[7] X. J. Li *et al.*, "CAD-vision-range-based self-localization for mobile robot using one landmark," *J. Intell. Robot. Syst.*, vol. 35, no. 1, pp. 61–82, 2002.
[8] J. M. Perez, C. Urdiales, A. Bandera, and F. Sandoval, "A Hough-based solution to the simultaneous localization and map building problem," in *Proc. 1st Eur. Conf. Mobile Robots (ECMR)*, Radziejowice, Poland, 2003, pp. 53–58.
[9] A. Bonci, G. Ippoliti, L. Jetto, T. Leo, and S. Longhi, "Methods and algorithms for sensor data fusion aimed at improving the autonomy of mobile robot," in *Advances in Control of Articulated Mobile Robots*, B. Siciliano *et al.*, Eds. Heidelberg, Germany: Springer-Verlag, Springer Tracts in Advanced Robotics (STAR), 2004, pp. 191–222.
[10] P. V. C. Hough, "Methods and means for recognising complex patterns," U.S. Patent 3 069 654, Dec. 18, 1962.
[11] R. O. Duda and P. E. Hart, "Use of the Hough Transform to detect lines and curves in pictures," *Commun. ACM*, vol. 15, no. 1, pp. 11–15, Jan. 1972.
[12] F. O'Gorman and M. B. Clowes, "Finding picture edges through collinearity of feature points," *IEEE Trans. Comput.*, vol. C-25, no. 4, pp. 449–454, Apr. 1976.
[13] R. M. Haralick and L. G. Shapiro, *Computer and Robot Vision*, vol. 1. Reading, MA: Addison-Wesley, 1992.
[14] Q. Ji and R. M. Haralick, "An improved Hough Transform technique based on error propagation," in *IEEE Int. Conf. Systems Man and Cybernetics*, San Diego, CA, 1998, vol. 5, pp. 4653–4658.
[15] ——, "Error propagation for the Hough Transform," *Pattern Recogn. Lett.*, vol. 22, no. 6–7, pp. 813–823, 2001.
[16] I. E. Sobel, "Camera models and machine perception," Ph.D. dissertation, Elect. Eng. Dept., Stanford Univ., Stanford, CA, 1970.
[17] A. H. Jazwinski, "Mathematics in sciences and engineering," in *Stochastic Processes and Filtering Theory*, vol. 64. New York: Academic, 1970.
[18] H. J. Larson and B. O. Shubert, "Probabilistic models in engineering sciences," in *Random Variables and Stochastic Processes*, vol. 1. New York: Wiley, 1979.
[19] E. A. Bender, *Mathematical Methods in Artificial Intelligence*. Los Alamitos, CA: IEEE Comput. Soc. Press, 1996.
[20] R. C. Luo and M. G. Kay, *Data Fusion and Sensor Integration, Data Fusion in Robotics and Machine Intelligence*, M. A. Abidi and R. C. Gonzales, Eds. Orlando, FL: Academic, 1992, pp. 54–56.
[21] A. Bonci, S. Longhi, A. Monteriù, and M. Vaccarini, "Motion control of a smart mobile manipulator," in *Int. Conf. Intelligent Manipulation and Grasping*, Genoa, Italy, Jul. 1–2, 2004, pp. 110–116.
[22] ——, "Navigation system of a smart wheelchair," *J. Zhejiang Univ. Sci.*, vol. 6A, no. 2, pp. 110–117, Feb. 2005.
[23] A. Bonci, G. Di Francesco, and S. Longhi, "A Bayesian approach to the Hough Transform for video and ultrasonic data fusion in mobile robot navigation," in *Proc. IEEE Int. Conf. Systems Man and Cybernetics*, Hammamet, Tunisia, 2002, vol. 3, pp. 354–359.

Linear Versus Nonlinear Neural Modeling for 2-D Pattern Recognition

Claudio A. Perez, Guillermo D. Gonzalez, Leonel E. Medina, and Francisco J. Galdames

***Abstract*—This paper compares the classification performance of linear-system- and neural-network-based models in handwritten-digit classification and face recognition. In inputs to a linear classifier, nonlinear inputs are generated based on linear inputs, using different forms of generating products. Using a genetic algorithm, linear and nonlinear inputs to the linear classifier are selected to improve classification performance. Results show that an appropriate set of linear and nonlinear inputs to the linear classifier were selected, improving significantly its classification performance in both problems. It is also shown that the linear classifier reached a classification performance similar to or better than those obtained by nonlinear neural-network classifiers with linear inputs.**

***Index Terms*—Face recognition, genetic selection of inputs, handwritten-digit classification, linear classifier, neural-network classifier, nonlinear inputs.**

I. Introduction

Modeling, based on linear systems and neural networks, has been widely used in a large number of applications [1], [17]. Although the linear-system and the neural-network approaches to modeling have grown independently, they make use of similar techniques. The present study compares linear-system-based models and neural-network-based models in two pattern-recognition problems: handwritten-digit classification and face recognition.

Neural networks have been used in modeling pattern-recognition problems in many industrial applications such as in woodboard-defects classification for the forestry industry and in the lithological-composition sensor for the mining industry [2], [6], [12]. Handwritten-digit recognition is an important task in automated document analysis. Applications have been developed to read postal addresses, bank checks, tax forms, and census forms, including reading aids for the visually impaired, among others [3]–[5]. Face recognition has become popular as a possible person-identification procedure based on biometrics. Several papers have used neural networks for face recognition in the past few years [8], [14]. Both problems, handwritten-digit classification and face recognition, are suitable for exploring new approaches in the design of two-dimensional (2-D) pattern-recognition classifiers, because they are inherently complex tasks, but they are restricted to only a few classes, enabling relatively simple implementation [20].

In the problem of handwritten-digit classification, published results on classification performance for handwritten digits fall in a broad range, from 68% [15] to 99% [5], [18], [26]. Results largely depend on the size of the database, type of partition, and rejection ratios employed [5]. In the face-recognition problem, performances have been reported reaching up to 95% on limited-size (less than 50

Manuscript received March 4, 2002; revised November 21, 2002 and August 27, 2003. This research was funded by CONICYT–Chile through project FONDECYT 1000977 and by the Department of Electrical Engineering, Universidad de Chile, Santiago, Chile. This paper was recommended by Associate Editor D. Zhang.

The authors are with the Department of Electrical Engineering, Universidad de Chile, Santiago, Chile (e-mail: clperez@cec.uchile.cl; gugonzal@cec.uchile.cl; lmedina@cec.uchile.cl; fgaldame@cec.uchile.cl).

