Evaluation of Character Recognition Systems

C. L. Wilson
National Institute of Standards and Technology
Gaithersburg, MD 20899

Neural Networks for Signal Processing III, IEEE, New York, pp. 485-496.

Abstract

At the first Census Optical Character Recognition Systems (COCRS) Conference, the National Institute of Standards and Technology (NIST) produced accuracy data for more than 40 character recognition systems. The recognition experiments were performed on sample sizes of 58,000 digits and 12,000 upper and lower case alphabetic characters. The algorithms used by the 26 conference participants included rule-based methods, image-based methods, statistical methods, and neural networks. The neural network methods included Multi-Layer Perceptrons, Learned Vector Quantitization, Neocognitrons, and cascaded neural networks.

In this paper 11 different COCRS systems are evaluated using correlations between the answers of different systems, comparing the decrease in error rate as a function of confidence of recognition, and comparing the writer-dependence of recognition. This comparison shows that methods that used different algorithms for feature extraction and recognition performed with very high levels of correlation. Subsquent experiments were performed by NIST to compare the OCR accuracy of various neural network and statistical classification systems. For each neural network system a statistical system of comparable accuracy was developed. These experiments tested seven different classifiers using 11 different feature sets and obtained OCR error levels between 2.5% and 5.1% for the best feature set sizes. This similarity in accuracy is true for neural network systems and statistically based systems and leads to the conclusion that neural networks have *not* yet demonstrated a clear superiority to more conventional statistical methods in either the COCRS test or in independent tests at NIST.

1 Introduction

At the first COCRS Conference a large number of systems (40 for digits) were used to recognize the same sample of characters [1]. A summary of these results is given in Table 1. Neural network systems, systems combining neural network methods with other methods (hybrid system), and systems based entirely on statistical pattern recognition methods were submitted to the COCRS conference. This provides a large test sample which can be used to detect differences between these various methods. In addition, subsequent test at NIST [2] using a seven different neural network (NN) and statistical classifiers confirmed that using a fixed set of features both types of methods have similar accuracies.

In this paper 11 different COCRS conference systems are discussed in some detail. These system are itemized by type in Table 2. These systems are broken into NN based systems, hybrid systems, and non-NN systems. The author realizes that this distinction is subject to interpretation, but it does allow some useful comparisons to be made. The COCRS conference systems were all designed to use different methods of feature extraction. In order to separate feature extraction and classification, the image recognition group at NIST performed classification experiments using seven methods of classification and a common set of Karhunen-Loève (KL) based features [3]. The results of these tests are shown in Table 3.

