Mygdalis, V., Iosifidis, A., Tefas, A., & Pitas, I. (2016). One Class Classification Applied in Facial Image Analysis. In Image Processing (ICIP), 2016 IEEE International Conference on 25-28 Sept. 2016. Institute of Electrical and Electronics Engineers (IEEE). DOI: 10.1109/ICIP.2016.7532637

ONE CLASS CLASSIFICATION APPLIED IN FACIAL IMAGE ANALYSIS

Vasileios Mygdalis[⋆] *Alexandros Iosifidis*[⋆†] *Anastasios Tefas*[⋆] *Ioannis Pitas*[⋆‡]

[⋆] Department of Informatics, Aristotle University of Thessaloniki, Thessaloniki, Greece
[†]Department of Signal Processing, Tampere University of Technology, Tampere, Finland
[‡]Department of Electical and Electronic Engineering, University of Bristol, UK

ABSTRACT

In this paper, we apply One-Class Classification methods in facial image analysis problems. We consider the cases where the available training data information originates from one class, or one of the available classes is of high importance. We propose a novel extension of the One-Class Extreme Learning Machines algorithm aiming at minimizing both the training error and the data dispersion and consider solutions that generate decision functions in the ELM space, as well as in ELM spaces of arbitrary dimensionality. We evaluate the performance in publicly available datasets. The proposed method compares favourably to other state-of-the-art choices.

Index Terms— Facial image analysis, one-class classification, regularization

1. INTRODUCTION

Face recognition is one of the most widely studied classification problems in the image analysis. A typical face recognition framework consists of four processing steps, i.e., face detection, feature extraction, dimensionality reduction and classification. In the classification step, face recognition is commonly addressed as a multi-class classification problem. In essence, a classifier is trained to recognize a set of individuals, given a training set consisting of vectorial facial image representations and labels. However, application scenarios of multi-class classification methods have limitations related to model expansion and class-specific classification. Expanding a a pre-trained multi-class classification model in order to include an additional class, essentially requires retraining of the entire model. As the model grows including more and more classes, the computational complexities of re-training, storing and testing/using the model increase as well. Finally, the multi-class classification model does not take class importance into consideration. For example, in movie post-production applications, recognizing the lead actor correctly all the time, might be more important than recognizing actors having a peripheral role. In order to overcome the above described limitations, we consider One-Class Classification (OCC) methods. The motivation of exploiting one-class classifiers for face detection is two-fold. First, by exploiting one classifier per ID class has the advantage that both face recognition and face verification can be considered. In our movie post-production example, one can either try to recognize the actor depicted in a video segment, or can determine if the person depicted in the segment is the lead actor or not. Second, the enrichment of the problem with new classes does not necessarily involve training the entire classification model from scratch, but new one-class models can be added to the existing model in order to include the new classes.

Perhaps the most widely adopted OCC method is the One-Class Support Vector Machines (OC-SVM) [1], discriminates the target class from the origin with a hyperplane. Another approach is the Support Vector Data Description (SVDD) [2], which generates a hypersphere that encloses the target class. Both OC-SVM and SVDD work in both input space and feature spaces of arbitrary dimensionality, by employing data mappings inherently obtained by using a kernel function, e.g., the Radial Basis Function (RBF) kernel. When the RBF kernel is employed, it has been found that both OC-SVM and SVDD provide equivalent solutions [2]. Moreover, methods based on the Kernel Principal Component Analysis (KPCA) have been proposed in [3, 4]. A proximity measure is calculated based on the reconstruction error in the kernel space [3], or by employing a mapping to the null space of the class scatter [4]. Finally, a single-hidden layer neural network-based method trained by using a variant of Extreme Learning Machines has been recently proposed in [5], namely the One Class Extreme Learning Machines (OC-ELM), having comparable performance to other state-of-the-art OCC methods. Applications of OCC methods have been found in failure detection in industrial systems [6], biomedical classification tasks [7], video surveillance/summarization [8, 9] and hyperspectral image classification [10]. Depending on the application type, the terms anomaly/novelty detection have been used to describe OCC problems [3, 4, 6, 11]. Reviews of OCC approaches and applications can be found in [6, 11, 12]. Recently, interest in OCC methods have also been found in visual data classification tasks [13].

