \eta \frac{\partial \mathcal{J}_t}{\partial \mathbf{v}_{k,t}} \tag{9}$$

$$\frac{\partial \mathcal{J}_t}{\partial \mathbf{v}_{k,t}} = \frac{\partial \mathcal{J}_t}{\partial x_{ik,t}} \frac{\partial x_{ik,t}}{\partial q_{ik,t}} \frac{\partial q_{ik,t}}{\partial d_{ijk,t}} \frac{\partial d_{ijk,t}}{\partial v_{k,t}} \tag{10}$$

where η is an update rate parameter. In order to avoid scaling issues, codebook vectors of both the initial and the updated codebooks, $\mathbf{v}_{k,0}$, $\mathbf{v}_{k,t}$ respectively, are normalized to have unit l_2 norm.

The two above described optimization steps are performed until $(\mathcal{J}_t - \mathcal{J}_{t+1})/\mathcal{J}_t < \epsilon$, where ϵ is a small positive value (equal to $\epsilon = 10^{-6}$ in our experiments). Since LDA is the most widely adopted discriminant learning technique, due to its effectiveness in many classification problems, we provide the update rule obtained by employing the LDA discrimination criterion in the following. The derivation of codebook adaptation processes exploiting different discrimination criteria is straightforward.

LDA determines an optimal discriminant space for data projection in which classes are better discriminated. The adopted criterion is the ratio of the within-class scatter to the between-class scatter in the projection space. LDA solves the optimization problem (8) by using the matrices:

$$\mathbf{A}_t = \sum_{\alpha=1}^{C} \sum_{i=1}^{N_T} b_i^{\alpha} \left(\mathbf{x}_{i,t} - \bar{\mathbf{x}}_t^{\alpha}\right) \left(\mathbf{x}_{i,t} - \bar{\mathbf{x}}_t^{\alpha}\right)^T, \tag{11}$$

$$\mathbf{B}_t = \sum_{\alpha=1}^{C} \left(\bar{\mathbf{x}}_t^{\alpha} - \bar{\mathbf{x}}_t\right)\left(\bar{\mathbf{x}}_t^{\alpha} - \bar{\mathbf{x}}_t\right)^T, \tag{12}$$

where $\mathbf{A}_t$, $\mathbf{B}_t$ are the within-class and between-class scatter matrices obtained by using $\mathbf{V}_t$, respectively. b_i^{α} is an index denoting if the normalized action vector $\mathbf{x}_{i,t}$ belongs to action class α, i.e., $b_i^{\alpha} = 1$ if $l_i = \alpha$ and $b_i^{\alpha} = 0$ otherwise. $\bar{\mathbf{x}}_t^{\alpha}$ is the mean vector of action class α and $\bar{\mathbf{x}}_t$ is the mean normalized action vector of the entire training set.

By using (11), (12), the gradient (10) is given by:

$$\begin{aligned}
\frac{\partial \mathcal{J}_t}{\partial \mathbf{v}_{k,t}} &= \left(a\tilde{\mathbf{W}}_{t(i,:)}(\mathbf{x}_{i,t} - \bar{\mathbf{x}}_t^{\alpha}) - c\tilde{\mathbf{W}}_{t(i,:)}\bar{\mathbf{x}}_t^{\alpha}\right) \\
&\cdot \left(\frac{1}{\tilde{s}_{k,t}} - \frac{s_{ik,t} - \bar{s}_{k,t}}{\tilde{s}_{k,t}^3}\right)\left(\frac{1}{\|\mathbf{q}_{i,t}\|_2} - \frac{q_{ik,t}^2}{\|\mathbf{q}_{i,t}\|_2^3}\right) \\
&\cdot \frac{N_T - 1}{N_T N_i}\left(\frac{1}{\|\mathbf{d}_{ij,t}\|_1} - \frac{d_{ijk,t}}{\|\mathbf{d}_{ij,t}\|_1^2}\right) \\
&\cdot -g\|\mathbf{v}_{k,t} - \mathbf{p}_{ij}\|_2^{-(g+2)}\left(\mathbf{v}_{k,t} - \mathbf{p}_{ij}\right),
\end{aligned} \tag{13}$$