Entered	Percentage Classification Error		
System	Digits	Uppers	Lowers
AEG	3.43 ± 0.23	3.74 ± 0.82	12.74 ± 0.75
ASOL	8.91 ± 0.39	11.16 ± 1.05	21.25 ± 1.36
ATT_1	3.16 ± 0.29	6.55 ± 0.66	13.78 ± 0.90
ATT_2	3.67 ± 0.23	5.63 ± 0.63	14.06 ± 0.95
ATT_3	4.84 ± 0.24	6.83 ± 0.86	16.34 ± 1.11
ATT_4	4.10 ± 0.16	5.00 ± 0.79	14.28 ± 0.98
COMCOM	4.56 ± 0.91	16.94 ± 0.99	48.00 ± 1.87
ELSAGB_1	5.07 ± 0.32		
ELSAGB_2	3.38 ± 0.20		
ELSAGB_3	3.35 ± 0.21		
ERIM_1	3.88 ± 0.20	5.18 ± 0.67	13.79 ± 0.80
ERIM_2	3.92 ± 0.24		
GMD_1	8.73 ± 0.35	14.04 ± 1.00	22.54 ± 1.22
GMD_2	15.45 ± 0.64	24.57 ± 0.91	28.61 ± 1.25
GMD_3	8.13 ± 0.39	14.22 ± 1.09	20.85 ± 1.25
GMD_4	10.16 ± 0.35	15.85 ± 0.95	22.54 ± 1.22
GTESS_1	6.59 ± 0.18	8.01 ± 0.59	17.53 ± 0.75
GTESS_2	6.75 ± 0.30	8.14 ± 0.59	18.42 ± 1.09
HUGHES_1	4.84 ± 0.38	6.46 ± 0.52	15.39 ± 1.10
HUGHES_2	4.86 ± 0.35	6.73 ± 0.64	15.59 ± 1.08
IBM	3.49 ± 0.12	6.41 ± 0.80	15.42 ± 0.95
IFAX	17.07 ± 0.34	19.60 ± 1.26	
KAMAN_1	11.46 ± 0.41	15.03 ± 0.79	31.11 ± 1.15
KAMAN_2	13.38 ± 0.49	20.74 ± 0.88	35.11 ± 1.09
KAMAN_3	13.13 ± 0.45	19.78 ± 0.60	33.55 ± 1.37
KAMAN_4	20.72 ± 0.44	27.28 ± 1.30	46.25 ± 1.23
KAMAN_5	15.13 ± 0.41	33.95 ± 1.22	42.20 ± 0.96
KODAK_1	4.74 ± 0.37	6.92 ± 0.78	14.49 ± 0.77
KODAK_2	4.08 ± 0.26		
MIME	8.57 ± 0.34	10.07 ± 0.81	
NESTOR	4.53 ± 0.20	5.90 ± 0.68	15.39 ± 0.90
NIST_1	7.74 ± 0.31	13.85 ± 0.83	18.58 ± 1.12
NIST_2	9.19 ± 0.32	23.10 ± 0.88	31.20 ± 1.16
NIST_3	9.73 ± 0.29	16.93 ± 0.90	20.29 ± 0.99
NIST_4	4.97 ± 0.30	10.37 ± 1.28	20.01 ± 1.06
NYNEX	4.32 ± 0.22	4.91 ± 0.79	14.03 ± 0.96
OCRSYS	1.56 ± 0.19	5.73 ± 0.63	13.70 ± 0.93
REI	4.01 ± 0.26	11.74 ± 0.90	
RISO	10.55 ± 0.43	14.14 ± 0.88	21.72 ± 0.98
SYMBUS	4.71 ± 0.38	7.29 ± 1.07	
THINK_1	4.89 ± 0.24		
THINK_2	3.85 ± 0.33		
UBOL	4.35 ± 0.20	6.24 ± 0.66	15.48 ± 0.81
UMICH_1		5.11 ± 0.94	15.08 ± 0.92
UPENN	9.08 ± 0.37		
VALEN_1	17.95 ± 0.59	24.18 ± 1.00	31.60 ± 1.33
VALEN_2	15.75 ± 0.32		

Table 1: Mean zero-rejection-rate error rates and standard deviations in percent calculated over 10 partitions of the COCRS conference test data. See [1] for details

In the past few years NN's have become important as a possible method for constructing computer programs that can solve problems, such as speech and character recognition, where "human-like" response or artificial intelligence is needed. The most useful characteristics of NN's are their ability to learn from examples, their ability to operate in parallel, and their ability to perform well using data that are noisy or incomplete. Many of these characteristics are shared by various statistical pattern recognition methods. These characteristics of pattern recognition systems are important for solving real problems from the field of character recognition exemplified by this paper.

It is important to understand that the accuracy of the trained OCR system produced will be strongly dependent on both the size and the quality of the training data. Many common test examples used to demonstrate the properties of pattern recognition system contain on the order of 10^2 examples. These examples show the basic characteristics of the system but provide only an approximate idea of the system accuracy.

