Dynamic LVQ Models for Classification of Spatiotemporal Patterns

Isaque Q. Monteiro, Guilherme A. Barreto, and Patrícia V. Nascimento

***Abstract*—This paper proposes the combination of three short-term memory (STM) mechanisms with the Learning Vector Quantization (LVQ) model for classifying spatiotemporal patterns. The goal is to investigate the ability of these dynamic models to acquire neural representations of faces that are invariant to changes in images caused by the movement of the subjects. The proposed models are evaluated by their ability to recognize faces in sequences of images, as well as by their sensitivity to memory parameters and image noise. A simple theoretical analysis for understanding the discriminative power of the proposed spatiotemporal classifiers is also provided. Through simulations, it is shown that the dynamic variants of LVQ perform considerably better than the static LVQ model, achieving classification rates similar to those obtained by a dynamic MLP network.**

***Index Terms*—Learning vector quantization, face recognition, spatiotemporal classifiers, short-term memory.**

I. Introduction

A long-standing problem in cognitive science has been the formation of *perceptual invariances*, such as the remarkable human ability to recognize objects (e.g. faces) independently of variations in their spatiotemporal representation, and their emergence in learning processes [1]. Despite recent advances in the computational intelligence field, such an ability is still very difficult to be reproduced by artificial learning systems, specially in real-time, unconstrained and unpredictable environments (see [2], [3], [4] for surveys). This explains in part why automated face recognition is still an attractive domain of research.

From an engineering-oriented perspective, invariant face recognition plays an important role in many applications, such as building access control, suspect identification and surveillance [5], [6]. All these promising applications have resulted in a significant increase of research activities in this area over the past few years. However, few can achieve a completely reliable performance. The problem arises due to the difficulty of distinguishing different individuals who have approximately the same facial configuration and yet contend with wide variations in the appearance of a particular face due to changes in pose, lighting, facial makeup, facial expression and, in a very important degree in real-world applications, temporal effects [7].

A key hypothesis that has been investigated to design artificial learning systems capable of building invariant representations assumes that when a system is supposed to recognize input patterns irrespective of certain transformations such as translation, rotation, and scaling, these transformations are actually learned from natural *sequences* of such patterns that are produced from each other by the same transformations [8]. Thus, it is important to investigate the possibility of learning, storing, and classifying patterns that occur in sequence, because many tasks performed by humans and animals involve decision-making and behavioral responses to spatiotemporal stimuli.

Artificial Neural Networks (ANN) have been successfully applied to several pattern recognition applications because of their powerful classification abilities and their inherent abstraction/generalization properties, which result from learning [9]. Despite the vast majority of ANN models have been concerned with the learning of static (memoryless) patterns, there are smaller group of neural algorithms that are able to deal with spatiotemporal (dynamic) patterns. The processing of these patterns differs fundamentally from that of static ones in that the temporal order and/or temporal correlation of the patterns being observed must be taken into account [10].

Some attempts showing how temporal correlations can lead to invariant recognition of faces or objects by neural networks are found in [11], [12], [13], [14], [7]. For example, in [14], the authors have tested a competitive neural method in recognizing invariance to pose (angle of rotation related to the front image) and invariance to facial expression, reporting classification rates ranging from 40% to 100%. All of these studies, however, have essentially used the same memory mechanism combined with simple competitive neural models to capture current and past information from the input sequence, despite the existence of many alternatives as reviewed in [10]. Furthermore, none of them have provided a unifying theoretical framework for analyzing and designing of spatiotemporal classifiers.

This paper proposes three dynamic LVQ models for classification of faces appearing in close temporal proximity and compare their performances. It is intended to investigate the ability of each model to acquiring representations that are tolerant to changes in the images. The proposed models are also evaluated by their by their sensitivities to memory parameters and image noise. In addition, a first attempt in setting up a theoretical framework for understanding the discriminative power of the proposed spatiotemporal classifiers is also provided. It is shown that the dynamic variants of LVQ perform considerably better than the static LVQ model, achieving classification rates similar to those obtained by a dynamic MLP network.