where $a = \frac{2b_i^{\alpha}}{trace(\mathbf{W}_t^T\mathbf{B}_t\mathbf{W}_t)}$, $c = \frac{2b_i^{\alpha}trace(\mathbf{W}_t^T\mathbf{A}_t\mathbf{W}_t)}{trace(\mathbf{W}_t^T\mathbf{B}_t\mathbf{W}_t)^2}$, $\tilde{\mathbf{W}}_{t(i,:)}$ is the i-th row of the matrix $\tilde{\mathbf{W}}_t = \mathbf{W}_t\mathbf{W}_t^T$ and $\bar{s}_{k,t}$, $\tilde{s}_{k,t}$ are the mean and standard deviation of the training action vectors in dimension k, respectively.

The update rate parameter value η in (9) can either be set to a fixed value, e.g., $\eta = 0.01$, or be dynamically determined. In order to accelerate the codebook adaptation process and (possibly) to avoid convergence on local minima, in our experiments we have employed a (dynamic) line search strategy. That is, in each iteration of the codebook adaptation process, the trace ratio criterion (8) was evaluated by using (9) and $\eta_0 = 0.1$. In the case where $\mathcal{J}_{t+1} < \mathcal{J}_t$, the trace ratio criterion was evaluated by using a codebook update parameter value $\eta_n = 2\eta_{n-1}$. This process is followed until $\mathcal{J}_{t+1} > \mathcal{J}_t$ and the codebook update parameter value providing the highest $\mathcal{J}$ decrease was employed for codebook adaptation. In the case where, by using a codebook update parameter value $\eta_0 = 0.1$, $\mathcal{J}_{t+1} > \mathcal{J}_t$, the trace ratio criterion was evaluated by using a codebook update parameter value $\eta_n = \eta_{n-1}/2$. This process is followed until $\mathcal{J}_{t+1} < \mathcal{J}_t$ and the codebook update parameter value providing $\mathcal{J}$ decrease was employed for codebook adaptation.

2.4. Action Recognition (Test Phase)

Let us denote by $\mathbf{V}_{opt}$, $\mathbf{W}_{opt}^*$ the codebook and the corresponding projection matrix obtained by applying the above described optimization process employing

the feature vectors $\mathbf{p}_{ij}$ describing the training action videos and the corresponding action class labels. Let $\mathbf{p}_{tj} \in \mathbb{R}^D,\ j = 1, \ldots, N_t$ be feature vectors describing a test action video. $\mathbf{p}_{tj}$ are employed in order to calculate the corresponding normalized action vector $\mathbf{x}_t \in \mathbb{R}^K$ using $\mathbf{V}_{opt}$. $\mathbf{x}_t$ can be either classified in this space, or be mapped to the discriminant space, determined in the training phase, by applying $\mathbf{z}_t = \mathbf{W}_{opt}^{*T}\mathbf{x}_t$. In either cases, action classification is performed by employing any, linear or non-linear, classifier, like K-NN, SVM and ANNs.

3. Experimental Evaluation

In this Section we present experiments conducted in order to evaluate the proposed discriminant codebook learning technique and the obtained discriminant BoWs-based action representation. In all the experiments we have employed the Harris3D detector ([21]) followed by HOG/HOF descriptors ([35]) calculation for action video description. We should note here that the proposed optimization framework for discriminant BoWs-based representation can be combined with any descriptor proposed for BoWs-based representation. The optimal values of parameters K and g have been determined by applying grid search using values $50 < K < 500$ and $g = [1, 2, 5, 10, 20]$, respectively. In order to limit the complexity, we cluster a subset of $100k$ randomly selected HOG/HOF descriptors for initial codebook calculation. To increase precision of the initial codebook, we initialize K-Means 10 times and keep the codebook providing the smallest error. In the test phase, classification is performed by employing a Single-hidden Layer Feedforward Neural Network trained by applying the recently proposed Extreme Learning Machine algorithm ([12]).