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).

In this paper, we are interested in the application of one-class classification in face recognition. We consider the cases where the available training data information originates from only one class, or an important class is present in the training set. We would like to create a classification model that will be able to recognize whether any test sample belongs to this class or not. Moreover, we propose a novel extension of the OC-ELM algorithm, namely the Minimum Variance One-Class Extreme Learning Machine classifier. This classifier solves a modified optimization problem, which emphasizes in minimizing the training error and considers the variance of the class data at the same time. We show that the proposed solution has the effect of regularization, which forces the network output weights to emphasize in low-variance directions. We evaluate the performance of the proposed method in publicly available face recognition datasets.

The rest of the paper is structured as follows. In Section 2, we provide an overview of related one-class classification methods. In Section 3, we describe in detail the proposed classifier. The conducted experiments are described in Section 4. Finally, conclusions are drawn in Section 5.

2. ONE CLASS EXTREME LEARNING MACHINES

Let a set of D-dimensional vectors $\boldsymbol{x}_i \in \Re^D, i = 1, \ldots, N$ be the training set, formed by N training samples of the target class. We employ them in order to train an One-Class classifier. We consider employing the recently proposed OC-ELM algorithm [5], which is a variant of the multi-class ELM algorithm. The multi-class ELM algorithm [14] is a fast algorithm that can be employed to train a Single-hidden Layer Feed-forward Neural network (SLFN) consisting of D input, L hidden and C output neurons, where C is the number of classes forming the multi-class classification problem. In the ELM algorithm, the network input weights $\boldsymbol{W}_{in}$ and the network hidden layer bias $\boldsymbol{b}$ are randomly assigned, while the network output weights $\boldsymbol{W}_{out}$ are analytically calculated. In the OCC case, the network output layer consists of a single neuron ($C = 1$), thus the network output weight is a vector $\boldsymbol{w} \in \Re^L$. Given an activation function $\Phi(\cdot)$ for the network hidden layer and using a linear activation function for the network output layer, the response o_i of the neural network corresponding to an input vector $\mathbf{x}_i$ is calculated by:

$$o = \sum_{j=1}^{L} w_j \, \Phi(\mathbf{v}_j, b_j, \mathbf{x}_i), \; i = 1, ..., N, \tag{1}$$

where $\mathbf{v}_j$ is the j-th column of $\boldsymbol{W}_{in}$ and w_j is the j-th element of $\boldsymbol{w}$. It has been shown that almost any non-linear piecewise continuous activation function $\Phi(\cdot)$ can be used for the calculation of the network hidden layer outputs, e.g., the sigmoid, polynomial, Radial Basis Function (RBF), RBF-χ^2, Fourier series, etc [15, 16, 17, 18].

The network output weight vector $\boldsymbol{w}$ is calculated by solving the following optimization problem:

$$\text{Minimize} \quad \frac{1}{2}\|\boldsymbol{w}\|_2^2 + \frac{c}{2}\sum_{i=1}^{N}\xi_i^2, \tag{2}$$

$$\text{Subject to} \quad \boldsymbol{w}^T\boldsymbol{\phi}_i = 1 - \xi_i, \quad i = 1, ..., N, \tag{3}$$

where $\boldsymbol{\phi}_i \in \Re^L$ are the hidden layer outputs corresponding to each training sample $\boldsymbol{x}_i$ and ξ_i are the slack variables. By observing the optimization problem of OC-ELM (2), it can be seen that in the special case where a kernel function is employed, instead of an activation function mapping the input vectors to the network's hidden layer, the solution of OC-ELM is equivalent to the unbiased version of the Least Squares One Class Support Vector Machines and Kernel Ridge Regression [19]. The output weight vector $\boldsymbol{w}$ of OC-ELM is given by:

$$\boldsymbol{w} = \left(\boldsymbol{\Phi}\boldsymbol{\Phi}^T + \frac{1}{c}\boldsymbol{I}\right)^{-1}\boldsymbol{\Phi}\mathbf{1}, \tag{4}$$

where $\mathbf{1} \in \Re^N$ is a vector of ones. When a non-linear activation function is employed, the solution can also be found using:

$$\boldsymbol{w} = \boldsymbol{\Phi}\left(\boldsymbol{K} + \frac{1}{c}\boldsymbol{I}\right)^{-1}\mathbf{1}, \tag{5}$$

where $\boldsymbol{K}$ is the so-called ELM kernel matrix, expressing data similarity between the training data representations in the ELM space such that $k_{ij} = \kappa(\boldsymbol{\phi}_i \cdot \boldsymbol{\phi}_j), i, j = 1, \ldots, N$. After the calculation of the network output weight $\boldsymbol{w}$, the network response for a given test datum $\boldsymbol{x}_t \in \Re^D$ is given by:

$$o = \boldsymbol{w}^T\boldsymbol{\phi}_t \tag{6}$$

and $\boldsymbol{x}_t$ is classified to the target class if it satisfies the following proximity measure:

$$(o - 1)^2 \leq \epsilon, \tag{7}$$

where $\epsilon \geq 0$ is a threshold that can be determined by using the network responses for the training data multiplied by a small number (i.e., a value of $\epsilon = 0.05 \times \bar{o}_t$ was used in all our experiments, where $\bar{o}_t$ are the mean network responses for the training data).

3. MINIMUM VARIANCE ONE CLASS EXTREME LEARNING MACHINES

In this Section, we describe in detail the proposed Minimum Variance One-Class Extreme Learning Machines (MV-OC-ELM) algorithm. The variance of the network output is given

by:

$$\begin{aligned}\boldsymbol{S}_w &= \frac{1}{N}\sum_{i=1}^{N}\left(o_i-\bar{o}\right)\left(o_i-\bar{o}\right)^T = \\ &= \frac{1}{N}\sum_{i=1}^{N}\left(\boldsymbol{w}^T\boldsymbol{\phi}_i-\boldsymbol{w}^T\bar{\boldsymbol{\phi}}\right)\left(\boldsymbol{w}^T\boldsymbol{\phi}_i-\boldsymbol{w}^T\bar{\boldsymbol{\phi}}\right)^T = \\ &= \boldsymbol{w}^T\left(\frac{1}{N}\sum_{i=1}^{N}\left(\boldsymbol{\phi}_i-\bar{\boldsymbol{\phi}}\right)\left(\boldsymbol{\phi}_i-\bar{\boldsymbol{\phi}}\right)^T\right)\boldsymbol{w} = \boldsymbol{w}^T\boldsymbol{S}\boldsymbol{w},\end{aligned} \tag{8}$$

where $\bar{o}=\frac{1}{N}\sum_{i=1}^{N}o_i$ is the mean output of the network for all training samples and $\bar{\boldsymbol{\phi}}=\frac{1}{N}\sum_{i=1}^{N}\boldsymbol{\phi}_i$ is the mean vector of the class in the feature space determined by the hidden layer outputs of the network. $\boldsymbol{S}$ is the scatter matrix of the training class. Thus, in order to both minimize data dispersion and minimize training error at the same time for one-class classification, we propose the following optimization problem:

$$minimize \quad \frac{1}{2}\boldsymbol{w}^T\boldsymbol{S}\boldsymbol{w}+\frac{c}{2}\sum_{i=1}^{N}\xi_i, \tag{9}$$

$$subject\ to \quad \boldsymbol{w}^T\boldsymbol{\phi}_i\leq 1-\xi_i. \tag{10}$$

The solution for the above described optimization problem can be found by finding the saddle points of the Lagrangian:

$$\mathcal{L}=\frac{1}{2}\boldsymbol{w}^T\boldsymbol{S}\boldsymbol{w}+\frac{c}{2}\sum_{i=1}^{N}\xi_i^2-\sum_{i=1}^{N}\alpha_i(\boldsymbol{w}^T\boldsymbol{\phi}_i-1+\xi_i) \tag{11}$$

where $\alpha_i,\ i=1,\ldots,N$ are the Lagrange multipliers corresponding to the constraints (10). By determining the saddle points of $\mathcal{L}$ with respect to $\boldsymbol{w}$, ξ and α_i, we obtain:

$$\frac{\vartheta\mathcal{L}}{\vartheta\boldsymbol{w}}=0\Rightarrow\boldsymbol{S}\boldsymbol{w}=\boldsymbol{\Phi}\boldsymbol{\alpha}, \tag{12}$$

$$\frac{\vartheta\mathcal{L}}{\vartheta\xi_i}=0\Rightarrow\boldsymbol{\xi}=\frac{1}{c}\boldsymbol{\alpha}, \tag{13}$$

$$\frac{\vartheta\mathcal{L}}{\vartheta\alpha_i}=0\Rightarrow\boldsymbol{\Phi}^T\boldsymbol{w}=\mathbf{1}-\boldsymbol{\xi}. \tag{14}$$

From (12), the network output weight vector is given by:

$$\boldsymbol{w}=\boldsymbol{S}^{-1}\boldsymbol{\Phi}\boldsymbol{\alpha}. \tag{15}$$

In the case where $L>N$, the matrix $\boldsymbol{S}$ will be singular. In order to avoid singularity issues, we adopt a regularized version of $\boldsymbol{S}$, such that:

$$\tilde{\boldsymbol{S}}=\boldsymbol{S}+r\boldsymbol{I}, \tag{16}$$

where $r>0$ is a regularization parameter and $\boldsymbol{I}$ the identity matrix of appropriate dimensions. In order to obtain the solution of the proposed OCC method, we substitute (13) and (14) in (15):

$$\begin{aligned}\boldsymbol{w} &= \left(\boldsymbol{\Phi}\boldsymbol{\Phi}^T+\frac{1}{c}\tilde{\boldsymbol{S}}\right)^{-1}\boldsymbol{\Phi}\mathbf{1} \\ &= \left(\boldsymbol{\Phi}\boldsymbol{\Phi}^T+\frac{1}{c}\boldsymbol{S}+\frac{r}{c}\boldsymbol{I}\right)^{-1}\boldsymbol{\Phi}\mathbf{1}.\end{aligned} \tag{17}$$

After the calculation of the network output weight $\boldsymbol{w}$, the network response for a test vector $\boldsymbol{x}_t\in\Re^D$ is given by (6) and $\boldsymbol{x}_t$ is classified to the target class by employing the proximity measure in (7). Here we should note that in order to include subclass information in the scatter matrix as in [18], a modified version of $\boldsymbol{S}$ can be employed:

$$\tilde{\boldsymbol{S}}_w=\sum_{i=1}^{N}\sum_{k=1}^{K}\frac{N_k}{N}\gamma_i^k\left(\boldsymbol{\phi}_i-\bar{\boldsymbol{\phi}}\right)\left(\boldsymbol{\phi}_i-\bar{\boldsymbol{\phi}}\right)^T, \tag{18}$$

where γ_i^k is an index denoting if the training sample $\boldsymbol{x}_i$ belongs to the subclass k and N_k is the subclass cardinality. In fact, we employed this version of $\boldsymbol{S}_w$ in all our experiments and determined subclasses by applying the k-means algorithm.

