Semi-supervised classification of human actions based on Neural Networks

Alexandros Iosifidis, Anastasios Tefas and Ioannis Pitas
Department of Informatics, Aristotle University of Thessaloniki, Greece
Email: {aiosif,tefas,pitas}@aiia.csd.auth.gr

***Abstract*—In this paper, we propose a novel algorithm for Single-hidden Layer Feedforward Neural networks training which is able to exploit information coming from both labeled and unlabeled data for semi-supervised action classification. We extend the Extreme Learning Machine algorithm by incorporating appropriate regularization terms describing geometric properties and discrimination criteria of the training data representation in the ELM space to this end. The proposed algorithm is evaluated on human action recognition, where its performance is compared with that of other (semi-)supervised classification schemes. Experimental results on two publicly available action recognition databases denote its effectiveness.**

I. 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 and games and automatic assistance in healthcare of the elderly for independent living to name a few. However, it is a challenging problem, because of the complexity of human actions. Challenges that action recognition methods should be able to face include variations in human body proportions and execution style between individuals, different observation angles, variations in occlusion levels and in the distance between the individual and the camera, cluttered backgrounds and moving cameras. Such variations lead to high intra-class and, possibly, small inter-class variations for human actions.

Popular action representations describe actions either as series of successive human body poses, or as collections of local shape and motion descriptors, or by exploiting holistic video representations. In the first case, human body silhouettes are evaluated by applying video frame segmentation techniques or background subtraction. However, such techniques are inappropriate in real applications involving scenes having cluttered background where multiple persons appear. he two remaining approaches do not require video frame segmentation as a preprocessing step, since they employ video representation evaluated directly to the color (grayscale) video frames. In the first case, shape and motion descriptors, like the Histogram of Oriented Gradients (HOG) and the Histogram of Optical Flow (HOF) are calculated on Space-Time Interest Points (STIPs) [1] and videos are represented by adopting the Bag of Features (BoFs) model [2]. In the later case, action videos are divided in multiple sub-videos and each sub-video is described by exploiting its similarity with reference ones in order to obtain a template-based action video representation [3].

After the determination of an appropriate action video representation, action classification is usually performed by adopting supervised classification schemes. This approach has been extensively studied in the last two decades, leading to high action classification rates in several action recognition datasets [4], [26]. However, labeled action training samples are, usually, difficult or expensive to obtain, since they require human effort (manual annotation). Therefore, good learning models using a limited number of labeled action videos are required. Despite the fact that action recognition has been extensively studied in the last two decades, there are few semi-supervised action recognition methods, which can exploit both labeled and unlabeled action videos in their training process. We briefly describe such methods in the following Section.

Extreme Learning Machine (ELM) [5] 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 process, like gradient descent. However, gradient descent learning techniques, like the Backpropagation algorithm, 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. ELM not only tends to reach a small training error, but also a small norm of output weights, indicating good generalization performance [6]. ELM has been successfully applied to many classification problems, including human action recognition [7], [8], [9], [12], [13].

In this paper we extend the ELM algorithm in order to exploit information appearing in both labeled and unlabeled action videos. This is achieved by introducing a proper regularization term on the MCVELM optimization process [10], which is an extension of the ELM network as will be described in the following Section. The proposed SLFN network training algorithm has been evaluated on two publicly available action recognition datasets, namely the KTH and the UCF50 datasets, where its performance is compared with that of other supervised and semi-supervised classification schemes. Experimental results denote the effectiveness of the proposed approach.

The rest of the paper is structured as follows. Section II discusses previous work related to the proposed approach. Section III describes the proposed classification algorithm in detail. Experimental results evaluating their performance are illustrated in Section IV. Finally, conclusions are drawn in Section V.

II. Related Work

As has been previously mentioned, despite the fact that action recognition has been extensively studied in the last two

decades, there are few methods approaching the problem from a semi-supervised point of view. Labeled Kernel Sparse Coding (LKSC) and l1 graphs are proposed in [14], in order to use unlabeled action videos in a sparsity-based action classification scheme. Semi-supervised Discriminant analysis with Global constraint (SDG) is proposed in [15]. SDG incorporates Linear Discriminant Analysis (LDA), Principal Component Analysis (PCA) and Locality Preserving Projections (LPP) in one optimization scheme, in order to fuse the information appearing in both labeled and unlabeled action videos.

