Multi-view Regularized Extreme Learning Machine 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 an extension of the ELM algorithm that is able to exploit multiple action representations. This is achieved by incorporating proper regularization terms in the ELM optimization problem. In order to determine both optimized network weights and action representation combination weights, we propose an iterative optimization process. The proposed algorithm has been evaluated by using the state-of-the-art action video representation on three publicly available action recognition databases, where its performance has been compared with that of two commonly used video representation combination approaches, i.e., the vector concatenation before learning and the combination of classification outcomes based on learning on each view independently.

Keywords: Extreme Learning Machine, Multi-view Learning, Single-hidden Layer Feedforward networks, Human Action Recognition

1 Introduction

Human action recognition is intensively studied to date due to its importance in many real-life applications, like intelligent visual surveillance, human-computer interaction, automatic assistance in healthcare of the elderly for independent living and video games, to name a few. Early human action recognition methods have been investigating a restricted recognition problem. According to this problem, action recognition refers to the recognition of simple motion patterns, like a walking step, performed by one person in a scene containing a simple background [1, 2]. Based on this scenario, most such methods describe actions as series of successive human body poses, represented by human body silhouettes evaluated by applying video frame segmentation techniques or background subtraction. However, such an approach is impractical in most real-life applications, where actions are performed in scenes having a complex background, which may contain multiple persons as well. In addition, actions may be observed by one or multiple, possibly moving, camera(s), capturing the action from arbitrary viewing angles. The above mentioned problem is usually referred to as 'action recognition in the wild' and is the one that is currently addressed by most action recognition methods.

The state-of-the-art approach in this, unrestricted, problem describes actions by employing the Bag-of-Features (BoF) model [3]. According to this model, sets of shape and/or motion descriptors are evaluated in spatiotemporal locations of interest of a video and multiple (one for each descriptor type) video representations are obtained by applying (hard or soft) vector quantization by employing sets of descriptor prototypes, referred to as codebooks. The descriptors that provide the current state-of-the-art performance in most action recognition databases are: the Histogram of Oriented Gradients (HOG), the Histogram of Optical Flow (HOF) and the Motion Boundary Histogram (MBH). These descriptors are evaluated on the trajectories of densely sampled video frame interest points, which are tracked for a number of consecutive video frames. The normalized location of the tracked interest points is also employed in order to form another descriptor type, referred to as Trajectory (Traj).

Since different descriptor types express different properties of interest for actions, it is not surprising the fact that a combined action representation exploiting all the above mentioned (single-descriptor based) video representations results to increased performance [3]. Such combined action representations are usually obtained by employing unsupervised combination schemes, like the use of concatenated representations (either on the descriptor, or on the video representation level), or by combining the outcomes of classifiers trained on different representation types [4], e.g., by using the mean classifier outcome in the case of SLFN networks [5]. However, the adoption of such combination schemes may decrease the generalization ability of the adopted classification scheme, since all the available action representations equally contribute to the classification result. Thus, supervised combination schemes are required in order to properly combine the information provided by different descriptor types.

Extreme Learning Machine (ELM) [6] is a, relatively, new algorithm for fast Single-hidden Layer Feedforward Neural (SLFN) networks training, requiring low human supervision. Conventional SLFN training algorithms require adjustment of the network weights and the bias values, using a parameter optimization approach, like gradient descent. However, gradient descent learning techniques are, generally, slow and may lead to local minima. In ELM, the input weights and the hidden layer bias values are randomly chosen, while the network output weights are analytically calculated. By using a sufficiently large number of hidden layer neurons, the ELM classification scheme can be thought of as being a non-linear mapping of the training data on a high-dimensional feature space, called ELM space hereafter, followed by linear data projection and classification. ELM not only tends to reach a small training error, but also a small norm of output weights, indicating good generalization performance [7]. ELM has been successfully applied to many classification problems, including human action recognition [8–11].