As an example, the first version of an OCR system was built at NIST using 1024 characters for training and testing. This system has an accuracy of 94%. As the sample size was increased the accuracy initially dropped as more difficult cases were included. As the test and training sample reached 10000 characters the accuracy began to slowly improve. The poorest accuracy achieved was with sample sizes near 10^4 and was 85%. The 58,000 digit sample discussed in this paper is well below the 10^5 character sample size which we have estimated is necessary to saturate the learning process of the NIST system [3]. The best system developed by NIST uses probabilistic NNs (PNN) [4] and achieved an error of 2.5% when trained on 7480 digits.

The goal of this paper is to discuss the different kinds of methods used at the COCRS Conference in a way that will illustrate why NN's and statistical methods achieved similar levels of performance. The various methods used are summarized in Figure 1 for classification and feature extraction. Most of the systems presented at the Conference used separate methods of feature extraction and classification. In the discussion presented here any image processing which preceded the feature extraction is combined with feature extraction. The results of these comparisons are presented in sections 2 by algorithm type and in section 3 for NN and statistical algorithms.

Since the results of the COCRS conference were not what was originally expected, NIST conducted a set of pattern classification experiments using KL features, sets of different sizes, and using seven different classification methods. These experiments confirm the COCRS conference results. These results are discusssed in section 5.

2 Types of Algorithms Used

The discriminant function and classification sections of the systems are of two types: adaptive learning based and rule-based. The most common approach to machine learning based systems used at the Conference was NNs. The neural approach to machine learning was originally devised by Rosenblat [5] by connecting together a layer of artificial neurons [6] on a perceptron network. The observations which were present in this approach were analyzed by Minski and Papert [7]. The results of this Conference suggest that many of these weaknesses are still relevant. The advent of new methods for network construction and training during the last ten years led to rapid expansions in NN research in the late 1980s. Many of the methods referred to in Figure 1 were developed in this period. Adaptive learning is further subdivided into two types, supervised learning and self-organization. The material presented in this paper does not cover the mathematical detail of these methods, but the bibliographic references provided with many of the systems [1] discuss these methods in detail.

The principal difference between NN methods and rule-based methods is that the former attempt to simulate intelligent behavior by using adaptive learning and the latter use logical symbol manipulation. The two most common rule-based approaches at the Conference were those derived from mathematical image processing and those derived from statistics. Image based methods are usually used for feature extraction while statistical methods are usually used for classification.

Most of the OCR implementations discussed in this report combine several methods to carry out preprocessing (filtering) and feature extraction. Many of the filtering methods used are based on methods described in texts on image processing such as [8] and on methods based on KL transforms [3]. In these methods, the recognition is done using features extracted from the primary image by rule based techniques. The filtering and feature extraction processes start with an image of a character. The features produced are then used as the input for classification.

In a self-organizing method, such as [9], data is applied directly to the NN and any filtering is learned as features are extracted. In a supervised method, the features are extracted using either rule-based or adaptive methods and classification is carried out using either type of method.

In Figure 1, rules based on mathematical image processing are distinguished from rules based on statistics. These two types of rules are similar in that they both derive features based on a model of the images. Statistical rules derive these model parameters based on the data presented. For example, typical model parameters might be sample means and variances. Mathematical rules operate on the data based on external model parameters or on the specific data being analyzed. The model parameters might be designed to detect strokes, curvature, holes, or concave or convex surfaces.

All of the methods shown in Figure 1 can also be categorized broadly into linear methods, such as LVQ [10], and nonlinear methods, such as Multi-Layer Perceptrons (MLPs) [11]. This separation into linear and non-linear algorithms also extends to mathematical and statistical methods. Many of the convolution and transform methods, such as combinations of Gabor transforms [12], are linear. Other methods start with linear operations such as correlation matrices and become non-linear by removing information with low statistical significance; KL transforms [8] and principal component analysis (PCA) [13] are examples of this.

