About the author - Alice J. O'Toole received a B.A. in Psychology and Philosophy from The Catholic University of America in Washington, D.C. in 1983. She received an M.S. and a Ph.D. from Brown University in 1985 and 1988, respectively. She completed a 2 year postdoctoral fellowship at the Université de Bourgogne and the Ecole Nationale Supérieure des Télécommunications, France. She is presently an assistant professor in the School of Human Development at The University of Texas at Dallas. Her research interests include human memory for faces, computational models of face perception and stereoscopic vision.

About the author - Garrison Cottrell received a BS in Mathematics and Sociology in 1972 and a MAT in Mathematics Education in 1975 at Cornell University, Ithaca, NY. He received an M.S. in 1981 and a Ph.D. in 1985 in Computer Science at the University of Rochester. He was an NIMH post-doctoral fellow at the Center for Human Information Processing at the University of California at San Diego (UCSD) between 1985-87 and is currently an associate professor in the Department of Computer Science at UCSD. His research interests include connectionist models of cognitive processes, especially language processing and procedural learning, parallel distributed processing, image compression, modeling of invertebrate circuits, and non-monotonic reasoning.

nent analysis using autoassociative neural networks, *AIChE J.* **37**, 233–243 (1991).

51. E. Oja, data compression, feature extraction, and autoassociation in feedforward neural networks, *Artificial neural networks*, T. Kohonen, O. Simula and J. Kangas, eds., pp. 737–745, Elsevier Science, New York (1991).
52. S. Usui, S. Nakauchi and M. Nakano, Internal color representation acquired by a five-layer neural network, *Artificial Neural Networks*, T. Kohonen, O. Simula and J. Kangas, eds., pp. 867–872, Elsevier Science, New York (1991).
53. T. Poggio and S. Edelman, A network that learns to recognize three-dimensional objects, *Nature*, **343**, 263–266 (1990).
54. R. Brunelli and T. Poggio, Face recognition: Features versus templates, *IEEE transactions on P.A.M.I*, **15**, 1042–1052 (1993).
55. J. Buhmann, J. Lange and C. v. d. Malsburg, Distortion invariant object recognition by matching hierarchically labeled graphs, *Proc. Int. Conf. on Neural Networks*, Vol. I, pp. 155–159, Washington, (1989).
56. J. Buhmann, J. Lange, C. v.d. Malsburg, J.C. Vorbrüggen and R.P. Würtz, Object recognition with Gabor functions in the dynamic link architecture — Parallel implementation on a transputer network, *Neural Networks for Signal Processing*, B. Kosko, Ed., pp. 121–159. Englewood Cliffs, NJ: Prentice Hall, (1991).
57. M. Lades, J.C. Vorbruggen, J. Buhmann, J. Lange, C. v.d. Malsburg, R.P Wurtz and W. Konen, Distortion invariant object recognition in the dynamic link architecture, *IEEE Transactions on Comput.* **3**, 300–311 (1993).
58. A.L. Yuille, Deformable templates for face recognition, *J. Cognitive Neurosci.* **3**, 59–70 (1991).
59. A.L. Yuille and P.W. Hallinan. Deformable templates, *Active Vision*, A. Blake and A.L. Yuille eds., MIT Press, Cambridge (1992).
60. R. McKendall and M. Mintz, Robust fusion of location information. Preprint. Dept. of Computer and Info. Sci., Univ. of Pennsylvania (1989).
61. P.W. Hallinan, Recognizing human eyes, *SPIE Geometric Method Comput. Vision* **1570**, 214–226 (1991).

About the author - Dominique Valentin received an M.S. degree in Psychology in 1986, from the Université of Bourgogne (France). She is currently a Ph.D. student in the School of Human Development at The University of Texas at Dallas under the supervision of the second author. Her research interests include human memory for faces and computational models of face perception.

About the author - Hervé Abdi received an M.S. in Psychology from the University of Franche-Comté (France) in 1975, an M.S. (D.E.A.) in Economics from the University of Clermond-Ferrand (France) in 1976, an M.S. (D.E.A.) in Neurology from the University Louis Pasteur in Strasbourg (France) in 1977, and a Ph.D. in Mathematical Psychology from the University of Aix-en-Provence (France) in 1980. He was an assistant professor in the University of Franche-Comté (France) in 1979, an associate professor in the University of Bourgogne at Dijon (France) in 1983, a full professor in the University of Bourgogne at Dijon (France) in 1988. He is currently an associate professor in the School of Human Development at the University of Texas at Dallas (since 1990). He was a visiting associate professor of Cognitive and Linguistic Sciences in Brown University in 1986 and 1987 and a Fullbright scholar. His research interests include neural networks, computational and statistical models of cognitive processes (especially memory and learning), experimental design, and multivariate statistical analysis.

and R.S. Jones, Distinctive features, categorical perception, and probability learning: Some applications of a neural model, *Psychol. Rev.* **84**, 413–451 (1977).

29. T. Kohonen, *Associative memory: A system theoretic approach.* Springer-Verlag, Berlin (1977).
30. T. Kohonen, E. Oja and P. Lehtiö, Storage and processing of information in distributed associative memory systems, *Parallel models of associative memory*, G.E. Hinton and J.A. Anderson, eds., pp. 49–81. Erlbaum, Hillsdale (1981).
31. J.P. Benzécri, *L'analyse des données*, Dunod, Paris (1977).
32. M.J. Greenacre, *Theory and applications of correspondence analysis*, Academic Press, London (1984).
33. R.O. Duda and P.E. Hart, *Pattern classification and scene analysis*, Wiley, New York (1973).
34. J.L. McClelland and D.E. Rumelhart, *Parallel distributed processing: Explorations in the microstructure of cognition*, MIT Press/Bradsford Books, Cambridge (MA) (1986).
35. M.I. Jordan, An introduction to linear algebra in parallel distributed processing, *Parallel distributed processing: Explorations in the microstructure of cognition* Vol I, D.E. Rumelhart and J.L. McClelland, eds., pp. 365–422, MIT Press/Bradsford Books, Cambridge (MA) (1986).
36. H. Abdi, *Les réseaux de neurones*, PUG, Grenoble (1994).
37. J.A. Anderson and M.C. Mozer, Categorization and selective neurons, *Parallel models of associative memory*, G.E. Hinton and J.A. Anderson, eds., pp. 213–236, Erlbaum, Hillsdale (1981).
38. H. Abdi, A generalized approach for connectionist auto-associative memories: Interpretation, implications and illustration for face processing, *Artificial intelligence and cognitive sciences*, J. Demongeot. ed., Manchester University Press, Manchester (1988).
39. P. Werbos, *Beyond regression: New tools for prediction and analysis in the behavioral sciences*, Ph.D. thesis, Applied Mathematics, Harvard University (1974).
40. D.B. Parker, A comparison of algorithms for neuron-like cells, *Neural networks for computing*, J.S. Denker, ed., American Institute of Physics: New York (1986).
41. D.E. Rumelhart, G.E. Hinton and R.J. Williams, Learning representations by back-propagating errors, *Nature* **323**, 533–536 (1986).
42. Y. Le Cun, Learning process in an asymmetric threshold network, *Disordered systems and biological organization*, E. Bienenstock, F. Fogelman Soulié and G. Weisbuch, eds., Springer Verlag, Berlin (1986).
43. H. Abdi, Précis de connexionisme, *Intelligence artificielle et intelligence naturelle*, J.F. Le Ny, ed., PUF, Paris (1993).
44. S.R. Searle, *Matrix algebra useful for statistics*, Wiley, New York (1982).
45. G.W. Cottrell, P. Munro and D. Zipser, Learning internal representations of gray scale images: An example of extensional programming, *Proc. 9th Annu. Cognitive Sci. Soc. Conf.*, pp. 462–473, Erlbaum, Hillsdale (1987).
46. H. Bourlard and Y. Kamp, (1988). Auto-association by multilayer perceptrons and singular value decomposition, *Biol. Cybern.* **59**, 291–294 (1988).
47. P. Baldi and K. Hornik, Neural networks and principal components analysis: Learning from examples without local minima, *Neural Networks* **2**, 53–58 (1989).
48. G.W. Cottrell and P. Munro, Principal component analysis of images via back-propagation, *Proc. Soc. Photo-Optical Instrumen. Eng.*, pp. 1070–1076, Cambridge (1988).
49. D. DeMers and G.W. Cottrell, Non-linear dimensionality reduction, *Advances in neural information processing systems* **5**, S.J. Hanson, J.D. Cowan and C.L. Giles, eds., pp. 580–587. Kaufmann, San Mateo (1993).
50. M. Kramer, Nonlinear principal compo-

Anderson, eds., pp 49–81. Erlbaum, Hillsdale (1981).

9. J.A. Feldman and D.H. Ballard, Connectionist models and their properties, *Cognitive Sci.* **6**, 205–254 (1982).
10. L. Sirovich and M. Kirby, Low-dimensional procedure for the characterization of human faces, *J. Opt. Soc. Am. A* **4**, 519–524 (1987).
11. M. Kirby and L. Sirovich, Application of the Karhunen-Loève procedure for the characterization of human faces, *IEEE Trans. Pattern Analysis Mach. Intell.* **12**, 103–108 (1990).
12. M. Turk and A. Pentland, Face processing: Models for recognition, *SPIE Intelligent Robots and Comput. Vision VII: Algorithms and Techniques* **1192**, 22–32 (1989).
13. A.M. Burton, V. Bruce, R.A. Johnston, Understanding face recognition with an interactive activation model, *Br. J. Psychol.* **81**, 361–380 (1990).
14. R. Millward and A.J. O'Toole, Recognition memory transfer between spatial-frequency analyzed faces, *Aspects of face processing*, H.D. Ellis, M.A. Jeeves, F. Newcombe, and A. Young, eds., pp. 34–44. Nijhoff, Dordrecht (1986).
15. A.J. O'Toole and H. Abdi, Connectionist approaches to visually based feature extraction, *Advances in cognitive psychology* (Vol. 2), G. Tiberghien, ed. pp 123–140. Wiley, London (1989).
16. A.J. O'Toole, H. Abdi, K.A. Deffenbacher and J.C. Bartlett, Classifying faces by race and sex using an autoassociative memory trained for recognition, *Proc. 13th Annu. Conf. Cognitive Sci. Soc.*, K.J. Hammomd and D. Gentner, eds., pp 847–851. Erlbaum, Hillsdale (1991).
17. A.J. O'Toole, H. Abdi, K.A. Deffenbacher and D. Valentin, A low-dimensional representation of faces in the higher dimensions of the space, *J. Opt. Soc. Am. A* **10**, 405–411. (1993).
18. A.J. O'Toole, K.A. Deffenbacher, H. Abdi and J.C. Bartlett, Simulating the other race effect as a problem in perceptual learning, *Connection Sci.* **3**, 163–178 (19-91).
19. A.J. O'Toole, R.B. Millward and J.A. Anderson, A physical system approach to recognition memory for spatially transformed faces. *Neural Networks* **1**, 179–199 (1988).
20. A. Schreiber, S. Rousset and G. Tiberghien, Context effects in face recognition: Below recognition bias. The contribution of a simulation, *Cognitive biases*, J.P. Caverni, J.M. Fabre and M. Gonzalez, eds., pp. 243–273. Elsevier Sciences Publishers B.V (North-Holland) (1990).
21. A. Schreiber, S. Rousset and G. Tiberghien, Facenet: A connectionist model of face identification in context, *Eur. J. Cognitive Psycho.* **3**, 177–198 (1991).
22. G.W. Cottrell and M.K. Fleming, Face recognition using unsupervised feature extraction, *Proc. Int. Conf. Neural Network*, pp 322–325, Paris (1990).
23. M.K. Fleming and G.W. Cottrell, Categorization of faces using unsupervised feature extraction, *Proc. Int. Joint Conf. on Neural Networks*, Vol. II, pp. 65–70, San Diego, (1990)
24. G.W. Cottrell and J. Metcalfe, EMPATH: Face, gender and emotion recognition using holons, *Advances in neural information processing systems* **3**, R.P. Lippman, J. Moody and D.S. Touretzky, eds., pp. 564–571. Kaufmann, San Mateo (1991).
25. B.A. Golomb, D.T. Lawrence and T.J. Sejnowski, Sexnet: a neural network identifies sex from human faces. *Advances in Neural Information Processing System* **3**, D.S. Touretzky and R. Lippman, eds., pp 572–577. Kaufman, San Mateo (1991).
26. V. Bruce and A. Young, Understanding face recognition, *Br. J. Psycho.* **77**, 363–383 (1986).
27. M.A. Shackleton and W.J. Welsh, Classification of facial features for recognition, *Proc. CVPR*, pp. 573–579. Maui, Hawaii, (1991).
28. J.A. Anderson, J.W. Silverstein, S.A. Ritz

ful choice of parameters, this approach is insensitive to scale variations. A disadvantage, however, results from the amount of calculation required to detect the features.