Digital Object Identifier 10.1109/TSMCA.2005.851268

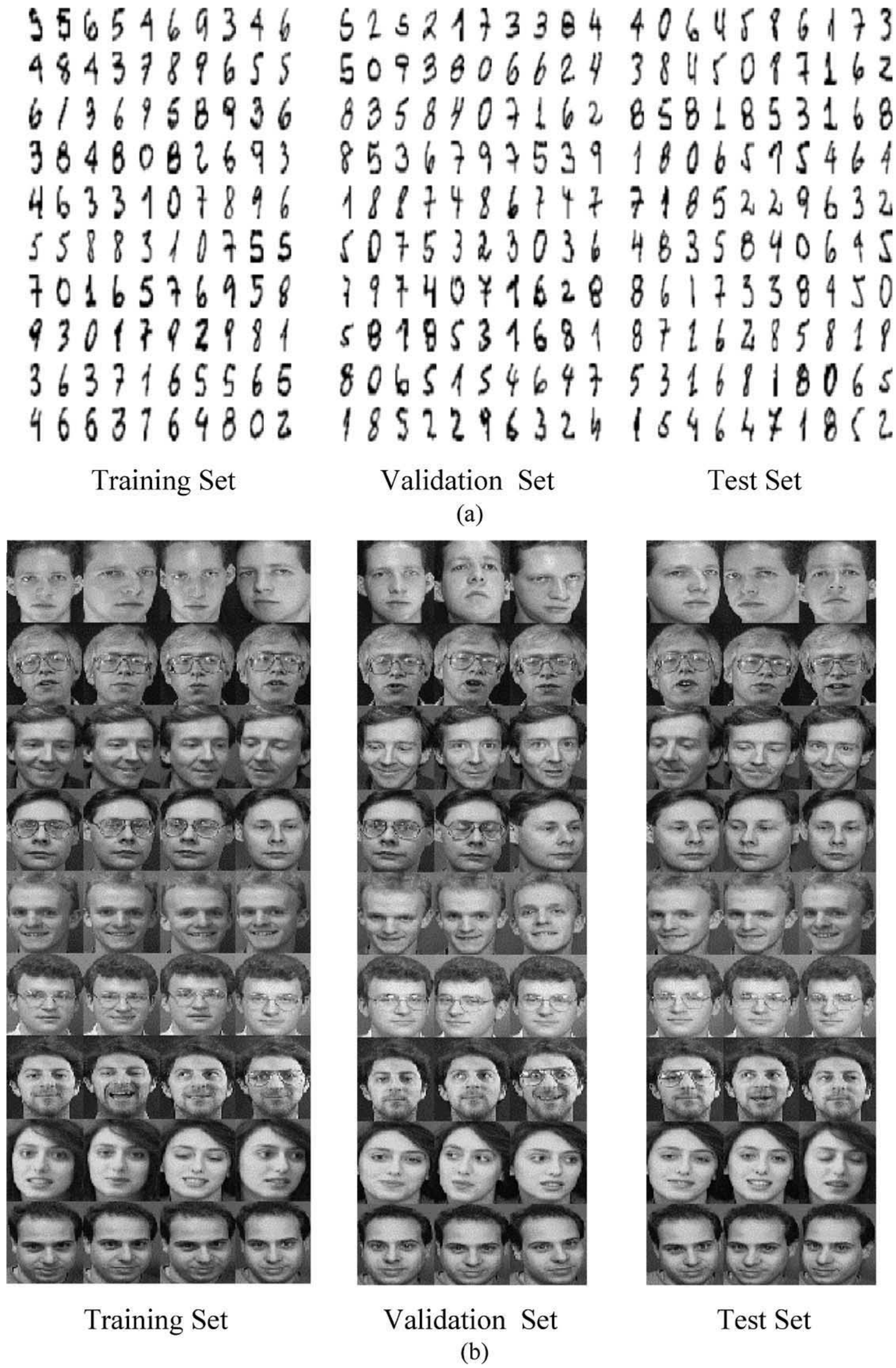


Fig. 1. (a) Samples of the handwritten-digit training, validation, and test sets. (b) Samples of the AT&T Laboratories, Cambridge, face database for the training, validation, and test sets.

individuals) databases [14], [23]. Therefore, to assess results of the new classification methods, a comparison to an international database or to a standard method is usually recommended.

The objective of the present study is to compare a linear system and a neural-network approach to modeling in 2-D pattern-recognition problems. It is shown that by introducing nonlinear operations among the inputs to the linear classifier, its performance can be enhanced. Nonlinear inputs to the linear classifier are generated by computing different types of products among linear inputs. These nonlinear inputs form a candidate set, from which nonlinear inputs are selected, using a genetic algorithm (GA) to improve linear-classifier performance. A comparison is made to neural-network classifiers using linear inputs. The preliminary results of the method proposed in this paper were presented in [21] and [22].

II. Methods

A. Databases

The method mentioned above is applied to the problems of handwritten-digit classification and face recognition. A database composed of 3674 handwritten digits was randomly partitioned into three sets for training, validation, and testing. The partition randomly separates digits from different persons, but leaves all digits from the same person in a single partition. Therefore, from the generalization point of view, this form of partition represents one of the most difficult possible cases. The training set was composed of 1837 handwritten digits, the validation set was composed of 918 digits, and the test set was composed of 919 digits. Each handwritten digit is in 256 levels of gray, composed of 15 $\times$ 23 pixels, and normalized in size. This database is available in [11]. Fig. 1(a) shows handwritten-digit samples of the training, validation, and test sets.

The second database is the face database from AT&T Laboratories, Cambridge, U.K., where there are ten different images of each of the 40 distinct subjects. The size of each image is 92 $\times$ 112 pixels, with 256 gray levels per pixel. For some subjects, the images were taken at different times, varying the lighting, facial expressions (open/closed eyes, smiling/not smiling), and facial details (glasses/no glasses). All the images were taken against a dark homogeneous background with the subjects in an upright, frontal position (with tolerance for some side movement) [7]. The face database was partitioned into three sets for training (160 faces), validation (120 faces), and test (120 faces). Since there are ten frontal faces for each subject, four images are in the training set, three in the validation set, and three in the test set. Fig. 1(b) shows samples of faces from the training, validation, and test sets.

B. Classification Method

The inputs to the linear and nonlinear classifiers are gray-level images of handwritten digits and frontal faces. Each pixel is considered as an input to the classifier. In this paper, we will designate each pixel in its original form as a *linear input* to the classifier. Therefore, each digit is composed of $15 \times 23 = 345$ linear inputs to the classifier and each face is composed of $92 \times 112 = 10\,304$ linear inputs to the classifier. The classifier has ten outputs, one for each handwritten digit class (0–9), and forty outputs in the face recognition problem, since there are 40 subjects on the database.

C. Linear System Modeling

1) Linear Inputs in the Linear System Model: Each of the linear-model outputs is determined according to (1). The simplest model contemplates only linear inputs, i.e., only the handwritten-digit pixels $x_1, \ldots, x_{345}$ or the face pixels $x_1, \ldots, x_{10\,304}$ are inputs to the linear units:

$$\hat{y}_j = \sum_{i=1}^{q} a_{ji} x_i + b_j. \tag{1}$$

The model is adjusted on the training set by finding the a_{ji} and b_j. In matrix form, the model can be expressed as

$$\hat{y} = A \cdot x + b \tag{2}$$

where A is a 10 $\times$ 345 matrix and b is a 10 $\times$ 1 vector for the handwritten-digit-classification problem. A is a 40 $\times$ 10 304 matrix and b is a 40 $\times$ 1 vector for the face-recognition problem. The weights can be found by minimizing the quadratic error given by

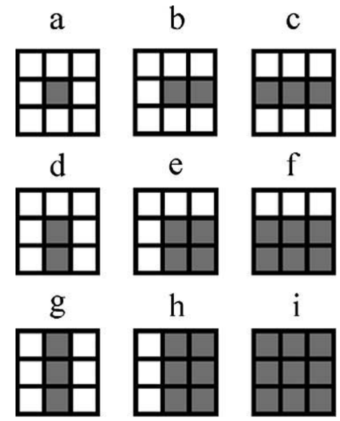


Fig. 2. Graphical representation of products among inputs for a 3 $\times$ 3 neighborhood. (a) No product, (b) 2 $\times$ 1 product, (c) 3 $\times$ 1 product, (d) 1 $\times$ 2 product, (e) 2 $\times$ 2 product, (f) 3 $\times$ 2 product, (g) 1 $\times$ 3 product, (h) 2 $\times$ 3 product, (i) 3 $\times$ 3 product.

$$E = \sum_{j=1}^{p} e_j^2 = \sum_{j=1}^{p} (y_j - \hat{y}_j)^2 \tag{3}$$

y_j are the known outputs of the model for the input pattern k, and $\hat{y}_j$ is the output of the model. Therefore, the error e_j depends on the response of the model to the kth pattern in the database. To determine the weights, the gradient-descent method is used [16], as stated in (4) and (5).

$$\Delta a_{ji} = -\eta \frac{\partial E}{\partial a_{ji}}. \tag{4}$$

Equations (1), (3), and (4) give

$$\Delta a_{ji} = -\eta e_j x_i. \tag{5}$$

This method is applied over the training set, online, with $\eta = 0.005$.

