Face Recognition Independent of Facial Expression Through SOM-based Classifiers

Isaque Q. Monteiro, Samy D. Queiroz, Alex T. Carneiro, Luis G. Souza and Guilherme A. Barreto

***Abstract*— In this paper, we evaluate four pattern classifiers built from the self-organizing map (SOM), a well-known neural clustering algorithm, in the recognition of faces independent of facial expression. The design of two of the classifiers involves post-training procedures for labelling the neurons, i.e. no class information is used prior to the training phase. The other two classifiers incorporate class information prior to the training phase. All the classifiers are evaluated using the well-known Yale face database and their performances compare favorably with standard neural supervised classifiers.**

***Index Terms*— Biometrics, self-organizing map, facial expression, face recognition, pattern classification.**

I. Introduction

Biometrics refers to the identification of an individual based on his/her physiological characteristics, like a fingerprint, face, eye, voice or behavior, like handwriting or keystroke patterns [1]. Because biometric characteristics are unique to each individual, they can be used to prevent theft of fraud. Unlike a password or a PIN, a biometric feature is much harder (if not impossible!) to be lost, stolen, or recreated.

Biometrics has long been an active research field, particularly because of all the attention focused on public and private security systems in recent years [2]. The goal of an automatic identity verification system is to either accept or reject the identity claim made by a given person. Such systems have many important applications, such as access control or transaction authentication. Advances in digital computers, software technologies, and embedded systems have further catalyzed increased interest in commercially available biometric application systems.

These advances in hardware and software technologies have also boosted the development of machine learning techniques, which have been at the forefront of scientific research in biometric authentication [3]. Unlike the conventional template matching approach, in which identification resorts to storing a considerable amount of example patterns of a class, the machine learning approach adopts representative statistical models to capture the characteristics of patterns in the feature domain. Among the several ML-based approaches, artificial neural networks (ANNs) have been successfully applied to several biometric identification tasks, such as text-independent speaker verification, fingerprint identification [4], [5], face localization [6], and face recognition [7].

Of particular interest to this paper is the face recognition problem under different lighting conditions, configurations (e.g. with or without glasses) and different facial expressions, feelings and moods including happy, sad, sleepy, surprised, and wink. This difficult face recognition task has been usually approached from the perspective of powerful supervised neural classifiers, such as the Multilayer Perceptron (MLP) and Radial Basis Functions (RBF) networks [8]–[12]. Despite the promising results obtained from this type of ANN-based classifier, the issue of face recognition independent of facial expressions is in its first infancy and much still remain to be done.

The authors are with the Department of Teleinformatics Engineering, Federal University of Ceará, C.P. 6007, CEP: 60555-760, Fortaleza-CE, Brazil. email: guilherme@deti.ufc.br

The Self-Organizing Map (SOM) [13], [14] is a neural clustering algorithm, which has been used as a viable alternative to MLP and RBF networks in a variety of pattern classification problems (see [15] for a review and references therein). For being an unsupervised technique, however, the SOM is mostly applied to image processing processing problems as a vector-quantization based compression algorithm [16]–[19]. Thus, it is usually used as a auxiliary tool for helping conventional supervised classifiers to deal with face recognition [20]. As such, very few applications of the SOM as a stand-alone face recognizer has been reported in the literature [21].

From the exposed above, the very goal of this paper is to evaluate the performance of SOM-based classifiers as stand-alone face recognizers. For that purpose, we evaluate four SOM-based pattern classifiers in the recognition of faces independent of facial expression. The design of two of the classifiers involves post-training procedures for labelling the neurons, i.e. no class information is used prior to the training phase. The other two classifiers incorporate class information prior to the training phase. All the classifiers are evaluated using the well-known Yale face database and their performances are discussed in the light of previous studies that used standard MLP- and RBF-based classifiers. To the best of our knowledge, this study is the first one to evaluate the SOM as a classifier *per se* in this type of face recognition task.

The remainder of the paper is organized as follows. In Section II we briefly describe the SOM algorithm and its main computational properties. In Section III we present four approaches to the problem of designing a supervised classifier based on the SOM. Simulations with the Yale face database and corresponding results are presented in Section IV. The paper is concluded in Section V.