While the deformable template approach seems promising, a good deal more data will be required to flesh out the limits of this approach. For example, a parametric exploration of the accuracy of the model to match faces as a function of the degree of pose change would be useful, as would an analysis of the accuracy of the matching with variations in the homogeneity of the stimulus set.

7. SUMMARY

We have provided a review of the recent attempts to model face processing using a statistical and/or a connectionist framework in conjunction with an image-based coding of the faces. These models use a distributed rather than a localized way of representing the faces. With this type of storage, faces share the same "storage space", and hence, representations of similar faces can interfere with each other. Therefore, the performance of the models is sensitive to the composition (or statistical structure) of the set of faces on which they are trained. We have shown that a common characteristic of these models is that they implement, explicitly or implicitly, a principal component analysis of the cross-product matrix of the set of faces on which they are trained. They represent faces as a weighted combination of "macrofeatures" (eigenvectors, eigenfaces, eigenpictures, or "holons", depending on the authors) derived from the statistical structure of a set of learned faces. When visually displayed, these "macrofeatures" span the entire face and appear face like. They provide enough information to categorize, recognize, and identify faces.

Clearly, although these models do not intend to solve the general problem of face processing, they do provide a practical solution to the tasks of face recognition and categorization, as well as an interesting tool for analyzing and quantifying information in faces. However, some problems such as the effect of change in size, orientation, background, and to a lesser degree lighting conditions, have not been completely resolved to date and merit further investigation.

Acknowledgments—Thanks are due to June Chance and Al Goldstein for providing the faces used in the simulations illustrating the linear autoassociator section and to Betty Edelman and Peter Assmann for helpful comments on a previous version of the manuscript.

REFERENCES

1. A. Samal and P.A. Iyengar, Automatic recognition and analysis of human faces and facial expressions: A survey, *Pattern Recognition* **25**, 65–77 (1992).
2. T. Sakai, M. Nagao and M. Kidode, Processing of multilevel pictures by computer —the case of photographs of human face, *Syst., Comput., Controls* **2**, 47–54 (1971).
3. Y. Kaya and K. Kobayashi, A basic study on human face recognition, *Frontiers of Pattern recognition*, S. Watanabe, ed., pp. 265–289. Academic press, New York (1971).
4. L.D. Harmon and W.K. Hunt, Automatic recognition of human face profiles, *Comput. Graphics Image Process.* **6**, 135–156 (1977).
5. L.D. Harmon, M.K. Khan, R. Lash and P.F. Ramig, Machine identification of human faces, *Pattern Recognition* **13**, 97–110 (1981).
6. G.J. Kaufman and K.J. Breeding, The automatic recognition of human faces from profile silhouettes, *IEEE Trans. Syst. Man Cybern.* **6**, 113–120 (1976).
7. M. Turk and A. Pentland, Eigenfaces for recognition, *J. Cognitive Neurosci.* **3**, 71–86 (1991).
8. J.A. Feldman, A connectionist model of visual memory, *Parallel models of associative memory*, G.E. Hinton and J.A.

views for a particular face. While it is possible to imagine that several views of a person could be learned by an autoassociative memory [7], to classify persons rather than views would probably require a nonlinear mapping mechanism such as the hyperBF network originally proposed by Poggio and Edelman [53,54] for the recognition of three-dimensional objects and faces.

6. OTHER APPROACHES

At this point it is necessary to digress briefly to describe another type of connectionist model that is aimed at classifying "people" from images taken from different view points.[55-57] As we shall see, these models do not fit easily into our connectionist model classification scheme since, in two senses, they are not classically "parallel distributed processing models". First, while the kinds of face representations they used can be computed in parallel, the storage of individual faces is "localized" rather than "distributed". "Templates" or "models" of individual faces are stored in discrete locations in the memory, rather than by the shared activities of a single set of units. Further, identification of a given face is achieved *via* a serial search through all of the stored models, using a kind of template matching scheme.

In two other senses, however, these models are very much in the spirit of "connectionist" models. First, the face representations are based on feature detectors that are strongly inspired by the properties of neurons in primary visual cortex. Second, the representations of the faces to be matched are allowed to "deform" in an elastic manner to better fit the stored templates. This is done by assessing a cost in the goodness of fit as a function of the degree to which distortion is required to achieve a match. This "deformable template" approach [55-57] appears very promising for solving some matching problems when the input face images do not match the stored templates due to small changes in view or scale.

Buhmann et al. [55-57] used such an approach with holistic face templates. A grid is superimposed over an example face, and Gabor jets (a set of coefficients of Gabor filters of various orientations, resolutions and frequencies) are extracted at each grid point. This is an unsupervised technique for extracting features from the data. The features are holistic at the level of Gabor jets because nearly the whole face can be regenerated from one Gabor jet. When a new view of a face (or a new face) is presented as input, the grid is deformed using an energy minimization approach until the best match is found. This results in the ability of the system to deal with orientation changes by producing the best match with the deformed template. Further, only one "training example" is necessary for each stored person. The disadvantage of this type of approach is that the system must potentially check the match to every stored template (corresponding to the number of known faces). However, it is likely that efficient data structures could be designed to store similar faces together and hence reduce the number of matches. Recent work by Lades et al.[57], using a gallery of 87 persons, indicates an 85% correct matching (averaged across 3 different criteria) when the face presented as memory key was rotated by 15 degrees.

A related approach has been applied by Yuille[58,59] to the problem of feature extraction. He constructed analytic templates of face features and parameterized them. The parameters were then used to define a Lyapunov function which was minimized when a match was found. In brief, a gradient descent algorithm is applied to the parameters of the templates to detect the features. By ordering the weights of the parameters in the successive minimizations, a nice sequential behavior results in which the eye is first located, then the template oriented, and finally a fine matching of features is performed. This approach is subject to local minima in the Lyapunov function, but a more sophisticated matching strategy avoids this problem[60]. Robust matching methods may be used[61] to give some ability to deal with occlusion[59]. An advantage of using this type of method is that with a carelike hidden unit receptive fields. The second one,[24] resulting from low learning rates, is basically linear. It is equivalent to a principal component analysis representation with the exception that the principal components are distributed across each hidden unit (i.e., the weight matrix is obtained as a rotation of the principal components). The third representation, [49] using 3 or more hidden layers, finds a parameterization of the underlying "face manifold" in appearance space. More work on this sort of representation is needed with larger data sets in order to determine the dimensionality and usefulness of models with several hidden layers.

5. Comparison between linear autoassociator and backpropagation networks

The first point of difference between the principal component and the backpropagation approaches is that the former involves simple linear units whereas the latter assumes the presence of nonlinear hidden units. However, this difference appears to be more theoretical than practical (at least when a single hidden layer is involved). As we have already noted, an analysis of the backpropagation network activity during learning showed that the hidden unit outputs tend to stay within the linear range of the logistic function.[22,23,46] This suggests that the problem at hand is basically a linear problem, thus making the nonlinear function of the hidden units unnecessary. [47] Further, the backpropagation algorithm is known to be very slow. In terms of gradient descent, this is due to the fact that although the landscape of error has a single global minimun[†] it also has many plateaux.[47] Therefore, it might be more practical simply to compute the principal components directly rather than to use a backpropagation network to derive an approximately equivalent solution.

A second difference between the models presented in this review concerns the type of error they try to minimize when adjusting the weights in the weight matrix. In many ways, this is related both to the task for which they have been designed and to some extent, the motivations of the researchers who have designed them. Researchers who have approached these problems from a purely computational point of view have been interested in having a model produce a "correct" classification response. For this reason, the final output of these models is a response where the activity of single units in the output layer indicate things like "male" or "toto" or "smiling". These networks, therefore, minimize the error of the *response* associated with a particular face. One advantage of these models is that they directly solve a particular or small sets of classification problem(s). Less positively, however, by training the model to produce a single or small set of classifications, it then becomes a very special purpose device.

By contrast to the previous models, simpler autoassociative or PCA models, by themselves, do not produce explicit responses to practical face classification tasks. They are trained only to minimize the least squares error between the original faces and the faces reconstructed by the memory. These models have been very useful, primarily as tools for the quantification of the highly complex information in sets of faces. They preserve and make use of information that is useful for most face processing tasks that rely on visually-based information. These tasks include recognition and classification into some visually derived semantic categories, such as sex and race. What is less clear is the ability of these models to solve some higher level classifications such as the classification by "person" from multiple views (e.g., profile, 3/4, full-face). This is certainly a nonlinear, classification task in that any given pair of "profiled" faces will be more similar to each other than will any given pair of radically different

[†] because the function to be minimized (i.e., the sum of squares of error) is a quadratic function it has a single minimum.

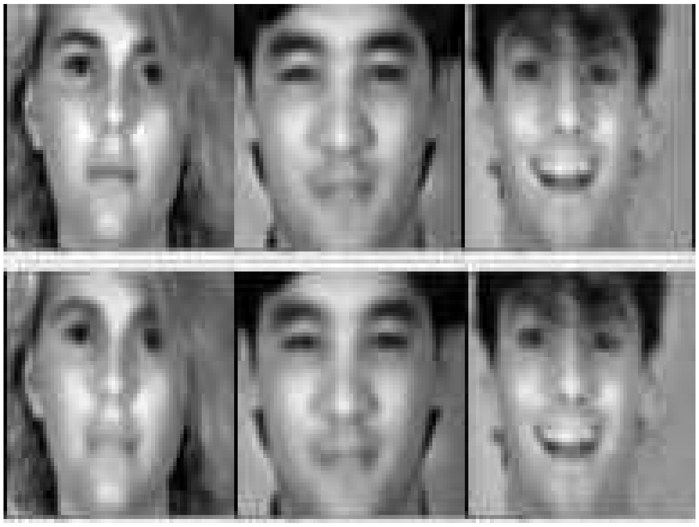

FIG. 14. — Three of the 160 face images used by DeMers and Cottrell[49] *a*) after reduction to the first 50 principal components (top panel) *b*) after reconstruction from a five dimensional encoding (bottom panel).

tion layer of 20 units, and a 30-unit decoding layer (cf. Figure 13). First, the network was trained using backpropagation to auto-encode the inputs. Then, dimensionality reduction was performed to reduce the number of units of the representation hidden layer. This was achieved by using a "greedy" algorithm which selects, in the initial representation layer, the hidden unit with the lowest variance across its outputs over the training set. A penalty term is then applied to the cost function for that unit so as to minimize its variance until the unit can be completely eliminated. The unit with the next smallest variance is then selected and its variance minimized. The process is repeated until the variance of the selected unit cannot be reduced further while maintaining low error. During this process, the output weight vector of these units must be kept to a fixed maximum size to prevent the learning algorithm from expanding them to compensate for the reduced variance.

