International Journal on Artificial Intelligence Tools

Human Action Recognition based on Multi-view Regularized Extreme Learning Machine

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*

Received (Day Month Year)
Revised (Day Month Year)
Accepted (Day Month Year)

In this paper, we employ multiple Single-hidden Layer Feedforward Neural Networks for multi-view action recognition. We propose an extension of the Extreme Learning Machine algorithm that is able to exploit multiple action representations and scatter information in the corresponding ELM spaces for the calculation of the networks' parameters and the determination of optimized network combination weights. The proposed algorithm is evaluated by using two state-of-the-art action video representation approaches on five publicly available action recognition databases designed for different application scenarios. Experimental comparison of the proposed approach with three commonly used video representation combination approaches and relating classification schemes illustrates that ELM networks employing a supervised view combination scheme generally outperform those exploiting unsupervised combination approaches, as well as that the exploitation of scatter information in ELM-based neural network training enhances the network's performance.

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 movie (post-)production, 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 investigated 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[1,4]. 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[8,19,22]. 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.

1.1. *Action recognition in the wild*

The state-of-the-art approach in this, unrestricted, problem describes actions by employing the Bag-of-Features (BoF) model[3,2]. 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 using 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) [5]. 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). While it has been shown that video representations exploiting dense sampling provide better action classification rates[5,6], their calculation is computationally expensive. When fast action recognition is important, visual information appearing in Space-Time Interest Points (STIPs) can be exploited. STIPs are video frame pixel locations that correspond to abrupt image intensity changes[39,40], hence containing motion information that is useful for action description.

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[5,6]. 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[7], e.g., by using the mean classifier outcome in the case of SLFN networks[10]. 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.

1.2. *Multi-view action recognition*

While the above-described approach has been shown to provide state-of-the-art performance in many realistic action databases, the fact that actions are observed by only one camera (viewpoint) complicates the recognition problem. This is due

to the fact that the visual appearance of actions is quite different, when observed from different view angles[32,33,34]. Therefore, single-view methods, i.e., methods employing one camera, usually perform well when the same view angle is used during both training and testing. If this assumption is not met, the performance of single-view action recognition methods decreases. Multi-view action recognition methods, i.e., methods exploiting visual information captured by multiple viewpoints, have been proposed in order to perform view-independent human action recognition. Two approaches have been investigated to this end. The first one exploits the enriched visual information obtained by multi-camera setup usage in order to determine view-independent action representations, like visual hulls[35], motion history volumes[36], multi-view postures[9], or skeletal and super-quadratic body models[37]. The use of such body representations assumes that the human body is visible from all the cameras forming the camera setup in both the training and test phases. In addition, most of these methods assume that both training and test camera setups are formed by the same number of (calibrated) cameras. In a different case, the obtained human body representation will be incomplete and, thus, action recognition performance will probably decrease. This fact renders such multi-view methods applicable only in some, rather restrictive, action recognition scenarios[38].

In order to overcome these restrictions, multi-view methods that are based on single-view ensembles have been proposed[10,21,19,22]. In these methods, multiple recognizers (experts) are trained in order to perform single-view view-independent human action recognition. In the test phase, single-view view-independent action recognition is performed by all the experts observing the performed action independently and their outcomes are properly combined in order to provide the final action classification result. While the performance of each single-view recognizer is usually low, action recognition outcomes combination is able to provide state-of-the-art performance in many multi-view action databases[10,21]. Thus, the creation of supervised combination schemes is important in order to properly combine the information provided by different single-view recognizers (experts). Such combination schemes are also referred to as multi-view learning techniques.

1.3. *Extreme Learning Machine*

Extreme Learning Machine (ELM)[11] is an algorithm for fast Single-hidden Layer Feedforward Neural (SLFN) networks training that belongs to the family of randomized neural networks[12,14,13,16,15]. 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[17]. ELM has been successfully applied to many classification problems, including human action recognition[18,19,21,22,23].

1.4. *Contributions*

In this paper we employ the ELM algorithm in order to perform human action recognition from videos. We investigate two action recognition scenarios. The first one refers to single-view recognition of human actions in unconstrained environments based on multiple video representations. For this case, we adopt the state-of-the-art BoF-based action video representation described above[6], 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. The second action recognition scenario refers to the recognition of actions when they are observed by multiple viewpoints. Since in these cases multiple video streams need to be simultaneously processed, fast single-view action recognition is required. In order to avoid the increased computational cost of video representations based on dense sampling, we employ STIP-based video representation for multi-view action recognition. In order to properly combine the information provided by different descriptor types and single-view recognizers, we exploit a variant of ELM that is able to incorporate multiple video representations in its optimization process[41]. In order to enhance action classification performance, we extend the MVRELM algorithm[41] so that to incorporate the (class) variance of the training data in each of the ELM spaces. An iterative optimization scheme is proposed, where the contribution of each video representation is appropriately weighted. We evaluate the performance of the proposed algorithm on three single-view and two multi-view databases, where we compare it with that of relating classification schemes and three commonly adopted video representation combination schemes.

The proposed approach is closely related to Multiple Kernel Learning (MKL)[28,29,30]. 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[31]. 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 (usually high-dimensional) ELM spaces, we aim at optimally weighting the contribution of each data representation in the outputs of the combined network outputs.