We have used five publicly available data sets aiming at different application scenarios, from single-view everyday actions, actions appearing in movies, complex ballet movements, multi-view everyday actions and, even, facial expressions. Comprehensive description of the data sets and information concerning the experimental setup used in each data set are provided in the following subsections. In all the experiments we compare the performance of the standard BoWs-based action video representation (BoWs), i.e., the BoWs-based action video representation adopting an unsupervised codebook, and the proposed discriminant BoWs-based action representation (DBoWs), i.e., the discriminant BoWs-based action representation obtained by applying the proposed supervised codebook learning optimization scheme. Furthermore, we provide comparison results of the proposed discriminant BoWs-based action video representation adopting the above mentioned descriptor-classifier combination with some recently proposed state-

of-the-art action recognition methods evaluating their performance on the adopted action recognition data sets.

3.1. The KTH Data Set

The KTH data set ([32]) consists of 600 videos depicting twenty-five persons, each performing six everyday actions: 'walking', 'jogging', 'running', 'boxing', 'hand waving' and 'hand clapping'. Four different scenarios have been recorded: (s1) outdoors, (s2) outdoors with scale variation, (s3) outdoors with different clothes and (s4) indoors. The persons are free to change motion speed and direction between different action realizations. Example video frames are illustrated in Figure 1. The most widely adopted experimental protocol in this data set is based on a split (16 training and 9 test persons) that has been used in ([32]). In order to limit the complexity of the proposed codebook adaptation process, each action video has been represented by $N_k = 500$ vectors obtained by clustering the HOG/HOF descriptors calculated on the detected STIP video locations. This choice clearly accelerates the training process of the proposed method, since codebook adaptation is performed by using only N_k vectors. However, such an approach may decrease action recognition performance. In our preliminary experiments we have not witnessed major performance drops by adopting this approach, since the vectors obtained by applying clustering on the HOG/HOF descriptors of a video can be considered to be representative of the video descriptors.

3.2. The Hollywood2 Data Set

The Hollywood2 data set ([27]) consists of 1707 videos collected from 69 Hollywood movies. The actions appearing in the data set are: 'answering phone', 'driving car', 'eating', 'fighting', 'getting out of car', 'hand shaking', 'hugging', 'kissing', 'running', 'sitting down', 'sitting up' and 'standing up'. Example video frames are illustrated in Figure 2. The most widely adopted experimental protocol in this data set is based on a split (823 training and 884 test videos) that has been used in ([32]). In order to limit the complexity of the proposed codebook adaptation process, each action video has been represented by $N_k = 1000$ vectors obtained by clustering the HOG/HOF descriptors calculated on the detected STIP video locations.

3.3. The Ballet Data Set

The Ballet data set ([5]) consists of 44 real video sequences depicting three actors performing eight ballet movements and has been collected from an instructional ballet DVD. The actions appearing in the data set are: 'left-to-right hand