The NLDR network was able to find a 5D representation of the face images. While the individual dimensions found by this method are difficult to characterize, all the representations within the convex hull of the region in the 5D space where the training data lie appear face-like when projected through the decoding layers. As an illustration, Figure 14 shows 3 face images reconstructed using the first 50 principal components (training input) of a covariance matrix created from 160 face images and the same images after reconstruction from a five dimensional encoding (NLDR network).

To show that the representation found by the NLDR network fared well in characterizing the faces, DeMers and Cottrell[49] trained a feed-forward network to classify the 5D face representations according to identity and gender. They used a randomly-chosen subset of 6 of the 8 face images of each subject as training set. The remaining 40 faces were used as testing set. The network was able to correctly identify and to determine the gender of 95% of the testing set.

In summary, three kinds of representations have been found by compression (or auto-encoding) neural networks. The first one, [22] resulting from high learning rates, is basically binary and is characterized by face-

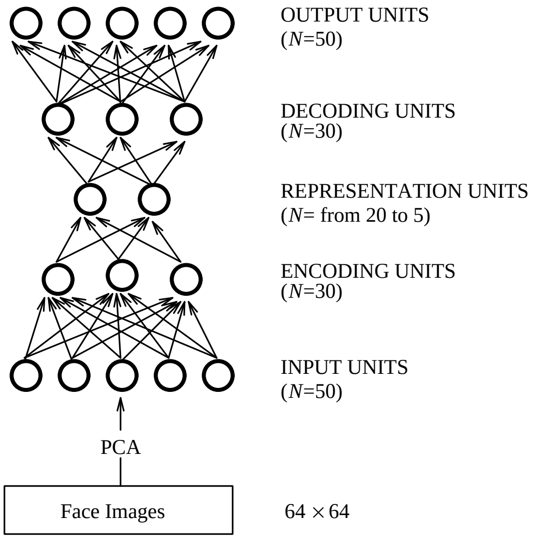

FIG. 13. — Architecture of the NLDR network used by DeMers and Cottrell.(49)

in a new set of 40-dimensional vectors, which could be displayed by running them through the decompression half of the network. The final images were normalized for brightness and variance in the same way as the training set. The images obtained with the first 6 eigenvectors are displayed in Figure 12.

The compression networks previously described used only one layer of weights between the input and the representation layer (the single hidden layer of dimension L). According to DeMers and Cottrell(49), with this type of architecture, only *linear* or *linearly separable* representations can be found. Hence, a compression network with a single hidden layer would be equivalent to PCA (i.e., to a linear representation of the data). Linear representations can overestimate the number of actual dimensions (or, for our purposes, *macrofeatures*) in the data. For example, suppose we were trying to find the principal components of a helix in 3D, PCA would find 3 significant components in spite of the fact that a spiral is clearly a 1D object. Having more than one hidden layer between the input and the output layers of a compression network, allows the network to learn a nonlinearly separable representation.(50-52) With additional hidden units, compression networks can find a 1D encoding of the helix.(49) More generally, they can perform a nonlinear analogue to PCA, and extract "principal manifolds". DeMers and Cottrell (49) referred to this technique as "nonlinear dimensionality reduction" (NLDR).

DeMers and Cottrell (49) used this approach to encode the faces previously used by Cottrell and Metcalfe(24). The face images were preprocessed using PCA. The projections of each face onto the first 50 principal components† were then used as training data for a NLDR network. The architecture of the NLDR network consisted of one input layer, one output layer and three hidden layers made of a 30-unit encoding layer, an initial representa-

†50 was chosen as criterion because it was at the 50th eigenvector that the eigenvalues began to flatten out.

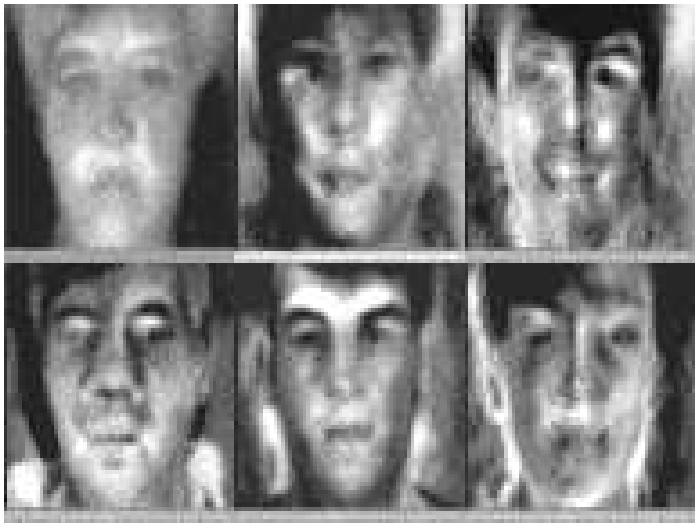

FIG. 12. — Six "holons" obtained by Cottrell and Metcalfe.[24]

ferred to by the authors as "holons" (cf. Figure 11).

Cottrell and Metcalfe[24] applied a similar network with limited success to the categorization of faces according to their sex, identity, and expression. The face images of 20 people taken with 8 different expressions were used, making a total of $K = 160$ stimuli. These images were first compressed using a 3 layer network ($4096 \times 40 \times 4096$). The representation formed in the hidden units was then used as input to a 2 layer network trained for sex and identity classification and a 3 layer classification network ($40 \times 20 \times 8$) that was trained to classify the faces according to facial expression. Results showed that the "sex and identity network" was able to identify 99% of the training set and to classify perfectly the faces according to their sex. The "emotion network" was able to distinguish some of the positive emotions (astonished, delighted, pleased, relaxed), but was unable to distinguish reliably the negative ones (sleepy, bored, miserable, angry). This could be due to the fact that people are better at displaying positive expressions on command than negative ones, and therefore negative expressions might be more similar to each other than the positive ones.

In contrast to Cottrell and Fleming[22,23] who used a very fast learning rate of 0.2 (as seen above), Cottrell and Metcalfe[24] used a very low learning rate ($\eta = .0001$) for the hidden layer. With this low learning rate, the activity of the hidden units tended to stay in the quasi-linear part of the squashing function as previously observed by Cottrell et al.[45] A "single cell recording" of one of these units would vary smoothly for different faces as opposed to the binary response found by Cottrell and Fleming[22,23]. Thus, with a low learning rate the network spanned the principal component subspace of the covariance matrix of the face images (i.e., the first eigenvectors of the covariance matrix). When displayed, the receptive fields of the hidden units resembled "white noise". In order to analyze the representation of the network, Cottrell and Metcalfe[24] first recorded the hidden unit activations for each of the 160 faces of the training set in a 160×40 matrix. From this matrix, they computed the 40×40 cross-product matrix for the hidden units. The eigendecomposition of this cross-product matrix resulted

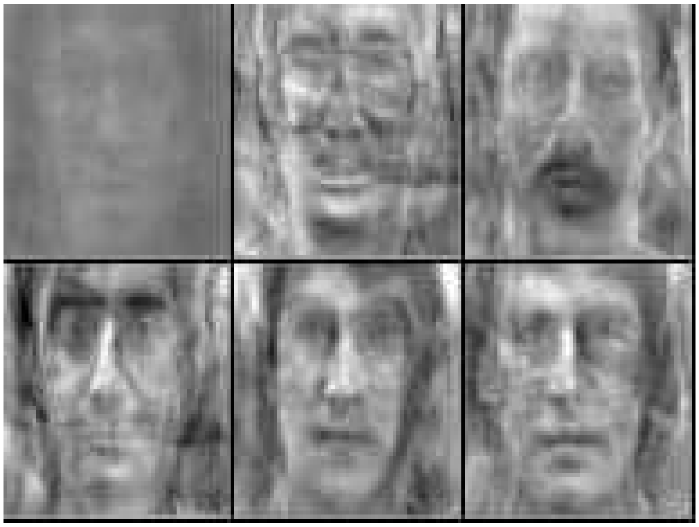

FIG. 11. — Six "holons" obtained by Cottrell and Fleming.(22,23) The first one represents the "bias" of the output units. This is the activation of the outputs from the vector of bias weights. The remaining 5 were chosen randomly from the set of hidden units.

model was able to classify correctly most of the new faces (8.1% error averaged across 8 trials). An interesting point mentioned by the authors is that a 3-layer network with 900 input units, 40 hidden units and 1 output unit, directly trained to produce a value of 1 for the male and 0 for the female faces, was able to categorize correctly the faces on which it was trained, but was also unable to generalize to new faces. However, the authors state this point as an observation and do not provide a statistical analysis.

As noted previously, an interesting aspect of compression networks is the nature of the representation developed by their hidden units. In particular, these representations vary depending on the learning rate used to compress the faces. Cottrell and Fleming,(22,23) used a fast learning rate ($\eta = 0.2$) that can be considered too high for the fan-in of the hidden units (4096). The resulting internal representation was basically binary (i.e., for a given face, the hidden units were either "on" or "off"). In order to gain a better understanding of the representation developed by the hidden units, Cottrell and Fleming(22,23) analyzed the set of weights for each hidden unit separately. More specifically, using the notation of Equations 15 to 21, they analyzed both matrices $\mathbf{W}$ (from the input to the hidden units) and $\mathbf{Z}$ (from the hidden to the output units). First, they consider the $L \times I$ (i.e., 80 hidden units $\times$ 4096 input cells) matrix $\mathbf{W}$, one row at a time. They call the ℓ-th row of $\mathbf{W}$ the "receptive field" of the ℓ-th input cell. They also considered the response of the network to a stimulation coming from a single hidden unit (i.e., with the ℓ-th hidden cell being "on" and all the other cells being "off"), which is equivalent to considering the ℓ-th column of the $J \times L$ (i.e., 4096 output cells $\times$ 80 hidden units) matrix $\mathbf{Z}$. By analogy with the name of receptive field used for the rows of $\mathbf{W}$, the columns of $\mathbf{Z}$ could be called the "projective field" of the hidden units. When transformed to gray scale and graphically displayed, the hidden unit receptive and projective fields looked "face-like" and showed some similarity to the eigenvectors or eigenfaces presented in the principal component approach section. Receptive and projective fields as well as eigenvectors are re-

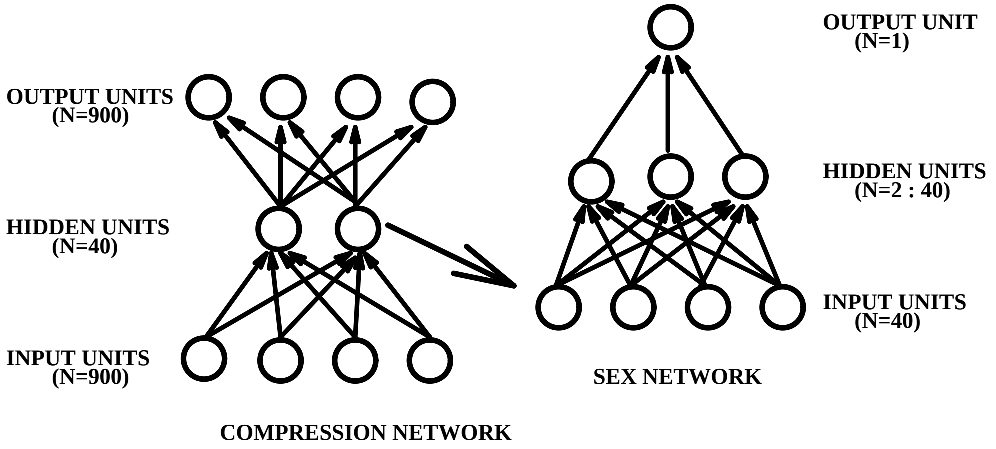

FIG. 10. — Architecture of SEX-Net.[25]

(1) The model identified and categorized perfectly the images on which it was trained, and classified nearly perfectly (3% error) the new instances of learned faces.