This paper extends our previous work[41] in that:

- We extend MVRELM algorithm so that to incorporate the (class) variance of the training data in each of the ELM spaces. This leads to enhanced

action classification performance.
- We investigate two action recognition scenarios, i.e., single-view action recognition based on multiple video representations and multi-view action recognition where each view corresponds to a specific viewpoint.
- We compare the performance of the proposed approach with that of relating classification schemes and three commonly adopted video representation combination schemes.

The remainder of the paper is structured as follows. In Section 2, we briefly describe the ELM algorithm. Multi-view Regularized ELM (MVRELM) algorithm is described in Section 3. The proposed Multi-view ELM algorithm exploiting the (class) variance of the data in each of the ELM spaces is described in Section 4. Experimental results evaluating its performance are illustrated in Section 5. Finally, conclusions are drawn in Section 6. In Table 1, we summarize the notation that will be used in the paper.

Table 1. Notation.

Symbol	Explanation
$\mathbf{x}_i$	BoF-based representation of video i.
c_i	Action class label of video i.
C	Number of action classes.
D	Dimensionality of $\mathbf{x}_i$.
H	Number of network hidden layer neurons.
$\mathbf{W}_{in}$	Network hidden layer weights.
$\mathbf{b}$	Network hidden layer bias values.
$\mathbf{W}_{out}$	Network output weights.
$\Phi(\cdot)$	Hidden layer activation function.
$\boldsymbol{\phi}_i$	Network hidden layer output for $\mathbf{x}_i$.
$\mathbf{v}_j$	j-th column of $\mathbf{W}_{in}$.
$\mathbf{u}_k$	k-th row of $\mathbf{W}_{out}$.
$\mathbf{o}_i$	Network output for $\mathbf{x}_i$.
$\mathbf{\Phi}$	Hidden layer output matrix.
$\mathbf{T}$	Network target matrix.
$\mathbf{S}_w$	Within-class scatter matrix.
$\mathbf{S}_T$	Between-class scatter matrix.
$\mathbf{W}_{in}^V$	Network hidden layer weights for view v.
$\mathbf{W}_{out}^v$	Network output weights for view v.
$\boldsymbol{\gamma}$	View combination weight vector.
$\mathbf{S}_{u\,w}$	Within-class scatter matrix for view v.
$\mathbf{S}_{u\,T}$	Between-class scatter matrix for view v.
c, λ	Regularization parameters.

2. Extreme Learning Machine

ELM has been proposed for single-view classification[11]. 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 $\mathbf{x}_i$ 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.:

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

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

leading to MLP and RBF networks, respectively. The RBF-χ^2 activation function can also be employed:

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

since it has been found to outperform both the above two alternative choices for BoF-based data representations[5,6,42].

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

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

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., $\mathbf{o}_i = \mathbf{t}_i$, $i = 1, ..., N$, $\mathbf{W}_{out}$ can be analytically calculated by solving for:

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

where $\mathbf{T} = [\mathbf{t}_1, \ldots, \mathbf{t}_N]$ 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[24], where the network output weights are obtained, according to a regularization paramter $c > 0$, by:

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

By exploiting the fact that the second processing step of ELM training corresponds to a linear projection of the (high-dimensional) data to a low-dimensional feature space determined by the network target vectors, the ELM algorithm has been extended in order to exploit the (class) variance of the data in the ELM space[21,42]. In this case the network output weights are obtained by solving for:

$$\textbf{Minimize: } \mathcal{J} = \frac{1}{2}\|\mathbf{S}^{\frac{1}{2}}\mathbf{W}_{out}\|_F^2 + \frac{c}{2}\sum_{i=1}^{N}\|\boldsymbol{\xi}_i\|_2^2 \tag{9}$$

$$\textbf{Subject to: } \mathbf{W}_{out}^T\boldsymbol{\phi}_i - \mathbf{t}_i = \boldsymbol{\xi}_i,\ \ i = 1, ..., N. \tag{10}$$

and are given by:

$$\mathbf{W}_{out} = \left(\mathbf{\Phi}\mathbf{\Phi}^T + \frac{1}{c}\mathbf{S}\right)^{-1} \mathbf{\Phi}\, \mathbf{T}^T, \tag{11}$$

where $\mathbf{S}$ can be either the within-class scatter matrix $\mathbf{S}_w$ given by:

$$\mathbf{S}_w = \sum_{j=1}^{C}\sum_{i=1}^{N}\frac{\beta_{ij}}{N_j}(\boldsymbol{\phi}_i - \boldsymbol{\mu}_j)(\boldsymbol{\phi}_i - \boldsymbol{\mu}_j)^T. \tag{12}$$

or the total scatter matrix of the data in the ELM space $\mathbf{S}_T$ given by:

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

In (12) and (13), β_{ij} is an index denoting if action vector $\mathbf{x}_i$ belongs the to action class j, i.e., $\beta_{ij} = 1$, if $c_i = j$ and $\beta_{ij} = 0$ otherwise and $N_j = \sum_{i=1}^{N}\beta_{ij}$ is the number of training action vectors belonging to action class j. $\boldsymbol{\mu}_j \in \mathbb{R}^H$ is the mean vector of class j and $\boldsymbol{\mu} \in \mathbb{R}^H$ is the mean vector of the entire training set in $\mathbb{R}^H$, i.e., $\boldsymbol{\mu}_j = \frac{1}{N_j}\sum_{i=1}^{N}\beta_{ij}\boldsymbol{\phi}_i$ and $\boldsymbol{\mu} = \frac{1}{N}\sum_{i=1}^{N}\boldsymbol{\phi}_i$.