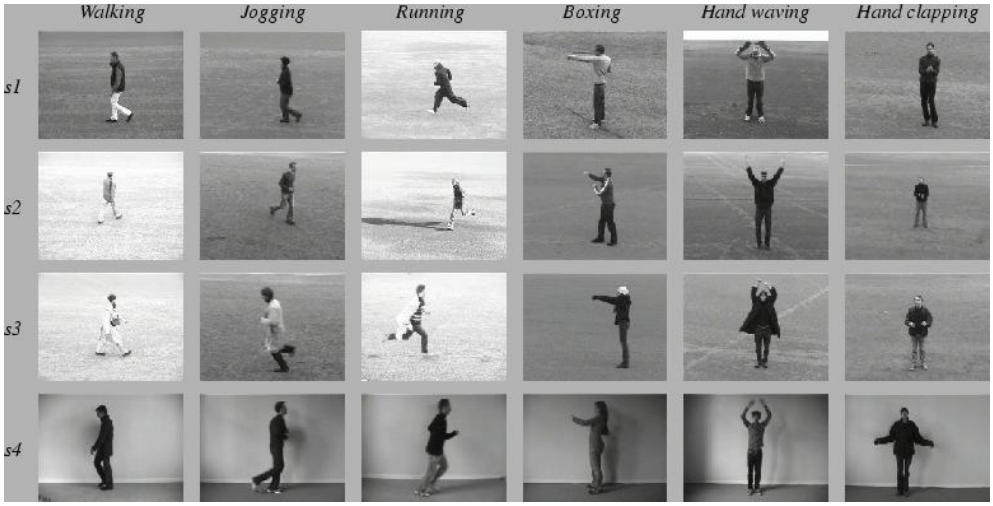

Figure 1: Example video frames of the KTH data set depicting instances of all the six actions from all the four experimental scenarios.

Figure 2: Example video frames of the Hollywood2 data set depicting instances of all the twelve actions.

opening', 'right-to-left hand opening', 'standing hand opening', 'leg swinging', 'jumping', 'turning', 'hopping' and 'standing still'. Example video frames are illustrated in Figure 3. The Leave-One-Video-Out cross-validation experimental protocol is, typically, used for this data set.

Figure 3: Video frames of the Ballet data set depicting instances of all the eight actions.

3.4. The i3DPost Data Set

The i3DPost data set ([9]) consists of 512 videos depicting eight persons from eight observation angles, each performing eight actions: 'walk', 'run', 'jump in place', 'jump forward', 'bend', 'fall down', 'sit on a chair' and 'wave one hand'. Example video frames depicting a person walking from all the eight cameras used in the data set are illustrated in Figure 4. The Leave-One-Person-Out cross-validation experimental protocol is, typically, used for this data set. In order to fuse the information coming from all available cameras, the action videos depicting the same test action instance from different viewing angles have been classified independently and the obtained action class labels were fused by using a simple majority voting fusion scheme, similar to ([15]).

3.5. The Facial Expression Data Set

The facial expression data set ([5]) consists of 192 videos depicting two persons expressing six different emotions under two lighting conditions. The expressions appearing in the data set are: 'anger', 'disgust', 'fear', 'joy', 'sadness' and 'surprise'. The persons always start with a neutral expression, show the emotion and return to neutral. Example video frames depicting all the expressions appearing in the data set are illustrated in Figure 5. Four experimental protocols have been used for this data set: 'Same person & lighting' (SpSl), 'Same person, Different lighting' (SpDl), 'Different person, Same lighting' (DpSl) and 'Different person & lighting' (DpDl).

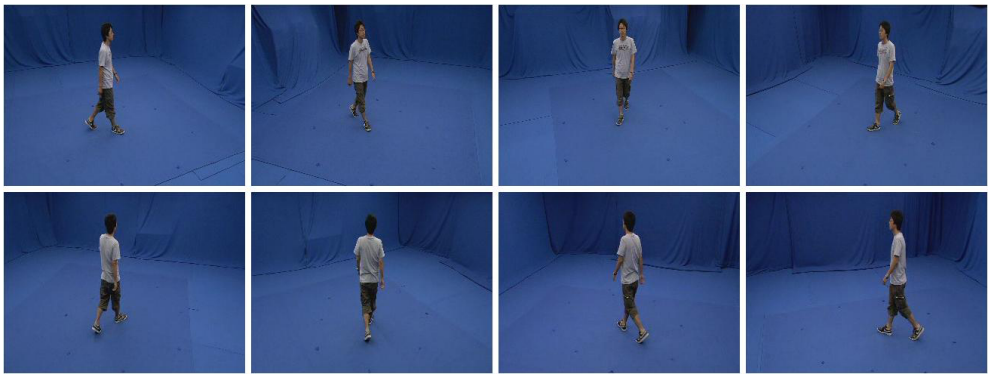

Figure 4: Example video frames of the i3DPost data set depicting a person walking from eight viewing angles.

Figure 5: Video frames of the facial expression data set depicting the emotion apex of all the six emotions.