The remainder of this paper is organized as follows. The proposed dynamic LVQ models are described in Section II. A theoretical analysis of the proposed models under the framework of statistical pattern recognition is carried out in Section III. In Section IV simulations with the three dynamic LVQ models are presented. The paper is concluded

The authors are with the Department of Teleinformatics Engineering, Federal University of Ceará, Campus do Pici, Av. Mister Hull, S/N, CP 6005, CEP 60455-760, Fortaleza, Ceará, BRAZIL.
Emails: {isaque, guilherme, patricia}@deti.ufc.br

in Section V.

II. Dynamic Learning Vector Quantization

The Learning Vector Quantization (LVQ) comprises an important family of neural classifiers built according to competitive learning principles. Each neuron i in LVQ models is associated with a weight vector $\mathbf{w}_i = [w_{i1}, w_{i2}, \ldots, w_{in}]^T \in \mathbb{R}^n$ and has a class-label $\mathcal{C}_i$ assigned to it at the beginning of the learning process. Through a supervised learning process, the output neurons become tuned after several presentations of the input data vectors $\mathbf{x} \in \mathbb{R}^n$ and their corresponding classes. The basic learning algorithm comprises two steps:

(i) The index of the winning neuron, $i^*(t)$, is found by computing the Euclidean distances between the current input vector and all weight vectors:

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

(ii) The weight vector of the winning neuron, $\mathbf{w}_{i^*}(t)$ is updated according to one of the following learning rules:

- If $\mathbf{x}(t)$ and $\mathbf{w}_{i*}(t)$ belong both to $\mathcal{C}_i$:

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

- If $\mathbf{x}(t)$ and $\mathbf{w}_{i*}(t)$ DO NOT belong both to $\mathcal{C}_i$:

$$\mathbf{w}_{i^*}(t+1) = \mathbf{w}_{i^*}(t) - \alpha(t)[\mathbf{x}(t) - \mathbf{w}_{i^*}(t)] \tag{3}$$

where t is the discrete time instant. Weight updating is usually done after the presentation of each input vector $\mathbf{x}(t)$ and the initial values of $\mathbf{w}_i(0)$ are usually random. For better convergence of weights, the learning rate should decrease with time according to $\alpha(t) = \alpha_0 \left(\alpha_T/\alpha_0\right)^{t/T}$, where α_0 and α_T are the initial and final values of the learning rate, respectively. The maximum number of training iterations (T) is the number of training epochs times the number of training vectors.

Once training is completed, a new incoming vector $\mathbf{x}(t)$ is then associated to the same class to which the nearest weight vector $\mathbf{w}_{i^*}(t)$ belongs. The learning procedure just described is known as the LVQ1 algorithm. Variants of it, called LVQ2 and LVQ3, can be found in [15], [16].

A. Short-Term Memory Models

The original LVQ classifiers are based on the matching of static patterns alone. However, the input patterns may, in addition to being spatially related, may occur in a sequence, as pointed out in the introduction. In this case, the temporal order in which the patterns are observed plays a major role and must be taken into account by the neural model.

In order to represent temporal associations between consecutive patterns in a temporal sequence, the network must be able to retain information about past sequence items. This type of retention mechanism, usually called *short-term memory* (STM), will be considered in this paper to design LVQ-based spatiotemporal classifiers. STM mechanisms can be realized in a number of ways (see [10], [17] for detailed reviews), but we will focus on just three possibilities.

Dynamic LVQ: Model 1

In object recognition applications, the conventional solution for the achievement of invariances in perception, such as the invariance with respect to movements of an object, is to provide a simple classifier with a heuristically designed *preprocessing stage* that extracts a set of *invariant features* from the primary signals [16]. In contemporary pattern recognition, certain local features, such as pieces of sinusoidal waveforms called the *wavelets* have become popular as invariant features. Classification, for instance by neural networks, is then based on these features.

For the purposes of this paper, a preprocessing procedure that captures the dynamic information in the input sequence can be achieved as follows [18]:

$$\bar{\mathbf{x}}(t) = (1-\lambda)\bar{\mathbf{x}}(t-1) + \lambda\mathbf{x}(t) \tag{4}$$

where the vector $\mathbf{x}(t)$ is the current sequence pattern, and $0 \leq \lambda \leq 1$ is the memory parameter, which determines the influence of past inputs. We usually set $\bar{\mathbf{x}}(0) = \mathbf{0}$ at the beginning of each sequence of vectors.