2) Nonlinear Inputs in the Linear-System Model: Nonlinear terms were introduced in the linear-system model by including products among inputs. Inputs are gray-level pixels from the input image. Only products among certain neighbors were considered, because they represent local spatial features that could be useful for classification. Products may represent lines, corners, or region detectors in any direction. For example, line-detector filters are found on cortical units in the visual system of mammals [9], [13]. Product terms are determined within a 3 $\times$ 3 pixel window. Any product among the pixels within the 3 $\times$ 3 window is possible. In Fig. 2, the graphical representation of products among pixels within a 3 $\times$ 3 window is shown, as used in [22]. In case (a), a single pixel at the center represents the no-product case or linear input. In the following, x_{ij} represents the center in each 3 $\times$ 3 neighborhood, and x_{ij}^{P} represents the corresponding result of products at (i, j). Case (b) corresponds to $x_{ij}^{\mathrm{P}} = x_{ij} x_{i+1j}$, which is a 2×1 product. In case (c), $x_{ij}^{\mathrm{P}} = x_{i-1j} x_{ij} x_{i+1j}$ corresponds to a 3×1 product. In case (d), $x_{ij}^{\mathrm{P}} = x_{ij} x_{ij-1}$ corresponds to a 1 $\times$ 2 product. In case (e), $x_{ij}^{\mathrm{P}} = x_{ij} x_{i+1j} x_{ij-1} x_{i+1j-1}$, which is a 2 $\times$ 2 product. In case (f), $x_{ij}^{\mathrm{P}} = x_{i-1j} x_{i-1j-1} x_{ij} x_{ij-1} x_{i+1j} x_{i+1j-1}$ corresponds to a 3 $\times$ 2 product. Case (g), $x_{ij}^{\mathrm{P}} = x_{ij+1} x_{ij} x_{ij-1}$, represents a 1 $\times$ 3 product. Case (h), $x_{ij}^{\mathrm{P}} = x_{ij+1} x_{ij} x_{ij-1} x_{i+1j+1} x_{i+1j} x_{i+1j-1}$,

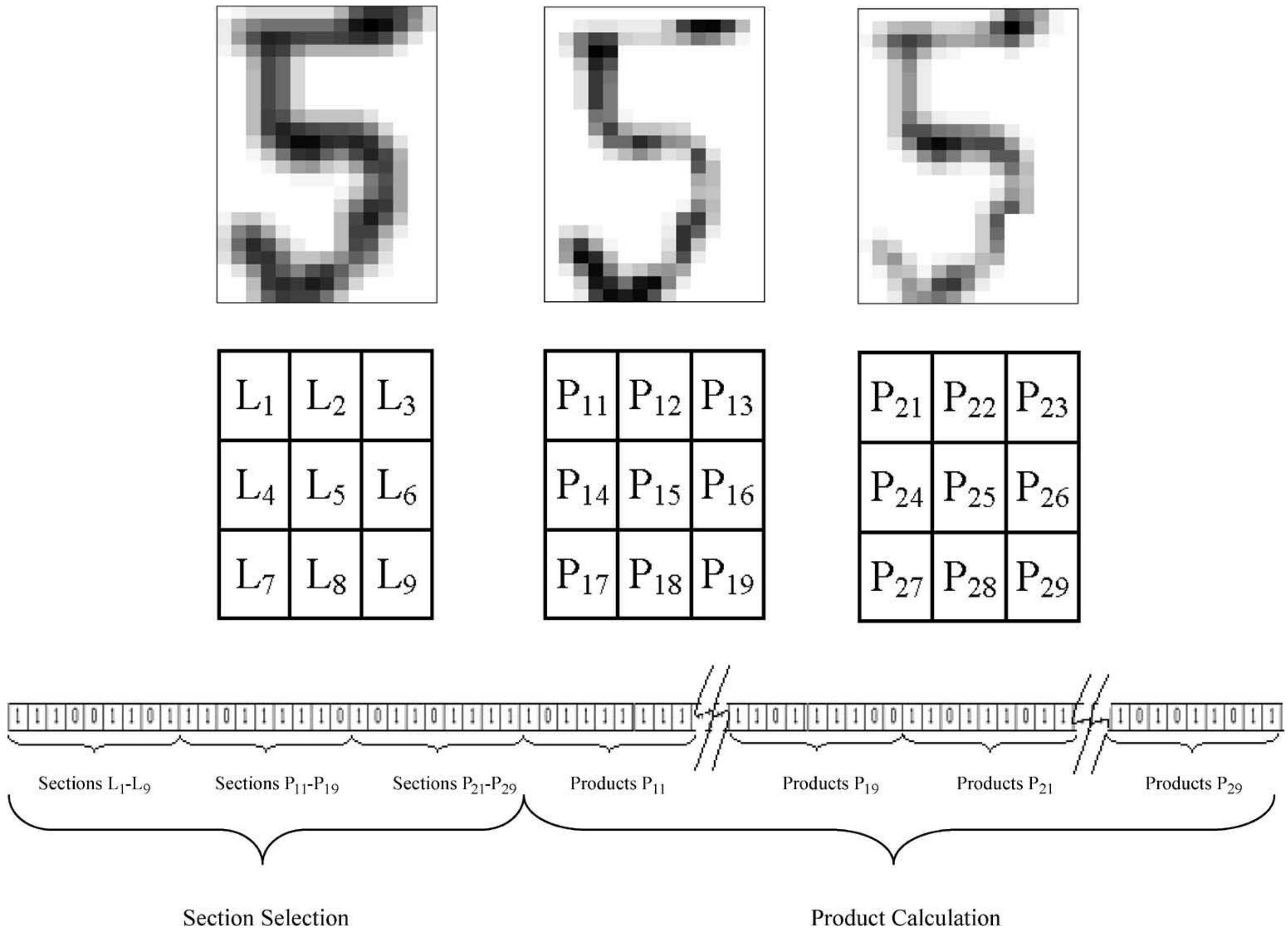


Fig. 3. Each image is divided into nine linear regions, $\mathrm{L}_1, \ldots, \mathrm{L}_9$. For each region, two different product terms are calculated originating images $\mathrm{P}_{11}, \ldots, \mathrm{P}_{19}$, and $\mathrm{P}_{21}, \ldots, \mathrm{P}_{29}$. The binary string represents the presence "1" or absence "0" of all possible sections as inputs to the linear classifier and encodes the products within each rectangular region.

TABLE I
SUMMARY OF THE PARAMETERS FOR THE GA

Number of Individuals per Population (N)	20
Crossover Probability ($P_{crossover}$)	0.5
Mutation Probability ($P_{mutation}$)	0.003

represents a 2 × 3 product. Case (i), $x_{ij}^{\mathrm{p}} = x_{i-1j+1}x_{i-1j}x_{i-1j-1} x_{ij+1}x_{ij}x_{ij-1}x_{i+1j+1}x_{i+1j}x_{i+1j-1}$, corresponds to a 3 × 3 product.

Each image is divided into nine rectangular regions $\mathrm{L}_1, \ldots, \mathrm{L}_9$ and two different product terms are calculated within each of these regions, creating two new images containing the products (see Fig. 3) $\mathrm{P}_{11}, \ldots, \mathrm{P}_{19}$ and $\mathrm{P}_{21}, \ldots, \mathrm{P}_{29}$. The selection of an appropriate set of linear inputs in regions $\mathrm{L}_1, \ldots, \mathrm{L}_9$ and the appropriate product terms in regions $\mathrm{P}_{11}, \ldots, \mathrm{P}_{19}$ and $\mathrm{P}_{21}, \ldots, \mathrm{P}_{29}$ are determined by a GA. Therefore, there are 18 different product terms for each image. The 18 different products allow the extraction of different features associated with spatial characteristics in each image. After the linear and nonlinear inputs have been selected by the GA, the weights of the linear classifier are adjusted with the training set. The generalization capacity of the linear model with linear and nonlinear inputs is measured on the test set, which was not used either to adjust the weights of the linear classifier or to select the nonlinear products by the GA.