3.6. Experimental Results

Tables 1 - 6 illustrate the performance obtained by using the proposed discriminant BoWs-based action video representation on the KTH, Hollywood2, Ballet, i3DPost and the facial expression data sets, respectively. As can be seen in these Tables, the adoption of a supervised learning process on the codebook adaptation enhances performance, when compared to the standard BoWs-based action video representation. Table 1 illustrates the classification rates obtained by using different values of N_k in KTH database. As can be seen, a value of $N_k = 500$ provides satisfactory performance. In Tables 2 - 6 we, also, compare the performance of the proposed action video recognition approach with that of some state-of-the-art methods, recently proposed in the literature.

In the KTH data set, the use of the BoWs-based action video representation led to a classification rate equal to 88.89%. By adopting the proposed DBoWs-based action video representation, an increased action classification rate, equal to 92.13%, has been obtained. As can be seen in Table 2, the proposed method outperforms other state-of-the-art methods employing low-level action video representations. In this Table, we also provide the performance of some state-of-the-art methods evaluating their performance on the KTH data set that exploit mid-level and high-level action video representations. It can be seen that the proposed approach exploiting a low-level action video representation provides performance comparable with that of the the two methods exploiting a mid-level action video representation. In addition, its performance is comparable with that of one method exploiting high-level action video representation, while it is inferior to that of the remaining three methods exploiting high-level representations. However, the calculation of high-level representations is computationally demanding, compared to the calculation of low-level ones and, thus, a comparison between the two approaches in terms of only the obtained action recognition performance is not fair.

Table 1: Performance on KTH for different values of N_k.

	200	300	400	500	600
BoWs	85.19%	86.11%	87.5%	88.89%	88.89%
DBoWs	**87.96%**	**89.35%**	**90.74%**	**92.13%**	**92.13%**

In the Hollywood2 data set, the use of the BoWs-based action video representation led to a performance equal to 41.5%. By adopting the proposed DBoWs-

Table 2: Classification rates on the KTH data set.

	Representation	Performance
[46]	low-level	82%
[19]	low-level	84.3%
[44]	low-level	87.3%
[18]	low-level	90.57%
[30]	low-level	91.1%
[22]	low-level	91.8%
[7]	mid-level	90.5%
[39]	**mid-level**	**92.4%**
[48]	high-level	93.25%
[24]	high-level	93.9%
[8]	high-level	94.5%
[20]	high-level	94.5%
[31]	high-level	98.9%
[16]	**high-level**	**99.54%**
BoWs	low-level	88.89%
DBoWs	**low-level**	**92.13%**

based action video representation, a performance equal to 45.8% has been obtained. As can be seen in Table 3, the methods evaluated on this data set can be roughly divided based on the employed action video description. On the one hand, methods employing densely sampled descriptors for action video representation, i.e., Cuboids, Dense and Regions, have been shown to outperform the ones employing descriptors calculated on STIPs, i.e., Harris3D and Hessian. On the other hand, it can be seen that the adoption of a higher number of descriptors, each describing a different action property, enhances action classification performance.

As can be seen in Table 3, the best approach in this data set is that of ([34]), which is based on an action video representation exploiting information appearing on the trajectories of densely sampled video frame interest points and is described by calculating multiple (five) descriptors, i.e., HOG, HOF, MBHx, MBHy and Traj, along these trajectories. However, this approach has the following two disadvantages: 1) It is very computationally demanding and 2) it requires multiple descriptors (as well as a good descriptor combination scheme) in order to achieve better performance, when compared to action video representations exploiting STIP-based information. It can be seen in Table 3 that in the cases where the Dense Trajectory-based and the Cuboid-based video descriptions exploit HOG/HOF descriptors, their performance is comparable with that of the

proposed method exploiting STIP-based visual information. Taking into account that STIP-based video representations are much faster, when compared to ones exploiting densely-sampled visual information, we can see that a comparison between the two approaches, in terms of only the obtained performance, is not fair.

