Regularized Extreme Learning Machine for large-scale media content analysis

Alexandros Iosifidis, Anastasios Tefas and Ioannis Pitas

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

Abstract
In this paper, we propose a new regularization approach for Extreme Learning Machine-based Single-hidden Layer Feedforward Neural network training. We show that the proposed regularizer is able to weight the dimensions of the ELM space according to the importance of the network's hidden layer weights, without imposing additional computational and memory costs in the network learning process. This enhances the network's performance and makes the proposed approach suitable for learning non-linear decision surfaces in large-scale classification problems. We test our approach in medium- and large-scale face recognition problems, where we observe its superiority when compared to the existing regularized Extreme Learning Machine classifier in both constrained and unconstrained problems, thus making our approach applicable in demanding media analysis applications such as those appearing in digital cinema production.

Keywords: Extreme Learning Machine, Regularization, Face Recognition, Large-scale learning

1 Introduction

Extreme Learning Machine has been proposed as an alternative algorithm for Single-hidden Layer Feedforward Neural (SLFN) networks training [8], towards overcoming the computational bottleneck of related SLFN network training algorithms which are based on gradient descend optimization, e.g. the Backpropagation [16] algorithm. The main idea of ELM is that the network hidden layer weights need not to be learned, but they can be randomly assigned instead. A learning process is applied only for the determination of the network output weights by solving an optimization problem that has a closed form solution. Such an approach has also been found to be efficient in earlier attempts on neural networks training of several topologies [1, 18, 21, 2]. By using a very large number of hidden layer neurons, ELM networks can achieve satisfactory performance in many classification problems [15]. It has been also proven that ELMs have the properties of global approximators in the case where the number of hidden layer neurons is equal to the cardinality of the training set [5, 22]. Recently, it has been shown that ELM networks can achieve state-of-the-art performance in many small- and medium-scale classification problems related to media content analysis, since for such problems the realization of networks having a number of hidden layer neurons comparable to the training set cardinality is possible [7].

In order to achieve satisfactory performance in large-scale classification problems involving high-dimensional data, ELM networks need to exploit regularized solutions for the calculation of the network output weights [7]. Regularized ELM networks minimizing both the network training error and the (Frobenius) norm of the network output weights have been proposed [7, 9]. Regularized ELM networks have been shown to outperform standard ELM networks, while not requiring additional computational cost. In this paper, we propose a regularized solution for the network output weights calculation of ELM networks. When compared to the standard (Frobenius) norm regularization, which leads to uniform network output weights regularization, the proposed solution can appropriately regularize the dimensions of the obtained network output weights, while not requiring additional training computational cost.

We test the proposed classifier in a media content analysis application, i.e. human face recognition. This problem has received much attention during the last two decades, since it is the first processing step towards semantic image/video analysis and visual content analytics [10]. However, much of the research conducted until now has been focused on a restricted application scenario, that involves lab-generated visual data having small to medium scale in both resolution and size. Resent advances in technological equipment (e.g. cameras and smartphones), as well as the accessibility of social and image/video sharing web applications in our daily lives have rejuvenated research interest in this area, since the problem to be solved has been extended in three directions, i.e. visual content resolution, data size and difficulty. In order to highlight these differences between the two application scenarios, recent works characterize the new face recognition problem as an open-universe problem (when compared to the restricted application scenario noted as closed-universe face recognition problem) [14]. Experimental analysis on five publicly available databases shows that the proposed classifier can achieve almost perfect classification performance in the closed-universe face recognition problem by exploiting very simple facial image representations (i.e. vectorized image intensity values). In the open-universe face recognition problem, the proposed classifier outperforms both standard and regularized ELM networks, while both its training and test complexities are the same with that of the ELM algorithm. These observations indicate that the proposed approach is appropriate for demanding large-scale media analysis applications such as digital cinema production where a large amount of video streams from multiple cameras is captured every shooting day and it should be analysed and described for the post-production. In such an application scenario, the proposed approach can be used for automatic actor recognition, something that can facilitate subsequent post-processing steps.