In order to extend the proposed method in order to exploit ELM spaces of arbitrary dimensionality [20], we work as follows. First, we describe the network output weight as a linear combination of the training data representations in the ELM space and a reconstruction vector $\boldsymbol{\beta}\in\Re^N$, by exploiting the Representer Theorem [21], such that:

$$\boldsymbol{w}=\boldsymbol{\Phi}\boldsymbol{\beta}. \tag{19}$$

We also decompose the matrix $\boldsymbol{S}$ as follows:

$$\begin{aligned}\boldsymbol{S} &= \frac{1}{N}\sum_{i=1}^{N}\left(\boldsymbol{\phi}_i-\bar{\boldsymbol{\phi}}\right)\left(\boldsymbol{\phi}_i-\bar{\boldsymbol{\phi}}\right)^T = \\ &= \frac{1}{N}\boldsymbol{\Phi}(\boldsymbol{I}-\frac{1}{N}\boldsymbol{e}\boldsymbol{e}^T)\boldsymbol{\Phi}^T=\boldsymbol{\Phi}\boldsymbol{M}\boldsymbol{\Phi}^T,\end{aligned} \tag{20}$$

where $\boldsymbol{e}$ is a N-dimensional vector of ones and $\boldsymbol{I}$ is a $N\times N$ identity matrix. $\tilde{\boldsymbol{S}}_w$ in (18) can be decomposed in a similar manner. Next, we substitute (19) and (20) in (12) and we obtain:

$$\boldsymbol{a}=\left(\boldsymbol{M}\boldsymbol{\Phi}^T\boldsymbol{\Phi}+r\boldsymbol{I}\right)\boldsymbol{\beta}=(\boldsymbol{M}\boldsymbol{K}+r\boldsymbol{I})\boldsymbol{\beta}, \tag{21}$$

where $\boldsymbol{K}\in\Re^{N\times N}$ is the ELM kernel matrix expressing data similarities between the training samples and $\boldsymbol{I}$ is an identity matrix. Finally, by substituting (21) in (14) we obtain:

$$\boldsymbol{\beta}=\left(\boldsymbol{K}+\frac{1}{c}\boldsymbol{M}\boldsymbol{K}+\frac{r}{c}\boldsymbol{I}\right)^{-1}\mathbf{1}. \tag{22}$$

Finally, the network response o for a test sample $\boldsymbol{x}_t$ is given by:

$$o=\boldsymbol{w}^T\boldsymbol{\phi}_t=\boldsymbol{\beta}^T\boldsymbol{k}_t, \tag{23}$$

where $\boldsymbol{k}_t\in\Re^N$ is a vector containing the similarities of $\boldsymbol{x}_t$ with the training samples. The classification decision for $\boldsymbol{x}_t$ is given through (7).

Table 1. Experimental Results in PubFig83 + LFW Dataset

Algorithm	Hidden layers (L)	Average g-mean rate
OC-ELM [5]	500	38.34
MV-OC-ELM	500	**39.43**
OC-ELM [5]	1000	41.12
MV-OC-ELM	1000	**42.16**
OC-ELM [5]	Infinite (RBF)	52.56
MV-OC-ELM	Infinite (RBF)	**63.56**

Table 2. Experimental Results in Standard Face Recognition Datasets

Algorithm	AR	ORL	YALE
method [1]	74.30	54.04	69.10
method [3]	74.01	94.55	77.22
method [4]	83.84	95.23	82.26
OC-ELM [5]	78.00	95.22	81.19
Proposed	**88.83**	**96.39**	**84.87**

4. EXPERIMENTS

This section presents the experiments conducted in order to evaluate the performance of the proposed OCC method in face recognition. To this end, we have employed publicly available datasets, which have been widely adopted in relevant work in face recognition.