3) Genetic Algorithm: A GA was used to select product terms, as well as linear terms, among the inputs. Each individual in a population represented a set of linear and nonlinear inputs to the linear classifier. The set of inputs was composed of three images divided into nine rectangular regions as illustrated in Fig. 3. The first image represents the linear inputs, and the second and third images represent nonlinear inputs in the form of product terms. Therefore, the selection is performed among 27 image regions. Within each region of the nonlinear inputs, a product term among pixels in a 3 × 3 window was defined. Therefore, a total of $9 \times 2 \times 3 \times 3 = 162$ bits are required to represent all product terms within the two images with nonlinear inputs. Therefore, each individual is encoded into $27 + 162 = 189$ bits, representing an image with linear terms and two images with nonlinear terms. An example of the string with binary coding is shown in Fig. 3.

TABLE II
NEURAL-CLASSIFIER TRAINING PARAMETERS

Parameter	Value
Number of Hidden Layers	1
Units in the Hidden Layer (handwritten digit problem)	1 to 120
Units in the Hidden Layer (face recognition problem)	1 to 130
Range of Initial Random Weights	[-0.5, 0.5]
Learning Rate	0.01
Momentum	0.2
Training Epochs	500

The fitness measurement is a number in the interval [0,1] representing the range between 0% and 100% recognition on the validation set.

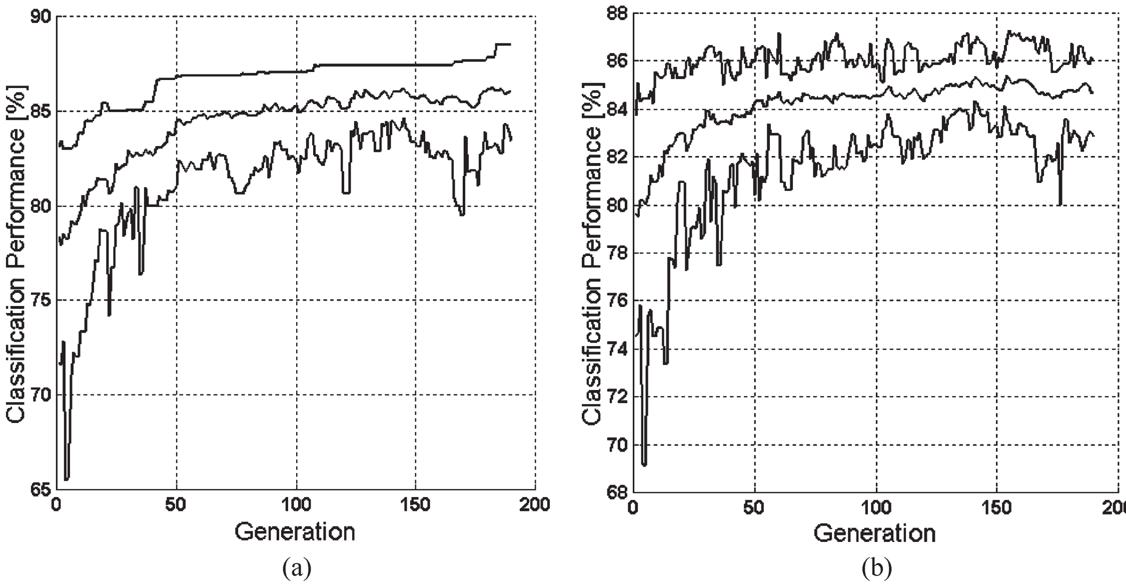


Fig. 4. Classification performance of the linear system model with a genetic selection of linear and nonlinear inputs for the handwritten classification problem: (a) validation set, and (b) test set. Best individual, average of the population, and worst individual.

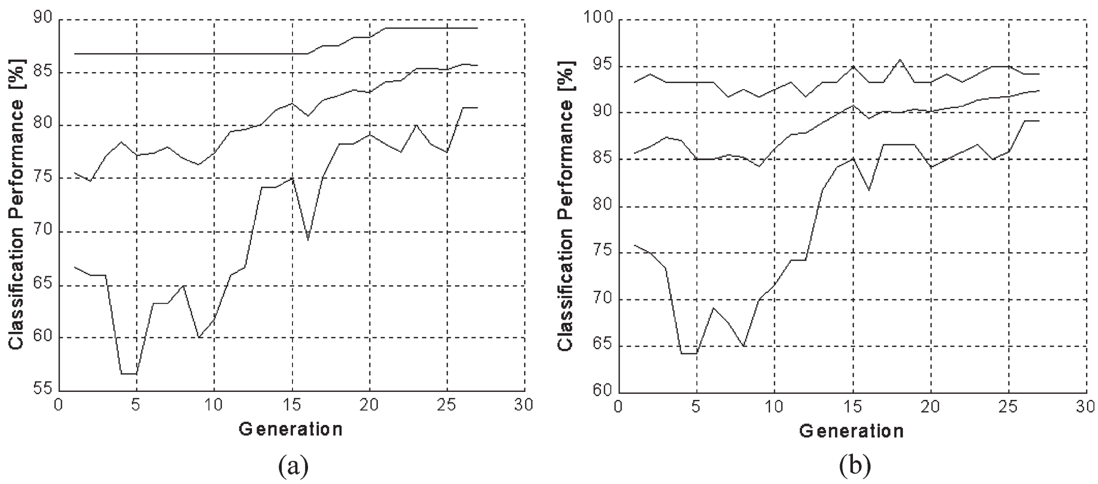


Fig. 5. Classification performance of the linear system model with a genetic selection of linear and nonlinear inputs for the face-recognition problem: (a) validation set, and (b) test set. Best individual, average of the population, and worst individual.

The GA uses proportional selection [10] and the stochastic universal sampling method [19] to assign an individual of the current generation a probability to be chosen as an individual of the next generation. Uniform crossover and mutation [19] are performed on each population after the selection and sampling processes. Table I shows a summary of the GA parameters employed on the simulations.

D. Neural-Network Modeling

Neural-network models were implemented for single-layer perceptron and multilayer feed-forward neural networks with one hidden layer. Units in these models are nonlinear because of the sigmoidal function.

1) Single-Layer Perceptron: The perceptron model with a single layer of nonlinear units and linear inputs for the ten classes differ from the linear system model in the nonlinear sigmoidal function. The equation for each of the outputs of the model is given by

$$\hat{y}_j = \tanh\left(\sum_{i=1}^{345} a_{ji}x_i + b_j\right) \tag{6}$$

and in matrix form

$$\hat{y} = \tanh(A \cdot x + b). \tag{7}$$

In (7) $\tanh$ is applied over each of the elements of vector $\hat{y} = [\hat{y}_1 \ \dots \ \hat{y}_\mathrm{p}]^\mathrm{T}$ and $x = [x_1 \ \dots \ x_k]^\mathrm{T}$.

2) Multilayer Perceptron: The equations in matrix form for a multilayer perceptron with one hidden layer, with layers $\hat{y}_{l1}$ and $\hat{y}_{l2}$ are given by

$$\hat{y}_{l1} = \tanh(A_1 \cdot x + b_1) \tag{8}$$

$$\hat{y}_{l2} = \tanh(A_2 \cdot \hat{y}_1 + b_2) \tag{9}$$

$$\hat{y}_{l2} = \tanh\left(A_2 \cdot \tanh(A_1 \cdot x + b_1) + b_2\right) \tag{10}$$

where $\tanh$ is applied over each component of the vector. Training was performed by backpropagation with momentum and bias [16]. Table II shows a summary of the neural-classifier training parameters employed on the simulations.