Regarding ELM-based classification schemes, many ELM variants have been proposed in the last few years, each extending properties of the ELM network in different directions [16], [17], [18], [19], [20], [21], [22], [23], [10]. From them, those that are more related to the proposed classification scheme are the the ones proposed in [23], [10], [24]. In [23], an optimization-based regularized version of the ELM algorithm, noted as ORELM hereafter, has been proposed in order to enhance the generalization performance of the ELM network and overcome the Small Sample Size problem which is related to the original ELM optimization problem. By using a sufficiently large number of hidden layer neurons, the ELM classification scheme can be thought of as being a two-step optimization process. The first step corresponds to a non-linear mapping of the training data to a high-dimensional feature space, called ELM space hereafter, while the second one corresponds to linear data projection and classification. Based on this observation, the optimization problem of the ORELM network has been extended in order to incorporate the within-class variance of the training data representation in the ELM space, leading to increased action classification performance [10]. This algorithm is noted as MCVELM hereafter. Finally, a semi-supervised version of the ELM network has been proposed in [24]. SELM incorporates a regularizer on the optimization process of the ELM network in order to exploit information coming from unlabeled data. As will be described in the following sections, the proposed classification scheme can be considered as an extension of both the MCVELM and SELM algorithms, which is able to exploit information coming from labeled and unlabeled data and, additionally, incorporate discrimination criteria on the ELM optimization process.

III. Proposed Method

In this Section, we describe in detail the proposed (SDELM) algorithm for semi-supervised SLFN network training. Since, as it has been already mentioned, it is an extension of ELM, SELM and MCVELM algorithms, we briefly describe them in the following subsections. Here, we introduce the notation that will be used in the following subsections.

Let $\mathbf{x}_i$ $i = 1, \ldots, l, \ldots, N$ be a set of vectors, each describing an action video. The first l vectors $\mathbf{x}_i$, $i = 1, \ldots, l$ are accompanied with action class labels $c_i \in \mathcal{A}$, while the action class of the remaining $u = N - l$ vectors is not a priori known. We would like to employ these vectors and the corresponding class labels in order to train a SLFN network. For a classification problem involving the D-dimensional vectors $\mathbf{x}_i$, each belonging to one of the C classes forming the action class set $\mathcal{A}$, the network should contain D input, H hidden and C output neurons. The number of the network hidden layer neurons is usually chosen to be much higher than the number of action classes, i.e., $H \gg C$. The network target vectors $\mathbf{t}_i = [t_{i1}, ..., t_{iC}]^T$, $i = 1, \ldots, l$, each corresponding to one labeled vector $\mathbf{x}_i$, are set to $t_{ij} = 1$ for vectors belonging to class j, i.e., when $c_i = j$, and to $t_{ij} = -1$ otherwise.

In ELM-based classification schemes, 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 row of $\mathbf{W}_{out}$ and u_{kj} be the j-th element of $\mathbf{u}_k$. For a given hidden layer activation function $\Phi()$ and by using a linear activation function for the output neurons, the output $\mathbf{o}_i = [o_1, \ldots, o_{N_A}]^T$ of the ELM network corresponding to training action vector $\mathbf{x}_i$ is given by:

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

Many activation functions $\Phi()$ can be employed for the calculation of the hidden layer output, such as sigmoid, sine, Gaussian, hard-limiting and Radial Basis (RBF) functions. Let us denote by $\mathbf{\Phi} \in \mathbb{R}^{H \times N}$ a matrix containing the hidden layer output vectors $\boldsymbol{\phi}_i$, $i = 1, \ldots, N$, each corresponding to an action vector $\mathbf{x}_i$. Let us also denote by $\mathbf{\Phi}_l \in \mathbb{R}^{H \times l}$ a sub-matrix of $\mathbf{\Phi}$ containing the network hidden layer outputs corresponding to the labeled action vectors $\mathbf{x}_i$ $i = 1, \ldots, l$.

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

$$c_t = arg \max_j o_{tj},\ j = 1, ..., C. \tag{2}$$

A. The ELM algorithm

ELM has been proposed for supervised classification [5]. It assumes that the predicted network outputs $\mathbf{O} \in \mathbb{R}^{C \times l}$ are equal to the desired ones, i.e., $\mathbf{o}_i = \mathbf{t}_i$, $i = 1, ..., l$. Given this assumption, $\mathbf{W}_{out}$ can be analytically calculated by solving for:

$$\mathbf{W}_{out}^T \mathbf{\Phi}_l = \mathbf{T}, \tag{3}$$

where $\mathbf{T} \in \mathbb{R}^{C \times l}$ is a matrix containing the network target vectors. The network output weights minimizing $\|\mathbf{W}_{out}^T \mathbf{\Phi}_l - \mathbf{T}\|_F^2$ are given by:

$$\mathbf{W}_{out} = \mathbf{\Phi}_l^{\dagger}\, \mathbf{T}^T, \tag{4}$$

where $\mathbf{\Phi}_l^{\dagger} = \left(\mathbf{\Phi}_l \mathbf{\Phi}_l^T\right)^{-1} \mathbf{\Phi}_l$ is the generalized pseudo-inverse of $\mathbf{\Phi}_l^T$.