In this paper we employ the ELM algorithm in order to perform human action recognition from videos. We adopt the state-of-the-art BoF-based action representation described above [3], in order to describe videos depicting actions, called action videos hereafter, by multiple vectors (one for each descriptor type), each describing different properties of interest for actions. In order to properly combine the information provided by different descriptor types, we extend the ELM algorithm in order to incorporate multiple video representations in its optimization process. An iterative optimization scheme is proposed to this end, where the contribution of each video representation is appropriately weighted. We evaluate the performance of the proposed algorithm on three publicly available databases, where we compare it with that of two commonly adopted video representation combination schemes.

The proposed approach is closely related to Multiple Kernel Learning (MKL) [16–18]. MKL methods aim at the determination of an "improved" feature space for nonlinear data mapping. This is usually approached by employing a linear combination of a set of kernel functions followed by the optimization of an objective function by employing the training data for the determination of the kernel combination weights. A recent review on MKL methods can be found in [19]. Our work differs from MKL in that in the proposed approach the feature spaces employed for nonlinear data mapping are determined by employing randomly chosen network weights. After obtaining the data representations in the high-dimensional ELM space, we aim at optimally weighting the contribution of each data representation in the outputs of the combined network outputs.

The remainder of the paper is structured as follows. In Section 2, we briefly describe the ELM algorithm. The proposed Multi-view Regularized ELM (MVRELM) algorithm is described in Section 3. Experimental results evaluating its performance are illustrated in Section 4. Finally, conclusions are drawn in Section 5.

2 Extreme Learning Machine

ELM has been proposed for single-view classification [6]. Let $\mathbf{x}_i$ and $c_i,\ i = 1, ..., N$ be a set of labeled action vectors and the corresponding action class labels, respectively. We would like to employ them in order to train a SLFN network. For a classification problem involving the D-dimensional action vectors $\mathbf{x}_i$, each belonging to one of the C action classes, the network should contain D input, H hidden and C output neurons. The number of the network hidden layer neurons is, typically, chosen to be higher than the number of action classes, i.e., $H \gg C$. The network target vectors $\mathbf{t}_i = [t_{i1}, ..., t_{iC}]^T$, each corresponding to one labeled action vector $\mathbf{x}_i$, are set to $t_{ij} = 1$ for vectors belonging to action class j, i.e., when $c_i = j$, and to $t_{ij} = -1$ otherwise.

In ELM, the network input weights $\mathbf{W}_{in} \in \mathbb{R}^{D \times H}$ and the hidden layer bias values $\mathbf{b} \in \mathbb{R}^H$ are randomly chosen, while the output weights $\mathbf{W}_{out} \in \mathbb{R}^{H \times C}$ are analytically calculated. Let $\mathbf{v}_j$ denote the j-th column of $\mathbf{W}_{in}$, $\mathbf{u}_k$ the k-th column of $\mathbf{W}_{out}$ and u_{kj} be the j-th element of $\mathbf{u}_k$. For a given hidden layer activation function $\Phi(\cdot)$ and by using a linear activation function for the output neurons, the output $\mathbf{o}_i = [o_{i1}, \ldots, o_{iC}]^T$ of the ELM network corresponding to training action vector $\mathbf{s}_i$ is given by:

$$o_{ik} = \sum_{j=1}^{H} u_{kj}\ \Phi(\mathbf{v}_j, b_j, \mathbf{x}_i), \quad k = 1, ..., C. \tag{1}$$

Many activation functions $\Phi(\cdot)$ can be employed for the calculation of the hidden layer output, such as sigmoid, sine, Gaussian, hard-limiting and Radial Basis (RBF) functions. The most popular choices are the sigmoid and the RBF functions, i.e.:

$$\Phi_{sigmoid}(\mathbf{v}_j, b_j, \mathbf{x}_i) = \frac{1}{1 + exp\left(-(\mathbf{v}_j^T\ \mathbf{x}_i + b_j)\right)}, \tag{2}$$

$$\Phi_{RBF}(\mathbf{v}_j, b_j, \mathbf{x}_i) = exp\left(-b_j \|\mathbf{x}_i - \mathbf{v}_j\|_2^2\right), \tag{3}$$

leading to MLP and RBF networks, respectively. However, since we are interested in BoF-based human action recognition, in this work we exploit the χ^2 activation function:

$$\Phi_{\chi^2}(\mathbf{v}_j, b, \mathbf{x}_i) = exp\left(- \frac{1}{2b_j} \sum_{d=1}^{D} \frac{(\mathbf{x}_{id} - \mathbf{v}_{jd})^2}{\mathbf{x}_{id} + \mathbf{v}_{jd}} \right), \tag{4}$$

which has been found to outperform both the above two alternative choices.