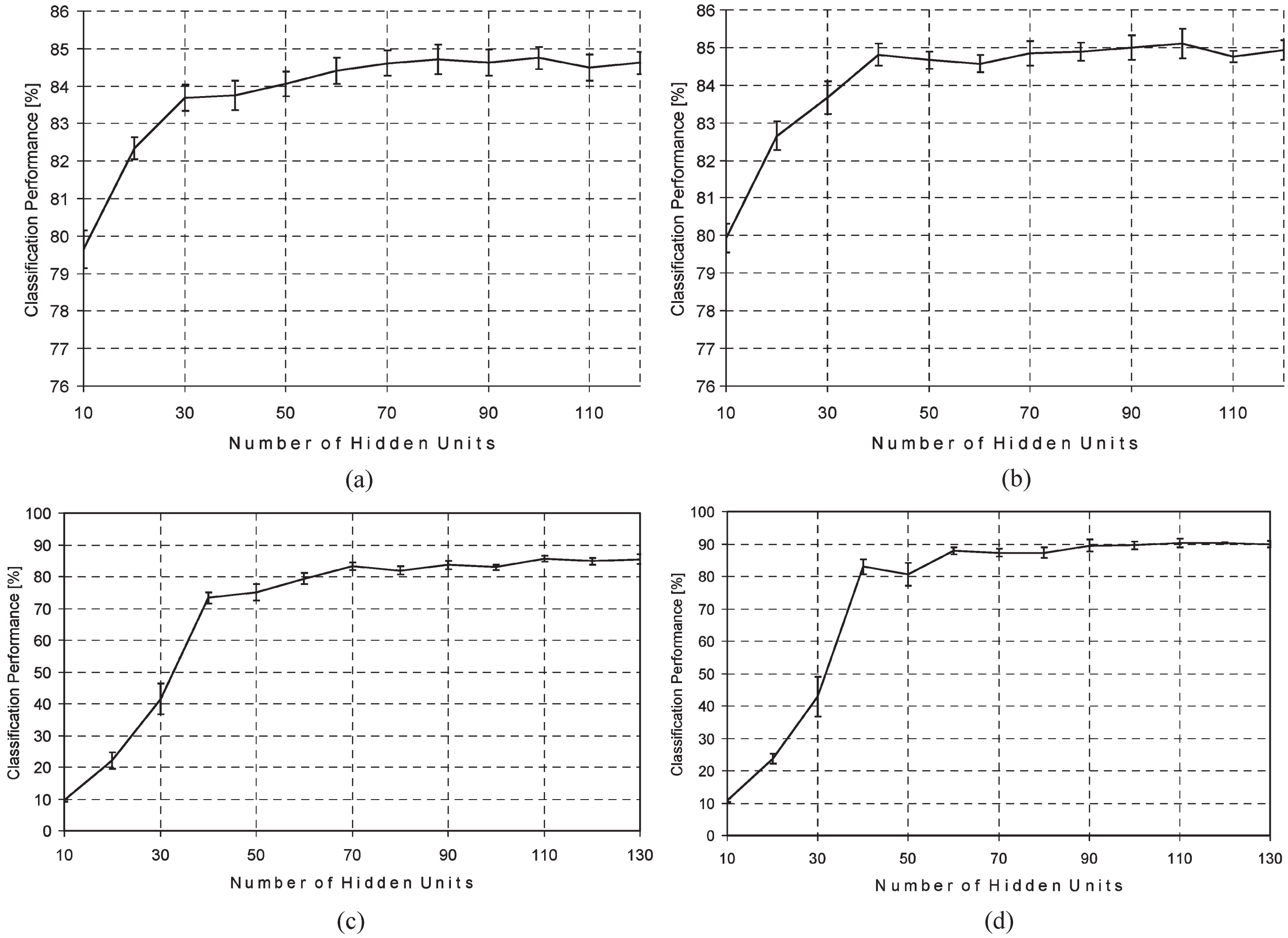


Fig. 6. Percentage of the validation and test sets correctly classified by a one-hidden-layer perceptron as a function of the number of hidden units. (a) and (b) Handwritten-digit problem. (c) and (d): Face-recognition problem.

E. Summary of Simulations Performed in Linear and Nonlinear Modeling

The simulations performed for the linear-system model and the nonlinear neural-network models were the following:

1) Linear system model with only linear inputs ($q = 345$ inputs and $p = 10$ outputs for handwritten digits and $q = 10\,304$ inputs and $p = 40$ outputs for faces).
2) Linear system model with genetic selection of linear and nonlinear inputs. The GA selects rectangular regions from the linear input image, as well as from the pair of nonlinear products in each of the nine sectors of the image. A maximum number of inputs for the case of all sections selected by the GA include 1035 inputs in the handwritten-digits problem and 30 912 for the face-recognition problem.
3) Single layer perceptron with linear inputs.
4) One-hidden-layer perceptron with linear inputs. The number of hidden units varied from 1 to 120 for the handwritten-digit-classification problem and from 1 to 130 for the face-recognition problem.

In the case of the neural models, each simulation was performed at least ten times with different starting sets of initial weights to avoid local minima.

III. Results

The linear-system model with linear inputs in the handwritten-classification problem (345 inputs) yielded a classification performance of 74.2% on the validation set and 79.0% on the test set. The face-recognition problem (10 304 inputs), on the other hand, reached a classification performance of 77.5% on the validation set and 87.5% on the test set.

Results of the linear-system model with genetic selection of linear and nonlinear inputs in the handwritten-digit-classification problem are shown in Fig. 4. In this figure, the recognition rates on (a) the validation set, and (b) the test set, are plotted as a function of the generation number. The three curves shown in each graph correspond to the classification performance of: (top) the best individual in each population; (middle) the average of the population; and (bottom) the worst individual in each population. The best classification performance reached 87.9% on the validation set and 87.3% on the test set. Therefore, the classification performance improved on the validation and test sets by 13.7 and 8.3 percentage points, respectively, relative to the case with only linear inputs.

In the face-recognition problem, Fig. 5 shows the linear-system recognition rate on the (a) validation, and (b) test sets as a function of the generation number when the GA selects linear and nonlinear inputs to the linear model. The three curves shown in each graph correspond to the classification performance of: (top) the best individual in each

TABLE III

COMPARISON OF BEST CLASSIFICATION PERFORMANCE FOR THE LINEAR MODEL WITH LINEAR INPUTS, LINEAR MODEL WITH GENETICALLY SELECTED INPUTS (LINEAR AND NONLINEAR), SINGLE LAYER PERCEPTRON AND ONE-HIDDEN-LAYER PERCEPTRON WITH LINEAR INPUTS. RESULTS ARE SHOWN FOR THE VALIDATION AND TEST SETS FOR THE HANDWRITTEN-DIGIT CLASSIFICATION AND FACE-RECOGNITION PROBLEMS

		Linear Model With Linear Inputs [%]	Linear Model With Genetic Selected Inputs (linear and non-linear) [%]	Single Layer Perceptron With Linear Inputs [%]	One-Hidden Layer Perceptron with Linear Inputs [%]
Handwritten Digit Database	Validation Set	74.2	87.9	80.5	84.8
	Test Set	79.0	87.3	80.9	85.1
AT&T Cambridge Face Database	Validation Set	77.5	89.2	85.8	87.5
	Test Set	87.5	95.8	90.0	90.8