B. The MCVELM algorithm

MCVELM algorithm has also be proposed for supervised classification [10]. MCVELM solves the following optimization problem:

$$\mathbf{W}_{out} = \underset{\mathbf{W}_{out}}{argmin}\ \frac{1}{2}\|\mathbf{S}_w^{\frac{1}{2}} \mathbf{W}_{out}\|_F^2 + \frac{\lambda}{2} \sum_{i=1}^{l} \|\boldsymbol{\xi}_i\|_2^2, \tag{5}$$

$$s.t.: \quad \mathbf{W}_{out}^T \boldsymbol{\phi}_i = \mathbf{t}_i - \boldsymbol{\xi}_i,\ i = 1, ..., l, \tag{6}$$

where $\boldsymbol{\xi}_i \in \mathbb{R}^C$ is the error vector corresponding to training vector $\mathbf{x}_i$ and λ is a parameter denoting the importance of

the training error in the optimization problem. $\mathbf{S}_w$ denotes the within-class scatter matrix of the training vectors in the ELM space is given by:

$$\mathbf{S}_w = \sum_{j=1}^{C} \sum_{i=1}^{l} \beta_{ij} (\boldsymbol{\phi}_i - \boldsymbol{\mu}_j)(\boldsymbol{\phi}_i - \boldsymbol{\mu}_j)^T, \quad (7)$$

where β_{ij} is an index denoting if vector $\mathbf{x}_i$ belongs the to class j, i.e., $\beta_{ij} = 1$, if $c_i = j$ and $\beta_{ij} = 0$ otherwise. $\boldsymbol{\mu}_j \in \mathbb{R}^H$ is the mean vector of class j in the ELM space. In the case where the number of vectors belonging to different classes vary, the contribution of each class to the calculation of $\mathbf{S}_w$ can be appropriately weighted. In addition, in the case where the training data form multimodal classes in the ELM space, i.e., classes formed by multiple subclasses, $\mathbf{S}_w$ can be modified in order to describe the within-subclass variance of the training data. By solving the optimization problem (5), $\mathbf{W}_{out}$ is given by:

$$\mathbf{W}_{out} = \left(\boldsymbol{\Phi}_l \boldsymbol{\Phi}_l^T + \frac{1}{c} \mathbf{S}_w \right)^{-1} \boldsymbol{\Phi}_l \mathbf{T}^T. \quad (8)$$

C. The SELM algorithm

SELM solves the following optimization problem:

$$\mathbf{W}_{out} = \underset{\mathbf{W}_{out}}{argmin} \, \|\mathbf{W}_{out}^T \boldsymbol{\Phi}_l - \mathbf{T}\|_F^2, \quad (9)$$

$$s.t. : \quad \sum_{i=1}^{N} \sum_{j=1}^{N} w_{ij} \|\mathbf{W}_{out}^T \boldsymbol{\phi}_i - \mathbf{W}_{out}^T \boldsymbol{\phi}_j\|_2^2 = 0, \quad (10)$$

where w_{ij} is a value denoting the similarity between $\boldsymbol{\phi}_i$ and $\boldsymbol{\phi}_j$. By solving (9), $\mathbf{W}_{out}$ is given by:

$$\mathbf{W}_{out} = \left(\left(\mathbf{J} + \lambda \mathbf{L} \right) \boldsymbol{\Phi} \right)^{\dagger} \mathbf{J} \mathbf{T}^T, \quad (11)$$

where $\mathbf{J} = diag(1, 1, \ldots, 0, 0)$ with the first l diagonal entries as 1 and the rest 0, $\mathbf{L}$ is the Graph Laplacian matrix [25] encoding the similarity between the training vectors.