II. The Self-Organizing Map

Competitive learning models constitute one of the main classes of unsupervised neural networks [22], [23]. This type of learning algorithm is based on the concept of *winning neuron*, defined as the one whose weight vector is the closest to the current input vector. Different metrics for measuring proximity between two vectors can be used, such as Euclidean distance or Inner-product. During the learning phase, the weight vectors of the winning neurons are modified incrementally in time in order to extract *average statistical features* from the set of input patterns.

Competitive learning rules allow a single-layer network to group and represent data samples that lie in a neighborhood of the input space. Each neighborhood is represented by a single prototype. This operation is commonly called *clustering* in pattern recognition [24], [25]. From the point of view of the input space, clustering is dividing the space in local regions, each of which is associated with an output neuron. If we join prototype vectors by a line, its perpendicular bisector will meet other bisectors forming a division that is called a Voronoi tessellation, or simply a tessellation. Data samples that fall inside the regions are assigned to the corresponding prototype vector. Clustering is also a form of nonparametric density estimation. In the absence of a desired response, the best we can do for categorization is to use the information about the input data distribution to separate inputs into groups that share the same region in data space. The basic idea of clustering is to seek regions of high sample density - data clusters - and represent their centers in the network.

Using Euclidean distance, one of the simplest strategy to find the winning neuron, $i^*(t)$, is given by the following expression:

$$i^*(t) = \arg\min_{\forall i} \|\mathbf{x}(t) - \mathbf{w}_i(t)\| \tag{1}$$

where $\mathbf{x}(t) \in \mathbb{R}^p$ denotes the current input vector, $\mathbf{w}_i(t) \in \mathbb{R}^p$ is the weight vector of neuron i, and t symbolizes the time steps associated with the iterations of the algorithm.

The *Self-Organizing Map* (SOM) is a well-known competitive learning algorithm. The SOM learns from examples a mapping (projection) from a high-dimensional continuous input space $\mathcal{X}$ onto a low-dimensional discrete space (lattice) $\mathcal{A}$ of N neurons which are arranged in fixed topological forms, e.g., as a rectangular 2-dimensional array. The map $i^*(\mathbf{x}) : \mathcal{X} \rightarrow \mathcal{A}$, defined by the weight matrix $\mathbf{W} = (\mathbf{w}_1, \mathbf{w}_2, \ldots, \mathbf{w}_q), \mathbf{w}_i \in \mathcal{X}$, assigns to each input vector $\mathbf{x} \in \mathcal{X}$ a neuron $i^* \in \mathcal{A}$. Adjustment of the weight vectors of the winning neuron and of those neurons belonging to its neighborhood:

$$\mathbf{w}_i(t+1) = \mathbf{w}_i(t) + \alpha(t) h(i^*, i; t)[\mathbf{x}(t) - \mathbf{w}_i(t)] \tag{2}$$

where $0 < \alpha(t) < 1$ is the learning rate and $h(i^*, i; t)$ is a weighting function which limits the neighborhood of the winning neuron. A usual choice for $h(i^*, i; t)$ is given by the Gaussian function:

$$h(i^*, i; t) = \exp\left(-\frac{\|\mathbf{r}_i(t) - \mathbf{r}_{i^*}(t)\|^2}{2\sigma^2(t)}\right) \tag{3}$$

where $\mathbf{r}_i(t)$ and $\mathbf{r}_{i^*}(t)$ are respectively, the positions of neurons i and i^* in a predefined output array, and $\sigma(t) > 0$ defines the radius of the neighborhood function at time t.