When training data is used to adjust statistical model parameters to train MLPs, certain methods may be classified as either NN or statistical methods. The PNN [4] is an example of this type of method. In another context PNN methods can be regarded as one class of a radial basis function (RBF) method [14]. The information in Figure 1 classifies methods of this kind in an arbitrary way when statistical accumulation or NN models of a given method are equivalent.

System	Features	Classification
	Neural Net	
ATT_2	receptor fields	MLP
Hughes_1	neocognitron	
Nestor	necognitron	MLP
Symbus	raw	self-Org. NN
	Hybrid	
ERIM_1	morophological	MLP
Kodak_2	Gabor	MLP
NYNEX	model	MLP
NIST_4	K-L	PNN
	Non Neural Net	
Think_1	template	distance maps
UBOL	rule based	KNN
Elsagb_1	shape func.	KNN

Table 2: Feature extraction and classification methods used for the 11 system discussed.

System	24	28	32	36	40	44	48	52	56	60	64
KNN:1	2.9	2.7	2.7	2.7	**2.6**	2.6	2.6	2.7	2.7	2.7	2.7
KNN:3	2.8	2.7	2.7	2.7	**2.6**	2.7	2.7	2.7	2.7	2.8	2.7
KNN:5	2.9	2.8	2.8	**2.7**	2.8	2.8	2.8	2.8	2.8	2.8	2.8
WSNN:1.1	2.8	2.7	2.6	2.6	**2.5**	2.6	2.6	2.6	2.6	2.5	2.6
PNN:3.0	2.7	2.7	2.6	2.6	**2.5**	2.6	2.6	2.6	2.6	2.5	2.5
MLP:32	5.8	5.6	5.7	5.5	5.6	5.5	**5.3**	5.4	5.4	5.3	5.4
MLP:48	5.2	5.2	5.0	4.7	4.9	5.0	4.7	**4.6**	4.9	5.0	4.9
MLP:64	4.6	4.5	4.6	4.5	4.5	4.5	4.5	**4.3**	4.5	4.4	4.5
RBF1:1	13.2	13.1	13.9	13.0	**12.6**	13.4	12.6	13.2	13.3	13.2	13.2
RBF1:2	8.5	8.5	8.4	8.2	8.4	8.2	8.1	8.3	8.1	**7.9**	7.9
RBF1:3	6.7	6.6	6.5	6.5	6.5	6.4	6.4	**6.2**	6.4	6.2	6.3
RBF1:4	5.7	5.5	5.5	5.5	5.4	5.5	5.4	5.4	**5.3**	5.3	5.4
RBF1:5	5.0	4.7	4.9	5.0	4.9	4.8	4.7	4.9	4.9	4.7	**4.6**
RBF1:6	4.6	4.4	4.3	4.5	4.3	4.3	**4.2**	4.2	4.4	4.3	4.4
RBF2:1	8.7	9.5	9.1	9.1	9.2	**8.6**	8.8	8.8	8.9	8.9	8.9
RBF2:2	6.7	6.4	**6.1**	6.1	6.3	6.3	6.2	6.3	6.2	6.2	6.5
RBF2:3	5.6	5.5	5.0	6.0	5.4	**4.9**	5.7	4.9	5.0	5.6	5.0
RBF2:4	4.4	5.6	5.0	**4.3**	4.5	4.6	4.6	4.5	4.4	4.8	4.7
RBF2:5	4.5	4.6	4.4	4.6	4.4	4.4	4.4	4.2	4.1	4.1	**4.0**
RBF2:6	4.3	4.5	4.0	4.0	4.2	**3.9**	4.2	4.0	3.9	4.0	4.0
EMD:1	15.2	15.1	15.0	15.0	14.9	14.9	**14.8**	14.8	14.8	14.8	14.8
EMD:2	11.0	10.8	10.7	10.7	10.7	10.7	10.7	**10.6**	10.6	10.6	10.6
EMD:3	8.8	8.8	8.7	**8.6**	8.6	8.7	8.7	8.7	8.7	8.7	8.7
EMD:4	7.3	7.3	7.4	7.3	**7.1**	7.2	7.1	7.1	7.1	7.1	7.1
EMD:5	6.7	6.6	6.6	6.5	6.3	6.7	6.6	**6.2**	6.2	6.2	6.3
EMD:6	6.1	5.9	6.1	6.0	**5.7**	6.0	5.8	5.9	6.0	5.9	6.1
EMD:7	5.6	5.3	5.5	5.3	5.2	5.4	**5.1**	5.2	5.4	5.4	5.6
QMD:1	**4.8**	4.9	5.1	5.1	5.2	5.3	5.6	5.6	5.8	5.8	5.9
QMD:2	**4.7**	4.9	4.9	5.0	5.2	5.3	5.5	5.6	5.7	5.8	5.9
QMD:3	**4.0**	4.5	4.7	4.9	5.1	5.3	5.4	5.6	5.9	6.0	6.3
QMD:4	**4.5**	4.9	5.0	5.3	5.5	6.1	6.3	6.5	6.9	7.2	7.6
NRML	**4.8**	4.9	5.0	5.0	5.2	5.3	5.5	5.6	5.5	5.5	5.6