(2) The new faces were categorized perfectly with respect to "faceness" but not according to sex (37% error: 26 out of 70 female images were classified as male, all the male images were correctly classified).

(3) The model was fairly robust to partial deterioration of the images. It was able to identify and categorize (with only 3% error) faces partially obscured by a horizontal gray bar (1/5 the height of the image) placed in the bottom 4/5 of the faces. In contrast, when the bar was placed in the top 1/5 of the faces, 29% identification error and 16% sex categorization error occurred. This suggests that the network is using the information in the forehead region to discriminate between faces. The importance of the forehead region is confirmed by the fact that the network was more disturbed when the top half of the face was obscured that when the bottom half was obscured (56% error *vs.* 0% for faceness, 70% *vs.* 50% for identity, and 55% *vs.* 29% for sex categorization).

(4) Modification in brightness (up to 70% increase) resulted in no more than 7% error for both identification and classification.

In summary, compression networks are able to act as content addressable memories (i.e., they are able to reconstruct a face when a noisy image of this face is given as input). Further, the internal representation they develop provides useful information to identify and categorize faces. A final point worth noting is that, in the set of simulations reported here, the quality of representation of new faces by the compression network was relatively poor, however, this was probably due to the small size of the training set (only 11 subjects).

Before discussing the properties of the internal representation developed by the Cottrell and Fleming network, it is worth noting that the ability of this type of compression networks to extract information useful for sex categorization has also been demonstrated by Golomb et al.[25] Following the scheme of Cottrell and Fleming,[22,23] they used a backpropagation network with 900 input units, 40 hidden units, and 900 output units, to compress a set of 80 face images (40 males and 40 females). The internal representation developed by the hidden units was then presented as input to a 3-layer classification network with 40 input units, 10 hidden units, and 1 output unit. This second network was trained using backpropagation to produce an output of 1 for male and 0 for female faces (cf. Figure 10). After learning completion, the performance of the network was tested for sex categorization using 10 new faces (i.e., not learned by the network) as input. Results showed that the

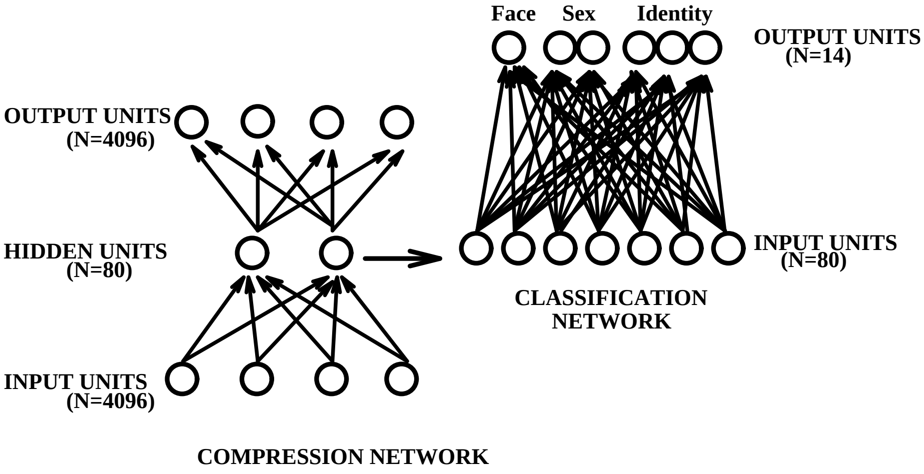

FIG. 9. — Architecture of the two-stage network used by Cottrell and Fleming.(22,23)

the projections of the inputs along the first M principal components. However, Cottrell and Munro(48) point out that nonlinear networks differ from linear networks in two ways.

(1) The first principal component is somewhat distorted, and the other ones slightly shortened.

(2) The variance is evenly distributed across the hidden units.

Cottrell et al.(45) suggested that "the hidden units span the space of the first M principal components, but are rotated so that each can have about equal variance" (p 471). This suggestion was later supported by further work.(48) This is not unlike what was observed for an autoassociator trained with Widrow-Hoff learning. Namely the fact that using a Widrow-Hoff learning rule is equivalent to equalizing the eigenvalues of the weight matrix.[‡]

4.3 Application to face processing.

The ability of compression networks to extract useful information for face processing was first investigated by Cottrell and Fleming.(22,23) They trained a 3 layer network with 4096 input units, 80 sigmoidal hidden units and 4096 output units to compress a set of faces and non-face images. This compression network differs from the one used by Cottrell et al.(45) in that the whole image is presented as input at once. A training set composed of 64 face images (5 or 6 different instances of 11 faces) and 13 non-face images (random shots) was extracted from a database of 231 digitized images[‡] consisting of 204 face images (5–20 pictures of 17 faces) and 27 non-face images. All faces in the training set were presented as input to the compression network. The representation formed in the hidden units was then used as input to a 2 layer network with 80 input units and 14 output units with range $[-1, 1]$, trained to classify the face images according to "faceness", sex, and identity (cf. Figure 9).

The ability of the model to generalize to new images was then tested by presenting the classification network with a testing set composed of new instances of learned faces, new faces, new non-faces, and a series of degraded faces (e.g., faces partially obscured, faces with different lighting). The results of the simulations are summarized below.

[‡] This is especially true because after sphericization any orthogonal set of vectors is a set of eigenvectors with eigenvalues equal to one.

[‡] The faces were digitized using 256 gray levels and then compressed by local averaging to a 64 × 64 pixel image.

where η is a small learning constant and k is randomly chosen.

The adjustment of weights in the hidden units is proportional to both the error in the output and the extent to which each specific unit contributes to this error. The error signal vector for the hidden units is denoted $\boldsymbol{\delta}_{\text{hidden}}$ and:

$$\begin{aligned}\boldsymbol{\delta}_{\text{hidden},k} &= f'(\mathbf{W}\mathbf{f}_k) \odot (\mathbf{Z}^T \boldsymbol{\delta}_{\text{output},\, k}) \\ &= \mathbf{h}_k \odot (\mathbf{1} - \mathbf{h}_k) \odot (\mathbf{Z}^T \boldsymbol{\delta}_{\text{output},\, k}) \ . \end{aligned} \qquad \mathbf{(20)}$$

Learning at time $t+1$ is given by:

$$\begin{aligned}\mathbf{W}^{(t+1)} &= \mathbf{W}^{(t)} + \eta \boldsymbol{\delta}_{\text{hidden},\, k} \mathbf{f}_k^T \\ &= \mathbf{W}^{(t)} + \boldsymbol{\Delta}_{\mathbf{W}}^{(t+1)} \ . \end{aligned} \qquad \mathbf{(21)}$$

4.2 *Compression Networks*

An interesting aspect of backpropagation networks is that during learning, the hidden layers build an internal representation of the input that is useful for producing the desired output. More specifically, a network with I input units, $J = I$ output units and L hidden units (with $L < I$) trained to reproduce its input at the output layer (i.e., to autoassociate its inputs) will develop a compact representation of the inputs in the hidden units. The fact that the hidden layer contains fewer units than the input and output layers forces the network to encode the inputs in a smaller dimensional subspace that retains most of the important information. In other words, the network is forced to capture the redundancy or regularities present in the input and to extract a set of statistically salient features in order to represent the input in a less correlated way. It is important to note that if the hidden units are linear, the best solution to this problem is the least squares solution (i.e., to have the hidden units span the L principal components with the highest eigenvalues). This type of network is often referred to in the literature as an *encoding* or *autoencoding* network. The input-to-hidden layer connections perform the encoding (or compression) of the information and the hidden-to-output layer connections do the decoding (or decompression).

Cottrell, Munro and Zipser[45] trained a 3 layer network with 64 units in the input layer, 16 hidden units, and 64 units in the output layer to reproduce a digitized image. The input was 150,000 8×8 square pixel patches randomly selected from the original image and successively presented. The network was trained by standard backpropagation with a logistic activation function rescaled to the range $[-1, 1]$. The learning constant (noted η in the previous equations) was .25 for the first 100,000 iterations and .10 for the remaining 50,000 iterations. The network was then tested on the entire image, patch by patch, using 8×8 non overlapping patches. The resulting image constituted an accurate reproduction of the original one. More interestingly, the network was also able to reconstruct several novel images.

In order to understand the way in which the hidden units represent the information, it is helpful to look at their activity during learning. Cottrell et al.,[45] found that the hidden unit outputs tend to stay in the linear range of the logistic function. Using an empirical approach, they showed that the weights of the hidden units span the space of the M first eigenvectors of the covariance matrix of the inputs (where M is smaller or equal to L, the number of hidden units). More specifically, using 16 hidden units they found that the network was spanning the first 13 eigenvectors (i.e., the ones with the highest eigenvalues). In other words, the network projects its inputs onto the subspace spanned by the principal components of the covariance matrix of the inputs[†]. Thus, the reconstructed image can be thought of as a linear combination of

[†] For a more formal demonstration of the relationship between backpropagation networks and principal component analysis, see, for example, references (46) and (47).

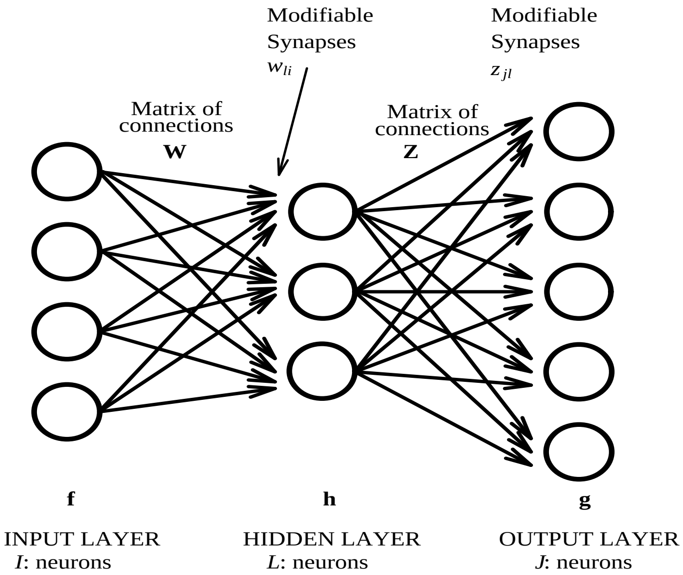

FIG. 8. — Illustration of a 3-layer backpropagation network

It has the interesting property of having a derivative easy to compute:

$$f'(x) = \frac{e^{-x}}{(1+e^{-x})^2} = f(x)[1-f(x)]. \quad \mathbf{(14)}$$

The backpropagation algorithm involves two phases. In the first phase, a forward flow of activation is generated from the input layer to the output layer *via* the hidden layer. Each unit in the hidden layer computes a weighted sum of its inputs and transforms it *via* the logistic function:

$$\mathbf{h}_k = f(\mathbf{W}\mathbf{f}_k) \ . \quad \mathbf{(15)}$$

The output for one layer is the input for the next layer. The responses of the output units are given by:

$$\hat{\mathbf{g}}_k = f(\mathbf{Z}\mathbf{h}_k) \ . \quad \mathbf{(16)}$$

In the second phase, the error, defined as the difference between the actual output and the desired output, is computed:

$$\mathbf{e}_k = (\mathbf{g}_k - \hat{\mathbf{g}}_k) \quad \mathbf{(17)}$$

and backpropagated through the network, layer by layer. During this phase, the weights of the connections are adjusted so as to minimize the mean-squared error between the network output and the desired output.