In our first set of experiments, we have employed the proposed MV-OC-ELM algorithm in the PubFig83 + LFW Dataset [22]. We have employed the feature vectors (HOG, LBP, and Gabor wavelet features reduced to 2048 dimensions with PCA), which were extracted from 13,002 facial images representing 83 individuals from PubFig83, divided into 2/3 training (8720 faces) and 1/3 testing set (4,282 faces), as well as 12,066 images representing over 5,000 faces which were used as a distractor set from LFW. For each of the 83 individuals, we have employed the training images for this class and tested on the respective test set of this class, as well as 200 randomly selected images for the distractor set. For each class, we have employed the proposed MV-OC-ELM algorithm along with the OC-ELM algorithm, using the same set of random bias and hidden layer neurons ($L = 500, 1000$). In order to eliminate randomness, we report the performance of a 10-fold cross-validation procedure. Finally, we have employed the kernel version of the proposed MV-OC-ELM algorithm, as well as the OC-ELM algorithm, by employing the RBF activation function. For all cases, we report the average obtained g-mean [23] for the 83 classes, which is a metric that contains both precision and recall measurements as follows :$g_{mean} = \sqrt{\text{precision} \cdot \text{recall}}$. Experimental results are provided in Table 1. As can be seen, the proposed MV-OC-ELM algorithm outperforms the OC-ELM in all cases.

In our second set of experiments, we have employed the proposed MV-OC-ELM algorithm in classic face recognition datasets. For comparison reasons, we have also trained the OC-ELM algorithm, as well as the OC-SVM algorithm [1], the Kernel PCA for novelty detection [3] and Kernel Null Space Methods for Novelty Detection [4]. The employed classic face recognition datasets include the AR [24], ORL [25] and Yale [26] datasets, which contain 2600, 400, and 2432 frontal facial images from 100, 40 and 38 subjects, respectively. As feature vectors, we have employed a $D = 1200$ dimensional vector corresponding to the gray-scale equivalent pixel luminosities of the resized 40×30 images. We have employed the 5-fold cross validation procedure, where we have employed $4/5$ of the dataset for training and $1/5$ for testing purposes. For each class, we have employed the positive training samples and tested on the test set (of each fold). Finally, we report the average obtained performance of all classes. Experimental results are provided in Table 2.

Table 3. Experimental Results in Facial Expression Datasets

Algorithm	BU	KANADE	JAFFE
method [1]	61.27	63.80	56.72
method [3]	50.71	49.97	56.33
method [4]	59.48	62.21	57.31
OC-ELM [5]	57.14	60.03	54.10
Proposed	**63.99**	**70.14**	**66.43**

Finally, we have employed proposed method as well as the above mentioned competing methods in facial expression datasets, namely the BU [27], KANADE [28] and JAFFE [29]. We followed the exact same experimental protocol as in our second set of experiments, in terms of feature vectors and cross validation. Experimental results are depicted in Table 3. In all cases, the proposed MV-OC-ELM outperformed the competition and in some cases, by a large extent.

5. CONCLUSION

In this paper, we described a novel method for one-class classification, which was evaluated in face recognition and facial expression recognition datasets, with favourable comparison to the state-of-the-art. The proposed method is based on the ELM algorithm, by emphasizing the minimization of the dispersion of the training data, during the ELM optimization process. The proposed method works in ELM spaces or known or arbitrary dimensionality, depending on the activation function choice.

6. REFERENCES

[1] B. Scholkopf, J.C. Platt, J. Shawe-Taylor, A. Smola, and R.C. Williamson, "Estimating the support of a high-dimensional distribution," *Neural computation*, vol. 13, no. 7, pp. 1443–1471, 2001.

[2] D.MJ. Tax and R.PW. Duin, "Support vector data de-

scription," *Machine learning*, vol. 54, no. 1, pp. 45–66, 2004.

[3] H. Hoffmann, "Kernel pca for novelty detection," *Pattern Recognition*, vol. 40, no. 3, pp. 863–874, 2007.

[4] P. Bodesheim, A. Freytag, E. Rodner, M. Kemmler, and J. Denzler, "Kernel null space methods for novelty detection," *IEEE CVPR*, 2013.