By storing the hidden layer neuron outputs in a matrix $\mathbf{\Phi}$:

$$\mathbf{\Phi} = \begin{bmatrix} \Phi(\mathbf{v}_1, b_1, \mathbf{x}_1) & \cdots & \Phi(\mathbf{v}_1, b_1, \mathbf{x}_l) \\ \ldots & \ddots & \ldots \\ \Phi(\mathbf{v}_H, b_H, \mathbf{x}_1) & \cdots & \Phi(\mathbf{v}_H, b_H, \mathbf{x}_l) \end{bmatrix}, \tag{5}$$

equation (1) can be written in a matrix form as $\mathbf{O} = \mathbf{W}_{out}^T \mathbf{\Phi}$. Finally, by assuming that the predicted network outputs $\mathbf{O}$ are equal to the desired ones, i.e., $\mathbf{o}_i = \mathbf{t}_i,\ i = 1, ..., l$, $\mathbf{W}_{out}$ can be analytically calculated by solving for:

$$\mathbf{W}_{out}^T \mathbf{\Phi} = \mathbf{T}, \tag{6}$$

where $\mathbf{T} = [\mathbf{t}_1, \ldots, \mathbf{t}_l]$ is a matrix containing the network target vectors. Using (6), the network output weights minimizing $\|\mathbf{W}_{out}^T \mathbf{\Phi} - \mathbf{T}\|_F$ are given by:

$$\mathbf{W}_{out} = \mathbf{\Phi}^{\dagger}\, \mathbf{T}^T, \tag{7}$$

where $\|\mathbf{X}\|_F$ is the Frobenius norm of $\mathbf{X}$ and $\mathbf{\Phi}^{\dagger} = \left(\mathbf{\Phi}\mathbf{\Phi}^T\right)^{-1} \mathbf{\Phi}$ is the generalized pseudo-inverse of $\mathbf{\Phi}^T$. By observing (8), it can be seen that this equation can be used only in the cases where the matrix $\mathbf{B} = \mathbf{\Phi}\mathbf{\Phi}^T$ is invertible, i.e., when $N > D$. A regularized version of the ELM algorithm addressing this issue has been proposed in [12], where the network output weights are obtained, according to a regularization paramter $c > 0$, by:

$$\mathbf{W}_{out} = \left(\mathbf{\Phi}\mathbf{\Phi}^T + \frac{1}{c}\mathbf{I}\right)^{-1} \mathbf{\Phi}\, \mathbf{T}^T. \tag{8}$$

After calculating the network output weights $\mathbf{W}_{out}$, a test action vector $\mathbf{x}_t$ can be introduced to the trained network and be classified to the action class corresponding to the maximal network output, i.e.:

$$c_t = arg \max_j o_{tj},\ j = 1, ..., C. \tag{9}$$

3 Multi-view Regularized Extreme Learning Machine

The above described ELM algorithm can be employed for single-view (i.e., single-representation) action classification. In this section, we describe an optimization process that can be used for multi-view action classification, i.e., in the cases where each action video is represented by multiple action vectors $\mathbf{x}_i^v,\ v = 1, \ldots, V$.

Let us assume that the N training action videos are represented by the corresponding action vectors $\mathbf{x}_i^v \in \mathbb{R}^{D_v}$, $i = 1, \dots, l, \dots, N$, $v = 1, \dots, V$. We would like to employ them, in order to train V SLFN networks, each operating on one view. To this end we map the action vectors of each view v to one ELM space $\mathbb{R}^{H_v}$, by using randomly chosen input weights $\mathbf{W}_{in}^v \in \mathbb{R}^{D_v \times H_v}$ and input layer bias values $\mathbf{b}^v \in \mathbb{R}^{H_v}$. H_v is the dimensionality of the ELM space related to view v.