If $\lambda = 1$, no past information is available and the system remains static (memoryless). If $\lambda < 1$, the actual input vector presented to the network $\bar{\mathbf{x}}(t)$ mixes information about the present $\lambda\mathbf{x}(t)$ with information from the past $(1-\lambda)\bar{\mathbf{x}}(t-1)$. By using this type of STM model, Equations (1) and (2), which define the search for the winning neuron and the updating of its weight vector, remain the same.

Dynamic LVQ: Model 2

The previous STM model is indeed a preprocessing method, since the network itself remains static. The main advantage of this STM mechanism is that it can be used by practically every neural network model to process temporal data.

An appealing alternative to extract information from sequence of vectors consists in modifying the neural network algorithm to directly cope with spatiotemporal information. Thus, the following STM mechanism can be equally used to design spatiotemporal LVQ classifiers [19]:

$$a_i(t) = (1-\lambda)a_i(t-1) - \frac{1}{2}\|\mathbf{x}(t) - \mathbf{w}_i(t)\|^2 \tag{5}$$

where $a_i(t)$ is called the temporal activation of neuron i, $\mathbf{w}_i(t)$ is its weight vector, and $\mathbf{x}(t)$ is the current input vector. As previously, $0 \leq \lambda \leq 1$ is the memory parameter. We assume $a_i(0) = 0$, $\forall i$, at the beginning of each sequence of input vectors.

It is worth noting that the winning neuron $i^*(t)$ is now chosen as follows:

$$a_{i^*}(t) = \max_{\forall i}\{a_i(t)\} \tag{6}$$

or, equivalently:

$$a_{i^*}(t) > a_i(t), \quad \forall i \neq i^*(t) \tag{7}$$

while, the weight updating equation (2) remains unchanged.

Dynamic LVQ: Model 3

The DLVQ-2 model modified only the procedure of search for the winning neuron. In [20], the authors proposed an STM model that captures temporal information by modifying both Equations (1) and (2). Thus, the winning neuron is now selected as follows:

$$i^*(t) = \arg\min_{\forall i} \|\mathbf{y}_i(t)\| \tag{8}$$

where $\mathbf{y}_i(t)$ is the difference vector of neuron i, which is given by:

$$\mathbf{y}_i(t) = (1-\lambda)\mathbf{y}_i(t-1) + \lambda[\mathbf{x}(t) - \mathbf{w}_i(t)] \tag{9}$$

where $\mathbf{w}_i(t)$ is the weight vector of unit i, and $0 \leq \lambda \leq 1$ is the memory parameter, which weighs the influence of past difference vectors in relation to the current input vector $\mathbf{x}(t)$. Accordingly, the weight vector of the winning neuron should be updated as a function of $\mathbf{y}_i(t)$ as follows:

- If $\mathbf{x}(t)$ and $\mathbf{w}_{i*}(t)$ belong both to $\mathcal{C}_i$:

$$\mathbf{w}_{i^*}(t+1) = \mathbf{w}_{i^*}(t) + \alpha(t)\mathbf{y}_{i^*}(t) \tag{10}$$

- If $\mathbf{x}(t)$ and $\mathbf{w}_{i*}(t)$ DO NOT belong both to $\mathcal{C}_i$:

$$\mathbf{w}_{i^*}(t+1) = \mathbf{w}_{i^*}(t) - \alpha(t)\mathbf{y}_{i^*}(t) \tag{11}$$

III. A Simple Theoretical Insight

The goal of this section is to give some insight about what kind of computation is being performed by the proposed DLVQ classifiers. For that, we will use the framework of statistical pattern recognition. Under this framework, a given feature vector $\mathbf{x}$ is said to belong to class $\mathcal{C}_k$, if the following general condition is observed:

$$g_k(\mathbf{x}) > g_i(\mathbf{x}), \qquad \forall i \neq k \tag{12}$$

where $g_i(\cdot)$ is the discriminant function associated to class $\mathcal{C}_i$ [21]. This condition is generally implemented through Fisher's criterium for optimal classification:

$$p(\mathcal{C}_k|\mathbf{x}) > p(\mathcal{C}_i|\mathbf{x}), \qquad \forall i \neq k \tag{13}$$

where $p(\mathcal{C}_i|\mathbf{x})$ is a posterior density function defining the probability that, given the feature vector $\mathbf{x}$, it belongs to class $\mathcal{C}_i$. With the help of Bayes rule, we rewrite (13) as follows:

$$p(\mathbf{x}|\mathcal{C}_k)p(\mathcal{C}_k) > p(\mathbf{x}|\mathcal{C}_i)p(\mathcal{C}_i), \qquad \forall i \neq k \tag{14}$$

where $p(\mathbf{x}|\mathcal{C}_i)$ is the likelihood function of class $\mathcal{C}_i$, which gives the probability that, given a certain class, it is this class that better "explain" the vector $\mathbf{x}$. The density function $p(\mathcal{C}_k)$ gives the prior probabilities of selecting class $\mathcal{C}_i$.

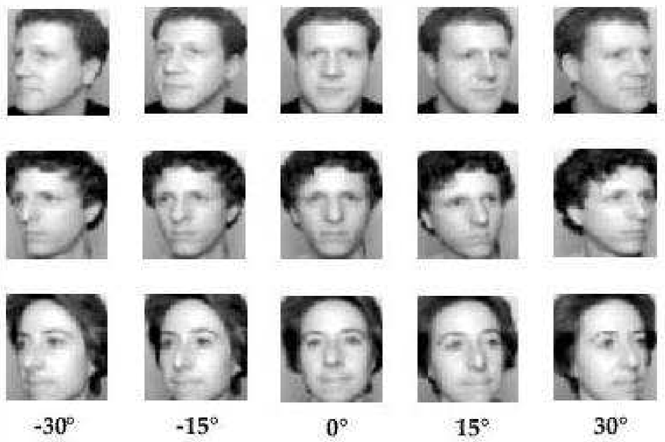

Fig. 1. Sample of the 100 images used in the simulation.

A classifier designed according to (14) is called a *Bayes Optimal Classifier* [9]. Assuming equal probability for each class and Gaussian likelihood functions for all classes, we get:

$$p(\mathbf{x}|\mathcal{C}_i) = \frac{1}{(2\pi)^{\frac{n}{2}}|\mathbf{C}_i|^{\frac{1}{2}}} \exp\left\{-\frac{1}{2}(\mathbf{x}-\boldsymbol{\mu}_i)^T\mathbf{C}_i^{-1}(\mathbf{x}-\boldsymbol{\mu}_i)\right\} \tag{15}$$

where $\boldsymbol{\mu}_i = E[\mathbf{x}|\mathcal{C}_i]$ is the *mean vector* and $\mathbf{C}_i = E[(\mathbf{x}-\boldsymbol{\mu}_i)(\mathbf{x}-\boldsymbol{\mu}_i)^T]$ is the *covariance matrix* of a given class $\mathcal{C}_i$, respectively. The factor $|\mathbf{C}_i|$ is the determinant of the covariance matrix.

Taking the natural logarithm of both sides of (15) and eliminating terms that are independent of the index i, we can write the discriminant function of class $\mathcal{C}_i$ as:

$$g_i(\mathbf{x}) = \ln p(\mathbf{x}|\mathcal{C}_i) = -\frac{1}{2}\ln(|\mathbf{C}_i|) - \frac{1}{2}(\mathbf{x}-\boldsymbol{\mu}_i)^T\mathbf{C}_i^{-1}(\mathbf{x}-\boldsymbol{\mu}_i) \tag{16}$$

If we further assume a diagonal form for $\mathbf{C}_i$ and a common variance σ^2 for all components of $\mathbf{x}$, i.e. $\mathbf{C}_i = \sigma^2\mathbf{I}$, the discriminant function reduces to:

$$g_i(\mathbf{x}) = -\frac{1}{2\sigma^2}(\mathbf{x}-\boldsymbol{\mu}_i)^T(\mathbf{x}-\boldsymbol{\mu}_i) = -\frac{1}{2\sigma^2}||\mathbf{x}-\boldsymbol{\mu}_i||^2 \tag{17}$$

where $||\cdot||$ is the Euclidean vector norm.

By comparing the discriminant function in (17) with the second term on the right-hand side of (5), we easily note that they are the same, except for the standard-deviation σ, which can be set to 1 without loss of generality. The first term on the right-hand side of (5) is responsible for the memory of past activations. Hence, the proposed DLVQ models, specially DLVQ-2 and DLVQ-3, are building *time-dependent discriminant functions*, i.e., discriminant functions which are sensitive to temporal dependencies! Further developments should be made to better understand the computational power of the proposed spatiotemporal classifiers, but a first insight has been given here.