The variables $\alpha(t)$ and $\sigma(t)$ should both decay with time to guarantee convergence of the weight vectors to stable steady states. In this paper, we adopt for both an exponential decay, given by:

$$\alpha(t) = \alpha_0 \left(\frac{\alpha_T}{\alpha_0}\right)^{(t/T)} \quad \text{and} \quad \sigma(t) = \sigma_0 \left(\frac{\sigma_T}{\sigma_0}\right)^{(t/T)} \tag{4}$$

where α_0 (σ_0) and α_T (σ_T) are the initial and final values of $\alpha(t)$ ($\sigma(t)$), respectively. The operations defined in Eqs. (1) and (4) are repeated until a steady state of global ordering of the weight vectors has been achieved. In this case, we say that the map has converged. In addition to usual clustering properties, the resulting map also preserves the topology of the input samples in the sense that adjacent patterns are mapped into adjacent regions on the map. Due to this topology-preserving property, the SOM is able to cluster input information and spatial relationships of the data on the map. Despite being very simple, the SOM algorithm is powerful and has become one of the most important ANN architectures. It has been applied to a variety of real-world problems [15], [26] and its computational properties are well-understood. Also, there are several theoretical studies on global ordering and convergence of the SOM [27].

III. SOM-based Pattern Classification

Since the SOM is formed through an unsupervised process that builds a topology-preserving representation of the statistical distribution of the data, it is particularly suited for unsupervised pattern recognition tasks, such as clustering or data visualization. However, the SOM architecture is flexible enough to allow users to apply it as well to supervised classification. Several approaches have been proposed throughout the years, and the four most common ones are described next. Since SOM-based classification is performed on the basis of the weight vector of the winning neuron, all resulting classifiers belong to the family of prototype-based classifiers [28].

A. SOM-based Classifier 1 (SOM-C1)

The first approach involves a post-training labelling phase, which should be performed for all neurons in the SOM before the testing phase [28]–[31]. No weight modifications are allowed and the original training data vectors should be used. Firstly, for each training input vector, the corresponding winning neuron should be found according to (1).

The labelling phase itself is carried out once the winning neurons of all input vectors have been found. It is worth noting that a given neuron can be selected the winner for input vectors belonging to different classes. However, among all the vectors a given neuron was selected the winner, the number of exemplars of a given class usually is higher that the number of exemplars of other classes. Hence, a label is assigned to a neuron on a majority voting basis, i.e. a neuron receives the label of the class with the highest number of exemplars. Ties can be broken by random selection of the competing labels.

It is also possible for a given neuron not to be selected the winner at all, for any of the input vectors. In this case, the neuron can inherit the label of its neighboring neurons or be tagged with an "unknown" or "rejection" class label. During testing, a new input vector inherits the label of its winning neuron.

B. SOM-based Classifier 2 (SOM-C2)

The second approach is a contribution of this paper. This method is also based on a post-training labelling procedure. However, in this case, we use the class centroids for labelling purpose, not the training data vectors themselves. Since the class labels are known beforehand, the class centroids (one for each available class) are first computed using the training data vectors. After SOM training is completed, a given weight vector is tagged with the class label of its nearest centroid. One advantage of this method over the SOM-C1 is that a neuron is labelled with one and only one class label, thus avoiding the occurrence of ties and unlabeled neurons.

C. SOM-based Classifier 3 (SOM-C3)

In the third approach, the SOM is made supervised by adding class information to each input data vector [13], [32], [33]. Specifically, the input vectors $\mathbf{x}(t)$ are now formed of two parts, $\mathbf{x}_p(t)$ and $\mathbf{x}_l(t)$, where $\mathbf{x}_p(t)$ is the pattern vector itself, while $\mathbf{x}_l(t)$ is the corresponding class label of $\mathbf{x}_p(t)$. During training, these vectors are concatenated to build augmented vectors $\mathbf{x}(t) = [\mathbf{x}_p(t) \;\; \mathbf{x}_l(t)]^T$ which are used as inputs to the SOM. The corresponding augmented weight vectors, $\mathbf{w}_i(t) = [\mathbf{w}_i^p(t) \;\; \mathbf{w}_i^l(t)]^T$, are adjusted as in the usual SOM training procedure.

Usually, the label vector $\mathbf{x}_l(t)$ is represented as a unit-length binary vector; that is, only one of its components is set to "1", while the others are set to "0". The index of the "1" position indicates the class of the pattern vector $\mathbf{x}_p(t)$. For example, if three classes are available, then three label vectors are possible: one for the first class ($[1\ 0\ 0]$), one for the second class ($[0\ 1\ 0]$) and one for the third class ($[0\ 0\ 1]$).