2 Previous Work

In this Section, we introduce notation that will be used throughout the paper and we briefly describe the ELM and regularized ELM algorithms. Let us assume that an annotated visual database contains facial images depicting C persons. By applying face detection and tracking techniques [23], facial images depicting the persons in the database can be extracted and pre-processed. This process leads to the determination of facial image vectors $\mathbf{x}_i \in \mathbb{R}^D,\ i = 1, \ldots, N$, which are accompanied by the corresponding person ID labels c_i. We would like to employ the data $\{\mathbf{x}_i, c_i\}_{i=1,\ldots,N}$ in order to train a SLFN network. In such classification problems, the SLFN network consists of D input, L hidden and C output neurons. The number of hidden layer neurons L is a parameter of any neural network training algorithm. We employ the person ID labels c_i in order to form target vectors $\mathbf{t}_i \in \mathbb{R}^C$. The elements of the target vectors are set equal to $t_{ik} = 1$, when $c_i = k$, and $t_{ik} = -1$, otherwise.

ELM assigns randomly the network hidden layer weights $\mathbf{W}_{in} \in \mathbb{R}^{L \times D}$. By exploiting an activation function $\phi(\cdot)$, the training data $\mathbf{x}_i$ are mapped to the so-called ELM space $\mathbb{R}^L$, i.e. $\mathbf{x}_i \in \mathbb{R}^D \overset{\phi(\cdot)}{\rightarrow} \boldsymbol{\phi}_i \in \mathbb{R}^L$. It has been shown that almost any nonlinear piecewise continuous activation functions $\Phi(\cdot)$ can be used for the calculation of the network hidden layer outputs, e.g. the sigmoid, sine, Gaussian, hard-limiting, Radial Basis Function (RBF), RBF-χ^2, Fourier series, etc [5, 7]. After the determination of the data representations in the ELM space and by using a linear activation function for the output layer neurons, the network output weights $\mathbf{W}_{out} \in \mathbb{R}^{L\times C}$ are analytically calculated so that to minimize the training error (in the sense of regression error), i.e. $\mathbf{W}_{out}$ is determined to be the matrix minimizing $\|\mathbf{W}_{out}^T\mathbf{\Phi} - \mathbf{T}\|_F^2$, where $\mathbf{\Phi} = [\boldsymbol{\phi}_1, \dots, \boldsymbol{\phi}_N]$, $\mathbf{T} = [\mathbf{t}_1, \dots, \mathbf{t}_N]$ and $\|\cdot\|_F$ denotes the matrix Frobenius norm. That is, $\mathbf{W}_{out}$ is obtained by the mean squares approximation, i.e. $\mathbf{W}_{out} = \left(\mathbf{\Phi}\mathbf{\Phi}^T\right)^{-1}\mathbf{\Phi}\mathbf{T}^T$. In order to avoid overfitting problems, RELM calculates the network output weights by using the Ridge Regression-based approximation and the network output weights are given by $\mathbf{W}_{out} = \left(\mathbf{\Phi}\mathbf{\Phi}^T + \frac{1}{c}\mathbf{I}\right)^{-1}\mathbf{\Phi}\mathbf{T}^T$, where $\mathbf{I} \in \mathbb{R}^{L\times L}$ is the identity matrix.

After the calculation of the network output weights $\mathbf{W}_{out}$, the network response for a (test) vector $\mathbf{x}_t \in \mathbb{R}^D$ is given by $\mathbf{o}_t = \mathbf{W}_{out}^T\boldsymbol{\phi}_t$ and $\mathbf{x}_t$ is usually classified to the (person ID) class corresponding to the maximal network output, i.e. $l_t = \arg\max_j \, o_{tj}$.