D. The proposed Semi-supervised Discriminant ELM algorithm

In this paper, we propose to solve the following optimization problem for the calculation of the network output weights $\mathbf{W}_{out}$:

$$\mathbf{W}_{out} = \underset{\mathbf{W}_{out}}{argmin} \, \frac{1}{2} \|\mathbf{S}_X^{\frac{1}{2}} \mathbf{W}_{out}\|_F^2 + \frac{\lambda_1}{2} \sum_{i=1}^{l} \|\boldsymbol{\xi}_i\|_2^2 \quad (12)$$

$$s.t. : \quad \mathbf{W}_{out}^T \boldsymbol{\phi}_i = \mathbf{t}_i - \boldsymbol{\xi}_i, \;\; i = 1, ..., l, \quad (13)$$

$$\sum_{i=1}^{N} \sum_{j=1}^{N} w_{ij} \|\mathbf{W}_{out}^T \boldsymbol{\phi}_i - \mathbf{W}_{out}^T \boldsymbol{\phi}_j\|_2^2 = 0. \quad (14)$$

By substituting (13) in (12) and taking the equivalent dual problem of (12) with respect to (14), we obtain:

$$\mathcal{J}_D = \frac{1}{2} \|\mathbf{S}_X^{\frac{1}{2}} \mathbf{W}_{out}\|_F^2 + \frac{\lambda_1}{2} \|\mathbf{W}_{out}^T \boldsymbol{\Phi}_l - \mathbf{T}\|_F^2 + \frac{\lambda_2}{2} Tr\left(\mathbf{W}_{out}^T \boldsymbol{\Phi} \mathbf{L} \boldsymbol{\Phi}^T \mathbf{W}_{out} \right). \quad (15)$$

Since action class discrimination in the projection space is handled by the second term in (15), $\mathbf{S}_X$ is chosen to describe action class compactness properties. That is, $\mathbf{S}_X$ can be either the within-(sub)class matrix (7) expressing (sub)class compactness in the ELM space, or the total scatter matrix $\mathbf{S}_T$ [11], given by:

$$\mathbf{S}_T = \sum_{j=1}^{C} \sum_{i=1}^{N} (\boldsymbol{\phi}_i - \boldsymbol{\mu})(\boldsymbol{\phi}_i - \boldsymbol{\mu})^T, \quad (16)$$

where $\boldsymbol{\mu} \in \mathbb{R}^H$ is the mean vector of the entire training set in the ELM space. $\mathbf{S}_T$ expresses the compactness of the entire training set in the ELM space. We should note here that the adoption of a matrix $\mathbf{S}_X = \mathbf{I}$ in (12), corresponds to an extension of the ORELM algorithm to semi-supervised SLFN network training.

By solving for $\frac{\theta \mathcal{J}_2}{\theta \mathbf{W}_{out}} = 0$, $\mathbf{W}_{out}$ is given by:

$$\mathbf{W}_{out} = \left[\boldsymbol{\Phi} \left(\mathbf{1} + \frac{\lambda_2}{\lambda_1} \mathbf{L} \right) \boldsymbol{\Phi}^T + \frac{1}{\lambda_1} \mathbf{S}_X \right]^{-1} \boldsymbol{\Phi}_l \mathbf{T}^T. \quad (17)$$

As can be seen in (16) the adoption of the proposed optimization scheme for the calculation of $\mathbf{W}_{out}$ leads to the determination of network output weights exploiting both information appearing in unlabeled training data (expressed by the Graph Laplacian matrix $\mathbf{L}$) and discrimination criteria (expressed by the matrix $\mathbf{S}_X$). It should be noted here that, the calculation of $\mathbf{S}_X$ and $\mathbf{L}$ in the ELM $\mathbb{R}^H$, rather than the input space $\mathbb{R}^D$, has the advantage that nonlinear relationships between the training vectors $\mathbf{x}_i$ can be described.

IV. Experiments

In this Section, we present experiments conducted in order to evaluate the performance of the proposed action classification algorithm. We have used two publicly available datasets to this end, namely the KTH and UCF50 databases. Comprehensive description of the databases used in our experiments are provided in the following subsections.

We employ the Action Bank [3] for action video representation. The dimensionality of the 14964-dimensional Action Bank vectors has been reduced by applying PCA, so that 98% of the energy is preserved, resulting to 91- and 467-dimensional feature vectors for the KTH and the UCF50 cases, respectively. We compare the performance of the proposed SDELM algorithm with that of ELM [5], ORELM [23][1], kernel Support Vector Machine employing RBF kernel (kSVM)[2], kernel Laplacian SVM employing RBF kernel (LapSVM) [25][3] and SELM [24] classifiers. The sigmoid activation function has been used for all the ELM-based classification schemes. The optimal parameter values for all the algorithms have been determined by applying a grid search strategy using the values $\lambda = 10^r$ for ORELM and SELM, $C = 10^r$ and $\sigma = 10^r$ for kSVM, $\gamma_A = 10^r$, $\gamma_I = 10^r$ and $\sigma = 10^r$ for LapSVM and $\lambda_1 = 10^r$, $\lambda_2 = 10^r$ for the proposed SDELM algorithm using the values $r = -6, \ldots, 6$. The number H of the network hidden layer neurons has been set equal to $H = 500$ and $H = 1000$ for all the ELM-based classification schemes on the KTH and the UCF50 cases, respectively. Finally, we compare the performance of the proposed action recognition method