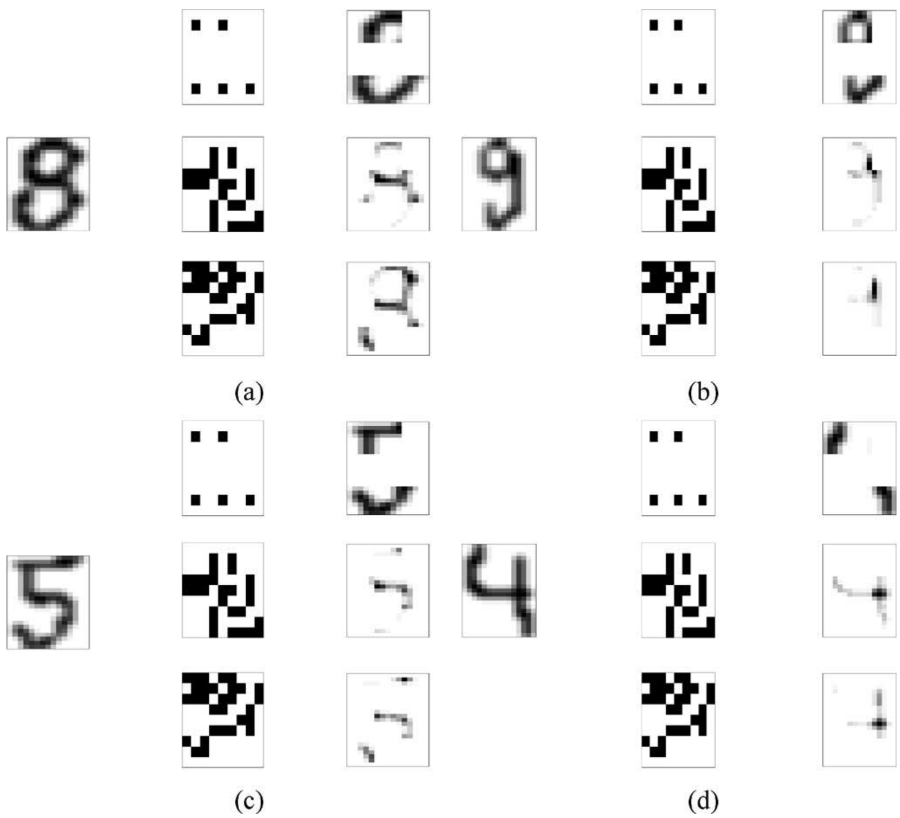
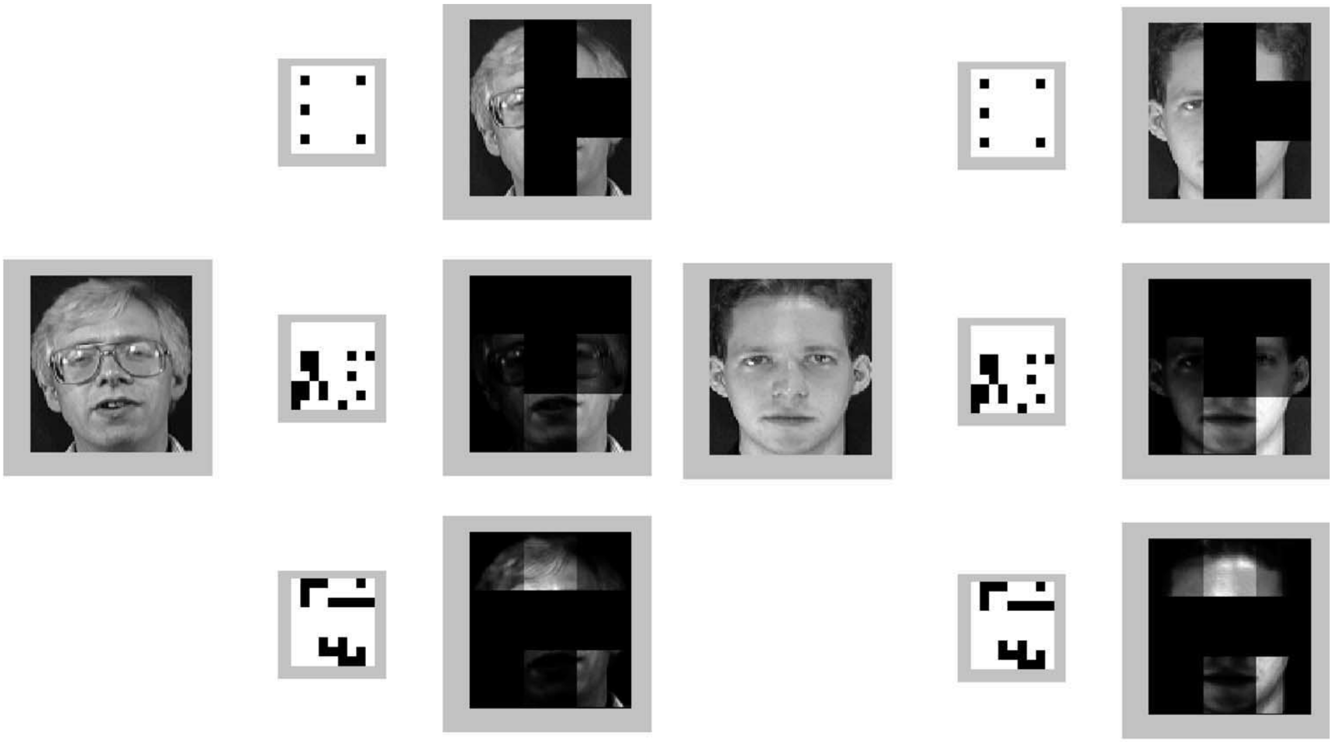


Fig. 7. Handwritten digits: (a) 8, (b) 9, (c) 5, and (d) 4, as linear and nonlinear inputs to the linear system (right column). On the column in the middle are the selected type of filtering: linear (top) and nonlinear (middle and bottom).

population; (middle) the average of the population; and (bottom) the worst individual in each population. The best recognition rate reached 89.2% on the validation set and 95.8% on the test set. Therefore, the classification performance improved on the validation and test sets by 11.7 and 8.3 percentage points, respectively, relative to the case with only linear inputs.

Results of the single-layer perceptron model with linear inputs in the problem of handwritten-digit classification reached a classification performance of 80.5% on the validation set and 80.9% on the test set. In the problem of face recognition, the single layer perceptron reached a classification performance of 85.8% on the validation set and 90.0% on the test set. In both problems, the single-layer perceptron model with linear inputs reached higher classification performance than those obtained by the linear-system model with linear inputs. Nevertheless, the single-layer perceptron reached lower classification performance than the linear model with combined inputs (linear and nonlinear) selected by the GA.

Results of the classification performance of the one-hidden-layer perceptron with linear inputs as a function of the number of hidden units are shown in Fig. 6 for the (a) validation set and (b) test set in the handwriting-classification problem. Results show the average classification performance and the standard deviation for ten simulations. The number of hidden units was varied in each hidden layer from 1 to 120, in steps of 10. It is observed that the classification performance improves for a larger number of hidden units until 40 hidden units, and thereafter, differences in classification performance are no longer statistically significant for a higher number of hidden units. The maximum classification performance for the one-hidden-layer perceptron reached 84.8% and 85.1% for the validation and test sets for 100 hidden units.



(a) (b)

Fig. 8. Two faces showing the linear and nonlinear inputs to the linear system (right column). On the column in the middle are the selected type of filtering: linear (top) and nonlinear (middle and bottom).

TABLE IV
NUMBER OF ADJUSTABLE PARAMETERS EMPLOYED IN EACH MODEL

Problem	Linear Model With Linear Inputs [N^o of parameters]	Linear Model With Genetic Selected Inputs (linear and non-linear) [N^o of parameters]	Single Layer Perceptron With Linear Inputs [N^o of parameters]	One-Hidden Layer Perceptron With Linear Inputs [N^o of parameters]
Digit Classification	3450	6500	3450	14 200
Face recognition	412 160	687 480	412 160	1 034 400

For the one-hidden-layer perceptron considering only linear inputs, Fig. 6(c) and (d) shows the classification performance in the face-recognition problem as a function of the number of hidden units. The average recognition rate and standard deviation after ten simulations are shown for each number of hidden units. The number of hidden units was varied in each hidden layer from 1 to 130 in steps of 10 units. The maximum recognition rate reached 87.5% for the validation set and 90.8% for the test set. Although the number of inputs is large for the nonlinear classifier, the training converged to solutions in all simulations. This can be observed from the generalization results shown in Fig. 6, where the standard deviation for the ten simulations computed for each number of hidden units was restricted to around 1%, specially in the case of a large number of hidden units. This result shows that the solution found after training did not reach a local minimum.