During recognition of an unknown vector $\mathbf{x}(t)$, the $\mathbf{x}_l(t)$ part is not considered, i.e. only its $\mathbf{x}_p$ part is compared with the corresponding part of the weight vectors. However, the class label of the unknown pattern vector is decided on the basis of the $\mathbf{w}_i^l(t)$ part of the winning weight vector $\mathbf{w}_{i^*}(t)$. The index of the component of $\mathbf{w}_{i^*}^l(t)$ with highest value defines the class label of the unknown pattern vector $\mathbf{x}_p$.

D. SOM-based Classifier 4 (SOM-C4)

The fourth approach uses one SOM network for each available class; for instance, if three classes of data are available, three SOMs will be trained, one for each class [34]. The several SOMs, however, are trained independently, using only the data vectors of the class it represents. There is no need for the several SOMs to have the same number of neurons, unless for the sake of simplicity. During testing, the winning neuron is searched among the neurons of all available SOM networks, so that its class label is assigned to the current input vector.

IV. Simulations

The Yale Face Database [35] contains 165 grayscale images in GIF format of 15 individuals. There are 11 images per subject, one per different facial expression or configuration: center-light, with glasses, happy, left-light, without glasses, normal, right-light, sad, sleepy, surprised, and wink. Each image has 243×320 pixels. Pixel intensities are rescaled to the range $[0-1]$. Figure 1 shows a sample of the images used in the simulations.

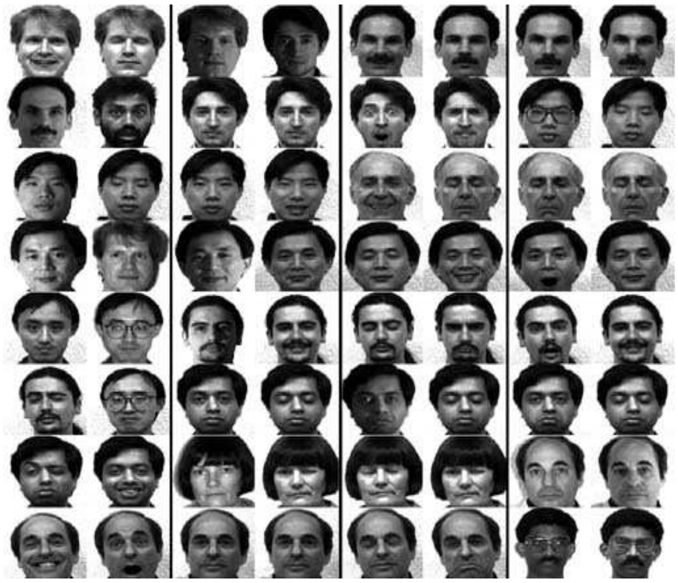
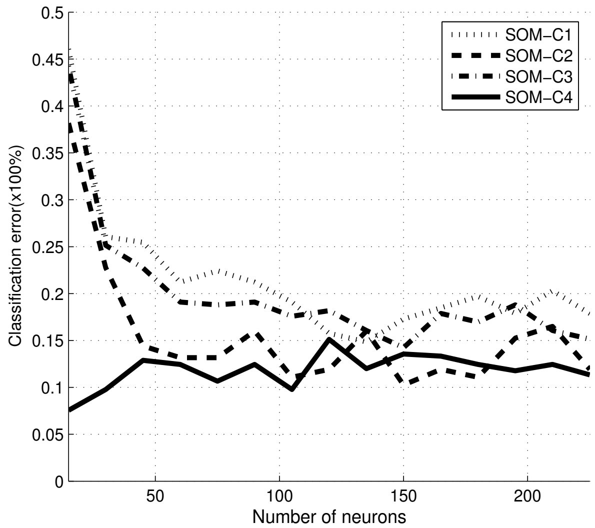


Fig. 1. Sample of images of the Yale face database.