[1] http://www.ntu.edu.sg/home/egbhuang/elm_codes.html

[2] http://www.csie.ntu.edu.tw/~cjlin/libsvm/

[3] http://manifold.cs.uchicago.edu/manifold_regularization/manifold.html

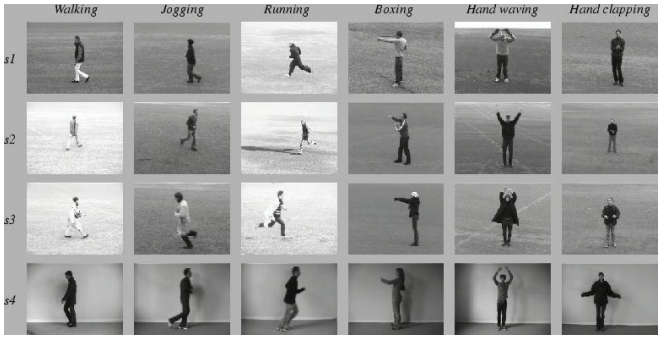

Fig. 1. *Video frames from the KTH action database for the four different scenarios.*

with that of other methods evaluating their performance on the above-mentioned databases.

A. The KTH action database

The KTH action database consists of 600 videos depicting 25 persons, performing six actions each [26]. The actions appearing in the database are: 'walking', 'jogging', 'running', 'boxing', 'hand waving' and 'hand clapping'. Four different scenarios have been recorded: outdoors (s1), outdoors with scale variation (s2), outdoors with different clothes (s3) and indoors (s4), as illustrated Figure 1. The persons are free to change motion speed and direction between different action realizations. The most widely adopted experimental setting on this data set is based on a split (16 training and 9 test persons) that has been used in [26].

B. The UCF50 action database

The UCF50 action database consists of 6680 realistic videos taken from YouTube, each belonging to one of 50 action classes. The database is very challenging, due to large variations in camera motion, subject appearance and pose, subject scale, view angle, cluttered background, illumination conditions, etc. For all the 50 categories, the videos are grouped into 25 groups, where each group consists of more than 4 action clips. The video clips in the same group may share some common features, such as the appearance of the same person, similar background, similar view angle, and so on. The most widely adopted experimental setting on this database is the 5-fold group-wise cross-validation procedure. That is, the videos are split on five sets, each containing 5 groups. On each fold of the cross-validation procedure, the videos belonging to 4 sets, i.e., 20 groups, are used for training and the videos belonging to the remaining set are used for testing. This procedure is performed 5 times, one for each test set. Example video frames from this database are illustrated in Figure 2.

C. Experimental Results

The mean action classification rates for the ELM, ORELM, kSVM algorithms and the proposed SDELM algorithm employing $\mathbf{S}_w$ and $\mathbf{S}_T$ for supervised action classification, i.e., by exploiting the available labeling information for the entire training set, are illustrated in Table I. As can be seen, the proposed SDELM algorithm outperforms all the competing