The rule used for backpropagating the error is a generalization of the delta or Widrow-Hoff learning rule presented in the linear autoassociator section. Formally, the error signal, denoted $\boldsymbol{\delta}_{\text{output}}$, for the output layer is given by:

$$\begin{aligned}\boldsymbol{\delta}_{\text{output},\ k} &= f'(\mathbf{Z}\mathbf{h}_k) \odot (\mathbf{e}_k) \\ &= \hat{\mathbf{g}}_k \odot (\mathbf{1} - \hat{\mathbf{g}}_k) \odot (\mathbf{g}_k - \hat{\mathbf{g}}_k) \ .\end{aligned} \quad \mathbf{(18)}$$

Where f' represents the derivative of the logistic function, $\odot$ the element-wise product of the vectors (or Hadamar product$^{(44)}$), and $\mathbf{1}$ a unit vector. As for the linear autoassociator, the difference between the response of the system and the expected response is corrected by iteratively changing the weights in the matrix $\mathbf{Z}$. At time $t+1$ the matrix is computed as:

$$\begin{aligned}\mathbf{Z}^{(t+1)} &= \mathbf{Z}^{(t)} + \eta \boldsymbol{\delta}_{\text{output},k} \mathbf{h}_k^T \\ &= \mathbf{Z}^{(t)} + \boldsymbol{\Delta}_{\mathbf{Z}}^{(t+1)}\end{aligned} \quad \mathbf{(19)}$$

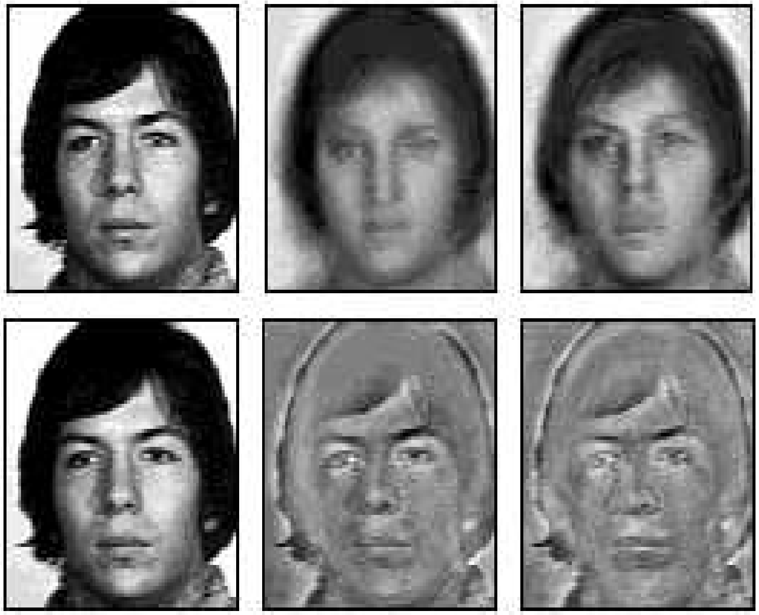

FIG. 7. — Illustration of the reconstruction of a face using: 1) top panels: *a*) all the eigenvectors *b*) the first 20 eigenvectors *c*) the first 40 eigenvectors 2) bottom panels: *d*) all the eigenvectors *e*) all but the first 20 eigenvectors *f*) all but the first 40 eigenvectors. Note that the eigenvectors with relatively large eigenvalues (panels *b* and *c*) provide mainly global shape information and the eigenvectors with relatively small eigenvalues (panels *e* and *f*) provide mostly identity-specific information.

ues is optimal for solving semantic categorization tasks such as sex or race categorization. In contrast, eigenvectors with small eigenvalues convey information about small, identity-specific details (cf. Figure 7, bottom panels). A low dimensional representation of the faces in the subspace associated with small eigenvalues is optimal for the identification of specific faces, or the discrimination between learned and unlearned faces.

4. BACKPROPAGATION NETWORKS—IMAGE COMPRESSION

In contrast to linear associative networks in which the input units are directly connected to the output units, backpropagation networks include nonlinear hidden units between input and output units. The backpropagation algorithm was originally proposed by Werbos[39] and was independently rediscovered by Parker [40], Rumelhart, Hinton and Williams[41], and Le Cun.[42] This algorithm adjusts the weights in any feedforward network trained to associate pairs of patterns. It can be easily described using a matrix notation equivalent to the one used in Section 3.1.

4.1 Algorithm description

Our notation will be as follows (cf. reference 43, Equations 28–34). The k-th stimulus described by I features is represented as an $I \times 1$ vector $\mathbf{f}_k$. The response of the hidden units for the k-th stimulus is represented by an $L \times 1$ vector $\mathbf{h}_k$, where L represents the number of hidden units. The response of the output units for the k-th stimulus is represented by a $J \times 1$ vector $\hat{\mathbf{g}}_k$, where J represents the number of output units. The desired output for the k-th stimulus is represented by a $J \times 1$ vector $\mathbf{g}_k$. The intensities of the connections between the input units and the hidden units are given by an $L \times I$ matrix $\mathbf{W}$. The intensities of the connections between the hidden units and the output units are given by a $J \times L$ matrix $\mathbf{Z}$. Several nonlinear functions can be used by the hidden and output layers, the most popular one is the *logistic function*:

$$f(x) = \frac{1}{1+e^{-x}} \ . \qquad \mathbf{(13)}$$

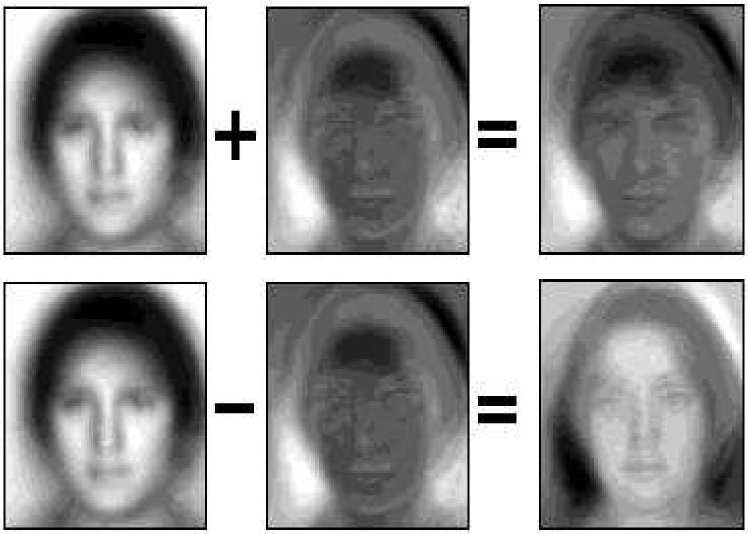

FIG. 6. — Adding the second eigenvector to the first eigenvector of a cross-product matrix of half male and half female faces gives rise to a masculine looking face (top panel). Subtracting the second eigenvector from the second one gives rise to a feminine looking face (bottom panel)

used in that simulation. An interesting aspect of this result is that the linear autoassociator was not trained for race classification, but rather, was trained to reconstruct the faces. The information necessary to classify faces according to their race, hence, emerged spontaneously from the memory representation.

In a second simulation, O'Toole et al.(17) were interested in examining the usefulness of different eigenvectors in categorizing the faces according to sex. When the model was trained with equal numbers of male and female faces of a single race, a correlational analysis showed that the eigenvectors with larger eigenvalues were more useful for predicting the sex of the faces than were the eigenvectors with smaller eigenvalues. Here, the second eigenvector provided the most important information for separating male and female faces. O'Toole et al.(17) have shown that although this eigenvector appears androgynous, adding it to the first eigenvector creates a masculine-looking face and subtracting it creates a feminine looking face (cf. Figure 6).

More generally, these results depend upon the racial and sexual homogeneity of the training set. From a psychological point, this is a nice feature of the model in that it is reminiscent of factors that affect human memory and might be part of the cause of phenomena such as the "other-race effect". The first eigenvector itself represents essentially the characteristics shared by all the faces in the training set. In other words, it represents the face category in general and can be interpreted as a face prototype.(38) This eigenvector could be used to categorize stimuli as faces as opposed to other object categories. The eigenvalue associated with the first eigenvector is generally very large (it explains between 90% and 99% of the variance). This strong value reflects the fact that faces constitute a class of highly similar objects.

An important implication of the O'Toole et al. (17) studies is that the optimal characteristic of a representation is not absolute but rather depends on specific task requirements. Eigenvectors with high eigenvalues convey mainly information about the general shape of the faces (cf. Figure 7, top panels). Therefore, a low dimensional representation of the faces in the subspace associated with high eigenval-

However, if the change were to become sufficiently large (e.g., if the face is presented upside down) then, performance is expected to decrease dramatically. Finally, recognition performance decreased dramatically with change in head size (64% correct classification averaged over size variations). However, this problem can be solved easily by automatically detecting the outline of the face in the picture and re-scaling it prior to recognition testing.(7) In brief, the presence of a face in an image can be detected by computing the distance between successive subsets of the image and the face space determined by principal component analysis. Because face images are less distorted than other images when projected onto the face space, small distance values indicate the presence of a face. Once the face is located, the background can be diminished by multiplying the face subimage by a two-dimensional Gaussian window centered on the face. The scaling problem can then be solved either by using eigenfaces (or eigenvectors) at different scales or by using different sizes of the input image.

It is important to note that the studies conducted by Sirovich and Kirby(10,11) and Turk and Pentland(7,12) were motivated from an information theory point of view. They were interested in finding the best way to *represent* a set of face images with the smallest possible number of parameters. Consequently, the low dimensional representation they proposed (i.e., the eigenvectors with the highest eigenvalues) is optimal in the least squares error sense. Recent work by O'Toole et al.,(17) demonstrated that while this low dimensional representation is optimal for reconstructing the faces, it is not the most useful representation for recognition. In a first simulation, they examined the importance of different ranges of eigenvectors for discriminating learned faces from unlearned ones (recognition task). An autoassociative matrix was created from 100 face images (50 females and 50 males) and was decomposed into its eigenvectors. Different ranges of 15 eigenvectors (sorted in decreasing order according to their eigenvalues) were used to reconstruct the 100 learned (or "old") faces and 59 new faces. The quality of reconstruction of each face was estimated by the cosine taken between the original and the reconstructed face. For the recognition task, the average cosine was used as a decision criterion. Faces with a cosine above the criterion were categorized as "old" and faces with a cosine below this criterion were categorized as "new". The results were as follows.

(1) The quality of representation (physical similarity as measured by a cosine taken between the original and the reconstructed face vectors) decreased as the range of eigenvectors used was shifted from the eigenvectors with larger eigenvalues to those with smaller eigenvalues.

(2) The ability of the model to discriminate between old and new faces did not follow the decrement in quality of representation. Any of the 15-eigenvector ranges between the 45th and the 80th eigenvectors provided better information for discriminating learned from unlearned faces (i.e., "recognizing" faces) than did the first 15-eigenvector range.