The adoption of the optimization scheme in (9) leads to the calculation of network output weights providing a compromise between the training error of the network and the within-class or total scatter of the network outputs for the training data.

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.:

$$c_t = arg \max o_{tj},\ j = 1, ..., C. \tag{14}$$

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 the MVRELM algorithm that can be used for multi-view action classification[41], i.e., in the cases where each action 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 $\mathbf{x}_i^v \in \mathbb{R}^{D_v},\ i = 1, \ldots, l, \ldots, N,\ v = 1, \ldots, V$. 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.

The networks output weights $\mathbf{W}_{out}^v \in \mathbb{R}^{H_v \times C}$ and the view combination weights $\gamma \in \mathbb{R}^V$ are determined by solving the following optimization problem:

$$\textbf{Minimize: } \mathcal{J} = \frac{1}{2}\sum_{v=1}^{V} \|\mathbf{W}_{out}^v\|_F^2 + \frac{c}{2}\sum_{i=1}^{N} \|\boldsymbol{\xi}_i\|_2^2 \tag{15}$$

$$\textbf{Subject to: } \left(\sum_{v=1}^{V} \gamma_v \mathbf{W}_{out}^{v\,T} \boldsymbol{\phi}_i^v\right) - \mathbf{t}_i = \boldsymbol{\xi}_i,\ \ i = 1, ..., N, \tag{16}$$

$$\|\boldsymbol{\gamma}\|_2^2 = 1, \tag{17}$$

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.

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

$$\mathbf{O} = \sum_{v=1}^{V} \gamma_v \mathbf{W}_{out}^{v\,T} \boldsymbol{\Phi}^v. \tag{18}$$

By substituting (16) in (15) and taking the equivalent dual problem, we obtain:

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

$$= \frac{1}{2}\sum_{v=1}^{V}\|\mathbf{W}_{out}^{v}\|_F^2 + \frac{c}{2}\boldsymbol{\gamma}^T\mathbf{P}\boldsymbol{\gamma} - c\mathbf{r}^T\boldsymbol{\gamma} + \frac{c}{2}tr\left(\mathbf{T}^T\mathbf{T}\right) + \frac{\lambda}{2}\boldsymbol{\gamma}^T\boldsymbol{\gamma}, \tag{19}$$

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

$$\boldsymbol{\gamma} = \left(\mathbf{P} + \frac{\lambda}{c}\mathbf{I}\right)^{-1}\mathbf{r}. \tag{20}$$

By substituting (16) in (15) 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:

$$\mathbf{W}_{out}^{v} = \left(\frac{2}{c\gamma_v}\mathbf{I} + \gamma_v\boldsymbol{\Phi}^v\boldsymbol{\Phi}^{v\,T}\right)^{-1}\boldsymbol{\Phi}^v(2\mathbf{T} - \mathbf{O})^T, \tag{21}$$

As can be observed in (20), (21), $\boldsymbol{\gamma}$ is a function of $\mathbf{W}_{out}^{v}$, $v = 1,\ldots,V$ 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. In order to determine both $\mathbf{W}_{out}^{v}$, $v = 1,\ldots,V$ and $\boldsymbol{\gamma}$, we employ an iterative optimization scheme formed by two optimization steps[41]. In the following, the index t is used in order to denote the iteration of the 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. $\mathbf{W}_{out,1}^{v}$ are initialized by using (8), while the values $\boldsymbol{\gamma}_{1,v} = 1/V$ is used for all the action video representations $v = 1,\ldots,V$. By using $\boldsymbol{\gamma}_t$, the network output weights $\mathbf{W}_{out,t+1}^{v}$ are updated by using (21). After the calculation of $\mathbf{W}_{out,t+1}^{v}$, $\boldsymbol{\gamma}_{t+1}$ is obtained by using (20). 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 $\mathbf{x}_l^v \in \mathbb{R}^D$ is given by:

$$\mathbf{o}_l = \sum_{v=1}^{V} \gamma_v \mathbf{W}_{out}^{v\,T} \boldsymbol{\phi}_l^v. \tag{22}$$

4. Exploiting the (class) variance in MVRELM

In this Section, we describe the proposed algorithm for Multi-view SLFN network training. By using the notations introduced in Section 3, the networks output weights $\mathbf{W}_{out}^v \in \mathbb{R}^{H_v \times C}$ and the view combination weights $\gamma \in \mathbb{R}^V$ can be determined by solving the following optimization problem:

$$\textbf{Minimize: } \mathcal{J} = \frac{1}{2} \sum_{v=1}^{V} \|\mathbf{S}_v^{\frac{1}{2}} \mathbf{W}_{out}^v\|_F^2 + \frac{c}{2} \sum_{i=1}^{N} \|\boldsymbol{\xi}_i\|_2^2 \tag{23}$$

$$\textbf{Subject to: } \left(\sum_{v=1}^{V} \gamma_v \mathbf{W}_{out}^{v\,T} \boldsymbol{\phi}_i^v \right) - \mathbf{t}_i = \boldsymbol{\xi}_i, \;\; i = 1, ..., N, \tag{24}$$

$$\|\boldsymbol{\gamma}\|_2^2 = 1, \tag{25}$$

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.