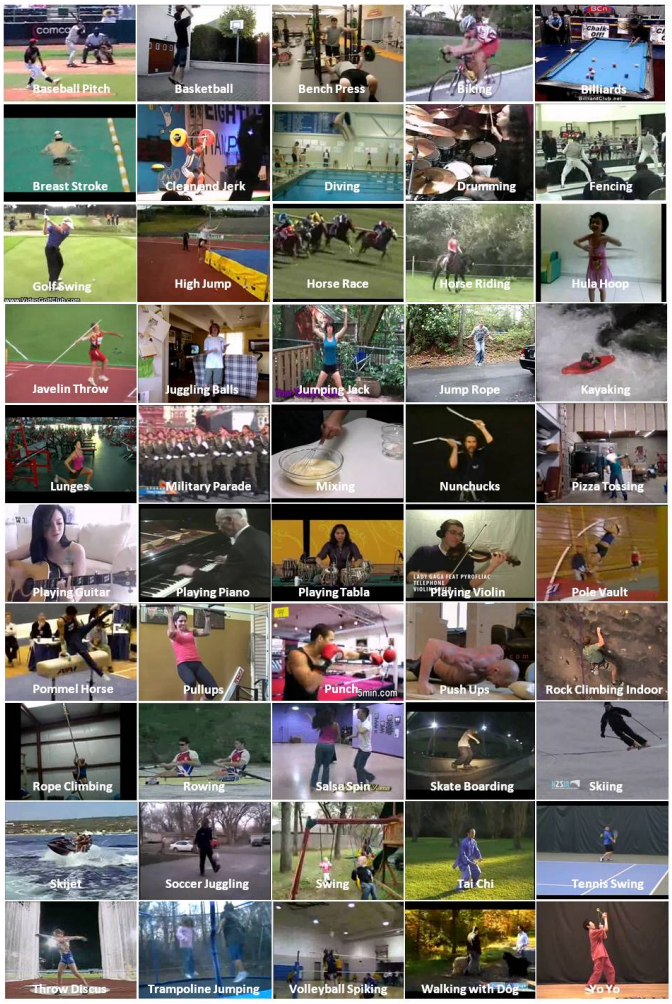

Fig. 2. *Video frames from the UCF50 action database.*

ones in both databases. The confusion matrix obtained for the KTH database is illustrated in Figure 3. In Table I, we also provide the mean training times for all the algorithms. All the experiments have been conducted on a 2.4GHz, 16GB, 64-bit Windows 8 PC, using a MATLAB implementation. As can be seen, the proposed DELM algorithm is computationally efficient, since its learning speed is comparable with that of ELM and ORELM, while the learning process kSVM is quite slow, since it is requires gradient descend based optimization.

We have also performed semi-supervised action classification on the KTH and UCF50 databases. We have ordered the training data forming the action classes of the KTH and UCF50 databases by using a random permutation of their indices and used 1% and 5% of them as labeled and the remaining samples as unlabeled data. The action classification rates obtained by following this process and applying the SELM, LapSVM and the proposed SDELM algorithms employing $\mathbf{I}$, $\mathbf{S}_w$ and $\mathbf{S}_T$ are illustrated in Table II. As can be seen, the proposed SDELM algorithm outperforms both the SELM and the LapSVM algorithms in most cases. Furthermore, it can be seen that the ELM-based classification schemes are computationally more efficient compared to the LapSVM algo-

TABLE I. ACTION CLASSIFICATION RATES AND MEAN TRAINING TIMES FOR SUPERVISED CLASSIFICATION ON THE KTH AND THE UCF50 ACTION DATABASES.

	KTH		UCF50	
	Accuracy	Training Time	Accuracy	Training Time
ELM	90.74%	89.8ms	60.6%	2.3s
ORELM	99.07%	98.4ms	56.28%	1.3522s
kSVM	98.15%	420ms	57.9%	30.864s
SDELM ($\mathbf{S}_w$)	98.61%	192.7ms	**61.21%**	1.475s
SDELM ($\mathbf{S}_T$)	**99.54%**	164.4ms	60.94%	1.096s

TABLE II. ACTION CLASSIFICATION RATES AND MEAN TRAINING TIMES FOR SEMI-SUPERVISED CLASSIFICATION ON THE KTH AND UCF50 DATABASES.

	KTH				UCF50			
	$l = 6$ (1 per action class)		$l = 18$ (3 per action class)		$l = 0.01N$		$l = 0.05N$	
	Accuracy	Training Time	Accuracy	Training Time	Accuracy	Training Time	Accuracy	Training Time
SELM	71.76%	106.7ms	82.87%	105.5ms	11.25%	4.7824s	17.01%	4.8281s
LapSVM	**82.41%**	203.9ms	**91.2%**	223.4ms	14.43%	30.8646s	31.54%	17.9869s
SDELM (**I**)	80.56%	115.1ms	90.74%	116.8ms	14.38%	3.6857s	32.2%	3.7191s
SDELM ($\mathbf{S}_w$)	80.09%	134.9ms	90.74%	145.1ms	**16.54%**	1.5262s	32.12%	1.7585s
SDELM ($\mathbf{S}_T$)	77.31%	126.6ms	**91.2%**	137ms	16.5%	1.3159s	**33.12%**	1.3091s

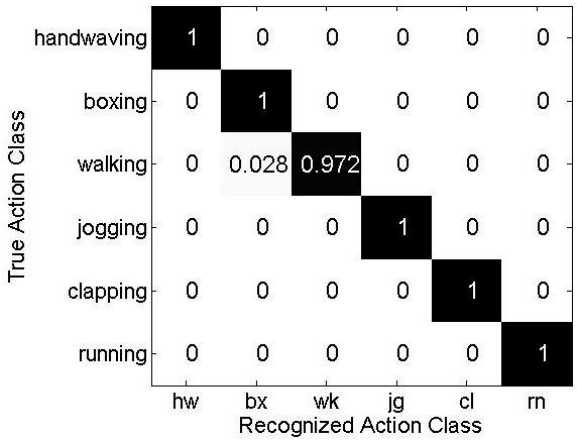

Fig. 3. *Confusion matrix for supervised action classification on the KTH database.*