[5] Q. Leng, H. Qi, J. Miao, W. Zhu, and G. Su, "One-class classification with extreme learning machine," *Mathematical Problems in Engineering*, pp. 1–12, 2014.

[6] M.AF Pimentel, D.A. Clifton, L. Clifton, and L. Tarassenko, "A review of novelty detection," *Signal Processing*, vol. 99, pp. 215–249, 2014.

[7] B. Krawczyk and M. Wozniak, "Handling label noise in microarray classification with one-class classifier ensemble," vol. 311, pp. 351–359, 2015.

[8] M. Markou and S. Singh, "A neural network-based novelty detector for image sequence analysis," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 28, no. 10, pp. 1664–1677, 2006.

[9] V. Mygdalis, A. Iosifidis, A. Tefas, and I. Pitas, "Video summarization based on subclass support vector data description," *IEEE Symposium Series on Computational Intelligence*, 2014.

[10] J. Muñoz-Marí, F. Bovolo, L. Gómez-Chova, L. Bruzzone, and G. Camp-Valls, "Semisupervised one-class support vector machines for classification of remote sensing data," *IEEE Transactions on Geoscience and Remote Sensing*, vol. 48, no. 8, pp. 3188–3197, 2010.

[11] X. Ding, Y. Li, A. Belatreche, and L.P. Maguire, "An experimental evaluation of novelty detection methods," *Neurocomputing*, vol. 135, pp. 313–327, 2014.

[12] M. Markou and Singh.S, "Novelty detection: a review-part 1: statistical approaches," *Signal Processing*, vol. 83, no. 12, pp. 2481 – 2497, 2003.

[13] V. Mygdalis, I. Alexandros, A. Tefas, and I. Pitas, "Large-scale classification by an approximate least squares one-class support vector machine ensemble," *IEEE BigDataSE*, vol. 2, pp. 6–10, 2015.

[14] G-B. Huang, Q-Y. Zhu, and C-K. Siew, "Extreme Learning Machine: a new learning scheme of feedforward neural networks," *IEEE IJCNN*, 2004.

[15] 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.

[16] G-B. Huang and L. Chen, "Convex incremental Extreme Learning Machine," *Neurocomputing*, vol. 70, no. 16, pp. 3056–3062, 2007.

[17] 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.

[18] 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.

[19] Y-S. Choi, "Least squares one-class support vector machine," *Pattern Recognition Letters*, vol. 30, no. 13, pp. 1236 – 1240, 2009.

[20] A. Iosifidis, A. Tefas, and I. Pitas, "On the kernel extreme learning machine classifier," *Pattern Recognition Letters*, vol. 54, pp. 11–17, 2015.

[21] B. Schölkopf, R. Herbrich, and A.J. Smola, "A generalized representer theorem," *Computational learning theory*, pp. 416–426, 2001.

[22] B. Becker and E. Ortiz, "Evaluating open-universe face identification on the web," *IEEE CVPR*, 2013.

[23] M. Kubat, R.C. Holte, and S. Matwin, "Machine learning for the detection of oil spills in satellite radar images," *Machine learning*, vol. 30, no. 2-3, pp. 195–215, 1998.

[24] Aleix M Martinez, "The ar face database," *CVC Technical Report*, vol. 24, 1998.

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

[26] A.S. Georghiades, P.N. Belhumeur, and D.J. Kriegman, "From few to many: Illumination cone models for face recognition under variable lighting and pose," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 23, no. 6, pp. 643–660, 2001.

[27] L. Yin, X. Wei, Y. Sun, J. Wang, and M.J. Rosato, "A 3d facial expression database for facial behavior research," *IEEE FG*, 2006.

[28] T. Kanade, J.F. Cohn, and Y. Tian, "Comprehensive database for facial expression analysis," *IEEE FG*, 2000.

[29] M. Lyons, S. Akamatsu, M. Kamachi, and J. Gyoba, "Coding facial expressions with gabor wavelets," *IEEE FG*, 1998.