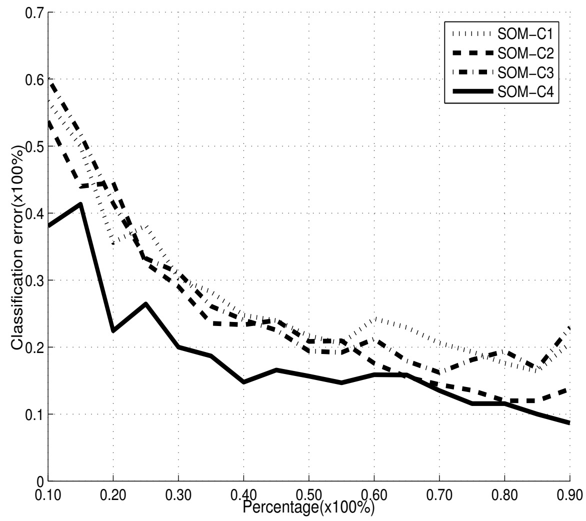
Image preprocessing: Before presentation to the classifiers each image in the dataset is vectorized first, i.e. its columns are rearranged, one beneath the other, into a single column-vector of dimension $p = 77,760$. Principal component analysis (PCA) is then performed over the resulting set of 165 vectors in order to reduce their dimensionality. After some experimentation it was found that the first 26 principal directions (eigenvectors of the data correlation matrix) suffice to explain approximately 90% of the variance of the dataset. Thus, the original set of 165 77,760-dimensional vectors is transformed into a set of 165 26-dimensional vectors. Finally, the 26 components of the transformed data vectors are normalized to zero mean and unity variance.

All the four SOM-based classifiers are evaluated in terms of recognition error rates each one reports during the testing phase. For the sake of completeness, the sensitivity of the error rates of the four classifiers to several design parameters, such as number of neurons, number of training epochs, size of the training set, and number of classes (individuals), are also assessed.

Firstly, we evaluate the sensitivity of classifiers to the number of neurons, q. For each value of q, the classifiers are trained and tested for 50 runs. At each run, the training/testing datasets are resampled, so that 80% of the 165 data vectors are randomly selected to compose the training set, while the remaining 20% are used for testing purposes. The reported error rate, for each q, is then averaged over the 50 training/testing runs. For this simulation, the classifiers are trained for 100 epochs and the number of classes is set to 15, i.e. all the individuals in the database are to be recognized. The learning rate and neighborhood function annealing parameters are set to $\alpha_0 = 0.5$, $\alpha_T = 0.001$, $\sigma_0 = q/2$, $\sigma_T = 0.001$ and



(a)



(b)

Fig. 2. Recognition error rate as a function of the (a) number of neurons, and (b) the size of training set.

$T = 100 \times 132 = 13,200$.

The results are shown in Figure 2(a). One can note that the performances of three of the classifiers (SOM-C1, SOM-C2, SOM-C3) improved as the number of neurons increases, while the performance of the SOM-C4 classifier deteriorates slightly. However, changes in the performances of all classifiers is more intense for $q < 150$. From this value on, their performances always degrade. For $100 < q < 150$, the performances of all classifiers are equivalent. For $50 < q < 100$, the performances of the SOM-C2 and SOM-C4 classifiers are the better ones. Recall $q = 50$ for the SOM-C4 classifier means that it uses indeed $15 \times 50 = 750$ neurons in total, a value much higher than the total number of images available. The value $q = 50$ for the SOM-C2 means that uses only 50 neurons to represent all the images available, providing at the same time information compression and high recognition rates.

The second simulation aim to evaluate the sensitivity of the classifiers to the size of the training set. The training parameters of the previous simulation are maintained, except for the percentage of data vectors in the training set, which varies from 10 to 90% of the available pattern vectors. The percentage of data vectors in the testing set varies accordingly from 90 to 10%. The results are shown in Figure 2(b). As expected, as more information about the individuals are provided during training, the error rates of all classifiers diminish. For a percentage value higher than 60%, the SOM-C2 and SOM-C4 performed much better that other classifiers.

The third simulation evaluates the sensitivity of the classifiers to the number of classes (individuals) to be recognized. The training parameters of the first simulation are maintained, except for the number of classes, which varies from 2 to 15. The error rates varies considerably throughout the range of interest. Only for a high number of classes (i.e. more than 13) the SOM-C2 and SOM-C4 classifiers presented better performances.