In (23), $\mathbf{S}_v \in \mathbb{R}^{H_v \times H_v}$ is a matrix describing the (class) variance of the data in $\mathbb{R}^{H_v}$. That is, it can be either the within-class scatter matrix evaluated on $\boldsymbol{\phi}_i^v$, i.e.,:

$$\mathbf{S}_{v,w} = \sum_{j=1}^{C} \sum_{i=1}^{N} \frac{\beta_{ij}}{N_j} (\boldsymbol{\phi}_i^v - \boldsymbol{\mu}_j^v)(\boldsymbol{\phi}_i^v - \boldsymbol{\mu}_j^v)^T. \tag{26}$$

or the total scatter matrix evaluated on $\boldsymbol{\phi}_i^v$, i.e.,:

$$\mathbf{S}_{v,T} = \sum_{i=1}^{N} (\boldsymbol{\phi}_i^v - \boldsymbol{\mu}^v)(\boldsymbol{\phi}_i^v - \boldsymbol{\mu}^v)^T. \tag{27}$$

In (26) and (27), $\boldsymbol{\mu}_j^v$ is the mean vector of class j and $\boldsymbol{\mu}^v$ is the mean vector of the entire training set in $\mathbb{R}^{H_v}$, i.e., $\boldsymbol{\mu}_j^v = \frac{1}{N_j} \sum_{i=1}^{N} \beta_{ij} \boldsymbol{\phi}_i^v$ and $\boldsymbol{\mu}^v = \frac{1}{N} \sum_{i=1}^{N} \boldsymbol{\phi}_i^v$.

By substituting (24) in (23) and taking the equivalent dual problem, we obtain:

$$\begin{aligned}
\mathcal{J}_D &= \frac{1}{2} \sum_{v=1}^{V} \|\mathbf{S}_v^{\frac{1}{2}} \mathbf{W}_{out}^v\|_F^2 + \frac{c}{2} \sum_{i=1}^{N} \| \left(\sum_{v=1}^{V} \gamma_v \mathbf{W}_{out}^{v\,T} \boldsymbol{\phi}_i^v \right) - \mathbf{t}_i \|_2^2 + \frac{\lambda}{2} \|\boldsymbol{\gamma}\|_2^2 \\
&= \frac{1}{2} \sum_{v=1}^{V} \|\mathbf{S}_v^{\frac{1}{2}} \mathbf{W}_{out}^v\|_F^2 + \frac{c}{2} \| \left(\sum_{v=1}^{V} \gamma_v \mathbf{W}_{out}^{v\,T} \boldsymbol{\Phi}^v \right) - \mathbf{T} \|_F^2 + \frac{\lambda}{2} \|\boldsymbol{\gamma}\|_2^2 \\
&= \frac{1}{2} \sum_{v=1}^{V} \|\mathbf{S}_v^{\frac{1}{2}} \mathbf{W}_{out}^v\|_F^2 + \frac{c}{2} \boldsymbol{\gamma}^T \mathbf{P} \boldsymbol{\gamma} - c \mathbf{r}^T \boldsymbol{\gamma} + \frac{c}{2} tr\left(\mathbf{T}^T \mathbf{T}\right) + \frac{\lambda}{2} \boldsymbol{\gamma}^T \boldsymbol{\gamma},
\end{aligned} \tag{28}$$

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:

$$\boldsymbol{\gamma} = \left(\mathbf{P} + \frac{\lambda}{c}\mathbf{I}\right)^{-1}\mathbf{r}. \tag{29}$$

By substituting (24) in (23) and taking the equivalent dual problem, we can also obtain:

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

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:

$$\mathbf{W}_{out}^{v} = \left(\frac{2}{c\gamma_v}\mathbf{S}_v + \gamma_v\mathbf{\Phi}^{v}\mathbf{\Phi}^{v\,T}\right)^{-1}\mathbf{\Phi}^{v}(2\mathbf{T} - \mathbf{O})^T, \tag{30}$$

Similar to the MVRELM case, $\boldsymbol{\gamma}$ is a function of $\mathbf{W}_{out}^{v}$, $v = 1, \ldots, V$ 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. In order to determine both $\mathbf{W}_{out}^{v}$, $v = 1, \ldots, V$ and $\boldsymbol{\gamma}$, we follow an iterative optimization process formed by two optimization steps. In the following, the index t is used in order to denote the iteration of the 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. $\mathbf{W}_{out,1}^{v}$ are initialized by using (11), while the values $\boldsymbol{\gamma}_{1,v} = 1/V$ is used for all the action video representations $v = 1, \ldots, V$. By using $\boldsymbol{\gamma}_t$, the network output weights $\mathbf{W}_{out,t+1}^{v}$ are updated by using (30). After the calculation of $\mathbf{W}_{out,t+1}^{v}$, $\boldsymbol{\gamma}_{t+1}$ is obtained by using (29). 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:

$$\mathbf{o}_l = \sum_{v=1}^{V}\gamma_v\mathbf{W}_{out}^{v\,T}\boldsymbol{\phi}_l^v. \tag{31}$$

5. Experiments

In this section, we present experiments conducted in order to evaluate the performance of the proposed algorithms. We test the proposed approach in two action recognition scenarios, i.e., single-view action recognition based on multiple video representations, view-independent action recognition based on multiple view representations. For the first case we have employed three publicly available databases, namely the Hollywood2, the Olympic Sports and the Hollywood 3D databases. For the latter ones, we have employed a new multi-view database created for the needs of the FP7 R&D European Project IMPART[a] and a publicly available multi-view database, namely the i3DPost database. In the following subsections, we describe the databases and evaluation measures used in our experiments. Experimental results are provided in subsection 5.6.