In order to determine both the networks output weights $\mathbf{W}_{out}^v \in \mathbb{R}^{H_v \times C}$ and appropriate view combination weights $\gamma \in \mathbb{R}^V$ we can formulate the following optimization problem:

$$\textbf{Minimize: } \mathcal{J} = \frac{1}{2}\sum_{v=1}^{V} \|\mathbf{W}_{out}^v\|_F^2 + \frac{c}{2}\sum_{i=1}^{N} \|\boldsymbol{\xi}_i\|_2^2 \tag{10}$$

$$\textbf{Subject to: } \left(\sum_{v=1}^{V} \gamma_v \mathbf{W}_{out}^{v\,T} \boldsymbol{\phi}_i^v\right) - \mathbf{t}_i = \boldsymbol{\xi}_i, \;\; i = 1, ..., N, \tag{11}$$

$$\|\boldsymbol{\gamma}\|_2^2 = 1, \tag{12}$$

where $\mathbf{t}_i \in \mathbb{R}^C$, $\boldsymbol{\phi}_i^v \in \mathbb{R}^{H_v}$ are target vector of the i-th action video and the representation of $\mathbf{x}_i^v$ in the corresponding ELM space, respectively. $\boldsymbol{\xi}_i \in \mathbb{R}^C$ is the error vector related to the i-th action video and c is a regularization parameter expressing the importance of the training error in the optimization process. Alternatively, we could employ the constraints $\gamma_v \geq 0, \; v = 1, \dots, V$ and $\sum_{v=1}^{V} \gamma_v = 1$ [19].

By setting the representations of $\mathbf{x}_i^v$ in the corresponding ELM space in a matrix $\mathbf{\Phi}^v = [\boldsymbol{\phi}_1^v, \dots, \boldsymbol{\phi}_N^v]$, the network responses corresponding to the entire training set are given by:

$$\mathbf{O} = \sum_{v=1}^{V} \gamma_v \mathbf{W}_{out}^{v\,T} \mathbf{\Phi}^v. \tag{13}$$

By substituting (11) in (10) and taking the equivalent dual problem, we obtain:

$$\begin{aligned}
\mathcal{J}_D &= \frac{1}{2}\sum_{v=1}^{V} \|\mathbf{W}_{out}^v\|_F^2 + \frac{c}{2}\sum_{i=1}^{N} \| \left(\sum_{v=1}^{V} \gamma_v \mathbf{W}_{out}^{v\,T} \boldsymbol{\phi}_i^v\right) - \mathbf{t}_i\|_2^2 + \frac{\lambda}{2}\|\boldsymbol{\gamma}\|_2^2 \\
&= \frac{1}{2}\sum_{v=1}^{V} \|\mathbf{W}_{out}^v\|_F^2 + \frac{c}{2}\| \left(\sum_{v=1}^{V} \gamma_v \mathbf{W}_{out}^{v\,T} \mathbf{\Phi}^v\right) - \mathbf{T}\|_F^2 + \frac{\lambda}{2}\|\boldsymbol{\gamma}\|_2^2 \\
&= \frac{1}{2}\sum_{v=1}^{V} \|\mathbf{W}_{out}^v\|_F^2 + \frac{c}{2}\boldsymbol{\gamma}^T \mathbf{P} \boldsymbol{\gamma} - c\mathbf{r}^T \boldsymbol{\gamma} + \frac{c}{2} tr\left(\mathbf{T}^T \mathbf{T}\right) + \frac{\lambda}{2}\boldsymbol{\gamma}^T \boldsymbol{\gamma},
\end{aligned} \tag{14}$$

where $\mathbf{P} \in \mathbb{R}^{V \times V}$ is a matrix having its elements equal to $[\mathbf{P}]_{kl} = tr\left(\mathbf{W}_{out}^{k\,T} \mathbf{\Phi}^k \mathbf{\Phi}^{l\,T} \mathbf{W}_{out}^l\right)$ and $\mathbf{r} \in \mathbb{R}^V$ is a vector having its elements equal to $\mathbf{r}_v = tr\left(\mathbf{T}^T \mathbf{W}_{out}^{v\,T} \mathbf{\Phi}^v\right)$. By solving for $\frac{\vartheta \mathcal{J}_D(\boldsymbol{\gamma})}{\vartheta \boldsymbol{\gamma}} = 0$, $\boldsymbol{\gamma}$ is given by:

$$\boldsymbol{\gamma} = \left(\mathbf{P} + \frac{\lambda}{c}\mathbf{I}\right)^{-1} \mathbf{r}. \tag{15}$$