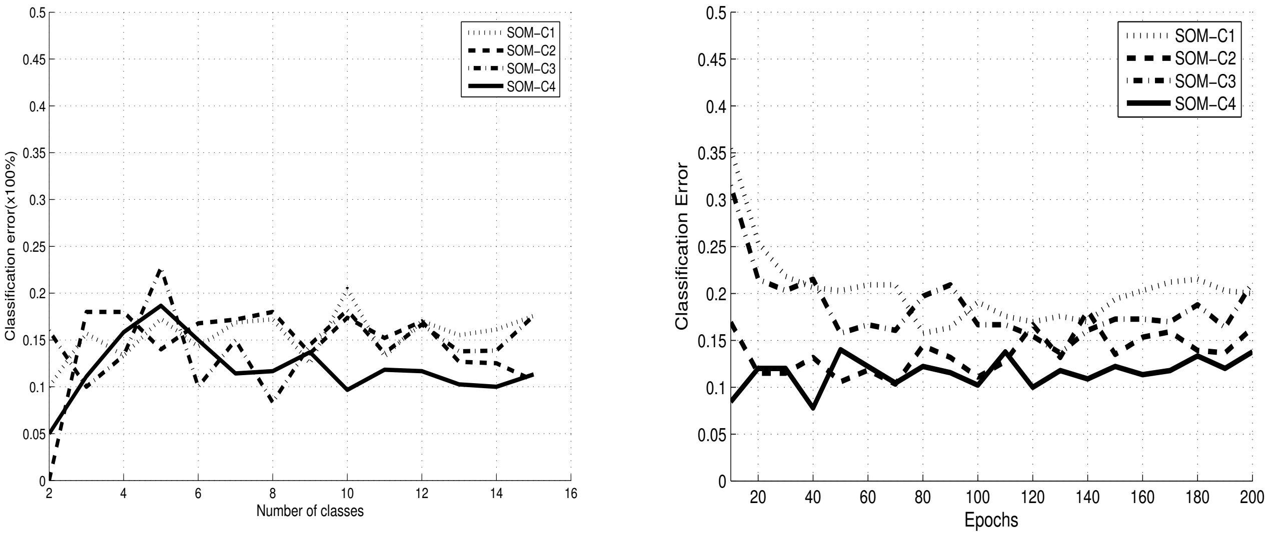
The fourth simulation evaluates the sensitivity of the classifiers to the number of training epochs. Ideally, a classifier should require just a few training epochs in order to provide a good testing performance. The training parameters of the first simulation are maintained, except for the number of training epochs, which varies from 10 to 200 (hence, the values of T changes accordingly). As a general result, the performances of all classifiers deteriorates for values higher than 100 epochs. Below, this value the performances of the SOM-C2 and SOM-C4 classifiers are again the better ones.

Finally, the best results reported for the SOM-based classifiers are shown in Table I. In this table we list the mean, minimum, maximum and average values of the recognition rates. For this simulation, the classifiers are trained for 100 epochs and the number of classes is set to 15. We set $q = 100$ and use 80% of the pattern vectors to compose the training set. The learning rate and neighborhood annealing parameters are the same of first simulation.

As expected the best performances were reported for the SOM-C2 and SOM-C4 classifiers. It is worth mentioning that these classifiers presented performances comparable to those previously reported in the literature for standard neural supervised classifiers engaged in face recognition tasks [8], [10]–[12].

V. Conclusions and Further Work

We evaluated four pattern classifiers built from the self-organizing map (SOM), a well-known neural clustering algorithm, in the recognition of faces independent of facial expression. The design of two of the classifiers involve post-training labelling of neurons, i.e. no class information is



(a) (b)

Fig. 3. Recognition error rate as a function of the (a) number of classes, and (b) the number of the training epochs.

TABLE I
BEST PERFORMANCES OF THE FOUR SOM-BASED CLASSIFIERS.