IV. Simulations

The simulations to be shown in this section aim to evaluate the three DLVQ models in the following tasks: (*i*)

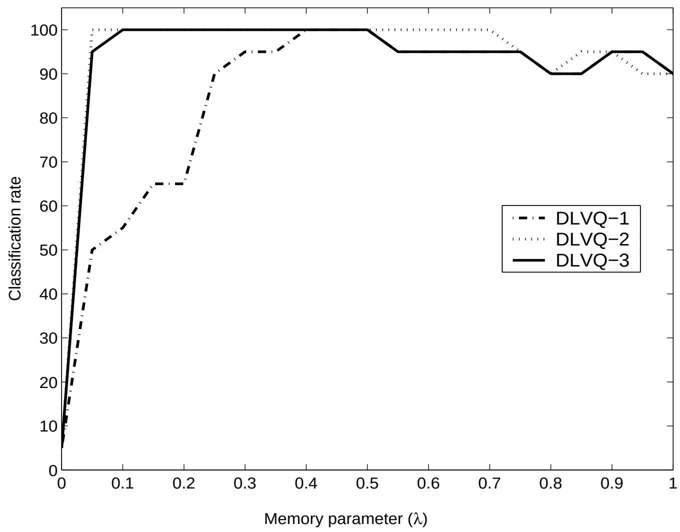

Fig. 2. Classification performance as a function of the memory parameter λ.

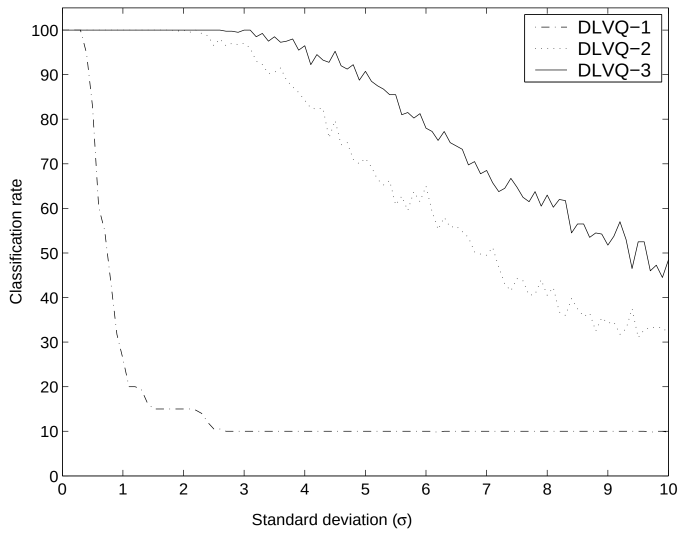

Fig. 3. Classification performance as a function of noise variance.

classification performance, (*ii*) dependence of the classification rate on the memory parameter λ, and (*iii*) sensitivity of the models to noise in the input images. Data for these simulations consisted of 80 images of faces undergoing a change in pose. There were sixteen individuals at each of five poses, ranging from -30^o to $+30^o$, as the subject change pose from left to right (Figure 1). Image set was provided by Marian Bartlett [14] with permission of David Beymer [22]. The faces were automatically located in the frontal view image by using the feature-based template matching algorithm proposed in [22]. The location of the face in the frontal view image defined a window for the other images in the sequence. Each input sequence then consisted of a single stationary window within which the subject moved the head. The images were normalized for luminance and scaled to 60×60 pixels.

Each 60×60 image within a sequence was then converted to a 3600-dimensional vector by concatenating each column of the original image one below the other. In order to reduce the dimension of the input vectors, they were preprocessed by the well-known *Principal Component Analysis* (PCA) method [21]. The dimension of each transformed input vector was chosen to be 2000, a value that corresponds to approximately 95% of the variance of the data.

For training the DLVQ models, we adopt a procedure different from that usually employed to train the static LVQ classifier. For the static case, the winning neuron is found after the presentation of *each* image in a sequence, and its weight vector is immediately updated according to (2). For the dynamic LVQ models, the winning neuron is found only at the end of a given sequence of images, i.e., only after the presentation of the 5-*th* image of a sequence. This is equivalent to say that the weights are updated only at the end of a sequence.

