Effectiveness of Feature and Classifier Algorithms in Character Recognition Systems

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

Abstract

At the first Census Optical Character Recognition Systems Conference, NIST generated accuracy data for more than 40 character recognition systems. Most system were tested on the recognition of isolated digits and upper and lower case alphabetic characters. 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 Perceptron's, Learned Vector Quantitization, Neocognitrons, and cascaded neural networks.

In this paper 11 different systems are compared 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. This is true for neural network systems, hybrid 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. Comparison of these results with the models of Vapnick (for estimation problems), MacKay (for Bayesian statistical models), Moody (for effective parameterization), and Boltzmann models (for information content) demonstrate that as the limits of training data variance are approached, all classifier systems have similar statistical properties. The limiting condition can only be approached for sufficiently rich feature sets because the accuracy limit is controlled by the available information content of the training set, which must pass through the feature extraction process prior to classification.

1 Introduction

At the first Census OCR System Conference a large number of systems (40 for digits) were used to recognize the same sample of characters [1]. Neural network systems, systems combining neural network methods with other methods (hybrid system), and systems based entirely on statistical pattern recognition methods were used. This provides a large test sample which can be used to detect differences between these various methods. In this paper 11 different systems are discussed. These system are itemized by type in Table 1. These systems are

broken into neural network based systems, hybrid systems, and non- neural network systems. The author realizes that this distinction is subject to interpretation, but it does allow some comparisons to be made.

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

In the past few years neural networks 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 neural networks 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 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 [6].

The goal of this paper is to compare the different methods used at the Census OCR Conference in a way that will illustrate why neural networks and rule based 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.

2 Types of Algorithms Used

2.1 Rule-based versus Machine learning

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 neural networks. The neural approach to machine learning was originally devised by Rosenblat [2] by connecting together a layer of artificial neurons [3] on a perceptron network. The weaknesses which were present in this approach were analyzed by Minski and Papert [4]. The results of this Conference suggest that many of these weaknesses are still important. The advent of new methods for network construction and training during the last ten years led to rapid expansions in neural network 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 neural network 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.

The alternate approach to recognition machine construction is rule-based. Rather than teaching the program to differentiate between characters, a rule- based program is constructed to distinguish among the various characters by writing rules to be followed by the system. These are explicitly programmed in the system in the form of mathematical formulas.

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 [5] and on methods based on Karhunan Loeve (KL) transforms [6]. 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 [7], data is applied directly to the neural network 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.

2.2 Statistical Rules versus Mathematical Rules

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.

2.3 Linear versus Non-linear Methods

All of the methods shown in Figure 1 can also be classed broadly into linear methods, such as LVQ [8], and nonlinear methods, such as Multi-Layer Perceptrons (MLPs) [9]. 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 [10] are linear. Other method start with linear operations such as correlation matrices and become non-linear by removing information with low statistical significance: KL transforms [5] and principal component analysis (PCA) [11] are examples of this.

2.4 Statistical and Neural Methods

When training data is used to adjust statistical model parameters to train MLPs, certain methods may be classed as either neural network or statistical methods. The probabilistic neural network (PNN) [12] 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 [13]. The information in Figure 1 classifies methods of this kind in an arbitrary way when statistical accumulation or neural network models of a given method are equivalent.

3 Comparison of Neural and Non-Neural Systems

Two types of data will be used to compare the neural and non-neural recognition system. 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. The reject accuracy data for the neural and hybrid systems is shown figure 2. Equivalent data for the non-neural systems and NIST_4 is shown in figure 3.

Comparison of Figure 2 with Figure 3 shows that with no reject 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 $\pm 0.3\%$ a significant overlap in performance exists. The best and worst neural systems are 4 standard deviations apart and the statistical system are about 2 standard deviations apart. Across the range of measured performance, the statistical systems can not be distinguished however the neural systems can. As the fraction of characters rejected increases, the variation in accuracy increases for the neural network system while the statistical systems remain tightly grouped. At 30% rejection the best neural network system has an error of 0.15% (ATT_2) and the worst neural network 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 better neural network systems and are significantly better than the worst neural network system.

The writer dependence data for the neural and hybrid systems is shown figure 4. Equivalent data for the non-neural systems and NIST_4 is shown in figure 5. For both kinds of

system 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 writer 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.