In addition to information in faces that is useful for identifying individuals, the model must also preserve visual categorical information about the sex, race, and age of faces. In previous work, O'Toole et al.(16) had examined the usefulness of the eigenvectors with relatively larger eigenvalues for categorizing faces according to visually-derived dimensions. For example, they reported that when a training set composed of Japanese and Caucasian faces was learned by the model, the 2nd eigenvector carried most of the information useful for predicting the race of the faces. This was shown by computing the projections of the faces onto the second eigenvector (Equation 9). The mean value of the projections was taken as a criterion value, and faces with projections larger than this value were categorized as Caucasian whereas faces with projections smaller than this value were categorized as Japanese. Using only this simple procedure, the second eigenvector by itself yielded correct race prediction for 88.6% of the faces

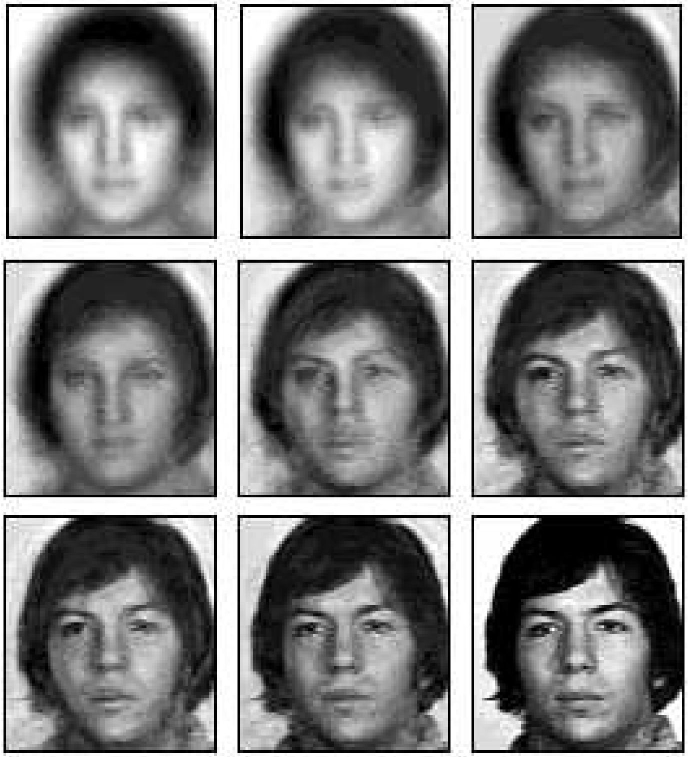

FIG. 5. — Illustration of the reconstruction of a face using (from left to right and top to bottom) *a*) the first eigenvector *b*) the first 5 eigenvectors *c*) the first 10 eigenvectors *d*) the first 20 eigenvectors *e*) the first 40 eigenvectors *f*) the first 60 eigenvectors *g*) the first 70 eigenvectors *h*) the first 100 eigenvectors *i*) all (160) the eigenvectors of a cross-product matrix created from 160 faces (80 males and 80 females). Note that the face is reconstructed exactly when all the eigenvectors are used and well approximated by a relatively small number of eigenvectors.

tilt of the head). Then, they selected several sets of 16 face images (one of each person), all taken under the same conditions, from the data base to serve as training sets. The remaining faces were used as test sets. For each training set a covariance matrix was created and the 8 eigenvectors with the highest eigenvalues were extracted. For recognition, faces from the testing sets were projected onto these first 8 eigenvectors. The resulting weight patterns were compared with the stored projections of the training faces and classified as known or unknown using a nearest neighbor (i.e., single link) algorithm based upon the Euclidean distance between the projection of the target image and the projection of each stored image. If the distance to the nearest neighbor was below some chosen threshold, the face was classified as known, otherwise it was classified as unknown.

A series of simulations carried out using various training and testing sets showed that the system was not extremely sensitive to changes in lighting condition (96% correct classification averaged over lighting variations), and 2D head orientation (85% correct classification averaged over orientation variations[†]). This makes sense since as long as the change in lighting or head orientation is not too large, the pixel-to-pixel correlation remains high.

[†] When performance is averaged only across nonzero head tilt conditions (i.e., the 0% tilt condition is not considered) head tilt yielded approximately 78% correct classification.

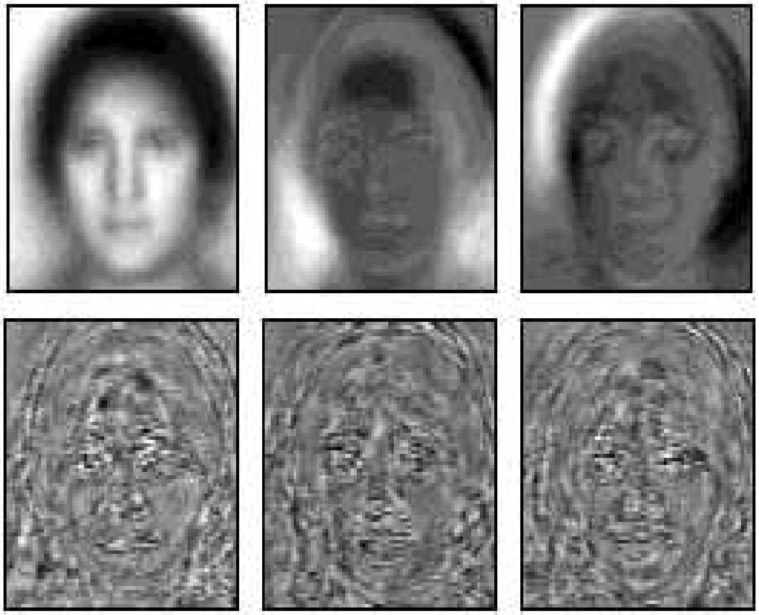

FIG. 4. — The first 3 (top panels) and last 3 (bottom panels) eigenvectors of a cross-product matrix created from 160 face images (80 males and 80 females). Note that all the eigenvectors span the entire face and are face-like.

between minority-race faces. O'Toole et al.$^{(18)}$ used this property of autoassociative memories to model the well known "other-race effect"† as a problem in perceptual learning.

An illustration of the usefulness of the eigenvectors (also referred to as eigenpictures or eigenfaces) for representing faces was provided by Sirovich and Kirby.$^{(10)}$ They analyzed the properties of a covariance matrix created from 115 faces images, cropped so as to include only the eyes and nose. The analysis showed the following:

(1) When graphically displayed, the eigenvectors of the covariance matrix of an ensemble of faces are face-like (cf. Figure 4).

(2) Any given stored face can be reconstructed exactly as a linear combination of the eigenvectors of the covariance matrix or approximated (in a least squares sense) using only the eigenvectors with the largest eigenvalues (cf. Figure 5).

They demonstrated both statistically and graphically that faces can be optimally reconstructed (to within 3% error) using roughly 40 parameters and the corresponding first 40 eigenvectors. The parameters correspond to the projections of the faces on the eigenvectors (cf. Equation 9). More recently, Kirby and Sirovich$^{(11)}$ extended these results, using a similar approach with 100 "cameo" of full-face including the eyes, nose and mouth. They showed that 82% of the variance of the set of faces is accounted for by the first 10 eigenvectors and 95% by the first 50 eigenvectors.

Turk and Pentland$^{(7,12)}$ analyzed the robustness of the eigenvector approach for face recognition. First, they collected a database of over 2500 full-face images of 16 subjects taken under different conditions of lighting, head scale, and 2D head orientation (sideways

†The "other-race effect" refers to the fact that people are better able to recognize faces of their own race than faces of other races.

faces, he demonstrated that an autoassociative memory could act as a content addressable memory for face images. In his demonstration, an autoassociative memory was first created by autoassociating the face images using a simple Hebbian learning rule (cf. Equation 1). The efficiency of the memory was tested then by presenting noisy or incomplete face images as input. The quality of the reconstructions was estimated by displaying the images reconstructed by the memory. Results showed that the images reconstructed by the system were convincingly similar to the original images. When incomplete or partially obliterated images are presented as memory keys, the memory fills in the missing parts of the image (cf. Figure 3).

Using a similar approach, Millward and O'Toole,(14) showed that autoassociative memories can act as an efficient system for recognizing faces (i.e., distinguishing learned from unlearned faces). In their study, an autoassociative memory was constructed by autoassociating a set of face vectors using a Widrow-Hoff learning rule. Face recognition was simulated by using a standard psychological memory paradigm called a "two-alternative forced choice task" (2AFC). This was done by testing the memory with pairs of face vectors, where each pair was composed of an old face (i.e., previously learned by the memory) and a new face (i.e., not learned by the memory). For each face vector (old and new), the quality of response of the model was estimated by computing the cosine between input and output vectors. The face in the pair with the highest cosine was said to be "recognized" by the memory. The results of the 2AFC showed that model performance was well above chance and hence that the linear autoassociator was able to "recognize" faces. Moreover, it was able to mimic the pattern of recognition transfer errors found with human subjects performing a recognition transfer task for spatially filtered faces.

Both Kohonen(29,30) and Millward and O'Toole(14) showed that a simple autoassociative memory is a useful tool for automatic face processing. More recent work (15-18) has attempted to analyze further the properties of this type of memory. Specifically, these studies try to describe and quantify the perceptual information in faces, and to model the learning of this information. This type of approach, generally referred to as the "principal component approach" to face modeling, relies essentially on the fact that faces can be represented implicitly (*via* a linear autoassociative memory) or explicitly (*via* a principal component analysis) as a weighted sum of eigenvectors extracted from the cross-product matrix of a set of faces. The eigenvectors can be thought of as a set of "features" or "basic components" from which a face is built. Anderson and Mozer(37) refer to this type of feature as a "macrofeature" to differentiate it from the more traditional use of the term "feature". They claim that "macrofeatures are what perception and feature analysis actually uses and are psychologically salient entities"(pp. 221-222).

Applied to face modeling, the "principal component approach" suggests a different definition of the "features" characterizing a face than the one assumed in the feature-based models described in the Samal and Iyengar review.(1) The advantage of representing faces in terms of macrofeatures rather than traditional features is that macrofeatures are not defined *a priori* but are generated *a posteriori* on a statistical basis. They reflect the statistical structure of the set of faces from which they are extracted. In other words, reconstructing faces from an autoassociative memory is like applying a filter to the face images (specifically, a Wiener filter(36)). The properties of the filter are determined by the "face history" of the matrix.

For example, if a cross-product matrix of faces is created from a heterogeneous set of faces, made up of a majority of faces of one race and a minority of faces of another race, the eigenvector representation will capture subtle variations in the form and configuration of the majority-race faces. However, this representation will be less able to capture featural information necessary to discriminate

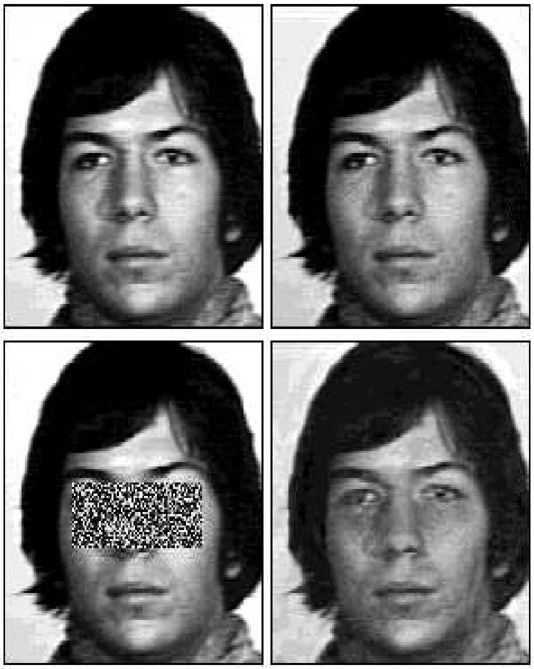

FIG. 3. — Illustration of a content addressable memory for faces. The left panels are the stimuli given as keys to the memory. The right panels are the responses of the memory. The memory is able to reconstruct, reasonably well, a face from an incomplete input (bottom panels). The memory was trained with 160 Caucasian faces (80 males and 80 females), using a Widrow-Hoff learning rule.

this paper, we follow the convention of calling the eigenvector with the largest eigenvalue the first eigenvector, the eigenvector with the second largest eigenvalue the second eigenvector, and so on. The largest eigenvalue is denoted $\lambda_{\max}$.