Neural Models	**Recognition rates (%)**			
	mean	*min*	*max*	*variance*
SOM-C1	82.6	64.0	100.0	0.64
SOM-C2	88.8	66.7	97.0	0.17
SOM-C3	82.0	72.7	90.9	0.32
SOM-C4	87.4	77.8	95.6	0.17

used prior to the training phase. The other two classifiers incorporate class information prior to the training phase.

All the classifiers were evaluated using the well-known Yale face database and their performances, despite being derived from an unsupervised neural algorithm, compared favorably with standard neural supervised classifiers, such MLP and RBF.

ACKNOWLEDGMENT

The authors would like to thank CNPq (grant #506979/2004-00) for the financial support.

REFERENCES

[1] J. D. Woodward Jr., N. M. Orlans, and P. T. Higgins, *Biometrics*, McGraw-Hill, 2002.
[2] P. Reid, *Biometrics for Network Security*, Prentice Hall, 2003.
[3] S. Y. Kung, M. W. Mak, and S. H. Lin, *Biometric Authentication: A Machine Learning Approach*, Prentice Hall, 2004.
[4] K. A. Toh and W. Y. Yau, "Fingerprint and speaker verification decisions fusion using a functional link network," *IEEE Transactions on Systems, Man and Cybernetics*, vol. C-35, no. 3, pp. 357–370, 2005.
[5] N. Yager and A. Amin, "Fingerprint classification: a review," *Pattern Analysis and Applications*, vol. 7, no. 1, pp. 77–93, 2004.
[6] M.-H. Yang, D. J. Kriegman, and N. Ahuja, "Detecting faces in images: A survey," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 24, no. 1, pp. 34–58, 2002.
[7] W. Zhao, R. Chellappa, P. J. Phillips, and A. Rosenfeld, "Face recognition: A literature survey," *ACM Computing Surveys*, vol. 35, no. 4, pp. 399–458, 2003.
[8] L. Ma and K. Khorasani, "Facial expression recognition using constructive feedforward neural networks," *IEEE Transactions on Systems, Man and Cybernetics*, vol. B-34, no. 3, pp. 1588–1595, 2004.
[9] A. Nes, "Hybrid systems for face recognition," M.S. thesis, Department of Computer and Information Science, Faculty of Information Technology, Mathematics and Electrical Engineering, Norwegian University of Science and Technology, Trondheim, Norway, 2003.
[10] M. J. Er, S. Wu, J. Lu, and H. L. Toh, "Face recognition with radial basis function (RBF) neural networks," *IEEE Transactions on Neural Networks*, vol. 13, no. 3, pp. 697–710, 2002.
[11] A. J. Howell, "Face recognition using rbf networks," in *Radial Basis Function Networks 2: New Advances in Design*, R. J. Howlett and L. C. Jain, Eds., pp. 103–142. Physica-Verlag, 2001.
[12] R. Q. Feitosa, M. M. B. R. Vellasco, D. V. Andrade, S. A. R. S. Maffra, and D. T. Oliveira, "Facial expression classification using RBF and back-propagation neural networks," in *Proceedings of the 6th International Conference on Information Systems Analysis and Synthesis (ISAS'2000)*, 2000, pp. 73–77.
[13] T. K. Kohonen, *Self-Organizing Maps*, Springer-Verlag, Berlin, Heidelberg, 2nd extended edition, 1997.
[14] T. Kohonen, "The self-organizing map," *Proceedings of the IEEE*, vol. 78, no. 9, pp. 1464–1480, 1990.
[15] M. Oja, S. Kaski, and T. Kohonen, "Bibliography of self-organizing map (som) papers: 1998-2001 addendum," *Neural Computing Surveys*, vol. 3, pp. 1–156, 2003.
[16] P. Navarrete and J. Ruiz del Solar, "Interactive face retrieval using self-organizing maps," in *Proceedings of the IEEE International Joint Conference on Neural Networks (IJCNN)'02*, 2002, pp. 687–691.
[17] J. Jiang, "Image compression with neural networks - a survey," *Signal Processing-Image Communication*, vol. 14, no. 9, pp. 737–760, 1999.
[18] C. Amerijckx, M. Verleysen, P. Thissen, and J.-D. Legat, "Image compression by self-organized kohonen map," *IEEE Transactions on Neural Networks*, vol. 9, no. 3, pp. 503–507, 1998.
[19] P. C. Cosman, K. L. Oehler, E. A. Riskin, and R. M. Gray, "Using vector quantization for image processing," in *Proceedings of the IEEE*, 1993, vol. 81, pp. 1326–1341.
[20] S. Lawrence, C. L. Giles, Ah Chung Tsoi, and A. D. Back, "Face recognition: a convolutional neural-network approach," *IEEE Transactions on Neural Networks*, vol. 8, no. 1, pp. 98–113, 1997.
[21] Xiaoyang Tan, Songcan Chen, Zhi-Hua Zhou, and Fuyan Zhang;,