Table 3: Dependence of Classification Error on KL Transform Feature Set Dimensionality. Given with the classifier acronym are: For k-NN the value of k, for WSNN the value of α, for PNN the value of σ, for MLP networks the number of hiddens units, for RBF networks the number of centers per class, and for EMD and QMD classifiers the number of clusters per class. Bold type indicates the dimensionality yielding minimum error for each classifier. See [2] for more detailed discussion.

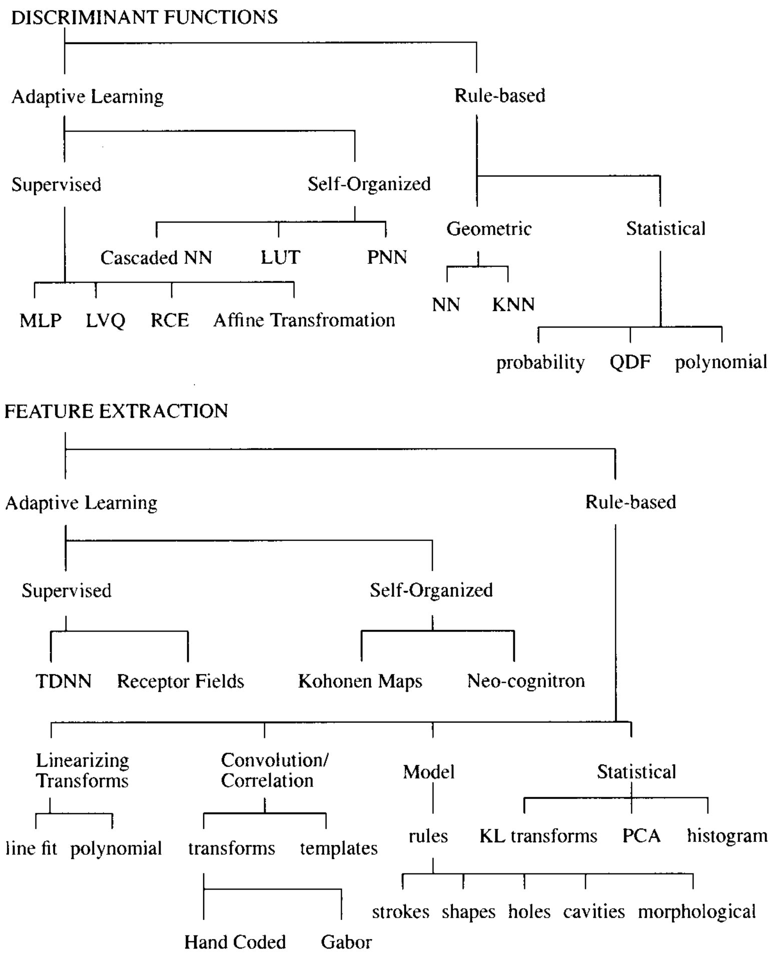

Figure 1: Types of methods used for feature extraction and classification.