4 System Speed

One of the ways neural networks might establish a technological edge over other methods is to achieve superior speed due to parallel implementation. The data from the systems conference illustrates the difficulty of evaluating speed differences.

Figure 6 shows the flow of data through a typical page level OCR system. The details of the particular system are discussed in [14]. The tests run for the OCR Systems Conference were conducted on a simplified problem in which the characters were isolated and segmented prior to being used by the conference participants. The only modules used for conference testing were normalization, filtering/feature extraction, recognition, and rejection. The load and store modules were present in either the full system or the simplified test system. The conference did not address field isolation and character segmentation.

Typical timings for a system of the type shown in Figure 6 are given in Table 2. The dominant times in this table are for image loading, field isolation, and character segmentation times. In the conference systems, field isolation and character segmentation times were not required so that the dominant time for the conference systems is the image loading time. Two times were tabulated: the total system time and the recognition time. In most cases, total system time is much longer than recognition time. This speed difference increases as recognition time decreases. Most systems have similar load times but recognition times vary by several orders of magnitude. The minimum recognition time is less than 1ms/character. The typical load time is near 100ms/character. These two times place distinct bounds on system performance. The recognition rate of the faster systems is near the present state-of-the-art for recognition performance and was achieved by neural network based systems. The system rate is near the typical speed that can be achieved loading and decompressing image data on common present-day desk-top systems.

In order to evaluate the performance bounds of possible systems, some knowledge of both algorithmic complexity and the importance of the algorithm in the overall system performance are needed. This can be accomplished by breaking the system into separate components each of which contains only one dominant algorithmic process or by measuring the full system performance on the specific application of interest. The importance of the scaling of algorithms in this context has been known since the early work on neural networks [4].

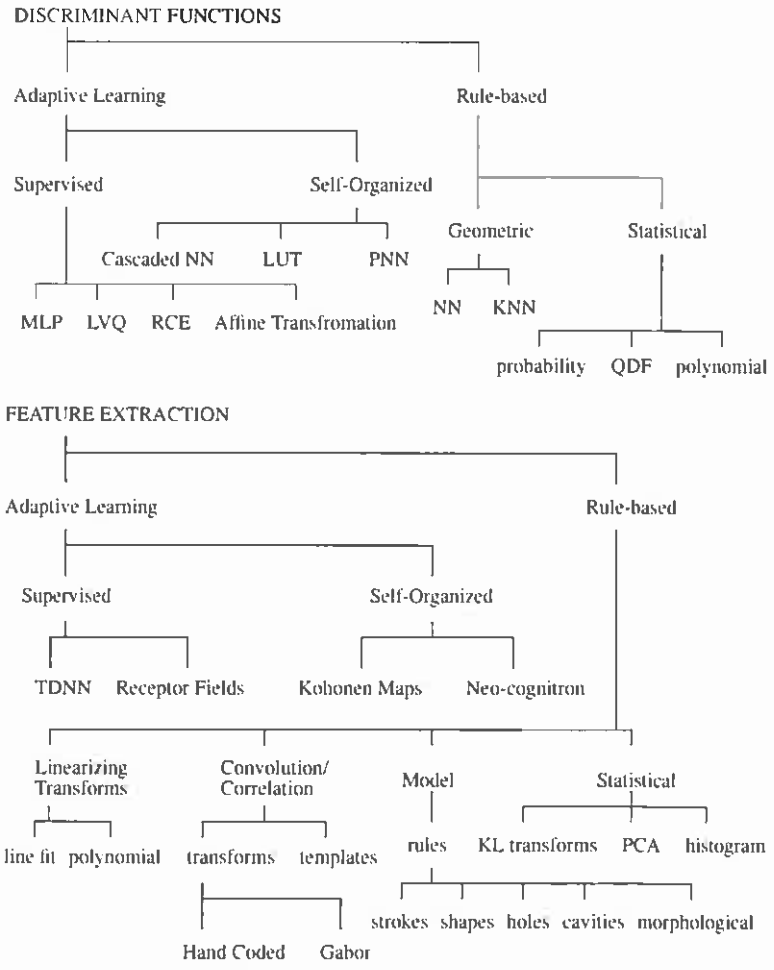

Figure 1: Types of methods used for feature extraction and classification.