3 Proposed Regularization

In order to obtain a robust solution for the network output weights $\mathbf{W}_{out}$ with respect to perturbations to the data representations in the ELM space, we solve the following optimization problem:

$$\mathcal{J} = \arg\min_{\mathbf{W}_{out}} \|\mathbf{W}_{out}^T\mathbf{\Phi} - \mathbf{T}\|_F^2, \tag{1}$$

$$\text{subject to}: \quad \mathbf{W}_{out}^T\boldsymbol{\phi}_i = \mathbf{W}_{out}^T\tilde{\boldsymbol{\phi}}_{i,m}, \;\; i = 1, \dots, N, \;\; m = 1, \dots, M, \tag{2}$$

where $\tilde{\boldsymbol{\phi}}_{i,m} \in \mathbb{R}^L$ is a perturbed copy of $\boldsymbol{\phi}_i$. That is, we would like to learn network output weights that lead to network outputs of perturbed samples $\tilde{\boldsymbol{\phi}}_{i,m}$ which are as close as possible to the network outputs for the original samples $\boldsymbol{\phi}_i$, i.e. $\mathbf{W}_{out}^T\tilde{\boldsymbol{\phi}}_{i,m} = \tilde{\mathbf{o}}_{i,m} \simeq \mathbf{o}_i = \mathbf{W}_{out}^T\boldsymbol{\phi}_i$, while (at the same time) the network training error is as low as possible.

A similar approach has been exploited in Autoencoders, leading to the so-caled Denoising Autoencoders [19]. Recently, it has been also shown that a similar approach can be exploited for training feedforward neural networks [4]. In both cases, it has been shown that the adoption of perturbed samples has an effect of regularization on the obtained network parameters, which enhances the generalization ability of the trained networks and leads to better generalization performance. In order to incorporate such a regularization approach in Backpropagation-based network training, the training set is usually enriched by generating random perturbations of the training data. However, as we shall show next, this is not the case for the proposed regularized ELM network, since the network output weights can be obtained based on closed form solution.

A perturbed sample $\tilde{\boldsymbol{\phi}}_{i,m}$ is obtained by copying the j-th element of $\boldsymbol{\phi}_i$ with a probability equal to p, or by setting the corresponding element equal to zero with probability equal to $(1-p)$. This process can be expressed as $\tilde{\boldsymbol{\phi}}_{i,m} = \mathbf{b}_{i,m} \circ \boldsymbol{\phi}_i$, where $\mathbf{b}_{i,m} \in \mathbb{R}^L$, having its elements equal to one with probability p, or zero with probability $(1-p)$ and $\circ$ denotes the element-wise product of two vectors. By setting $\boldsymbol{\phi}_{i,m} = \boldsymbol{\phi}_i - \tilde{\boldsymbol{\phi}}_{i,m}$, the constraint (2) can be replaced with the following one:

$$\mathbf{W}_{out}^T\boldsymbol{\phi}_{i,m} = 0, \;\; i = 1, \dots, N, \;\; m = 1, \dots, M. \tag{3}$$

By substituting (3) in (1) and taking the equivalent dual problem, we obtain:

$$\mathcal{J}_D = \arg\min_{\mathbf{W}_{out}} \|\mathbf{W}_{out}^T\mathbf{\Phi} - \mathbf{T}\|_F^2 + \frac{c}{M}\sum_{m=1}^{M}\|\mathbf{W}_{out}^T\mathbf{\Phi}_m\|_F^2, \tag{4}$$

where $\mathbf{\Phi}_m = [\boldsymbol{\phi}_{1,m}, \ldots, \boldsymbol{\phi}_{N,m}]$. From (4), the network output weights are obtained by:

$$\mathbf{W}_{out} = \left(\mathbf{\Phi}\mathbf{\Phi}^T + \frac{c}{M} \sum_{m=1}^{M} \mathbf{\Phi}_m \mathbf{\Phi}_m^T \right)^{-1} \mathbf{\Phi}\mathbf{T}^T. \tag{5}$$