Table III shows a summary of results for the linear model with linear inputs, the linear model with genetically selected linear and nonlinear inputs, the single layer perceptron with linear inputs, and the one-hidden-layer perceptron with linear inputs. It is observed that a linear classifier with an appropriate set of nonlinear inputs reached a classification performance similar or better than those obtained by nonlinear classifiers, such as single-layer perceptron or one-hidden-layer perceptron with linear inputs.

Fig. 7 shows four examples of handwritten digits as a linear input to the system: a) digit eight; b) digit nine; c) digit five; and d) digit four. The three images in the middle column with nine sections for each one identify the best combination of linear (top) and nonlinear (middle and bottom) inputs selected by the GA. The three images on the right column represent the result of applying these linear and nonlinear filters over the input image. These are the new inputs to the linear classifier that improve its classification performance. As observed in Fig. 7, several of the products selected by the GA correspond to products among pixels in the vertical or horizontal direction. These operations can be interpreted as feature extraction in the vertical and horizontal directions. Also, other features, such as corners, are also selected as shown in Fig. 7. It is interesting to see that these features are associated with certain positions. For example, in Fig. 7, a corner is selected in the center of the digit. In Fig. 8, a horizontal product was also selected in the region of the left eye and feature detectors of the corner type were selected in the right forehead. This feature allows differentiation among faces with/without hair covering the right forehead. Also a "U"-type feature was selected in Fig. 8, in the region of the mouth that allows emphasis on certain mouth shapes.

Fig. 8 shows two faces as linear inputs to the system. The three images in the middle column with nine sections each identify the best combination of linear (top) and nonlinear (middle and bottom) inputs selected by the GA. The three images on the right column represent the results of applying these filters over the input image. These are the new inputs to the linear classifier that improve its classification performance.

The total number of inputs to the linear-system model selected by the GA is 650 (185 linear and 475 nonlinear) for the handwritten-digit problem and 17 187 (5766 linear and 11 421 nonlinear) for the face-recognition problem. The linear model with linear and nonlinear inputs employs 650×10 and $17\,187 \times 40$ parameters in each problem, respectively. The neural network models (single-layer perception and one-hidden-layer perception) were evaluated with linear inputs only, i.e., 345 inputs for the handwritten-digit-classification problem and 10 304 inputs for the face-recognition problem. In this case, the single-layer perceptron model requires 345×10 and $10\,304 \times 40$ parameters in each problem, respectively. For the handwritten-digit-classification problem, the number of adjustable parameters in the one-hidden-layer perceptron model is the weights between the input units $(15 \times 23 = 345)$ and the hidden units (40) plus the weights between the hidden units (40) and the output units (10). Thus, for the handwritten-digit-classification problem, there are a total of $345 \times 40 + 40 \times 10$ adjustable parameters. For the face-recognition problem, the number of adjustable parameters in the one-hidden-layer perceptron model is the weights between the input units $(92 \times 112 = 10\,304)$ and the hidden units (100) plus the weights between the hidden units (100) and the output units (40). Thus, for the face-recognition problem, there are a total of $10\,304 \times 100 + 100 \times 40$ adjustable parameters.

In addition, the linear model was also evaluated with linear inputs only. The total number of adjustable parameters in the linear model is 345×10 in the handwritten-digit problem and $10\,304 \times 40$ in the face-recognition problem. Table IV shows a summary of the total number of adjustable parameters in each model for both problems.

The main objective of this research is to show that by increasing the input space in a linear classifier by adding nonlinear inputs, the classification performance of the linear classifier is significantly improved to the level of a nonlinear classifier. Two applications in 2-D pattern recognition are used to show this result. In the handwritten-digit-recognition problem, ten classes are required. In the face-recognition problem, a database with forty different individuals was used, and from this point of view, the method is applicable to a restricted domain in the number of classes. However, face recognition in a restricted domain may be useful in security applications, where only a limited number of persons are allowed into a facility. Scalability to larger number of classes is not shown in this paper. Although this maybe viewed as a restriction, according to the non-free-lunch theorem, it is not possible to find an algorithm to solve all possible problems that is superior to any competitor [25], therefore all algorithms are restricted in their application domain. From another point of view, the computation of the parameters required by the linear classifier is the result of solving a linear equation system and therefore, scalability to a larger number of classes is a linear problem. This may be considered an advantage relative to the nonlinear classifier. Nevertheless, the total number of possible products may increase exponentially with the number of classes, but the GA performs a selection among them. A similar approach involving product terms is employed in support vector machines (SVMs), where kernels are used to produce a separation among classes, allowing a linear classification [24].

Computation time was measured with a Pentium IV, 1.5-GHz computer, with the code written in C++. In the handwritten-digit-recognition problem, the classification time for a single digit was 0.63 ms in the linear model. This time includes computations for the nonlinear inputs to the linear model. In the case of the perceptron model, the classification time for one digit was 1.1 ms. In the case of the face-recognition problem, the recognition time for a single face image was 50 ms in the linear model. This time also includes the time required to compute the nonlinear inputs to the linear classifier. The perceptron model took 75 ms to recognize a single face. In both problems, recognition time is shorter for the linear system, which may be an advantage relative to the nonlinear classifier. The genetic selection of linear and nonlinear inputs to the classifier takes 200 min per generation in the handwriting-classification problem and 910 min for the face-recognition problem in a Pentium IV, 1.5-GHz computer. Nevertheless, for a given problem, the genetic selection is performed only once, in offline mode. After the linear and nonlinear inputs have been determined by the GA, the time required for classification in online mode is 0.63 ms for the handwritten-digit problem and 50 ms for the face-recognition problem. These computations include the time required to compute the nonlinear inputs to the linear classifier.

IV. Conclusion

A comparison was performed among a linear-system model and two nonlinear neural-network models in the problems of handwritten-digit classification and face recognition. In both problems, results showed that the linear-classifier performance could be enhanced by selecting, as a new set of inputs, an appropriate set of linear and nonlinear inputs. This selection was performed using a GA. Nonlinear inputs were generated using different forms of generating products among linear inputs. Nonlinear inputs were generated as products among groups of pixels within a 3×3 window. A total of 18 different product terms were allowed for each one associated to a region of the image.

After the selection of an appropriate set of linear and nonlinear inputs, the classification performance of the linear classifier, on the test set, improved from 79.0% to 87.3% in the handwritten-digit-classification problem and from 87.5% to 95.8% in the face-recognition problem. A single-layer perceptron with linear inputs reached 80.9% and 90.0% of correct classification on the test sets in both problems, respectively. A multilayer perceptron model with one hidden layer reached the best classification performance of 85.1% on the handwritten-digit-classification problem and 90.8% on the face-recognition-problem on the same test sets.

The products among inputs can be interpreted as nonlinear filtering over the input image to detect features such as lines in different directions, producing an overall improvement in the classification performance of the models. These nonlinear filters improve the linear separation among classes because the classification performance of the linear classifier improved when these inputs were selected as new inputs.

The linear classifier with linear and nonlinear inputs shows similar or better results than the nonlinear classifier. The nonlinearities introduced at the input of the linear system replace advantageously the nonlinearities introduced by the nonlinear units in the neural-network model. The linear and nonlinear inputs to the linear model were selected by the GA, and therefore, the amount of information processed by the linear model was reduced. In the nonlinear model, the nonlinearities are introduced by every unit and no selection takes place. Moreover, the total number of parameters of the linear model is lower than that of the nonlinear model. The nonlinear neural model provides a solution, but this may not be the best one, and the results could be improved on, for example, by pruning. SVMs take a similar approach by mapping the training data into a higher dimensional feature space, where linear separation is possible with high performance [24]. SVM has also used product features to map the input space into a linear separable one [24]. As proposed in this paper, the nonlinear terms produced as products among inputs are good feature extractors because information in images is highly correlated.