By substituting (11) in (10) and taking the equivalent dual problem, we can also obtain:

$$\begin{aligned}\mathcal{J}_D &= \frac{1}{2}\sum_{v=1}^{V}\|\mathbf{W}_{out}^{v}\|_F^2 + \frac{c}{2}\sum_{i=1}^{N}\|\left(\sum_{v=1}^{V}\gamma_v \mathbf{W}_{out}^{v\,T}\boldsymbol{\phi}_i^v\right) - \mathbf{t}_i\|_2^2 + \frac{\lambda}{2}\|\boldsymbol{\gamma}\|_2^2 \\ &= \frac{1}{2}\sum_{v=1}^{V}\|\mathbf{W}_{out}^{v}\|_F^2 + \frac{c}{2}\|\left(\sum_{v=1}^{V}\gamma_v \mathbf{W}_{out}^{v\,T}\boldsymbol{\Phi}^v\right) - \mathbf{T}\|_F^2 + \frac{\lambda}{2}\|\boldsymbol{\gamma}\|_2^2 \\ &= \frac{1}{2}\sum_{v=1}^{V} tr\left(\mathbf{W}_{out}^{v\,T}\mathbf{W}_{out}^{v}\right) + \frac{c}{2} tr\left(\sum_{v=1}^{V}\sum_{l=1}^{V}\gamma_v\gamma_l \mathbf{W}_{out}^{v\,T}\boldsymbol{\Phi}^v\boldsymbol{\Phi}^{l\,T}\mathbf{W}_{out}^{l}\right) \\ &- c\sum_{v=1}^{V} tr\left(\gamma_v \mathbf{W}_{out}^{v\,T}\boldsymbol{\Phi}^v\mathbf{T}^T\right) + \frac{c}{2} tr\left(\mathbf{T}^T\mathbf{T}\right) + \frac{\lambda}{2}\boldsymbol{\gamma}^T\boldsymbol{\gamma}.\end{aligned}$$

By solving for $\frac{\vartheta \mathcal{J}_D(\mathbf{W}_{out}^v)}{\vartheta \mathbf{W}_{out}^v} = 0$, $\mathbf{W}_{out}^v$ is given by:

$$\mathbf{W}_{out}^v = \left(\frac{2}{c\gamma_k}\mathbf{I} + \gamma_k \boldsymbol{\Phi}^v \boldsymbol{\Phi}^{v\,T}\right)^{-1} \boldsymbol{\Phi}^v (2\mathbf{T} - \mathbf{O})^T, \tag{16}$$

As can be observed in (15), (16), $\boldsymbol{\gamma}$ is a function of $\mathbf{W}_{out}^v$, $v = 1, \dots, V$ and $\mathbf{W}_{out}^v$ is a function of $\boldsymbol{\gamma}$. Thus, a direct optimization of $\mathcal{J}_D$ with respect to both $\{\gamma_v, \mathbf{W}_{out}^v\}_{v=1}^V$ is intractable. Therefore, we propose an iterative optimization scheme formed by two optimization steps. In the following, we introduce a index t denoting the iteration of the proposed iterative optimization scheme.