Similarly, the Widrow-Hoff learning rule can be expressed in terms of the eigenvectors and eigenvalues of a weight matrix at a given time t:[36]

$$\mathbf{W}^{(t)} = \mathbf{U}\mathbf{\Phi}^{t}\mathbf{U}^{T} \qquad \text{with} \quad \mathbf{\Phi}^{t} = [\mathbf{I}-(\mathbf{I}-\eta\mathbf{\Lambda})^{t}] \,. \tag{10}$$

This formulation clearly shows that the Widrow-Hoff error correction rule affects only the eigenvalues of $\mathbf{W}$. More specifically, it follows from Equation 10 that when η is smaller than $2\lambda_{\max}^{-1}$, using the Widrow-Hoff rule will equalize all the eigenvalues of $\mathbf{W}$,[29,30] or in other words will sphericize the weight matrix. Hence, at convergence, $\mathbf{W}$ reduces to:

$$\mathbf{W}^{\infty} = \mathbf{U}\mathbf{U}^{T} \tag{11}$$

and recall can be rewritten by dropping the eigenvalues in Equation 8 (cf. also Equation 9).

$$\hat{\mathbf{x}}_k = \sum_{r}^{R} (\mathbf{u}_r^{T}\mathbf{x}_k)\mathbf{u}_r \ . \tag{12}$$

Intuitively, this is equivalent to increasing the relative importance of the eigenvectors that explain smaller amounts of the variance.

In summary, using a linear autoassociator to store and recall faces is equivalent to computing the principal component analysis of the cross-product matrix of a set of faces and reconstructing the faces as a weighted sum of eigenvectors. Hence, the term "principal component analysis" (PCA) is generally used in the literature to describe this type of model.

3.3 Application to face processing.

As noted previously, Kohonen[29,30] was the first to use an autoassociative memory to store and recall face images. Using a sample of 100 as follows:

$$\begin{aligned}\mathbf{W}^{(t+1)} &= \mathbf{W}^{(t)} + \eta(\mathbf{x}_k - \mathbf{W}^{(t)}\mathbf{x}_k)\mathbf{x}_k^T \\ &= \mathbf{W}^{(t)} + \eta(\mathbf{x}_k - \hat{\mathbf{x}}_k)\mathbf{x}_k^T \\ &= \mathbf{W}^{(t)} + \mathbf{\Delta}_{\mathbf{W}}^{(t+1)} \qquad (5)\end{aligned}$$

where η is a small learning constant and k is randomly chosen. Intuitively, the weight matrix is updated at time $t+1$ by computing the difference between the actual output of the memory (i.e., $\hat{\mathbf{x}}_k$) and the desired output (i.e., $\mathbf{x}_k$) at time t, and by re-teaching this difference to the memory. Since the order in which the stimuli are learned is of no importance, Equation 5 can be rewritten in a more practical way as:

$$\mathbf{W}^{(t+1)} = \mathbf{W}^{(t)} + \eta(\mathbf{X} - \mathbf{W}^{(t)}\mathbf{X})\mathbf{X}^T \ . \qquad (6)$$

Note that the term *iteration* should be differentiated from the term *epoch.* Iteration refers to learning a single stimulus whereas epoch refers to learning the full set of stimuli. Hence, Equation 5 describes an iteration, whereas Equation 6 describes an epoch.

3.2 Model properties

Anderson et al.,[28] and Kohonen[29,30] have provided an interesting analysis of the properties of the weight matrix obtained by autoassociation. They noted that Hebbian learning in an autoassociative system creates a cross-product matrix of the input vectors, ($\mathbf{W} = \mathbf{X}\mathbf{X}^T$) which is equivalent to the sample cross-product matrix of factor analysis. In their demonstration, they pointed out that, since $\mathbf{W}$ is semi-positive definite, it can be reconstructed therefore as:

$$\mathbf{W} = \sum_r^R \lambda_r \mathbf{u}_r \mathbf{u}_r^T = \mathbf{U}\mathbf{\Lambda}\mathbf{U}^T \qquad (7)$$

where $\mathbf{U}$ represents the matrix of eigenvectors $\mathbf{u}_r$ of $\mathbf{W}$ and $\mathbf{\Lambda}$ the diagonal matrix of (positive) eigenvalues λ_r[†] with R being the rank of the matrix $\mathbf{W}$ (i.e., the number of eigenvectors with a non zero eigenvalue).

Further, since the eigenvectors of $\mathbf{W}$ are mutually orthogonal, each input vector $\mathbf{x}_k$ can be expressed as a linear combination of the eigenvectors of $\mathbf{W}$:

$$\hat{\mathbf{x}}_k = \sum_r^R \lambda_r c_{k,r} \mathbf{u}_r \qquad (8)$$

where the $c_{k,r}$ coefficients are the coordinates or projections of the stimulus vector $\mathbf{x}_k$ along the R eigenvectors. These projections are obtained as the dot product between the stimulus vector and each eigenvector. Formally:

$$c_{k,r} = \mathbf{u}_r^T \mathbf{x}_k \ . \qquad (9)$$

The eigenvalues (i.e., the λ_r's) correspond to the variance of the projections on the eigenvectors. The eigenvector with the largest eigenvalue accounts for the largest proportion of variance, the eigenvector with the second largest eigenvalue accounts for the second largest proportion of variance and so on. Because the eigenvectors are orthogonal, the amount of variance explained by a set of eigenvectors is the sum of their eigenvalues. As a consequence, the maximum number of eigenvectors needed to account for all of the variance is equal to the rank of the cross-product matrix. In addition, a large proportion of the variance can be explained by a small number of eigenvectors, the ones with the largest eigenvalues. Therefore, eigenvectors with large eigenvalues are the most useful for making discriminations among members of the input set. In

[†] Eigenvectors of a matrix are vectors that have the property that when multiplied by the matrix, they are unchanged up to a scaling factor (i.e., only their length is changed, not their direction). Thus, if $\mathbf{u}$ is an eigenvector of $\mathbf{W}$, then $\mathbf{W}\mathbf{u} = \lambda\mathbf{u}$. The eigenvalue λ associated with $\mathbf{u}$ indicates how much the length of $\mathbf{u}$ is changed when multiplied by $\mathbf{W}$. For a review of linear algebra applied to neural networks see, for example, reference (35) or (36).

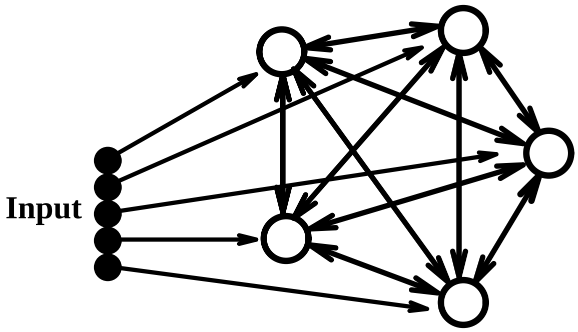

FIG. 2. — Architecture of an autoassociative memory: all cells connect to all other cells with modifiable weights. The input pattern is associated to itself.

viewed as a neural network (cf. Figure 2), the values in the weight matrix **W** correspond to the connection strengths between the cells (or units) of the memory (i.e., $w_{i,i'}$ represents the intensity of the connection between units i and i').

Learning may be achieved by using a simple Hebbian learning rule. This amounts to successively autoassociating each face vector, $\mathbf{x}_k$, and summing the resultant outer-product matrices:

$$\mathbf{W} = \sum_k^K \mathbf{x}_k \mathbf{x}_k^T = \mathbf{X}\mathbf{X}^T \qquad (1)$$

where $\mathbf{X}^T$ denotes the transpose of matrix $\mathbf{X}$.

The retrieval of the k-th face stored in **W** is achieved by postmultiplying the matrix **W** by the input vector $\mathbf{x}_k$ as follows:

$$\hat{\mathbf{x}}_k = \mathbf{W}\mathbf{x}_k \qquad (2)$$

where $\hat{\mathbf{x}}_k$ represents the response or "echo" of the memory. The response can be interpreted as the estimation of the k-th face by the memory. The quality of the image reconstructed by the system can be measured by computing the cosine of the angle between $\mathbf{x}_k$ and $\hat{\mathbf{x}}_k$:

$$\cos(\hat{\mathbf{x}}_k, \mathbf{x}_k) = \frac{\hat{\mathbf{x}}_k^T \mathbf{x}_k}{||\hat{\mathbf{x}}_k|| \, ||\mathbf{x}_k||} \qquad (3)$$

where $||\mathbf{x}_k||$ is the euclidean norm of the vector $\mathbf{x}_k$ (i.e., $||\mathbf{x}_k|| = \sqrt{\mathbf{x}_k^T \mathbf{x}_k}$). A cosine of 1 indicates a perfect reconstruction of the stimulus.

When all the input vectors stored in the memory are mutually orthogonal, recall is perfect ($\hat{\mathbf{x}}_k = \mathbf{x}_k$). If the input vectors are not orthogonal, the system will add noise (or cross-talk) to the correct pattern. Formally:

$$\begin{aligned}\hat{\mathbf{x}}_\ell &= \sum_k (\mathbf{x}_k^T \mathbf{x}_\ell)\mathbf{x}_k \\ &= (\mathbf{x}_\ell^T \mathbf{x}_\ell)\mathbf{x}_\ell + \sum_{k \neq \ell} (\mathbf{x}_k^T \mathbf{x}_\ell)\mathbf{x}_k \\ &= \mathbf{x}_\ell + \sum_{k \neq \ell} \cos(\mathbf{x}_k, \mathbf{x}_\ell)\mathbf{x}_k \\ &\quad (\text{because} \quad \mathbf{x}_k^T \mathbf{x}_k = 1) \ . \end{aligned} \qquad (4)$$

If the cross-talk or interference due to the presence of other patterns learned by the system is not too large, the response of the system is a relatively good estimate of the pattern originally learned. When the vectors are close to orthogonality, the cross-talk is reduced, and hence the response of the system is improved. The performance of the memory can be improved by using the Widrow-Hoff learning rule,(33) also known as the delta rule.(34) The Widrow-Hoff learning rule corrects the difference between the response of the system and the desired response by iteratively changing the weights in the matrix **W** of the persons, their sex or race).

Two principal categories of networks have been used to simulate the "learning" of face images. First, linear autoassociators, which are known to implement principal component analysis,[28-30] have been used for several tasks including recognition and categorization of faces by sex and race.[15-19] Second, nonlinear three-layer autoassociative networks trained with a backpropagation algorithm have been used as "compression networks". These networks are trained to reproduce their inputs through a narrow channel of hidden units. A compressed face code emerges at the level of the hidden units and then is used as input to a second classification network. These two-stage models have been applied to the tasks of face identification, and classification by sex and expression. [22–25]

Figure 1 shows a typology of the different connectionist and statistical models of face processing reported in the literature. It employs a correspondence analysis (or dual scaling analysis) of the models, using the criteria previously presented as classification descriptors (i.e., motivation, task, face representation, learning paradigm). Correspondence analysis is a multivariate statistical technique that gives a graphical description of a data matrix. The rows and the columns in the matrix are represented as 2 sets of points in a unique space so that the rows or the columns that are similar are displayed close together in the space map.[31,32]

Three principal classes of models emerge roughly from this typology. The first class is constituted by the models using an image-based coding of the faces in conjunction with a linear autoassociative learning or a principal component approach. These models have been applied to perceptual aspects of face processing including face recognition and categorization.[7,10-12,14-19] The second class includes models that use an image-based coding of the face as input to a compression and/or a backpropagation network. These models have been applied to both the perceptual and the semantic aspects of face processing.[22-25] The models composing the third class differ from the other models in two principal ways.

(1) They employ arbitrary codings to represent the faces.

(2) They have been designed to simulate high level cognitive phenomena such as semantic priming[13] or context effects in face memory.[20,21]

In what follows, only models that use an image-based coding of the faces are presented. As noted previously, other models[13,20,21] use codings that are not related to faces *per se* (i.e., arbitrary codes for hypothetical features). These models are, therefore, unable to give insights into the computational challenges posed by faces as pictorial or perceptual stimuli and are beyond the scope of this review.