Thus, in order to calculate the network output weights by using the proposed regularization approach, we can enrich our dataset by adding M random perturbations of each sample (when represented in the ELM space). While this approach can be easily implemented in small- and medium-scale problems, it's application in large-scale problems is difficult. However, by assuming that the number of employed perturbations is high ($M \rightarrow \infty$) [3], based on the weak law of large numbers the regularization term $\mathbf{R} = \frac{1}{M}\sum_{m=1}^{M} \mathbf{\Phi}_m \mathbf{\Phi}_m^T$ in (5) converges to its expected value:

$$\mathbf{R} \rightarrow E\left[\frac{1}{M} \sum_{m=1}^{M} \mathbf{\Phi}_m \mathbf{\Phi}_m^T \right]_{M \rightarrow \infty} = (\mathbf{\Phi}\mathbf{\Phi}^T) \circ \mathbf{P}, \tag{6}$$

where $\mathbf{P} = (1-p)^2\mathbf{1} + (1-p)^2\mathbf{I}$ and $\mathbf{1} \in \mathbb{R}^{L \times L}$ is a matrix of ones. By using (6), (5) can be expressed as:

$$\mathbf{W}_{out} = \left(\mathbf{\Phi}\mathbf{\Phi}^T + c\left[\mathbf{\Phi}\mathbf{\Phi}^T\right] \circ \mathbf{P} \right)^{-1} \mathbf{\Phi}\mathbf{T}^T = \left(\left[\mathbf{\Phi}\mathbf{\Phi}^T\right] \circ [\mathbf{1} + c\mathbf{P}] \right)^{-1} \mathbf{\Phi}\mathbf{T}^T. \tag{7}$$

Thus, we can observe that both the time and memory complexity of the proposed regularized ELM is the same with that of ELM and RELM.

4 Comparison of the two regularization terms

Here we compare the regularization terms of RELM [7] with that of the proposed regularization approach. The regularizer used in standard RELM is a uniform regularizer, i.e. all the dimensions of $\mathbf{W}_{out}$ are regularized by using the same quantity $\frac{1}{c}$. Thus, except of leading to more stable solutions for the network output weights calculation, it does not highlight the relevant importance of different output network weight dimensions (note that each dimension of the network output weight is related to the corresponding hidden layer weight).

The solution for the network output weights given in (7) can be expressed as follows:

$$\mathbf{W}_{out} = \left(\alpha \mathbf{\Phi}\mathbf{\Phi}^T + \beta \mathbf{D} \right)^{-1} \mathbf{\Phi}\mathbf{T}^T, \tag{8}$$

where $\alpha = 1 + c(1-p)^2$, $\beta = c(1-p)^2$ and $\mathbf{D} = \left[\mathbf{\Phi}\mathbf{\Phi}^T\right] \circ \mathbf{I}$. By analysing the elements of the diagonal matrix $\mathbf{D}$ we obtain $D_{ll} = \sum_{i=1}^{N} \phi_{il}^2,\ \ j = 1, \ldots, L$, where ϕ_{il} denotes the l-th element of $\boldsymbol{\phi}_i$. Let us denote by $\mathbf{w}_l \in \mathbb{R}^D$ the l-th hidden layer weight (i.e. the l-th column of $\mathbf{W}_{in}$). Then, we have:

$$D_{ll} = \sum_{i=1}^{N} \phi(\mathbf{w}_l, \mathbf{x}_i)^2,\ \ l = 1, \ldots, L. \tag{9}$$

D_{ll} can be considered to be a measure of the importance of $\mathbf{w}_l$ in the network learning process. For instance, let us consider one of the most frequently used activation functions in feedforward neural networks, i.e. the Radial Basis Function (RBF). Since the network hidden layer weights are randomly chosen, some of the weights may be chosen to be far from most of (or all) the training samples, leading to ELM dimensions that do not contribute to the discrimination of the various classes forming the classification problem at hand. This is due to that the responses of the corresponding hidden layer neurons for all the training samples will be similar (close to zero). On the other hand, appropriately chosen hidden

Table 1: Facial image datasets information.