Let us denote by $\mathbf{W}_{out,t}^v$, $\boldsymbol{\gamma}_t$ the network output and combination weights determined for the iteration t, respectively. We initialize $\mathbf{W}_{out,1}^v$ by using (8) and set $\gamma_{1,v} = 1/V$ for all the action video representations $v = 1, \dots, V$. By using $\boldsymbol{\gamma}_t$, the network output weights $\mathbf{W}_{out,t+1}^v$ are updated by using (16). After the calculation of $\mathbf{W}_{out,t+1}^v$, $\boldsymbol{\gamma}_{t+1}$ is obtained by using (15). The above described process is terminated when $(\mathcal{J}_D(t) - \mathcal{J}_D(t+1))/\mathcal{J}_D(t) < \epsilon$, where ϵ is a small positive value equal to $\epsilon = 10^{-10}$ in our experiments. Since each optimization step corresponds to a convex optimization problem, the above described process is guaranteed to converge in a local minimum of $\mathcal{J}$.

After the determination of the set $\{\gamma_v, \mathbf{W}_{out}^v\}_{v=1}^V$, the network response for a given set of action vectors $\mathbf{x}_l \in \mathbb{R}^D$ is given by:

$$\mathbf{o}_l = \sum_{v=1}^{V} \gamma v \mathbf{W}_{out}^{v\,T}\boldsymbol{\phi}_l^v. \tag{17}$$

4 Experiments

In this section, we present experiments conducted in order to evaluate the performance of the proposed MVELM algorithms. We have employed three publicly available databases, namely the Hollywood2, the Olympic Sports and the Hollywood 3D databases. In the following subsections, we describe the databases and evaluation measures used in our experiments. Experimental results are provided in subsection 4.4.

We employ the state-of-the-art action video representation proposed in [3], where each video is represented by five 4000-dimensional BoF-based vectors, each evaluated by employing a different descriptor type, i.e., HOG, HOF, MBHx, MBHy and Traj. We evaluate two commonly used unsupervised video representation combination schemes, i.e., the concatenation of all the available video representations before training a SLFN network by using the regularized ELM algorithm (eq. (8)) and the mean output of V SLFN networks, each trained by using one video representation using the regularized ELM algorithm (eq. (8)). The performance of these combination schemes is compared to that of the proposed MVRELM algorithm.

Regarding the parameters of the competing algorithms used in our experiments, the optimal value of parameter c used by both regularized ELM and MVRELM has been determined by linear search using values $c = 10^q$, $q = -5, \dots, 5$. The optimal value of the parameter λ used by the proposed MVRELM algorithm has also be determined by applying linear search, using values $\lambda = 10^l$, $l = -5, \dots, 5$. Finally, the parameters b_j used in the χ^2 activation function (4) have been set equal to the mean value of the χ^2 distances between the training action vectors and the network input weights. The number of network hidden neurons has been set equal to 500 in all the cases.

4.1 The Hollywood2 database

The Hollywood2 database [13] consists of 1707 videos depicting 12 actions. The videos have been collected from 69 different Hollywood movies. The actions appearing in the database are: answering the phone, driving car, eating, ghting, getting out of car, hand shaking, hugging, kissing, running, sitting down, sitting up and standing up. Example video frames of the database are illustrated in Figure 1. We used the standard training-test split provided by the database (823 videos are used for training and performance is measured in the remaining 884 videos). Training and test videos come from different movies. The performance is evaluated by computing the average precision (AP) for each action class and reporting the mean AP over all classes (mAP), as suggested in [13]. This is due to the fact that some sequences of the database depict multiple actions.

4.2 The Olympic Sports database

The Olympic Sports database [14] consists of 783 videos depicting athletes practicing 16 sports, which have been collected from YouTube and annotated using Amazon Mechanical Turk. The actions appearing in the database are: high-jump, long-jump, triple-jump, pole-vault, basketball lay-up, bowling, tennis-serve, platform, discus, hammer, javelin, shot-put, springboard, snatch, clean-jerk and vault. Example video frames of the database are illustrated in Figure 2. The database has rich scene context information, which is helpful for recognizing sport actions. We used the standard training-test split provided by the database (649 videos are used for training and performance is measured in the remaining 134 videos). The performance is evaluated by computing the mean Average Precision (mAP) over all classes, as suggested in [14]. In addition,

Fig. 1. *Video frames of the Hollywood*2 *database depicting instances of all the twelve actions.*

since each video depicts only one action, we also measured the performance of each algorithm by computing the classification rate (CR).