"Recognizing partially occluded, expression variant faces from single training image per person with SOM and soft k-NN ensemble," *IEEE Transactions on Neural Networks*, vol. 16, no. 4, pp. 875–886, 2005.

[22] J. C. Principe, N. R. Euliano, and W. C. Lefebvre, *Neural Adaptive Systems: Fundamentals Through Simulations*, John Willey and Sons, 2000.

[23] S. Haykin, *Neural Networks: A Comprehensive Foundation*, Macmillan Publishing Company, Englewood Cliffs, NJ, 1994.

[24] R. Xu and D. Wunsch, "Survey of clustering algorithms," *IEEE Transactions on Neural Networks*, vol. 16, no. 3, pp. 645–678, 2005.

[25] A. K. Jain, M. N. Murty, and P. J. Flynn, "Data clustering - A survey," *ACM Computing Surveys*, vol. 31, no. 3, pp. 264–323, 1999.

[26] T. K. Kohonen, E. Oja, O. Simula, A. Visa, and J. Kangas, "Engineering applications of the self-organizing map," *Proceedings of the IEEE*, vol. 84, no. 10, pp. 1358–1384, 1996.

[27] M. Cottrell, J. C. Fort, and G. Pagès, "Theoretical aspects of the SOM algorithm," *Neurocomputing*, vol. 21, pp. 119–138, 1998.

[28] A. Laha and N. R. Pal, "Some novel classifiers designed using prototypes extracted by a new scheme based on self-organizing feature map," *IEEE Transactions on Systems, Man, and Cybernetics*, vol. B-31, no. 6, pp. 881–890, 2001.

[29] B. Wyns, S. Sette, L. Boullart, D. Baeten, I. E. A. Hoffman, and F. De Keyser, "Prediction of diagnosis in patients with early arthritis using a combined Kohonen mapping and instance-based evaluation criterion," *Artificial Intelligence in Medicine*, vol. 31, no. 1, pp. 45–55, 2004.

[30] C. I. Christodoulou, S. C. Michaelides, and C. S. Pattichis, "Multifeature texture analysis for the classification of clouds in satellite imagery," *IEEE Transactions on Geoscience and Remote Sensing*, vol. 41, no. 11, pp. 2662–2668, 2003.

[31] P. N. Suganthan, "Hierarchical overlapped som's for pattern classification," *IEEE Transactions on Neural Networks*, vol. 10, no. 1, pp. 193–196, 1999.

[32] Y.-D. Xiao, A. Clauset, R. Harris, E. Bayram, P. Santago, and J. D. Schmitt, "Supervised self-organizing maps in drug discovery. 1. robust behavior with overdetermined data sets," *Journal of Chemical Information and Modeling*, vol. 45, no. 6, pp. 1749–1758, 2005.

[33] R. del-Hoyo, D. Buldain, and A. Marco, "Supervised classification with associative SOM," in *Proceedings of the 7th International Work-Conference on Artificial and Neural Networks, (IWANN)'03*, 2003, pp. 334–341.

[34] E. Biebelmann, M. Köppen, and B. Nickolay, "Pratical aplications of neural networks in texture analysis," *Neurocomputing*, vol. 13, no. 2–4, pp. 261–279, 1996.

[35] P. N. Belhumeur, J. P. Hespanha, and D. J. Kriegman, "Eigenfaces vs. Fisherfaces: Recognition using class specific linear projection," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 19, no. 7, pp. 711–720, 1997.