In ([34]), the adopted action video representation exploits the BoW model for each of the five descriptor types (i.e., HOG, HOF, MBHx, MBHy and Traj) and combines the obtained five (single-channel) action video representations on the classification process by using a multi-channel kernel Support Vector Machine approach. Thus, the action video representation of ([34]) employs multiple (five) codebooks (each calculated by applying K-Means clustering on the feature vectors calculated for the training videos corresponding to one descriptor type). From this, it can be seen that an extension of the proposed approach for unified codebook adaptation, in the case where multiple descriptor types are exploited, would probably enhance the performance of such methods. Overall, it can be seen that the proposed DBoWs action representation provides comparable performance with that of other state-of-the-art methods employing action video representations evaluated on STIPs.

Table 3: Classification rates on the Hollywood2 data set.

	Representation	Performance
[27]	Harris3D+HOG+HOF	32.4%
[27]	Harris3D+HOG+HOF+SIFT	32.6%
[27]	Harris3D+HOG+HOF+SIFT+Scene	35.5%
[35]	Harris3D+HOG/HOF	**45.2%**
[35]	Hessian+HOG/HOF	**46%**
[35]	Cuboids+HOG/HOF	46.2%
[24]	Cuboids+ISA	53.3%
[34]	Dense+HOG	41.5%
[34]	Dense+HOF	50.8%
[35]	Dense+HOG/HOF	47.4%
[34]	Dense+HOG+HOF+MBH+Traj	**58.3%**
[2]	Regions+HOG+HOF+OF	41.34%
BoWs	Harris3D+HOG/HOF	41.5%
DBoWs	Harris3D+HOG/HOF	**45.8%**

In the Ballet data set, the use of the BoWs-based action video representation led to an action classification rate equal to 86.3%. The use of the proposed DBoWs-based action video representation increased the action classification rate

to 91.1%, which is comparable to the performance of the two competing methods presented in Table 4.

Table 4: Classification rates on the Ballet data set.

[10]	[38]	BoWs	**DBoWs**
91.1%	**91.3%**	86.3%	**91.1%**

In the i3DPost data set, the use of the BoWs-based action representation led to an action classification rate equal to 95.31%, while the adoption of the proposed DBoWs-based action video representation led to an action classification rate equal to 98.44%, equal to that of ([11]). We should note here that the method in ([11]) employs a computationally expensive 4D optical flow-based action video representation and, thus, its operation is slower compared with the proposed one in the test phase.

Table 5: Classification rates on the i3DPost data set.

[15]	[11]	BoWs	**DBoWs**
94.87%	**98.44%**	95.31%	**98.44%**

Finally, in the facial expression data set, it can be seen that the proposed DBoWs-based action representation, clearly, outperforms the BoWs-based action representation providing up to 5% increase on the classification performance. The proposed method, clearly, outperforms the method in ([5]), while it provides comparable classification rates with that in ([10]). However, we should note that the method in [10] involves an optimization process during testing and, thus, its operation is slower compared to the BoWs-based action classification approach.

4. Conclusions

In this paper we proposed a novel framework for human action recognition unifying discriminative codebook generation and discriminant subspace learning. An iterative optimization scheme has been proposed for sequential discriminant BoWs-based action video representation calculation and codebook adaptation based on action classes discrimination. The proposed framework is able to,

Table 6: Classification rates on the facial expression data set for different experimental protocols.

	SpSl	SpDl	DpSl	DpDl
[5]	97.9%	89.6%	75%	69.8%
[10]	**100%**	**93.7%**	**91.7%**	72.9%
BoWs	98.96%	87.5%	78.65%	73.44%
DBoWs	**100%**	92.71%	83.33%	**77.6%**

naturally, incorporate several (linear or non-linear) discrimination criteria for discriminant BoWs-based action video representation. Experiments conducted by employing the LDA criterion on five publicly available data sets aiming at different application scenarios demonstrate that the proposed unified approach increases the codebook discriminative ability providing enhanced performance.

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).

[1] Belkin, M., Niyogi, P., 2001. Laplacian eigenmaps and spectral techniques for embedding and clustering. Advances in Neural Information Processing Systems 14, 585–591.