Two types of data will be used to compare the neural and non-neural recognition systems. First the recognition accuracy as a function of reject rate is used and second the writer dependence as a function of reject rate is used.

Comparison of NN and statistical systems shows that with no rejection the neural and hybrid systems have errors between 3.67% (ATT_2) and 4.84% (HUGHES_1). The statistical systems have errors between 4.35% (UBOL) and 5.07% (ELSAGB_1). Since the standard deviations on these numbers is typically ±0.3% a significant overlap in performance exists. The best and worst neural systems are 4 standard deviations apart and the statistical systems are about 2 standard deviations apart. Across the range of measured performance, the statistical systems can not be distinguished from each other. Across this same range of performance the neural systems can be distinguished from each other. As the fraction of characters rejected increases, the variation in accuracy increases for the NN system while the statistical systems remain tightly grouped. At 30% rejection the best NN system has an error of 0.15% (ATT_2) and the worst NN system has an error of 0.52% (SYMBUS). At the same rejection rate THINK_1 has an error of 0.27% and NIST_4 has an error rate of 0.21%. At high reject rates the statistical systems are nearing the performance of the better NN systems and are significantly better than the worst NN system. For further details see [1],[15].

For the writer dependence of NN and statistical systems, the greatest writer differentiation, 50 writers, occurs at a reject rate of 5%. The best systems in terms of error have the least writer sensitivity. This is not because these systems get more writers correct at zero reject but because no system from either group gets over 80 writers correct at zero rejection. This separation of systems exists because, when the worst characters from each writer are removed, the best system from each group obtains a 50 writer advantage as the first 5% of the characters are rejected. Writer dependence is less significant in distinguishing systems than error performance. For further details see [1],[15].

3 NIST Classification Experiments

NIST evaluated four statistical classifiers and three NN classifiers. The statistical classifiers are Euclidean Minimum Distance (EMD), Quadratic Minimum Distance (QMD), Normal (NRML), and k-Nearest Neighbor (k-NN). The three neural classifiers included in the evaluation are the MLP, RBF, and PNN. For a given application, all the classifiers were given the same feature sets. Misclassification errors using a 23140 dataset are tabulated as a function of feature dimension and classifier parameters such as the number of prototypes. Table 3 shows for each classifier the estimated probabilities of *error*, expressed as percentages, for increasing dimensionality of the KL feature set. Note that the optimal number of features yielding lowest classification error (shown in bold) is not the same for all classifiers, the parametric classifiers, QMD and NRML, being noticeably more parsimonious in the number of features required. It is also apparent that most of the classifiers essentially attain a plateau as the number of features reaches approximately 32 thereafter only gaining several tenths of a percent. The best classifiers are the computationally expensive nearest neighbor classifiers and the related PNN. They achieve one third less errors than the NNs and parametric classifiers. The optimum value of $\alpha = 1.1$ for WSNN corresponds to a 1-NN scheme for most test patterns. Accordingly, k-NN is seen to have a higher error rate for increasing k.

Two caveats should be made about the table. First, the MLP and RBF results depend on the initial guesses for the parameters. Often a number of different random guesses are tried assess the effect of the initial guess; for this table, because of the magnitude of the calculation necessary, only one initial guess was used.

These results show that for character classification accuracy NN methods and statistical methods have comparable accuracies confirming the COCR results.

4 Conclusions

Examination of the results of 11 OCR systems using a wide variety of recognition algorithms has shown that in accuracy and writer independence NN systems have not demonstrated a clear cut superiority over statistical methods. Some neural systems have higher accuracy than statistical methods; others have lower accuracy. The performance of statistical methods is more closely grouped and is approximately the same as the performance of an average NN system considered here. One area where NN's may have an advantage is in speed of implementation and recognition.