3. AUTOASSOCIATIVE MEMORY—PRINCIPAL COMPONENT APPROACH

The current models of face processing using a linear autoassociative learning are derived from older work by Anderson et al.[28] and Kohonen.[29] Kohonen demonstrated that an autoassociative memory could be used to store and retrieve face images (i.e., as a content addressable memory for faces). This section provides a brief description of the model and its properties, followed by a presentation of its diverse applications to face processing.

3.1 Model description

We first define the notation that will be used in this section. Any particular face (say the k-th face) is represented by a column vector $\mathbf{x}_k$ of dimensionality $I \times 1$. This vector consists of the concatenated rows of pixel intensities extracted from digitized images of faces. For convenience, the vectors are assumed normalized so that $\mathbf{x}_k^T\mathbf{x}_k = 1$. The set of K faces to be stored in the memory is represented by an $I \times K$ matrix $\mathbf{X} = [x_{i,k}]$, with $x_{i,k}$ being the intensity of the i-th pixel for the k-th face and so $\mathbf{x}_k$ is the k-th column of $\mathbf{X}$. The autoassociative memory (or weight matrix) is represented by an $I \times I$ matrix $\mathbf{W} = [w_{i,i'}]$, where $w_{i,i'}$ represents the covariance between pixels i and i'. When an autoassociative memory is

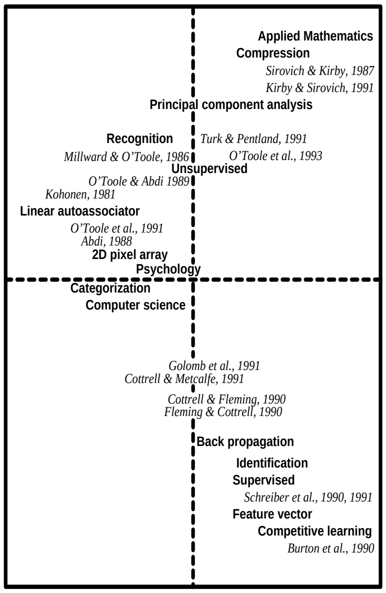

FIG. 1. — Typology of connectionist models of face processing.

face can be more similar to a particular unfamiliar face than to any other familiar face). In contrast, categorization is based on the physical similarity within subsets of faces. More specifically, it is based on the fact that faces within a category (e.g., female) share a set of common physical properties and therefore are in some way more similar to each other than to faces of other categories (e.g., male).

(2) Recognition and identification of a face constitute two dissociable processes: a face can be recognized as a familiar one, but not identified as the face of a specific person. This corresponds to the often reported feeling of being sure that we recognize a person, without being able to recall any information about the person (e.g., their name, from where we know them).

(3) In contrast with identification and recognition, categorization of a face according to its sex, race, or age does not require the face to be familiar.

Another way of classifying the models is to look at the different steps involved in face processing. The first and most problematic step in developing a model of face processing is to represent the faces in a way that makes it possible to distinguish a particular face from all other faces, including those not learned by the model. Whereas most connectionist models operate on image-based coding of the faces (2D pixel intensity array), some models represent the faces using arbitrary vectors representing hypothetical features of the faces.(13,20,21) These models are used to simulate cognitive and semantic tasks, rather than perceptual ones.

The second step involved in any computational model of face processing is to find an economical way of "storing" the faces in memory. In a connectionist network, due to the distributed representation of the stimuli, the faces on which the network has been trained can be thought of metaphorically as "stored" in the memory. This is a simplification of the actual process occurring during learning. In fact, the learning of a set of stimuli is the process by which the connection strengths are adjusted, based on these stimuli, to minimize some error criterion. For example, the error minimized for an autoassociative network is the error in reconstructing the input stimuli themselves. In contrast, the error that is minimized for a heteroassociative memory, is the error between the actual network response and the desired associated response (e.g., the name posed throughout the entire network, rather than stored in a localized area of the memory. With this method of storage the individual traces can be "retrieved" from the memory in the same way that a filter can selectively extract a simple frequency component from a complex acoustic waveform. Both nonconnectionist and connectionist approaches have contributed, in recent years, to our understanding of the properties of faces as complex visual patterns and have led to new insights into the way humans are able to process faces so effortlessly and efficiently.

In the following sections, we review a number of connectionist models of face processing that have been proposed in the literature. Because these models have been developed simultaneously in many fields with different motivations and objectives, it is difficult to determine an appropriate order of presentation. So, before going into the details of particular models, we shall present in Section 2 a brief classification scheme which yields two principal families of models: (a) autoassociative/principal component networks and (b) backpropagation/image compression networks. These two families of network models will be described in Sections 3 and 4, respectively, in terms of their applicability to face identification, recognition, and categorization. In Section 5, the computational efficiency of the different models will be compared. Finally, in Section 6, we shall discuss two additional models that do not fit neatly into our classification scheme. These "hybrid" models embody some of the computational aspects of connectionist models, as well as other more traditional aspects of template matching schemes.

2. CLASSIFICATION OF THE CONNECTIO NIST MODELS OF FACE PROCESSING

It is possible to classify connectionist models of face processing in a number of ways. A first important criterion is their *motivation.* The way a model is constructed depends largely on the motivations and objectives of its designers. For example, some face processing models are created to solve computational problems such as image compression or transmission.(7,10-12) These models are designed to produce efficient automatic systems of face recognition and to minimize the number of recognition errors produced by the system. Other models are directed at understanding and replicating the human ability to process faces. They are designed to simulate human behavior, including errors as well as correct performance.(13-21) Finally, some models are applied to both computational and cognitive issues.(22-25)

A second criterion that follows from the motivation is the *task* to be simulated by the model. The human ability to process faces is not limited to the *identification* of a face as the face of a specific person, it also includes different tasks such as the *recognition* of a face as a "known" face as opposed to an "unknown" face (feeling of familiarity) or the *categorization* of a face according to "visually derived dimensions"(26) such as sex, race, or age. Some additional models have been applied to other tasks such as face detection(7) or feature extraction.(27) We shall limit this discussion to face identification, recognition, and categorization. Three points are worth noting relative to these tasks.

(1) The kind of information necessary to identify a face is different from the kind of information that is needed to recognize or categorize a face. Whereas identification of a face involves a reference to semantic information (e.g., name of the person, context of acquaintance) in addition to visual information, both recognition and categorization are based solely on visual information. Recognition involves the comparison of a target face with the faces stored in memory. However, it is based not on the physical similarity within the category of familiar faces but on episodic information[†] associated with the faces (i.e., a familiar

[†] Episodic information is a psychological term referring to memory for autobiographical events or episodes.

tems that assume a feature-based representation of the faces. Typically, in these models, faces are represented in terms of distances, angles, and areas between elementary features such as the eyes, nose, chin, etc.,[2,3,4,5] or in terms of more complex mathematical functions like autocorrelation or moment invariants.[6] These descriptors are extracted from full facial views,[2,3] or from silhouette profiles[4,5,6]. Each face is then stored as a "feature vector" in a separate location in memory. Access to the memory is achieved by the computation of similarity between perceived items and memory traces. In brief, most of the effort in the feature-based models presented by Samal and Iyengar[1] is focused on finding individual features (e.g., eyes, mouth, head outline, etc.) and measuring statistical parameters to describe those features and their relationship. This type of approach constitutes an economical and efficient way of storing facial information as long as the features selected are appropriate and the task simulated by the model is limited to matching a target face with a face present in the database. However, selecting a set of features that captures the information appropriate for a given task is not easy. Despite the large number of algorithms proposed in the literature, no completely satisfactory solution has been found.

By contrast with non-connectionist models, which typically use geometrical face codings, connectionist models have generally operated directly on an image-based representation of faces (i.e., 2D pixel intensity array). Although this type of representation does not create an invariant 3D representation, when used with full-faces, it preserves both configural (i.e., the relationship between features) and feature-based information. Thus, geometric representations are coded implicitly, but in addition, texture and detailed shape information are preserved. An oft-cited limitation of pixel intensity representations taken from images of faces is that they are sensitive to substantial variations in lighting condition, head orientation, and size. To avoid these problems, a preprocessing of the faces (i.e., normalization for size and position) is necessary. However, this can be done in a relatively straightforward manner by using automatic algorithms for locating the faces in the images and rescaling them.[7]

The purpose of this review is to complement the Samal and Iyengar[1] survey with a review of connectionist or statistical approaches to the problems of automatic face recognition and categorization. Connectionist approaches to pattern recognition are defined generally as those using computational algorithms that can be carried out in parallel and that use distributed or non-localized mechanisms of memory storage. While there is no *a priori* reason for these models to use an image-based coding of the faces, this is the kind of coding that has been employed most frequently with connectionist models of face processing‡. One advantage of using connectionist models with an image-based coding is that the information used in these models is automatically derived from the statistical structure of the faces, and therefore, the difficult problem of selecting individual features is avoided.

The term "connectionist model" was first proposed by Feldman[8,9] to describe a class of models suggested by the anatomy and physiology of the brain. These models are networks of simple interconnected processing units that operate in parallel. The processing units are often referred to as "neuron-like", since their function is metaphorically analogous to that of neurons. Each unit receives inputs from other units, integrates (e.g., sums) them, and generates an output that is sent to other units. Information processing occurs in parallel through the interactions among the units. Learning results from the modification of the weights of the connections between units. Connectionist models of face processing differ from the models presented in the Samal and Iyengar[1] survey in at least two important ways. First, information is processed in parallel, rather than sequentially, by a large number of highly interconnected simple units. Second, memory traces are spatially distributed and superim-

IN: *Pattern Recognition. Vol. 27, pp. 1209–1230, 1994.*

CONNECTIONIST MODELS OF FACE PROCESSING: A SURVEY

DOMINIQUE VALENTIN[†], HERVÉ ABDI[†‡], ALICE J. O'TOOLE[†], GARRISON W. COTTRELL[§]

[†] School of Human Development, The University of Texas at Dallas, Richardson, TX 75083-0688. [‡] Dept. of Psychology, Université de Bourgogne à Dijon, 21004, Dijon Cedex, France. [§] Dept. of Computer Science & Engr., 0114, Institute for Neural Computation, University of California at San Diego, La Jolla, CA, 92093-0114.

Received 1 February 1994, received for publication 24 January 1994

Abstract—Connectionist models of face recognition, identification, and categorization have appeared recently in several disciplines, including psychology, computer science, and engineering. We present a review of these models with the goal of complementing a recent survey by Samal and Iyengar(1) of nonconnectionist approaches to the problem of the automatic face recognition. We concentrate on models that use linear autoassociative networks, nonlinear autoassociative (or compression) and/or heteroassociative backpropagation networks. One advantage of these models over some nonconnectionist approaches is that analyzable "features" emerge naturally from image-based codes, and hence the problem of feature selection and segmentation from faces can be avoided.

Faces   Pattern recognition   Image-based coding   Neural network   Back-propagation
Principal component analysis   Macrofeatures

1. INTRODUCTION

To interact socially with people we must be able to process faces in a variety of ways. There is a vast literature in social and cognitive psychology attesting to the impressive abilities of human observers at recognizing and identifying familiar faces, as well as extracting additional information from both familiar and unfamiliar faces, including sex, approximate age, race, and current emotional state of the person[†]. While we accomplish these tasks with ease, the automatic recognition and categorization of faces pose a rather unique set of computational challenges, which Samal and Iyengar(1) have detailed in their recent survey of automatic face recognition systems. In their review, Samal and Iyengar(1) have concentrated on automatic face identification sys-

[†] Henceforth we shall refer globally to these different tasks as face processing tasks or face processing.
[‡] Some connectionist models using a different type of coding are cited at the end of this paper.