For comparison purpose, we trained a Multilayer Perceptron (MLP) network using the STM model described in (4). We refer to the MLP network thus trained as the *Dynamic MLP* (DMLP) model. The DMLP had 10 neurons in the hidden layer (found by trial-and-error) and 4 neurons in the output layer (simple binary encoding of the 16 output classes), all of them using logistic activation functions. The usual backpropagation algorithm with momentum term was used to adjust the weights of the DMLP model, only after the presentation of the 5-*th* image of a sequence of faces.

Sixteen neurons (one for each individual) were used for the classification task. The weight vector of a given neuron is initialized to an image randomly selected from the five available for the class it represents. The training parameters common to all neural models are the following: 200 epochs, $\alpha_0 = 0.1$, $\alpha_T = 0.0001$ and $T = 200 * 20 = 4000$. The task of all the DLVQ classifiers is to recognize that a sequence of face images (feature vectors), corresponding to sequential views of a person's face, contains images of the same person (class), irrespective to changes caused by the movement of the subjects.

The first set of simulations evaluates the classification performance of the proposed DLVQ models as a function of the memory parameter λ. With these tests we want to confirm that the DLVQ models perform better that the static LVQ classifier on spatiotemporal data, as well as to find an optimal value for the memory parameter λ, if possible, that is common to all DLVQ models. Figure 2 shows the results for λ ranging from 0 to 1.

According to this figure, the DLVQ-2 model performed better than the other two DLVQ models, i.e., its classification rate is 100% for a wider range of values of the memory parameter ($\lambda \in [0.05, 0.7]$). A 100% rate means that the classifier indicated correctly the class (individual) that a given sequence of images belongs to. The worst classifier was the DLVQ-1 model (100% classification rate only for $\lambda \in [0.4, 0.5]$). We can observe that exactly for this range all DLVQ models have a 100% classification rate. The DLVQ-3 model had a 100% classification rate for $\lambda \in [0.1, 0.5]$).

λ	**LVQ**	**DMLP**	**DLVQ-1**	**DLVQ-2**	**DLVQ-3**
0.05	−−	95%	50%	100%	95%
0.25	−−	100%	90%	100%	100%
0.50	−−	100%	100%	100%	100%
0.65	−−	100%	95%	100%	95%
0.80	−−	100%	90%	90%	90%
1.00	90%	95%	90%	90%	90%

TABLE I
Classification rates for the DLVQ and DMLP models.

From Figure 2 one can also note that the static case, corresponding to $\lambda = 1$ for all DLVQ models, achieved a 90% classification rate. Table I shows numerical values of the classification rates of the DLVQ and DMLP models for some values of λ. From this table that the DLVQ models performed similarly to the powerful DMLP model for almost the entire range of variation of λ. Furthermore, it is important to emphasize that the DMLP is computationally much more expensive than the proposed DLVQ models. Thus, the DLVQ models are more suitable for real-time face recognition applications than the DMLP.

The last set of simulations evaluates the robustness of the DLVQ models with respect to the presence of gaussian white noise in the input images. Based on Table I we adopt $\lambda^* = 0.5$ as the optimum value of λ. The classification rates were computed for the standard deviation (σ) ranging from 0 to 10. For each value of σ, 100 independent classification tests were performed and the resulting classification rates were averaged to give the final rate. Figure 3 shows the results for the three dynamic classifiers. From that we conclude that the DLVQ-3 model is less sensitive to noise than the others, since its classification rates remain the highest for all the range of interest. Again, the DLVQ-1 model had the worst performance. By combining the results of Figures (2) and (3), we elect the DLVQ-3 model as the best one, suggesting its use also for classifying other spatiotemporal data sets.

V. Conclusion

This paper proposed three dynamic Learning Vector Quantization (LVQ) models for classifying spatiotemporal patterns. The goal was to investigate their abilities to acquire neural representations of faces that are invariant to changes in images caused by the movement of the subjects.

The proposed models were evaluated by their ability to recognize faces in sequences of images, as well as by their sensitivity to memory parameters and noise. A first step into the proposal of a theoretical framework to describing the discriminative power of the proposed spatiotemporal classifiers under the framework of statistical pattern recognition was also provided. All the dynamic variants of LVQ were shown to perform considerably better than the static LVQ model, achieving classification rates similar to those obtained by a dynamic MLP network.