Examination of Table 3 show that on OCR classification the ranking of the methods is similar. The neighbor-based methods are the most accurate with PNN being the best of these. The comparison of MLP and RBF methods shows that RBF is usually the better method. When MLP and RBF methods are compared to multicluster EMD and QMD methods the NN methods are more straightforward to implement but do not show a clear accuracy advantage. All of the experiments presented here also suggest that the training set sizes used, although large, are not sufficient to fully saturate most of the machine learning methods studied here.

Acknowledgement

The author would like to acknowledge Patrick Grother for providing tables 1 and 3, Jon Geist for assistance in interpretation of results of classifier comparisons, and Rama Chellappa for help in designing the comparison experiments.

References

[1] R. A. Wilkinson, J. Geist, S. Janet, P. J. Grother, C. J. C. Burges, R. Creecy, B. Hammond, J. J. Hull, N. J. Larsen, T. P. Vogl, and C. L. Wilson. The First Optical Character Recognition Systems Confernce. Technical Report NISTIR 4912, National Institute of Standards and Technology, August 1992.

[2] Patrick J. Grother and Gerald T. Candela. Comparison of Handprinted Digit Classifiers. Technical Report NISTIR 5209, National Institute of Standards and Technology, June 1993.

[3] P. J. Grother. Karhunen Loève feature extraction for neural handwritten character recognition. In *Proceedings: Applications of Artificial Neural Networks III*. Orlando, SPIE, April 1992.

[4] Donald F. Specht. Probabilistic neural networks. *Neural Networks*, 3(1):109–118, 1990.

[5] F. Rosenblatt. The perceptron: a probabilistic model for information storage and organization in the brain. *Psychological Review*, 65:386–408, 1958.

[6] W. S. McCulloch and W. Pitts. A logical calculus of the ideas immanent in nervous activity. *Bull. Math. Biophysics*, 9:115–133, 1943.

[7] M. Minsky and S. Papert. *Perceptrons*. MIT Press, Cambridge, MA, 1969.

[8] Anil K. Jain. *Fundamentals of Digital Image Processing*, chapter 5.11, pages 163–174. Prentice Hall Inc., prentice hall international edition, 1989.

[9] K. Fukushima. Neocognitron: A self-organizing neural network model for mechanism of pattern recognition unaffected by shift in position. *Biological Cybernetics*, 36:193–202, 1980.

[10] T. Kohonen. *Self-Organization and Associative Memory.* Springer-Verlag, Berlin, second edition, 1988.

[11] D. E. Rumelhart, G. E. Hinton, and R. J. Williams. Learning internal representations by error propagation. In D. E. Rumelhart and J. L. McClelland, et al., editors, *Parallel Distributed Processing: Explorations in the Microstructure of Cognition. Volume 1: Foundations*, chapter 8, pages 318–362. MIT Press, Cambridge, MA, 1986.

[12] J. G. Daugman. Complete discrete 2-d Gabor transform by neural networks for image analysis and compression. *IEEE Trans. on Acoustics, Speech, and Signal Processing*, 36:1169–1179, 1988.

[13] T. P. Vogl, K. L. Blackwell, S. D. Hyman, G. S. Barbour, and D. L. Alkon. Classification of Japanese Kanji using principal component analysis as a preprocessor to an artificial neural etwork. In *International Joint Conference on Neural Networks*, volume 1, pages 233–238. IEEE and International Neural Network Society, 7 1991.

[14] T. Poggio and F. Girosi. Networks for approximation and learning. *Proceedings of the IEEE*, 78(9):1481–1497, 1990.

[15] Charles L. Wilson. Effectiveness of Feature and Classifier Algorithms in Character Recognition Systems. In D. P. D'Amato, editor, , volume 1906. SPIE, San Jose, 1993.