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 and the mean output of V SLFN networks, each trained by using one video representation. We also evaluate another unsupervised video representation combination scheme, i.e. ELM network training by using the (element-wise) product of the hidden-layer outputs obtained for all the V action representations. The latter choice has been inspired by the element-wise kernel matrix multiplication combination scheme[20]. Specifically, for RBF-based hidden layer output calculation approaches, the element-wise multiplication combination scheme has the physical meaning of calculating the mean (Euclidean or χ^2) distance between the training data and the hidden layer weight vectors and subsequently applying the exponential operator for the calculation of the (combined) hidden layer output that will be used for the calculation of the network output weights.

We compare the performance of these combination schemes for the cases of Regularized ELM (RELM)[24] (eq. (8)), Minimum Class Variance ELM (MCVELM)[21] (eq. (11) using (12)) and Minimum Variance ELM (MVELM)[42] (eq. (11) using (13)) with that of MVRELM (eq. (21) and (20)) and the proposed MVRELM variant (eq. (30) and (29) using (26) or (27)). On each database, we perform ten experiments, and report the mean performance of each algorithm. For fair comparison, on each experiment we first initialize the networks hidden layer parameters and use them in order to map the training data in the corresponding ELM spaces. Subsequently, we calculate the remaining networks' parameters by employing the equations corresponding to each of the competing classification schemes.

Regarding the parameters of the competing algorithms used in our experiments, the optimal value of parameter c used in all the ELM variants has been determined by linear search using values $c = 10^q$, $q = -3, \ldots, 3$. The optimal value of the parameter λ used by both the MVRELM variants has also be determined by applying linear search, using values $\lambda = 10^l$, $l = -3, \ldots, 3$.

[a]http://impart.upf.edu/

5.1. *The Hollywood2 database*

The Hollywood2 database[25] 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)[25]. This is due to the fact that some sequences of the database depict multiple actions.

Fig. 1. *Video frames of the Hollywood2 database depicting instances of all the twelve actions.*

5.2. *The Olympic Sports database*

The Olympic Sports database[26] 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[26]. In addition, since each video depicts only one action, we also measured the performance of each algorithm by computing the classification rate (CR).

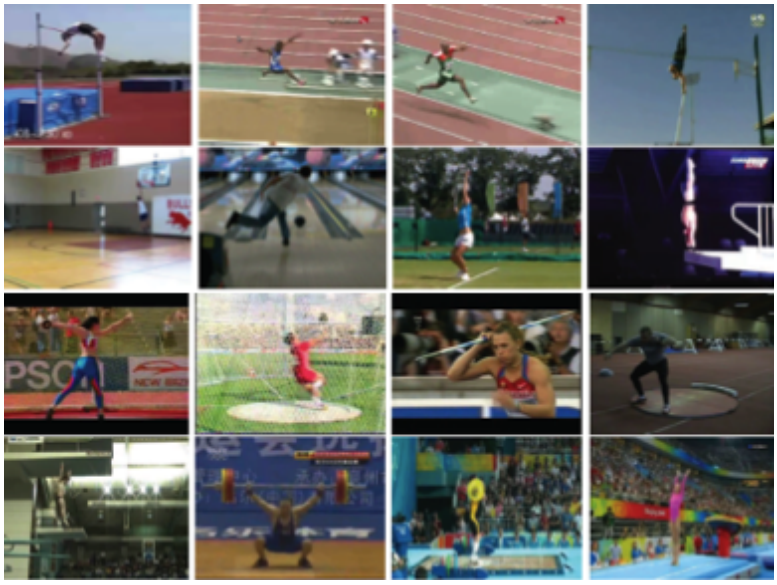

Fig. 2. *Video frames of the Olympic Sports database depicting instances of all the sixteen actions.*

5.3. *The Hollywood 3D database*

The Hollywood 3D database[27] 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[27].

5.4. *The i3DPost database*

The i3DPost multi-view action database[43] consists of 512 high-resolution (1080 $\times$ 1920 pixel) videos depicting eight persons performing eight actions. The database camera setup consists of eight cameras placed in the perimeter of a ring at a height of 2 meters above the studio floor. The actions appearing in the database are: walk, run, jump in place, jump forward, bend, fall down, sit on a chair and wave one hand. The Leave-One-Person-Out cross-validation procedure is used by most action recognition methods evaluating their performance on this data set. That is, the algorithms are trained by using the action videos of seven persons and tested on the action videos of the eighth one. Eight training-test rounds (folds) are performed,

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

one for each test person, in order to complete an experiment. The mean action classification rate over all folds is used in order to evaluate the performance of each method. Example video frames depicting a person walking as viewed from all $N_C = 8$ available view angles are illustrated in Figure 4.