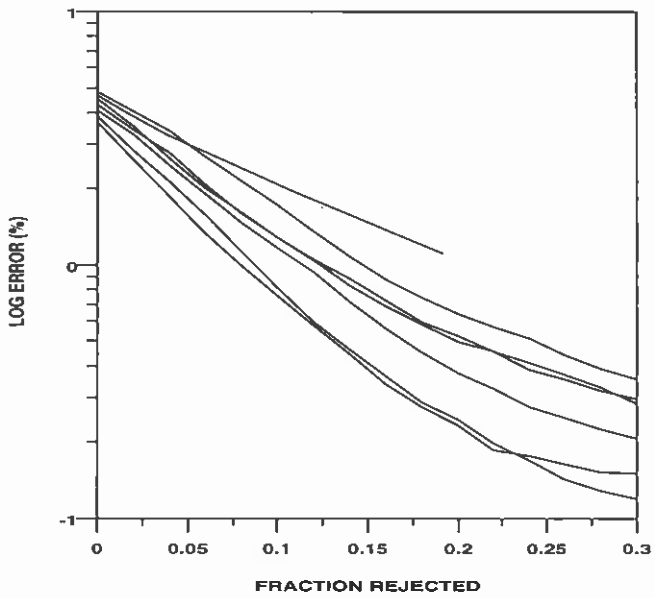

Figure 2: Reject versus error curves for six neural network based OCR systems.

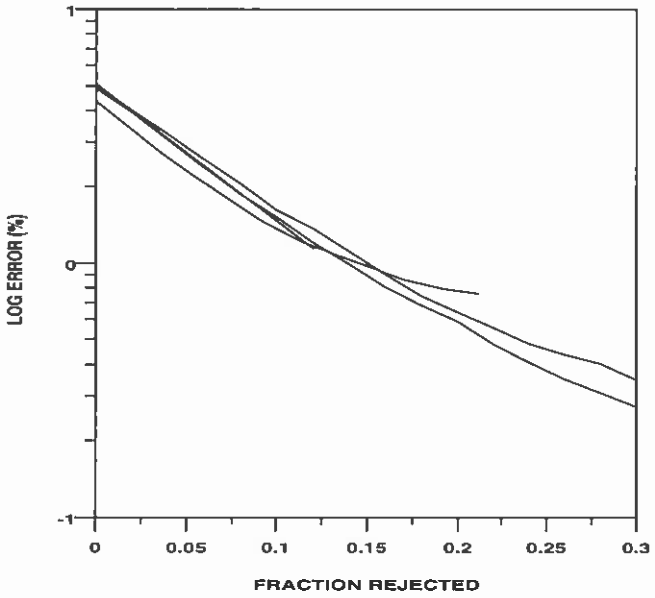

Figure 3: Reject versus error curves for four non-neural network based OCR systems.

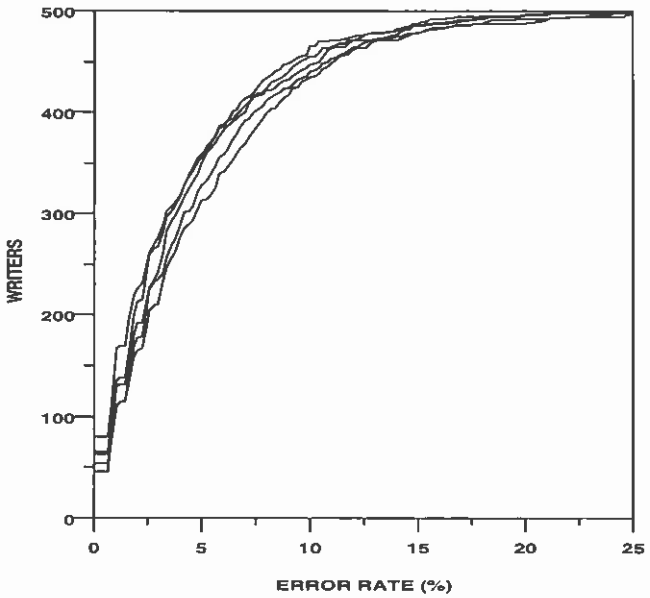

Figure 4: Writer dependence of error for six neural network based OCR systems.