Acknowledgments

The authors thank Dr. Marian S. Bartlett for kindly providing the image set, and the CNPq (DCR and UFC/PIBIC research programs) for the financial support.

References

[1] T. Kohonen, S. Kaski, and H. Lappalainen, “Self-organized formation of various invariant-feature filters in the adaptive-subspace SOM,” *Biological Cybernetics*, vol. 9, no. 6, pp. 1321–1344, 1997.
[2] W. Zhao, R. Chellappa, J. Phillips, and A. Rosenfeld, “Face recognition: A literature survey,” *ACM Computing Surveys*, vol. 35, no. 4, pp. 399–458, 2003.
[3] M. A. Grudin, “On internal representation in face recognition systems,” *Pattern Recognition*, vol. 33, pp. 1161–1177, 2000.
[4] R. Chellapa, C. L. Wilson, and S. Sirohey, “Human and machine recognition of faces: A survey,” *Proceedings of the IEEE*, vol. 83, pp. 705–740, 1995.
[5] S.Y. Kung, M.W. Mak, and S.H. Lin, *Biometric authentication: A machine learning approach*, Prentice Hall, 2004.
[6] A. Sato, H. Imaoka, T. Suzuki, and T. Hosoi, “Advances in face detection and recognition technologies,” *NEC Journal of Advanced Technology*, vol. 2, no. 1, pp. 28–34, 2005.
[7] M. S. Bartlett, *Face Image Analysis by Unsupervised Learning*, Kluwer Academic Publishers, Boston, 2001.
[8] P. Földiák, “Learning invariance from transformation sequences,” *Neural Computation*, vol. 3, no. 2, pp. 194–200, 1991.
[9] J. C. Príncipe, N. R. Euliano, and W. C. Lefebvre, *Neural and Adaptive Systems: Fundamentals through Simulations*, John Wiley & Sons, 2000.
[10] G. A. Barreto and A. F. R. Araújo, “Time in self-organizing maps: an overview of models,” *International Journal of Computer Research*, vol. 10, no. 2, pp. 139–179, 2001.
[11] J. Stone and A. Bray, “A learning rule for extracting spatio-temporal invariances,” *Network: Computation in Neural Systems*, vol. 6, no. 3, pp. 429–436, 1995.
[12] J. Stone, “Learning perceptually salient visual parameters using spatiotemporal smoothness constraints,” *Neural Computation*, vol. 8, no. 7, pp. 1463–1492, 1996.
[13] G. Wallis, “Spatio-temporal influences in object recognition,” *Network: Computation in Neural Systems*, vol. 9, pp. 265–278, 1998.
[14] M. S. Bartlett and T. J. Sejnowski, “Learning viewpoint-invariant face representations from visual experience in an attractor network,” *Network: Computation in Neural Systems*, vol. 9, no. 3, pp. 399–417, 1998.
[15] T. Kohonen, “The self-organizing map,” *Proceedings of the IEEE*, vol. 78, pp. 1464–1480, 1990.
[16] T. Kohonen, *Self-Organizing Maps*, Springer-Verlag, Berlin, 2nd edition, 1997.
[17] G. A. Barreto, A. F. R. Araújo, and S. C. Kremer, “A taxonomy for spatiotemporal connectionist networks revisited: The unsupervised case,” *Neural Computation*, vol. 15, no. 6, pp. 1255–1320, 2003.
[18] J. Kangas, “Time-delayed self-organizing maps,” in *Proc. IEEE/INNS International Joint Conf. on Neural Networks (IJCNN)*, 1990, vol. 2, pp. 331–336.
[19] G. J. Chappell and J. G. Taylor, “The temporal Kohonen map,” *Neural Networks*, vol. 6, no. 3, pp. 441–445, 1993.

[20] T. Koskela, M. Varsta, J. Heikkonen, and S. Kaski, "Time series prediction using recurrent SOM with local linear models," *International Journal of Knowledge-based Intelligent Engineering Systems*, vol. 2, no. 1, pp. 60–68, 1998.
[21] A. Webb, *Statistical Pattern Recognition*, Wiley & Sons, 2002.
[22] D. Beymer, "Face recognition under varying pose," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 1994, pp. 756–761.