Dataset	Dimensionality (D)	# of classes (C)	# of data	# of training data (N)	# of test data
ORL	1200	40	400	320	80
Yale	1200	38	2432	1945	487
AR	1200	100	2600	2080	520
PubFig+LFW	1536	200	47189	35469	11720
YTFaces	1770	340	370319	259223	111096

(a)
(b)
(c)

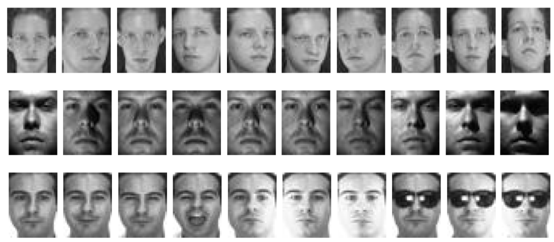

Figure 1: *Facial images depicting persons of the a) ORL, b) Extended YALE-B and c) AR datasets.*

layer weights will lead to ELM space dimensions having varying values for different training samples, thus, highlighting geometric properties of the ELM space. The adoption of the regularization term in (8) will result to a lower regularization factor for the ELM space dimensions corresponding to the hidden layer weights of the first category, when compared to ELM space dimensions corresponding to the latter ones.

5 Experiments

In this section, we present experiments conducted in order to test the proposed ELM network training method in facial image classification problems. Information concerning these datasets is illustrated in Table 1. All experiments have been conducted on a 4-core, i7–3630, 2.4GHz PC with 8GB RAM using single floating point precision and a MATLAB implementation.

The ORL dataset [17] consists of 400 facial images depicting 40 persons (10 images each). The images were captured at different times and with different conditions, in terms of lighting, facial expressions (smiling/not smiling) and facial details (open/closed eyes, with/without glasses). Facial images were taken in frontal position with a tolerance for face rotation and tilting up to 20 degrees. The Extended YALE-B dataset [12] consists of facial images depicting 38 persons in 9 poses, under 64 illumination conditions. In our experiments we have used the frontal cropped images provided by the database. The AR dataset [13] consists of over 4000 facial images depicting 70 male and 56 female faces. In our experiments we have used the preprocessed (cropped) facial images provided by the database, depicting 100 persons (50 males and 50 females) having a frontal facial pose, performing several expressions (anger, smiling and screaming), in different illumination conditions (left and/or right light) and with some occlusions (sun glasses and scarf). Each person was recorded in two sessions, separated by two weeks. Example images of these datasets are illustrated in Figure 1.

The PubFig+LFW dataset [14] has been created by combining the Public Figures (PubFig) [11] and the Labeled Faces in the Wild (LFW) [6] datasets, in order to mimic a web-scale face recognition scenario of finding specific celebrities while ignoring all other faces. It consists of five datasets depicting

(a) (b)

Figure 2: *Facial images depicting persons of the a) PubFig+LFW and b) YouTube datasets.*

Table 2: Classification performance (%) (mean and standard deviation).

Dataset	ELM	RELM	Proposed	SVM
ORL	98.7 $\pm$0.41%	98.75 $\pm$0.41%	**98.8$\pm$0.27**%	98.55 $\pm$0.84%
Yale	99.48 $\pm$0.01%	99.48 $\pm$0.01%	**99.5$\pm$0.01**%	98.98 $\pm$0.01%
AR	97.53 $\pm$0.13%	97.69 $\pm$0.34%	**98.2$\pm$0.15**%	97.72 $\pm$0.01%
PubFig+LFW	58.01 $\pm$0.33%	60.89 $\pm$0.23%	**65.05$\pm$0.12**%	-
YTFaces	76.83 $\pm$0.12%	78.1 $\pm$0.12%	**86.82$\pm$0.13**%	-

200 persons of PubFig with a random $75\%/25\%$ train/test split. All faces of the LFW dataset (except from the facial images depicting the 138 overlapping person ID classes) were added as distractors, thus, converting the closed-universe face recognition problem of the PubFig dataset to an open-universe one. The Youtube Faces dataset [20] consists of 621126 facial images depicting 1595 persons. All images have been downloaded from YouTube. In our experiments we have employed the facial images depicting persons in at least 500 images, resulting to a dataset of 370319 images and 340 classes. Example images of these datasets are illustrated in Figure 2.