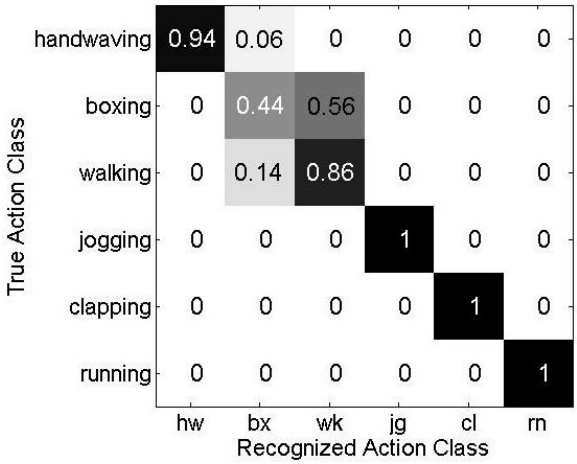

Fig. 4. *Confusion matrix for semi-supervised action classification on the KTH database ($l = 0.05N_I$).*

TABLE III. COMPARISON RESULTS ON THE KTH DATABASE.

	Accuracy
Method [27]	94.3%
Method [28]	94.5%
Method [29]	94.5%
Method [30]	94.5%
Method [3]	98.2%
SDELM ($\mathbf{S}_T$)	**99.54%**

TABLE IV. COMPARISON RESULTS ON THE UCF50 DATABASE.

	Accuracy
Method [31]	38.8%
Method [2]	47.9%
Method [3]	57.9%
SDELM ($\mathbf{S}_w$)	**61.21%**

rithm. The confusion matrix obtained by applying the proposed SDELM algorithm, employing $\mathbf{S}_T$, on the KTH database for the case of $l = 0.05N_I$ is illustrated in Figure 4.

Finally, we compare the performance of the proposed action classification scheme with that of others evaluating their performance on the KTH and UCF50 databases in Tables III, IV, respectively. As can be seen, the proposed SDELM algorithm combined with the Action Bank action video representation leads to state-of-the-art performance in both databases.

V. CONCLUSION

In this paper, we proposed an extension of the ELM algorithm for semi-supervised SLFN network training. The proposed algorithm both exploits information coming from both labeled and unlabeled data and incorporates discrimination criteria describing compactness properties of the training set in the ELM space in the ELM optimization process. The proposed algorithm has been evaluated in human action recognition where its performance has been compared with that of other (semi-)supervised classification schemes. Experimental results on two publicly available datasets denote that it is able to provide state-of-the-art performance for both supervised and

semi-supervised action classification.

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] I. Laptev, *On space-time interest points*, International Journal of Computer Vision, vol. 64, no. 2, pp. 107-123, 2005.

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

[3] S. Sadanand and J. J. Corso, *Action Bank: A High-Level Representation of Activity in Video*, IEEE Conference on Computer Vision and Pattern Recognition, 2012.

[4] L. Gorelick, M. Blank, E. Shechtman, M. Irani and R. Basri, *Actions as space-time shapes*, IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 29, no. 12, pp. 2247-2253, 2007.

[5] G. B. Huang, Q. Y. Zhu and C. K. Siew, *Extreme learning machine: a new learning scheme of feedforward neural networks*, IEEE International Joint Conference on Neural Networks, 2004.

[6] P. L. Bartlett, *The sample complexity of pattern classification with neural networks: the size of the weights is more important than the size of the network*, IEEE Transactions on Information Theory, vol. 44, no. 2, pp. 525-536, 1998.

[7] R. Minhas, A. Baradarani, S. Seifzadeh, Q. M. Jonathan, *Human action recognition using extreme learning machine based on visual vocabularies*, Neurocomputing, vol. 73, no. 10, pp. 1906-1917, 2010.

[8] A. Iosifidis, A. Tefas and I. Pitas, *Multi-view Human Action Recognition Under Occlusion based on Fuzzy Distances and Neural Networks*, European Signal Processing Conference, 2012.

[9] A. Iosifidis, A. Tefas and I. Pitas, *Dynamic action recognition based on Dynemes and Extreme Learning Machine*, Pattern Recognition Letters, vol. 34, pp. 1890-1898, 2013.

[10] A. Iosifidis, A. Tefas and I. Pitas, *Minimum Class Variance Extreme Learning Machine for Human Action Recognition*, IEEE Transactions on Circuits and Systems for Video Technology, vol. 23, no. 11, pp. 1968-1979, 2013.