[2] Bilen, H., Namboodiri, V., Gool, L., 2011. Action recognition: a region based approach. Applications of Computer Vision .

[3] Boweils, S., Saul, L., 2000. Nonlinear dimensionality reduction by locally linear embedding. Science 290, 2323–2326.

[4] Csurka, G., Bray, C., Dance, C., Fan, L., 2004. Visual categorization with bags of keypoints. European Conference on Computer Vision .

[5] Dollar, P., Rabaud, V., Cottrell, G., Belongie, S., 2005. Behavior recognition via sparse spatio-temporal features. International Workshop on Visual Surveillance and Performance Evaluation of Tracking and Surveillance .

[6] Duda, R., Hart, P., Stork, D., 2000. Pattern classification, 2nd ed. Wiley-Interscience.

[7] Fathi, A., Mori, G., 2008. Action recognition by learning mid-level motion features. Computer Vision and Pattern Recognition .

[8] Gilbert, A., Illingworth, J., Bowden, R., 2009. Fast realistic multi-action recognition using mined dense spatio-temporal features. Computer Vision and Pattern Recognition .

[9] Gkalelis, N., KIm, H., Hilton, A., Nikolaidis, N., Pitas, I., 2009. The i3dpost multi-view and 3d human action/interaction database. Conference on Visual Media Production , 159–168.

[10] Guha, T., Ward, R., 2011. Learning sparse representations for human action recognition. IEEE Transactions on Pattern Analysis and Machine Intelligence 34, 1576–1588.

[11] Holte, M., Chakraborty, B., Gonzalez, J., Moeslund, T., 2012. A local 3d motion descriptor for multi-view human action recognition from 4d spatio-temporal interest points. IEEE Journal of Selected Topics in Signal Processing 6, 553–565.

[12] Huang, G., Zhou, H., Ding, X., Zhang, R., 2012. Extreme learning machine for regression and multiclass classification. IEEE Transactions on Systems, Man, and Cybernetics, Part B: Cybernetics 42, 513–529.

[13] Huang, Y., Huang, Y., Tan, T., 2011. Salient coding for image classification. Computer Vision and Pattern Recognition .

[14] Huang, Y., Wu, Z., Wang, L., Tan, T., 2014. Feature coding in image classification: A comprehensive study. IEEE Transactions on Pattern Analysis and Machine Intelligence 36, 493–506.

[15] Iosifidis, A., Tefas, A., Pitas, I., 2012. View-invariant action recognition based on artificial neural networks. IEEE Transactions on Neural Networks and Learning Systems 23, 412–425.

[16] Iosifidis, A., Tefas, A., Pitas, I., 2014. Regularized extreme learning machine for multi-view semi-supervised action recognition. Neurocomputing DOI: 10.1016/j.neucom.2014.05.036.

[17] Jia, Y., Nie, F., Zhang, C., 2009. Trace ratio problem revisited. IEEE Transactions on Neural Networks 20, 729–735.

[18] Kaaniche, M., Bremond, F., 2010. Gesture recognition by learning local motion signatures. Computer Vision and Pattern Recognition .

[19] Klaser, A., Marszalek, M., Schmid, S., 2009. A spatio-temporal descriptor based on 3d-gradients. British Machine Vision Conference , 995–1004.

[20] Kovashka, A., Grauman, K., 2010. Learning a hierarchy of discriminative spacetime neighborhood features for human action recognition. Computer Vision and Pattern Recognition .

[21] Laptev, I., Lindeberg, T., 2003. Space-time interest points. International Conference on Computer Vision , 432–439.

[22] Laptev, I., Marszalek, M., Schmid, C., Rozenfeld, B., 2008. Learning realistic human actions from movies. Computer Vision and Pattern Recognition .

[23] Laptev, J., 2005. On space-time interest points. International Journal on Computer Vision 64, 107–123.

[24] Le, Q., Zou, W., Yeung, S., Ng, A., 2011. Learning hierarchical invariant spatio-temporal features for action recognition with independent subspace analysis. Computer Vision and Pattern Recognition .