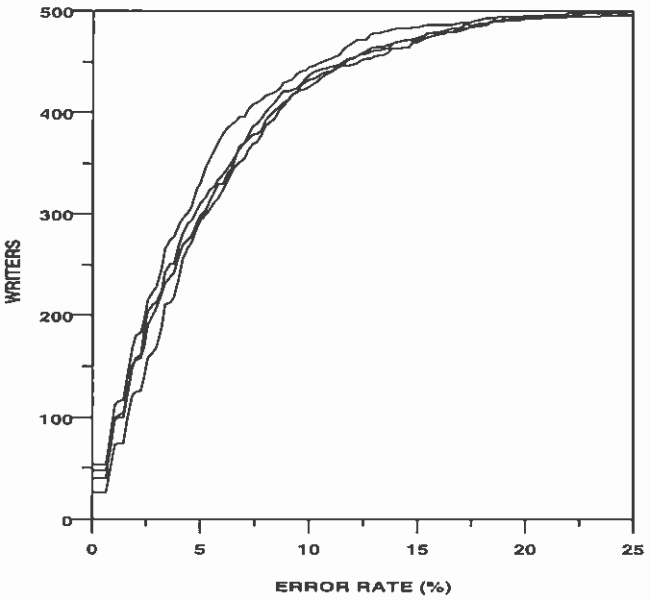

Figure 5: Writer dependence of error for four non-neural network based OCR systems.

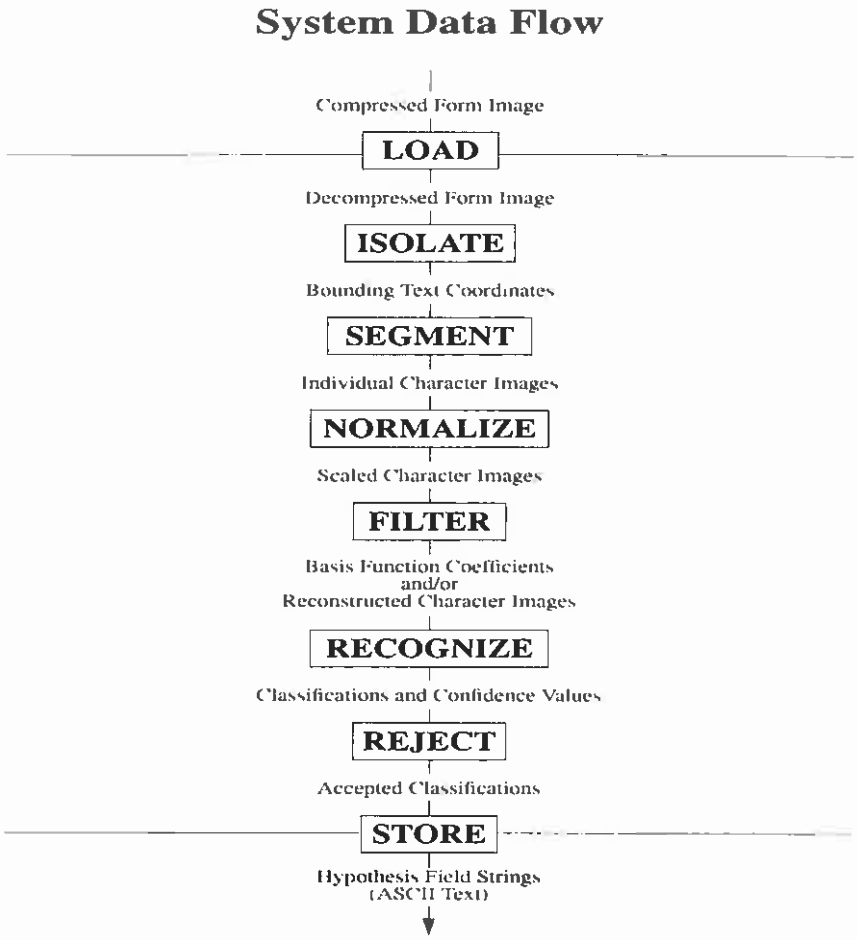

Figure 6: Data flow in a complete recognition system.

COMPONENT	OVERALL	PER FORM	
Load:	18668.328	8.889680	(58.54%)
Isolate:	3669.375	1.747321	(11.51%)
Segment:	4773.691	2.273186	(14.97%)
Normalize:	854.941	0.407115	(2.68%)
Filter:	3013.547	1.435023	(9.45%)
Recognize:	250.982	0.119515	(0.79%)
Reject:	50.900	0.024238	(0.16%)
Store:	609.079	0.290038	(1.91%)
Total:	31890.845	15.186117	(100.00%)

Table 2: System times in seconds for 2100 forms on a parallel computer.