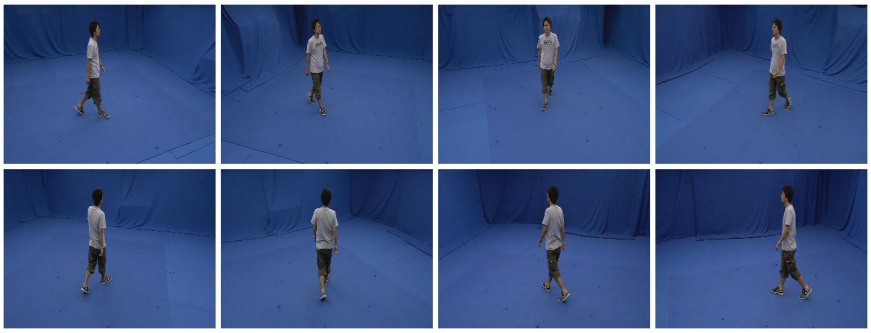

Fig. 4. *Video frames of the i3Dpost eight-view database depicting a person walking.*

5.5. *The IMPART database*

The IMPART multi-view action database consists of 504 high-resolution (1080×1920 pixel) videos depicting three persons performing twelve actions and interactions. The database camera setup consists of fourteen cameras placed in the perimeter of a ring at a height of 2 meters above the studio floor. The actions appearing in the database are: bend, fall, hand waving, jump forward, jump in place, sit, run and walk. Sequences depicting successive actions, namely run-jump-walk and walk-sit, are also provided. The persons also perform two interactions, namely pull and

hand shaking. Similar to the i3DPost database, we apply the Leave-One-Person-Out cross-validation procedure. That is, the algorithms are trained by using the action videos of two persons and tested on the action videos of the third one. Three training-test rounds (folds) are performed, one for each test person, in order to complete an experiment. The mean action classification rate over all folds is used in order to evaluate the performance of each method

5.6. *Experimental Results*

In our first set of experiments we applied the competing algorithms on the single-view action recognition scenario. We employed the state-of-the-art action video representation[6], 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 employed the RBF-χ^2 activation function (4) in order to calculate the network hidden layer outputs. The parameter b used in the RBF-χ^2 activation function (4) has been set equal to the mean value of the χ^2 distances between the training action vectors and the network input weights, which is the natural scale factor for the χ^2 distances between $\mathbf{x}_i$ and $\mathbf{v}_j$. On each dataset, we applied ten random initializations for each algorithm and we measure its performance by calculating the mean performance over all experiments and the corresponding standard deviation. Unless otherwise stated, the number of network hidden neurons has been set equal to $H = 1000$, a value that has been shown to provide satisfactory performance in many classification problems[24,21].

Tables 2 and 3 illustrate the performance of the competing algorithms on the Hollywood2, the Olympic Sports and the Hollywood 3D databases. We denote by 'Conc. Input' the classification scheme employing the concatenation of all the available video representations before training a SLFN network, by 'Mul. ELM' the classification scheme employing the (element-wise) multiplication of the hidden layer outputs corresponding to all the V data representations for training and 'Mean Out' the classification scheme employing the mean output of V SLFN networks, each trained by using one video representation. As can be seen in these Tables, the adoption of supervised combination schemes generally leads to better classification performance. Overall, the MVRELM algorithm exploiting the within-class scatter matrix $\mathbf{S}_w$ (eq. (26)) provides the best performance in two, out of three, databases.

In our second experiment we applied the competing algorithms on the multi-view (multi-camera) action recognition scenario. We performed view-independent action recognition on the IMPART database by using the action video representation[6], 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. RBF activation function (4) in order to calculate the network hidden layer outputs. The parameter b used in the RBF activation function (3) has been set equal to the mean Euclidean distance between the training action vectors and the network

Table 2. Action Recognition Performance (mAP) on the Hollywood2, Olympic Sports and Hollywood 3D databases.

Method	Hollywood2	Olympic Sports	Hollywood 3D
Conc. Input	52.38(±0.01) %	67.11(±0.01) %	25.27(±0.04) %
Mul. ELM	55.05(±0.27) %	77.78(±0.22) %	28(±0.19) %
Mean Out	57.57(±0.07) %	81.73(±0.1) %	**29.7(**±0.12**)** %
Conc. Input ($\mathbf{S}_w$)	54.71(±0.11) %	79.58(±0.15) %	24.39(±0.03) %
Mul. ELM ($\mathbf{S}_w$)	55.08(±0.16) %	83.24(±0.19) %	28.75(±0.12) %
Mean Out ($\mathbf{S}_w$)	57.58(±0.11) %	86.4(±0.13) %	**29.85(**±0.1**)** %
Conc. Input ($\mathbf{S}_T$)	54.04(±0.01) %	79.17(±0.03) %	25.27(±0.03) %
Mul. ELM ($\mathbf{S}_T$)	55.08(±0.19) %	83.24(±0.16) %	28(±0.14) %
Mean Out ($\mathbf{S}_T$)	57.57(±0.12) %	85.19(±0.1) %	28.75(±0.1) %
MVRELM	57.44(±0.12) %	85.19(±0.14) %	29.42(±0.21) %
MVRELM ($\mathbf{S}_T$)	57.44(±0.24) %	86.39(±0.2) %	29.42(±0.2) %
MVRELM ($\mathbf{S}_w$)	**57.77(**±0.21**)** %	**86.66(**±0.18**)** %	29.45(±0.18) %