[11] A. Iosifidis, A. Tefas and I. Pitas, *Minimum Variance Extreme Learning Machine for Human Action Recognition*, IEEE International Conference on Acoustics, Speech and Signal Processing, 2014.

[12] A. Iosifidis, A. Tefas and I. Pitas, *Active Classification for Human Action Recognition*, IEEE International Conference on Image Processing, 2013.

[13] A. Iosifidis, A. Tefas and I. Pitas, *Person Identification from Actions based on Artificial Neural Networks*, IEEE Symposium Series on Computational Intelligence, 2013.

[14] S. Yang, X. Wang, L. Yang, Y. Han, and L. Jiao, *Semi-supervised action recognition in video via Labeled Kernel Sparse Coding and sparse L1 graph*, Pattern Recognition Letters, vol. 33, pp. 19511956, 2012.

[15] X. Zhao, X. Li, C. Pang and S. Wang, *Human action recognition based on semi-supervised discriminant analysis with global constraint*, Neurocomputing, in press.

[16] M. B. Li, G. B. Huang, P. Saratchandran and N. Sundararajan, *Fully complex extreme learning machine*, Neurocomputing, vol. 68, no. 13, pp. 306-314, 2005.

[17] N. Y. Liang, B. B. Huang, P. Saratchandran and N. Sundararajan, *A fast and accurate on-line sequantial learning algorithm for feedforward networks*, IEEE Transactions on Neural Networks, vol. 17, no. 6, pp. 1411-1423, 2006.

[18] G. B. Huang, L. Chen and C. K. Siew, *Universal Approximation Using Incremental Constructive Feedforward Networks with Random Hidden Nodes*, IEEE Transactions on Neural Networks, vol. 17, no. 4, pp. 879-892, 2006.

[19] G. B. Huang and L. Chen, *Convex incremental extreme learning machine*, Neurocomputing, vol. 70, no. 16, pp. 3056-3062, 2008.

[20] G. Feng, G. B. Huang, Q. Lin, Q. and R. Gay., *Error minimized extreme learning machine with growth of hidden nodes and incremental learning*, IEEE Transactions on Neural Networks, vol. 20, no. 8, pp. 1352-1357, 2009.

[21] Y. Miche, A. Sorjamaa, P. Bas, O. Simula, C. Jutten and A. Lendasse, *OP-ELM: Optimally pruned extreme learning machine*, IEEE Transactions on Neural Networks, vol. 21, no. 1, pp. 158-162, 2010.

[22] Y. Wang, F. Cao and Y. Yuan, *A study on effectiveness of extreme learning machine*, Neurocomputing, vol. 74, no. 16, pp. 2483-2490, 2011.

[23] G. B. Huang, H. Zhou, X. Ding and R. Zhang, *Extreme learning machine for regression and multiclass classification*, IEEE Transactions on Systems, Man, and Cybernetics, Part B: Cybernetics, vol. 42, no. 2, pp. 513-529, 2012.

[24] J. Liu, Y. Cheng, M. Liu and Z. Zhao, *Semi-supervised ELM with application in sparse calibrated location estimation*, Neurocomputing, vol. 74, pp. 2566-2572, 2011.

[25] M. Belkin, P. Niyogi and V. Sindhwani, *Manifold regularization: A geometric framework for learning from labeled and unlabeled examples*, Journal of Machine Learning Research, vol. 7, pp. 2399-2434, 2006.

[26] C. Schuldt, I. Laptev and B. Caputo, *Recognizing human actions: A local SVM approach*, International Conference on Pattern Recognition, 2004.

[27] J. Liu and M. Shah, *Learning human actions via information maximization*, IEEE Conference on Computer Vision and Pattern Recognition, 2008.

[28] A. Gilbert, J. Illingworth and R. Bowden, *Fast realistic multi-action recognition using mined dense spatio-temporal features*, International Conference on Computer Vision, 2009.

[29] A. Kovashka and K. Grauman, *Learning a hierarchy of discriminative space-time neighborhood features for human action recognition*, IEEE Conference on Computer Vision and Pattern Recognition, 2010.

[30] X. Wu, D. Xu, L. Duan and J. Luo, *Action recognition using context and appearance distribution features*, IEEE Conference on Computer Vision and Pattern Recognition, 2011.

[31] A. Olivia and A. Torralba, *Modeling the shape of the scene: A holistic representation of the spatial envelope*, International Journal of Computer Vision, 2001.

[32] F. Samaria and A. Harter, *Parameterisation of a stochastic model for human face identification*, IEEE Workshop on Applications of Computer Vision, 1994.