Advantages of using a linear classifier instead of a nonlinear one include a significant reduction in training time, online propagation time, and required computational resources such as memory. In future research, different optimization algorithms may be employed and compared in selecting linear and nonlinear inputs to the linear classifier. Besides GAs, simulated annealing, tabu search, and mutual information may be used to select the inputs to the linear classifier.

ACKNOWLEDGMENT

The authors wish to thank Research Assistant C. Salinas for his initial contribution to this work. The authors would like to thank the reviewers for their comments.

REFERENCES

[1] C. M. Bishop, *Neural Networks for Pattern Recognition*. Oxford, U.K.: Oxford Univ. Press, 1995.
[2] A. Casali, G. Gonzalez, G. Vallebuona, C. Perez, and R. Vargas, "Grindability soft-sensors based on lithological composition and on-line measurements," *Miner. Eng.*, vol. 14, no. 7, pp. 689–700, 2001.
[3] K. F. Chan and D. Y. Yeung, "Recognizing on-line handwritten alphanumeric characters through flexible structural matching," *Pattern Recogn.*, vol. 32, no. 7, pp. 1099–1114, 1999.
[4] Z. Chi, J. Wu, and H. Yan, "Handwritten numeral recognition using self-organizing maps and fuzzy rules," *Pattern Recogn.*, vol. 28, no. 1, pp. 59–66, 1995.
[5] S. B. Cho, "Neural-network classifiers for recognizing totally unconstrained handwritten numerals," *IEEE Trans. Neural Netw.*, vol. 8, no. 1, pp. 43–53, Jan. 1997.
[6] R. W. Conners, C. W. McMillin, K. Lin, and R. E. Vasquez-Espinosa, "Identifying and locating surface defects in wood: Part of an automated lumber processing system," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 5, no. 6, pp. 573–583, Nov. 1983.
[7] (2002). *Face Database*. Cambridge, U.K.: AT&T Laboratories [Online]. Available: http://www.uk.research.att.com/facedatabase.html
[8] G. C. Feng and P. C. Yuen, "Recognition of head-and-shoulder face image using virtual frontal-view image," *IEEE Trans. Syst. Man Cybern., Part A, Syst. Humans*, vol. 30, no. 6, pp. 871–883, Nov. 2000.
[9] K. Fukushima, "Neocognitron: A hierarchical neural network capable of visual pattern recognition," *Neural Netw.*, vol. 1, no. 2, pp. 119–130, 1988.
[10] D. Goldberg, *Genetic Algorithms in Search, Optimization & Machine Learning*. Reading, MA: Addison-Wesley, 1989.
[11] (2002). Handwritten Digit Database. [Online]. Available: http://alcatraz.die.uchile.cl/handwritten_digit.htm
[12] S. Haykin, "Self-organizing systems II: Competitive learning," in *Neural Networks: A Comprehensive Foundations*, J. Griffin, Ed. New York: IEEE Press/Macmillan, 1994, pp. 408–430.
[13] D. Hubel, "The primary visual cortex," in *Eye, Brain and Vision.* New York: Scientific Amer. Library, Series No. 22, 1995, pp. 59–91.
[14] S. Lawrence, C. L. Giles, A. C. Tsoi, and A. D. Back, "Face recognition: A convolutional neural-network approach," *IEEE Trans. Neural Netw.*, vol. 8, no. 1, pp. 98–113, Jan. 1997.
[15] D. S. Lee, S. N. Srihari, and T. Pawlicki, "Experiments with neural network models for handwritten digit recognition," in *Systems and Signal Processing*, R. N. Madan, N. Viswanadham, and R. L. Kashyap, Eds. New Delhi, India: Oxford and IBH Publishing, 1991, pp. 757–774.
[16] R. P. Lippmann, "An introduction to computing with neural nets," *IEEE ASSP Mag.*, vol. 4, no. 2, pp. 4–22, Apr. 1987.
[17] C. G. Looney, *Pattern Recognition Using Neural Networks: Theory and Algorithms for Engineers and Scientists*. New York: Oxford Univ. Press, 1997.
[18] G. Mayraz and G. E. Hinton, "Recognizing hand-written digits using hierarchical products of experts," *Advances in Neural Information Processing Systems*, vol. 13. Cambridge, MA: MIT Press, 2001.
[19] M. Mitchell, "Implementing a genetic algorithm," in *An Introduction to Genetic Algorithms*. Cambridge, MA: MIT Press, 1996, pp. 155–178.
[20] C. A. Perez, C. A. Salinas, P. A. Estévez, and P. M. Valenzuela, "Genetic design of biologically inspired receptive fields for neural pattern recognition," *IEEE Trans. Syst. Man Cybern., Part B*, vol. 33, no. 2, pp. 258–270, Apr. 2003.
[21] C. A. Perez, G. D. Gonzalez, and C. Salinas, "Genetic selection of non-linear product terms in the inputs to a linear classifier for handwritten digit recognition," in *IEEE Int. Conf. Systems, Man & Cybernetics*, Tucson, AZ, Oct. 2001, pp. 2337–2342.
[22] C. A. Perez, G. Gonzalez, and C. Salinas, "Neural versus difference equation modeling for 2D pattern recognition problems," in *IEEE Int. Conf. Systems, Man & Cybernetics*, Nashville, TN, Oct. 8–11, 2000, pp. 2851–2856.
[23] P. J. Phillips, H. Wechsler, J. Huang, and P. J. Rauss, "The FERET database and evaluation procedure for face-recognition algorithms," *Image Vis. Comput.*, vol. 16, no. 5, pp. 295–306, 1998.
[24] B. Scholkopf and A. J. Smola, *"Kernels" in Learning with Kernels: Support Vector Machines, Regularization, Optimization and Beyond*. Cambridge, MA: MIT Press, 2002, pp. 25–60.
[25] H. P. Schwefel, "Advantages of evolutionary computation over other approaches," in *Evolutionary Computation 1: Basic Algorithms and Operators*, T. Back, D. B. Fogel, and T. Michalewicz, Eds. Philadelphia, PA: Inst. of Physics Publishing, 2000, pp. 20–22.
[26] B. Zhang, M. Fu, and H. Yan, "A non-linear neural network model of mixture of local principal component analysis: Application to handwritten digits recognition," *Pattern Recogn.*, vol. 34, no. 2, pp. 203–214, 2001.

Intermediating User–DSS Interaction With Autonomous Agents

Rustam Vahidov

***Abstract*—The use of advanced decision support system (DSS) capabilities is hampered by the inadequacy of a "toolbox" organization of DSS from the user's perspective. In such a setup, the user is assumed to have all the knowledge and skills necessary to appropriately use the tools provided by the system in the decision-making process. This paper proposes a model for the use of autonomous agents as intermediaries between the users and the system. The model is organized around the human problem-solving process. The paper elaborates on the types of intermediary agents and the architecture for a DSS. The approach is illustrated using the prototype for an investment DSS.**

***Index Terms*—Decision support systems, human problem solving, software agents.**

I. INTRODUCTION

The advance of the digital era brought about new challenges and opportunities for organizational and individual decision makers. The economic globalization processes, the increased complexity of the competitive environments, the flattening of organizational structures, and the explosive growth of potentially useful information on the Internet complicate significantly and place a heightened pressure on managerial decision making. It is therefore critical to many businesses to have adequate means in place for transforming the vast volumes of data into information relevant for decision making.

Decision support systems (DSSs) have traditionally been a tool for supporting managerial decision making [1]. A recent study of academic articles suggests that DSS has been one of the most popular fields in the past [2]. However, a closer look reveals that the interest in DSS (as manifested by the number of publications) has been somewhat waning recently. In our view, this observation is a bit paradoxical:

Manuscript received December 20, 2001; revised May 2, 2003. This paper was recommended by Associate Editor L. Fang.

The author is with the Department of Decision Sciences and MIS, John Molson School of Business, Concordia University, Montreal, QC H3G 1M8, Canada (e-mail: rvahidov@jmsb.concordia.ca).

Digital Object Identifier 10.1109/TSMCA.2005.851292