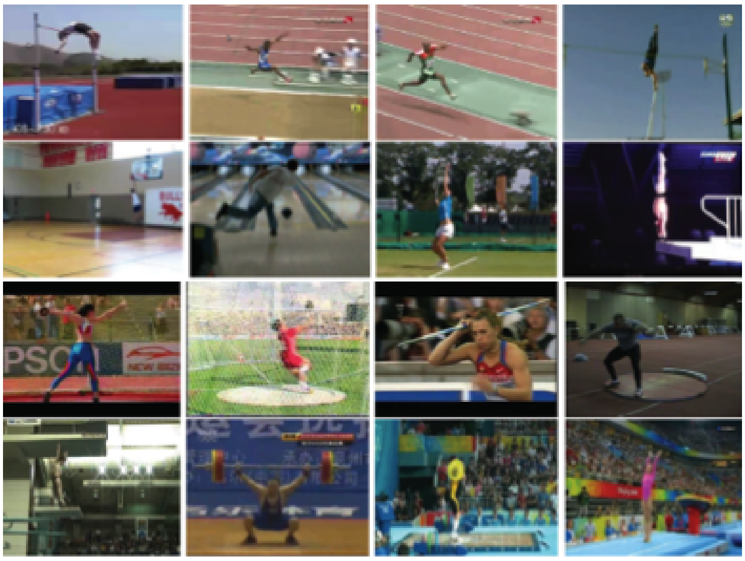

Fig. 2. *Video frames of the Olympic Sports database depicting instances of all the sixteen actions.*

4.3 The Hollywood 3D database

The Hollywood 3D database [15] consists of 951 video pairs (left and right channel) depicting 13 actions collected from Hollywood movies. The actions appearing in the database are: dance, drive, eat, hug, kick, kiss, punch, run, shoot, sit down, stand up, swim and use phone. Another class referred to as 'no action' is also included in the database. Example video frames of this database are illustrated in Figure 3. We used the standard (balanced) training-test split provided by the database (643 videos are used for training and performance is measured in the remaining 308 videos). Training and test videos come from different movies. The performance is evaluated by computing both the mean AP over all classes (mAP) and the classification rate (CR) measures, as suggested in [15].

Fig. 3. *Video frames of the Hollywood* 3*D database depicting instances of twelve actions.*

4.4 Experimental Results

Tables 1, 2 illustrate the performance of the competing algorithms on the Hollywood2, the Olympic Sports and the Hollywood 3D databases. We denote by 'Conc. ELM' the classification scheme employing the concatenation of all the available video representations before training a SLFN network by using the regularized ELM algorithm (eq. (8)) and by 'ELM Mean' the classification scheme employing the mean output of V SLFN networks, each trained by using one video representation using the regularized ELM algorithm (eq. (8)).

As can be seen, use of the mean SLFN network outcome outperforms the use of an action video representation obtained by concatenating all the available action vectors before training in the Olympic Sports and the Hollywood 3D databases, while they achieve comparable performance on the Hollywood2 database. The proposed MVRELM algorithm outperforms both of them in all the three databases. When the performance is measured by using the mean average precision metric, it achieves performance equal to 56.26%, 80.53% and 29.86% on the Hollywood2, the Olympic Sports and the Hollywood 3D databases, respectively. In the case where the performance is measured by using the classification rate, it achieves performance equal to 74.63% and 33.44% on the Olympic Sports and the Hollywood 3D databases, respectively.

Table 1. Action Recognition Performance (mAP) on the Hollywood2, Olympic Sports and Hollywood 3D databases.

	Conc. ELM	ELM Mean	**MVRELM**
Hollywood2	55.97 %	55.65 %	**56.26** %
Olympic Sports	77.39 %	79.09 %	**80.53** %
Hollywood 3D	28.26 %	28.73 %	**29.86** %

Table 2. Action Recognition Performance (CR) on the Olympic Sports and Hollywood 3D databases.