In our first set of experiments we have applied the ELM, RELM and the proposed regularized ELM algorithms on the ORL, Yale and AR datasets by using the image intensity values to represent the facial images. We have used the values $L = 2000$, $c = 10^{\{-5:5\}}$ and $p = \{0.2, \dots, 0.8\}$. We performed the five-fold cross-validation procedure, by taking into account the person ID labels. That is, we randomly split the facial images depicting the same person in five sets and we use four sets of all the persons to train the algorithms and measure their performance on the remaining set. This process is performed five times in order to complete an experiment, one for each evaluation set. We applied five experiments and measured the performance of each algorithm by using the mean classification rate over all experiments and the corresponding standard deviation. We also tested the performance of the kernel Support Vector Machine (SVM) classifier. The performance and the mean training time of each algorithm are illustrated in Tables 2 and 3, respectively. All the four algorithms achieve satisfactory (near perfect) performance in these three (closed-universe) face recognition problems. The training time of all the ELM variants is approximately the same for all three datasets, since the heaviest computational step of these experiments is the inversion of the $L \times L$ matrix, and is the same for all the methods.

In our second set of experiments we have applied the competing algorithms to the two large-scale (and open-universe) face recognition problems of the PubFig+LFW and YouTube Faces datasets. For the determination of the network hidden layer weights, we applied K-Means clustering on a subset of $40k$ randomly chosen training data. We have experimentally found that this additional processing step enhances performance. On the PubFig+LFW dataset we have applied the five-fold cross-validation procedure by using the standard training/test partitions provided by the database and using the 1536-dimensional facial image representations suggested by [14]. On the YouTube Faces dataset we ran-

Table 3: Training times (in seconds).

Dataset	ELM	RELM	Proposed	SVM
ORL	2.71	2.98	3.06	2.39
Yale	2.87	3.06	3.17	7.38
AR	3.01	3.21	3.46	8.68
PubFig+LFW	43.62	45.52	45.65	-
YTFaces	62.96	64.2	65.52	-

domly keep 70% of the facial images depicting each person in order to form our training set and test each algorithm on the remaining 30% of the database. The facial images were represented by using the 1770-dimensional representations suggested by [20]. We applied five experiments and measured the performance of each algorithm by calculating the mean classification rate and the corresponding standard deviation over all experiments. The performance and the mean training time of each algorithm are illustrated in Tables 2 and 3, respectively. In these two datasets we omit reporting the performance of the standard kernel SVM classifier, due to its higher computational cost. The training times observed for these two datasets is mainly caused from the K-Means clustering preprocessing step applied for the initialization of the network hidden layer weights. Compared to the closed-universe face recognition problems, the performance obtained on these two (open-universe) face recognition problems is lower. This drop is smaller on the YouTube Faces dataset probably due to the fact that this face recognition problem is more restricting, when compared to the completely unrestricted face recognition problem of the PubFig+LFW dataset. In terms of classification algorithms comparison, the proposed regularized ELM network achieves better performance when compared to the two remaining ELM variants. This difference in performance is higher in the unconstrained face recognition problems.

6 Conclusions

In this paper we proposed a new regularization scheme for Extreme Learning Machine-based Single-hidden Layer Feedforward Neural network training. We have shown that the standard regularization approach minimizing the (Frobenius) norm of the network output weights corresponds to a uniform regularization of the ELM space dimensions and that the proposed regularization approach is able to appropriately weight the dimensions of the ELM space. We have tested the performance of the proposed approach in small-, medium- and large-scale face recognition problems in both closed- and open-universe settings, where it achieves satisfactory performance and outperforms relating techniques, without requiring additional costs in both training and test phases.