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

Method	Olympic Sports	Hollywood 3D
Conc. Input (**I**)	70.9(±0.01) %	27.6(±0.01) %
Mul. ELM (**I**)	71.65(±0.23) %	30.52(±0.2) %
Mean Out (**I**)	80.6(±0.15) %	31.49(±0.13) %
Conc. Input ($\mathbf{S}_w$)	73.13(±0.01) %	22.4(±0.01) %
Mul. ELM ($\mathbf{S}_w$)	73.88(±0.23) %	30.52(±0.13) %
Mean Out ($\mathbf{S}_w$)	81.34(±0.15) %	31.49(±0.13) %
Conc. Input ($\mathbf{S}_T$)	73.13(±0.01) %	27.6(±0.03) %
Mul. ELM ($\mathbf{S}_T$)	73.88(±0.23) %	30.52(±0.15) %
Mean Out ($\mathbf{S}_T$)	81.34(±0.11) %	**34.74(**±0.13**)** %
MVRELM (**I**)	82.09(±0.11) %	30.52(±0.21) %
MVRELM ($\mathbf{S}_T$)	82.81(±0.18) %	31.49(±0.2) %
MVRELM ($\mathbf{S}_w$)	**82.84(**±0.19**)** %	33.12(±0.17) %

input weights, which is the natural scale factor for the Euclidean distances between $\mathbf{x}_i$ and $\mathbf{v}_j$. The performance of each algorithm is illustrated in the first column of Table 4. As can be seen, the MVRELM algorithm exploiting the within-class and the total scatter matrices, $\mathbf{S}_w$ (eq. (26)) and $\mathbf{S}_T$ (eq. (27)) respectively, provide the best performance (equal to 72.42%).

Finally, we applied the competing algorithms on the i3DPost, where we investigate the case where different views correspond to different viewpoints. We employed a STIP-based action video representation[21], where each video is represented one 200-dimensional vector obtained by applying soft vector quantization on concatenated HOG/HOF descriptors evaluated on video STIP locations. We employed the RBF activation function (4) in order to calculate the network hidden layer outputs. The parameter b used in the RBF activation function (3) has been set equal to the mean Euclidean distance between the training action vectors and the network input weights, which is the natural scale factor for the Euclidean distances between $\mathbf{x}_i$ and $\mathbf{v}_j$. The number of network hidden neurons has been set equal to $H = 100$ for all

the competing algorithms, a value that has been found to provide satisfactory performance. We chose a smaller value of H on the i3DPost database, due to the small training set cardinality and low dimensionality of the data. A higher value of H would probably lead to overfitting. The performance of each algorithm is illustrated in the second column of Table 4. As can be seen, the MVRELM algorithm exploiting the within-class scatter matrix $\mathbf{S}_w$ (eq. (26)) provides the best performance (equal to 100%).

Table 4. Action Recognition Performance on the i3DPost and IMPART databases.

Method	IMPART	i3DPost
Conc. Input	62.49(±1.25) %	81.05(±2.5) %
Mul. ELM	64.49(±1.65) %	83(±1.12) %
Mean Out	65.28(±1.04) %	95.23(±1.16) %
Conc. Input ($\mathbf{S}_w$)	63.09(±1.35) %	79.69(±2.3) %
Mul. ELM ($\mathbf{S}_w$)	64.5(±1.23) %	92.19(±1.47) %
Mean Out ($\mathbf{S}_w$)	65.08(±1.15) %	98.44(±0.12) %
Conc. Input ($\mathbf{S}_T$)	62.89(±1.31) %	82.81(±1.36) %
Mul. ELM ($\mathbf{S}_T$)	64.29(±1.65) %	92.01(±0.27) %
Mean Out ($\mathbf{S}_T$)	65.28(±1.12) %	98.44(±0.05) %
MVRELM	70.92(±1.92) %	97.06(±0.17) %
MVRELM ($\mathbf{S}_T$)	**72.42**(±2.3) %	98.44(±0.03) %
MVRELM ($\mathbf{S}_w$)	**72.42**(±2.3) %	**100**(±0.0) %

6. Conclusions

In this paper, we described a classification scheme for multi-view human action recognition that is based on multiple Single-hidden Layer Feedforward Neural Networks. We proposed an extension of the Extreme Learning Machine algorithm that is able to exploit multiple action representations and scatter information of the training data in the corresponding ELM spaces on its optimization process. The proposed algorithm has been evaluated by using two state-of-the-art action video representation approaches on four publicly available action recognition databases, where its performance has been compared with that of three commonly used video representation combination approaches and relating classification schemes.

Some useful observations are the following: a) multi-view neural networks can be exploited in application scenarios involving both multi-camera setups where each action is captured from multiple viewpoints and single-view action recognition where multiple action descriptions are used in order to express different action properties (e.g. shape and motion). b) While the adoption of a supervised combination scheme for multi-view neural networks training generally leads to enhanced performance, it requires the application of an iterative optimization scheme, which might be time consuming compared to the unsupervised view combination approach. c) the exploitation of the within-class and total scatter information of the training data

can enhance the performance of the network. d) In order to further enhance the performance of multi-view neural networks nonlinear view combinations or linear combinations expressing higher order relationships between the various views, e.g. relationships expressed in a matrix $\mathbf{G} \in \mathbb{R}^{V \times V}$, may be exploited. The exploitation of such combination schemes, as well as the derivation of faster optimization schemes, could be a future research direction.

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).