	Conc. ELM	ELM Mean	**MVRELM**
Olympic Sports	70.9 %	73.13 %	**74.63** %
Hollywood 3D	29.22 %	32.47 %	**33.44** %

5 Conclusions

In this paper, we proposed an extension of the ELM algorithm that is able to exploit multiple action representations. Proper regularization terms have been incorporated in the ELM optimization problem in order to extend the ELM algorithm to multi-view action classification. In order to determine both optimized network weights and action representation combination weights, we proposed an iterative optimization process. The proposed algorithm has been evaluated on three publicly available action recognition databases, where its performance has been compared with that of two commonly used video representation combination approaches, i.e., the vector concatenation before learning and the combination of classification outcomes based on learning on each view independently.

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. Turaga, P., Chellappa, R., Subrahmanian, V., Udrea, O.: Machine recognition of human activities: A survey. IEEE Transactions on Circuits and Systems for Video Technology, 18(11), 1473–1488 (2008)
2. Ji, X., Liu, H.: Advances in View-Invariant Human Motion Analysis: Review. IEEE Transactions on Systems, Man and Cybernetics Part–C, 40(1), 13–24 (2010)
3. Wang, H., Klaser, A., Schmid, C., Liu, C.L.: Dense Trajectories and Motion Boundary Descriptors for Action Recognition. International Journal of Computer Vision, 103(60), 1–20 (2013)
4. Kittler, J., Hatef, M., Duin, R., Matas, J.: On combining classifiers. IEEE Transactions on Pattern Analysis and Machine Intelligence, 20(3), 226–239 (1998)
5. Iosifidis, A., Tefas, A., Pitas, I.: View-invariant action recognition based on Artificial Neural Networks. IEEE Transactions on Neural Networks and Learning Systems, 23(3), 412–424 (2012)
6. Huang, G., Zhu, Q., Siew, C.: Extreme Learning Machine: a new learning scheme for feedfowrard neural networks. IEEE International Joint Conference on Neural Networks (2004)
7. Bartlett, P.L.: The sample complexity of pattern classification with neural networks: the size of the weights is more importantthan the size of te network. IEEE Transactions on Information Theory, 44(2), 525–536 (1998)
8. Minhas, R., Baradarani, A., Seifzadeh, S., Wu, Q.J.: Human action recognition using Extreme Learning Machine based on visual vocabularies. Neurocomputing 73(10), 1906–1917 (2010)
9. Iosifidis, A., Tefas, A., Pitas, I.: Multi-view Human Action Recognition under Occlusion based on Fuzzy Distances and Neural Networks. European Signal Processing Conference (2012)
10. Iosifidis, A., Tefas, A., 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)
11. Iosifidis, A., Tefas, A., Pitas, I.: Dynamic action recognition based on Dynemes and Extreme Learning Machine. Pattern Recognition Letters, 34, 1890–1898 (2013)
12. Huang, G., Zhou, H., Ding, Z., Zhang, R.: Extreme Learning Machine for Regression and Multiclass Classification. IEEE Transactions on Systems, Man and Cybernetics Part–B, 42(2), 513–529 (2012)
13. Marszalek, M., Laptev, I., Schmid, C.: Actions in context. IEEE Conference on Computer Vision and Pattern Recognition (2009)
14. Niebles, J.C., Chend, C.W., Fei-Fei, L.: Modeling Temporal Structure of Decomposable Mition Segemnts for Activity Classification. European Conference on Computer Vision (2010)
15. Hadfield, S., Bowden, R.: Hollywood 3D: Recognizing Actions in 3D Natural Scenes. IEEE Conference on Computer Vision and Pattern Recognition (2013)
16. Lanckriet, G.R.G., Cristianini, N., Ghaoui, L.E., Bartlett, P., Jordan, M.I.: Learning the kernel matrix with semidefinite programming. Journal of Machine Learning Research, 5, 27–72 (2013)
17. Bach, F.R., Lanckriet, G.R.G., Jordan M.I.: Multiple kernel learning, conic duality, and the SMO algorithm. International Conference on Machine Learning (2004)
18. Damoulas, T., Girolami, M.A.: Combining feature spaces for classification. Pattern Recognition, 42(11), 2671–2683 (2009)
19. Gonen, M., Alpaydin, E.: Multiple Kernel Learning Algorithms. Journal of Machine Learning Research, 12, 2211-2268 (2011)