5 Information Content and Network Performance

The systems submitted for testing at the Conference used all four combinations of rule-based and learning-based feature extraction and classification. Each combination yielded at least one low error rate system. The most common combination was the use of a mathematically based feature extractor with a MLP classifier. At least one system combined feature extraction with classification [15]. One major surprise was that linear methods, such as Learned Vector Quantitatization (LVQ) [8] and PNN [12] performed as well as highly non-linear methods such as MLPs.

A possible explanation for this can be found in Bayesian models of the learning and recognition process [16], [17], and [18]. The relationship between testing error, E_{tst} and training error E_{trn} is given by:

$$E_{tst} = E_{trn} + 2\sigma^2_{eff}\frac{p_{eff}}{n}$$

where σ^2_{eff} is the effective noise in the network variables, p_{eff} is the effective number of network parameters, and n is the size of the training sample.

The noise in the network is learned from the training sample and should be similar for all participants. Most participants achieved training errors of less than 0.5%. The strong similarity of accuracy results suggest that all of the methods used maintain a fixed ratio of complexity to sample size. This would suggest that, in noisy samples of the kind used in the Conference tests, learning can not remove sample noise injected into the classification system from the training data because the excess complexity of the network is used to track the noise in the data. This is not unexpected since the systems have no mechanism for evaluating "bad" writing except by statistical frequency.

An alternate explanation for correlated system performance is that as feature set size is expanded the ability of the feature set to span the feature space is limited. This limitation occurs because for features with a scale, S, and dimension, n, the size of the feature space expands as S^n. If the fractal dimension of the feature set is f, then the space that is covered by the features is of size S^f. This differs from Vapnik's argument in that the fractal

dimension is calculated in the limit of an infinite feature set. This limitation is a property of the distribution of features in space and cannot be solved by adding more training examples.

6 Conclusions

Examination of the results of 11 OCR systems using a wide variety of recognition algorithms has shown that in accuracy and writer independence neural network systems have not demonstrated a clear cut superiority over statistical methods. Some neural system have higher accuracy than statistical methods: other have lower accuracy. The performance of statistical methods is more closely grouped and is approximately the same as the performance of an average neural network system considered here. One area where neural networks may have an advantage is in speed of implementation and recognition. Analysis of a recognition system developed at NIST shows that at the systems level the OCR application is currently dominated by the speed of processing the image prior to recognition. This leads to the conclusion that neural networks have not yet demonstrated a clear superiority to more conventional statistical methods.

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 Conference. Technical Report NISTIR 4912, National Institute of Standards and Technology, August 1992.

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

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

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

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

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

[7] 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.

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

[9] 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.

[10] 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.

[11] 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.

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

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

[14] M. D. Garris, C. L. Wilson, J. L. Blue, G. T. Candela, P. Grother, S. Janet, and R. A. Wilkinson. Massivelly parallel implementation of character recognition systems. In *Conference on Character Recognition and Digitizer Technologies*, volume 1661, pages 269–280, San Jose California, February 1992. SPIE.

[15] Y. Le Cun, B. Boser, J. S. Denker, D. Henderson, R. E. Howard, W. Hubbard, and L. D. Jackel. Handwritten digit recognition with a back-propagation network. In D. Touretzky, editor, *Advances in Neural Information Processing Systems*, volume 2, pages 396–404. Morgan Kaufman, 1990.

[16] David J. C. MacKay. Bayesian model comparison and backprop nets. In R. Lippmann, editor, *Advances in Neural Information Processing System*, volume 4, pages 839–846. Morgan Kauffman, 1992.

[17] V. Vapnik. Principals of risk minimization for learning theory. In R. Lippmann, editor, *Advances in Neural Information Processing System*, volume 4, pages 831–838. Morgan Kauffman, 1992.

[18] J. E. Moody. The effective numbers of parameters: an analysis of generalization and regularization in nonlinear systems. In R. Lippmann, editor, *Advances in Neural Information Processing System*, volume 4, pages 847–854. Morgan Kauffman, 1992.