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. Huang, Y., Wu, Z., Wang, L., Tan, T.: Feature coding in image classification: A comprehensive study. IEEE Transactions on Pattern Analysis and Machine Intelligence 36, 493506 (2014).
3. G., Bray, C., Dance, C., Fan, L.: Visual categorization with bags of keypoints. European Conference on Computer Vision (2004).
4. 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)
5. 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)
6. Wang, H., Schmid, C.: Action Recognition with Improved Trajectories. IEEE International Conference on Computer Vision (2013)
7. Kittler, J., Hatef, M., Duin, R., Matas, J.: On combining classifiers. IEEE Transactions on Pattern Analysis and Machine Intelligence, 20(3), 226–239 (1998)
8. Gorelick, L., Blank, M., Shechtman, E., Irani, M., Basri, R.: Actions as Space-Time Shapes. IEEE Transactions on Pattern Analysisw and Machine Intelligence, 29(12), 2247–2253, 2007.
9. Iosifidis, A., Tefas, A., Pitas, I.: Multi-view Human Movement Recognition based on Fuzzy Distances and Linear Discriminant Analysis. Computer Vision and Image Understanding, 116, 347–360 (2012).
10. 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)
11. 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)
12. Schmidt, W.F., Kraaijveld, M.A., Duin, R.P.W.: Feed Forward Neural Networks With Random Weights. International Conference on Pattern Recognition, (1992)
13. Igelnik, B., Pao, Y.H.: Stochastic Choice of Basis Functions in Adaptive Function Approximation and the Functional-Link Net. IEEE Transactions on Nenral Networks, 6(6), 1320–1329 (1995)

14. Pao, Y.H., Park, G.H., Sobajic, J.: Learning and generalization characteristics of the random vector Functional-link net. Neurocomputing, 6 163–180 (1994)
15. McLoon, S.F., Irwin, G.W.: Improving neural network training solutions using regularisation. Neurocomputing, 37(1), 71–90 (2001)
16. McLoone, S.F., Brown, M.D., Irwin, G.W., Ligtbody, G.: A hybrid linear nlinear training algorithm for feedforward neural networks. IEEE Transactions on Neural Networks, 9(4), 669-684 (1998)
17. 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)
18. 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)
19. 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)
20. Wang, H., Ullah, M.M., Klaser, A., Laptev, I., Schmid, C.: Evaluation of Local Spatio-temporal Features for Action Recognition. British Machine Vision Conference, 1–11 (2009)
21. 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)
22. Iosifidis, A., Tefas, A., Pitas, I.: Dynamic action recognition based on Dynemes and Extreme Learning Machine. Pattern Recognition Letters, 34, 1890–1898 (2013)
23. Iosifidis, A., Tefas, A., Pitas, A.: Graph Embedded Extreme Learning Machine, IEEE Transactions on Cybernetics, doi:10.1109/TCYB.2015.2401973, (2015)
24. 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)
25. Marszalek, M., Laptev, I., Schmid, C.: Actions in context. IEEE Conference on Computer Vision and Pattern Recognition (2009)
26. 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)
27. Hadfield, S., Bowden, R.: Hollywood 3D: Recognizing Actions in 3D Natural Scenes. IEEE Conference on Computer Vision and Pattern Recognition (2013)
28. 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)
29. 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)
30. Damoulas, T., Girolami, M.A.: Combining feature spaces for classification. Pattern Recognition, 42(11), 2671–2683 (2009)
31. Gonen, M., Alpaydin, E.: Multiple Kernel Learning Algorithms. Journal of Machine Learning Research, 12, 2211-2268 (2011)
32. Yu, S., Tan, D., Tan, T.: Modelling the effect of view angle variation on appearance-based gait recognition. Asian Conference on Computer Vision (2006).
33. Poppe, R.: A survey on vision-based human action recognition. Image and Vision Computing, 28, 976–990 (2010).
34. Weinland, D., Ronfard, R., Boyer, E.: A survey of vision-based methods for action

representation, segmentation and recognition. Computer Vision and Image Understanding, 104(2), 224-241 (2011).
35. Kilner, J., Guillemaut, J., Hilton, A.: 3D action matching with keypose detection. International Conference on Computer Vision Workshops (2009)
36. Weinland, D., Ronfard, R., Boyer, E.: Free viewpoint action recognition using motion history volumes. Computer Vision and Image Understanding, 104(2), 249-257 (2006)
37. Tran, D., Sorokin, A.: Human activity recognition with metric learning. European Conference on Computer Vision (2008).
38. Qureshi, F., Terzopoulos, D.: Surveillance camera scheduling: A virtual vision approach. Multimedia Systems, 12(3), 269-283 (2006).
39. Laptev, I., Marszalek, M., Schmid, C., Rozenfeld, B.: Learning realistic human actions from movies. IEEE Conference on Computer Vision and Pattern Recognition (2008).
40. Dollar, P., Rabaud, V., Cottrell, G., Belongie, S.: Behavior recognition via sparse spatio-temporal features. Joint IEEE International Workshop on Visual Surveillance and Performance Evaluation of Tracking and Surveillance (2005).
41. Iosifidis, A., Tefas, A., Pitas, I.: Multi-view Regularized Extreme Learning Machine for Human Action Recognition. Artificial Intelligence: Methods and Applications (2014).
42. Iosifidis, A., Tefas, A., Pitas, I.: Minimum Variance Extreme Learning Machine for Human Action Recognition. IEEE International Conference on Accoustics, Speech and Signal Processing (2014).
43. Gkalelis, N., Kim, H., Hilton, A., Nikolaidis, N., Pitas, I.: The i3DPost multi-view and 3D human action/interaction database. Conference on Visual Media Production (2009).