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] D.S. Broomhead and D. Lowe. Multivariable functional interpolation and adaptive networks. *Complex Systems*, 2:321–355, 1988.

[2] C.L.P. Chen. A rapid supervised learning neural network for function interpolation and approximation. *IEEE Transactions on Neural Networks*, 7(5):1220–1230, 1996.

[3] M. Chen, K. Weinberger, F. Sha, and Y. Bengio. Marginalized denoising autoencoders for nonlinear representation. *International Conference on Machine Learning*, 2014.

[4] G.E. Hinton, N. Srivastava, A. Krizhevsky, I. Sutsekever, and R.R. Salakhutdinov. Improving neural networks by preventing co-adaptation of feature detectors. *arXiv:1207.0580 [cs.NE]*, 2012.

[5] 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*, 17(4):879–892, 2006.

[6] G.B. Huang, M. Ramesh, T. Berg, and E. Learned-Miller. Labeled faces in the wild: A database for studying face recognition in unconstrained environments. *Technical report, University of Massachusetts*, 2007.

[7] 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*, 42(2):513–529, 2012.

[8] 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*, 2:985–990, 2004.

[9] A. Iosifidis, A. Tefas, and I. Pitas. Regularized Extreme Learning Machine for multi-view semi-supervised Action Recognition,. *Neurocomputing*, 145:250–262, 2014.

[10] R. Jafri and H.R. Arabnia. A survey of face recognition techniques. *Journal of Information Processing Systems*, 5(6):41–68, 2009.

[11] N. Kumar, A. Berg, P. Belhumeur, and S. Nayar. Acquiriing linear subspaces for face recognition under variable lighting. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 33(10):1962–1977, 2011.

[12] K.C. Lee, J. Ho, and D. Kriegman. Acquiriing linear subspaces for face recognition under variable lighting. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 27(5):684–698, 2005.

[13] A. Martinez and A. Kak. PCA versus LDA. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 23(2):228–233, 2001.

[14] E.G. Ortiza and B.C. Beckerb. Face recognition for web-scale datasets. *Computer Vision and Image Understanding*, 118:153–170, 2014.

[15] J.S. Prakash and R. Rajesh. Extreme Learning Machines - a review and state-of-the-art. *International Journal of Wisdom based Computing*, 1(1):35–49, 2011.

[16] D.E. Rumelhart, G.E. Hinton, and R.J. Williams. Learning representations by back-propagating errors. *Nature*, 323(6088):533–536, 1986.

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

[18] W.F. Schmidt, M.A. Kraaijveld, and R.P.W. Duin. Feedforward neural networks with random weights. *International Conference on Pattern Recognition*, 1992.

[19] P. Vincent, H. Larochelle, Y. Bengio, and P.A. Manzagol. Extracting and composing robust features with denoising autoencoders. *International Conference on Machine Learning*, 2008.

[20] L. Wolf, T. Hassner, and I. Maoz. Face recognition in unconstrained videos with matched background similarity. *Computer Vision and Pattern Recognition*, 2011.

[21] Pao. Y.H., G.H. Park, and D.J. Sobajic. Learning and generalization characteristics of random vector functional-link net. *Neurocomputing*, 6:163–180, 1994.

[22] R. Zhang, Y. Lan, G.B. Huang, and Z.B. Zu. Universal approximation of Extreme Learning Machine with adaptive growth of hidden nodes. *IEEE Transactions on Neural Networks and Learning Systems*, 23(2):365–371, 2012.

[23] O. Zoidi, A. Tefas, and I. Pitas. Visual object tracking based on local steering kernels and color histograms. *IEEE Transactions on Circuits and Systems for Video Technology*, 23(5):870–882, 2013.