[25] Lian, X., Li, Z., Lu, B., Zhang, L., 2010. Max-margin dictionary learning for multiclass image categorization. European Conference on Computer Vision .

[26] Lu, J., Plataniotis, K., Venetsanopoulos, A., 2003. Face recognition using lda-based algorithms. IEEE Transactions on Neural Networks 14, 195–200.

[27] Marszalek, M., Laptev, I., Schmid, C., 2009. Actions in context. Computer Vision and Pattern Recognition .

[28] Perronnin, F., Dance, C., 2007. Fisher kernels on visual vocabularies for image categorization. Computer Vision and Pattern Recognition .

[29] Perronnin, F., Dance, C., Csurka, G., Bressan, M., 2005. Adapted vocacbularies for generic visual categorization. IEEE European Conference on Computer Vision .

[30] Ryoo, M., Aggarwal, J., 2009. Spatio-temporal relationship match: Video structure comparison for recognition of complex human activities. International Conference on Computer Vision .

[31] Sadanand, S., Corso, J., 2012. Action bank: A high-level representation of activity in video. Computer Vision and Pattern Recognition .

[32] Schuldt, C., Laptev, I., Caputo, B., 2004. Recognizing human actions: A local svm approach. International Conference on Pattern Recognition , 32–36.

[33] Tenenbaum, J., Silva, V., Langford, J., 2000. A global geometric framework for nonlinear dimensionality reduction. Science 290, 2319–2323.

[34] Wang, H., Klaser, A., Schmid, C., Liu, C., 2011. Action recognition by dense trajectories. Computer Vision and Pattern Recognition .

[35] Wang, H., Ullah, M., Klaser, A., Laptev, I., Schmid, C., 2009. Evaluation of local spatio-temporal features for action recognition. British Machine Vision Conference , 1–11.

[36] Wang, H., Yan, S., Xu, D., Tang, X., Huang, T., 2007. Trace ratio vs. ratio trace for dimensionality reduction. Computer Vision and Pattern Recognition .

[37] Wang, J., Yang, J., Yu, K., Lv, F., Huang, T., Gong, Y., 2010. Locality-constrained linear coding for image classication. Computer Vision and Pattern Recognition .

[38] Wang, Y., Mori, G., 2009. Human action recognition by semilatent topic models. IEEE Transactions on Pattern Analysis and Machine Intelligence 31, 1762–1774.

[39] Weinland, M., Ozuysal, D., Fua, P., 2010. Making action recognition robust to occlusions and viewpoing changes. European Conference on Computer Vision Workshops .

[40] Winn, J., Criminisi, A., Minka, T., 2005. Object categorizations by learned universal visual dictionary. International Conference on Computer Vision .

[41] Yan, S., Xu, D., Zhang, B., Zhang, H., Yang, Q., Lin, S., 2007. Graph embedding and extentions: A general framework for dimensionality reduction. IEEE Transactions on Pattern Analysis and Machine Intelligence 29, 40–51.

[42] Yang, J., Yu, K., Y., G., Huang, T., 2009. Linear spatial pyramid matching using sparse coding for image classication. Computer Vision and Pattern Recognition .

[43] Yang, L., Jin, R., Sukthankar, R., Jurie, F., 2008. Unifying discriminative visual codebook generation with classifier training for object category recognition. Computer Vision and Pattern Recognition .

[44] Yang, W., Wang, Y., G., M., 2010. Recognizing human actions from still images with latent poses. Computer Vision and Pattern Recognition .

[45] Yeffet, L., Wolf, L., 2009. Local trinary patterns for human action recognition. International Conference on Computer Vision , 492–497.

[46] Yin, J., Meng, Y., 2010. Human activity recognition in video using a hierarchical probabilistic latent model. Computer Vision and Pattern Recognition .

[47] Yu, K., Zhang, T., 2010. Improving local coordinate coding using loca tangents. International Conference on Machine Learning .

[48] Zhou, G., Wang, Q., 2012. Atomic action features: A new feature for action recognition. European Conference on Computer Vision Workshops .